

libquentier

Generated by Doxygen 1.9.4

1 libquentier	1
1.1 What's this	1
1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!	1
1.2 How to build/install	1
1.3 How to contribute	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 File Index	11
4.1 File List	11
5 Class Documentation	15
5.1 quentier::Account Class Reference	15
5.1.1 Detailed Description	16
5.1.2 Member Function Documentation	17
5.1.2.1 displayName()	17
5.1.2.2 evernoteAccountType()	17
5.1.2.3 evernoteHost()	17
5.1.2.4 id()	17
5.1.2.5 isEmpty()	18
5.1.2.6 name()	18
5.1.2.7 print()	18
5.1.2.8 setDisplayNames()	18
5.1.2.9 shardId()	18
5.1.2.10 type()	19
5.2 quentier::utility::cancelers::AnyOfCanceler Class Reference	19
5.2.1 Member Function Documentation	20
5.2.1.1 isCanceled()	20
5.3 quentier::utility::ApplicationSettings Class Reference	20
5.3.1 Detailed Description	21
5.3.2 Constructor & Destructor Documentation	21
5.3.2.1 ApplicationSettings() [1/4]	21
5.3.2.2 ApplicationSettings() [2/4]	22
5.3.2.3 ApplicationSettings() [3/4]	22
5.3.2.4 ApplicationSettings() [4/4]	22
5.3.2.5 ~ApplicationSettings()	23
5.3.3 Member Function Documentation	23
5.3.3.1 beginGroup() [1/3]	23
5.3.3.2 beginGroup() [2/3]	23

5.3.3.3 beginGroup() [3/3]	24
5.3.3.4 beginReadArray() [1/3]	24
5.3.3.5 beginReadArray() [2/3]	24
5.3.3.6 beginReadArray() [3/3]	25
5.3.3.7 beginWriteArray() [1/3]	25
5.3.3.8 beginWriteArray() [2/3]	25
5.3.3.9 beginWriteArray() [3/3]	26
5.3.3.10 contains() [1/3]	26
5.3.3.11 contains() [2/3]	27
5.3.3.12 contains() [3/3]	27
5.3.3.13 print()	27
5.3.3.14 remove() [1/3]	27
5.3.3.15 remove() [2/3]	28
5.3.3.16 remove() [3/3]	28
5.3.3.17 setValue() [1/3]	28
5.3.3.18 setValue() [2/3]	29
5.3.3.19 setValue() [3/3]	29
5.3.3.20 value() [1/3]	29
5.3.3.21 value() [2/3]	30
5.3.3.22 value() [3/3]	30
5.4 quantier::utility::ApplicationSettings::ArrayCloser Struct Reference	31
5.4.1 Detailed Description	31
5.5 quantier::synchronization::AuthenticationExpiredError Struct Reference	32
5.5.1 Detailed Description	32
5.6 quantier::synchronization::ISyncConflictResolver::ConflictResolution Struct Reference	32
5.6.1 Detailed Description	32
5.7 quantier::ErrorString Class Reference	33
5.7.1 Detailed Description	34
5.7.2 Member Function Documentation	34
5.7.2.1 print()	34
5.8 quantier::utility::EventLoopWithExitStatus Class Reference	35
5.9 quantier::utility::IKeychainService::Exception Class Reference	36
5.9.1 Detailed Description	37
5.9.2 Member Function Documentation	37
5.9.2.1 exceptionDisplayName()	37
5.10 quantier::utility::FileIOProcessorAsync Class Reference	37
5.10.1 Detailed Description	38
5.10.2 Member Function Documentation	39
5.10.2.1 onReadFileRequest	39
5.10.2.2 onWriteFileRequest	39
5.10.2.3 readFileRequestProcessed	39
5.10.2.4 setIdleTimePeriod()	40

5.10.2.5 writeFileRequestProcessed	40
5.11 quantier::utility::FileSystemWatcher Class Reference	41
5.12 quantier::utility::cancelers::FutureCanceler< T > Class Template Reference	42
5.12.1 Detailed Description	42
5.12.2 Member Function Documentation	43
5.12.2.1 isCanceled()	43
5.13 quantier::utility::ApplicationSettings::GroupCloser Struct Reference	43
5.13.1 Detailed Description	44
5.14 quantier::synchronization::IAuthenticationInfo Class Reference	44
5.14.1 Detailed Description	45
5.14.2 Member Function Documentation	45
5.14.2.1 authenticationTime()	45
5.14.2.2 authToken()	45
5.14.2.3 authTokenExpirationTime()	45
5.14.2.4 noteStoreUrl()	45
5.14.2.5 shardId()	46
5.14.2.6 userId()	46
5.14.2.7 userStoreCookies()	46
5.14.2.8 webApiUrlPrefix()	46
5.15 quantier::synchronization::IAuthenticationInfoBuilder Class Reference	46
5.16 quantier::synchronization::IAuthenticator Class Reference	47
5.17 quantier::ResourceRecognitionIndexItem::IBarcodeItem Struct Reference	47
5.18 quantier::utility::cancelers::ICanceler Class Reference	47
5.18.1 Detailed Description	48
5.19 quantier::enml::IConverter Class Reference	48
5.19.1 Detailed Description	49
5.19.2 Member Function Documentation	49
5.19.2.1 convertEnmlToHtml()	49
5.19.2.2 convertEnmlToPlainText()	50
5.19.2.3 convertEnmlToWordsList()	50
5.19.2.4 convertHtmlToDoc()	50
5.19.2.5 convertHtmlToEnml()	51
5.19.2.6 convertHtmlToXhtml()	51
5.19.2.7 convertHtmlToXml()	51
5.19.2.8 convertPlainTextToWordsList()	52
5.19.2.9 exportNotesToEnex()	52
5.19.2.10 importEnex()	53
5.19.2.11 validateAndFixupEnml()	53
5.19.2.12 validateEnml()	53
5.20 quantier::enml::IDecryptedTextCache Class Reference	54
5.21 quantier::synchronization::IDownloadNotesStatus Class Reference	54
5.21.1 Detailed Description	55

5.22	quentier::synchronization::IDownloadResourcesStatus Class Reference	56
5.23	quentier::utility::IEncryptor Struct Reference	57
5.23.1	Detailed Description	57
5.23.2	Member Enumeration Documentation	57
5.23.2.1	Cipher	57
5.23.3	Member Function Documentation	58
5.23.3.1	decrypt()	58
5.23.3.2	encrypt()	58
5.24	quentier::enml::IENMLTagsConverter Class Reference	58
5.24.1	Detailed Description	59
5.24.2	Member Function Documentation	59
5.24.2.1	convertDecryptedText()	59
5.24.2.2	convertEncryptedText()	59
5.24.2.3	convertEnToDo()	60
5.24.2.4	convertResource()	60
5.25	quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine Struct Reference	61
5.25.1	Detailed Description	61
5.26	quentier::enml::IHtmlData Struct Reference	61
5.26.1	Detailed Description	62
5.26.2	Member Function Documentation	62
5.26.2.1	html()	62
5.26.2.2	numEnCryptNodes()	62
5.26.2.3	numEnDecryptedNodes()	63
5.26.2.4	numEnToDoNodes()	63
5.26.2.5	numHyperlinkNodes()	63
5.26.2.6	print()	63
5.27	quentier::utility::IKeychainService Class Reference	63
5.27.1	Detailed Description	64
5.27.2	Member Enumeration Documentation	64
5.27.2.1	ErrorCode	64
5.27.3	Member Function Documentation	65
5.27.3.1	deletePassword()	65
5.27.3.2	readPassword()	65
5.27.3.3	writePassword()	65
5.28	quentier::local_storage::ILocalStorage Class Reference	66
5.28.1	Member Enumeration Documentation	70
5.28.1.1	Affiliation	70
5.28.1.2	TagNotesRelation	70
5.28.2	Member Function Documentation	71
5.28.2.1	notifier()	71
5.29	quentier::local_storage::ILocalStorageNotifier Class Reference	71
5.30	quentier::INoteEditorBackend Class Reference	73

5.31 quentier::synchronization::INoteStoreFactory Class Reference	76
5.32 quentier::InvalidArgument Class Reference	76
5.32.1 Member Function Documentation	77
5.32.1.1 exceptionDisplayName()	77
5.33 quentier::ResourceRecognitionIndexItem::IObjectItem Struct Reference	77
5.34 quentier::local_storage::IPatch Class Reference	78
5.34.1 Detailed Description	78
5.34.2 Member Function Documentation	78
5.34.2.1 apply()	78
5.34.2.2 backupLocalStorage()	78
5.34.2.3 fromVersion()	79
5.34.2.4 patchLongDescription()	79
5.34.2.5 patchShortDescription()	79
5.34.2.6 removeLocalStorageBackup()	79
5.34.2.7 restoreLocalStorageFromBackup()	80
5.34.2.8 toVersion()	80
5.35 quentier::IQuentierException Class Reference	80
5.35.1 Detailed Description	81
5.35.2 Member Function Documentation	81
5.35.2.1 print()	81
5.36 quentier::synchronization::ISendStatus Class Reference	82
5.36.1 Detailed Description	83
5.36.2 Member Function Documentation	83
5.36.2.1 failedToSendNotebooks()	83
5.36.2.2 failedToSendNotes()	83
5.36.2.3 failedToSendSavedSearches()	84
5.36.2.4 failedToSendTags()	84
5.36.2.5 needToRepeatIncrementalSync()	84
5.36.2.6 stopSynchronizationError()	84
5.36.2.7 totalAttemptedToSendNotebooks()	84
5.36.2.8 totalAttemptedToSendNotes()	85
5.36.2.9 totalAttemptedToSendSavedSearches()	85
5.36.2.10 totalAttemptedToSendTags()	85
5.36.2.11 totalSuccessfullySentNotebooks()	85
5.36.2.12 totalSuccessfullySentNotes()	86
5.36.2.13 totalSuccessfullySentSavedSearches()	86
5.36.2.14 totalSuccessfullySentTags()	86
5.37 quentier::ResourceRecognitionIndexItem::IShapeItem Struct Reference	86
5.38 quentier::enml::conversion_rules::ISkipRule Class Reference	87
5.38.1 Detailed Description	88
5.38.2 Member Enumeration Documentation	88
5.38.2.1 Target	88

5.38.3 Member Function Documentation	88
5.38.3.1 caseSensitivity()	88
5.38.3.2 includeContents()	88
5.38.3.3 matchMode()	89
5.38.3.4 print()	89
5.38.3.5 target()	89
5.38.3.6 value()	89
5.39 quantier::enml::conversion_rules::ISkipRuleBuilder Class Reference	89
5.40 quantier::synchronization::ISyncChunksDataCounters Struct Reference	90
5.40.1 Detailed Description	91
5.40.2 Member Function Documentation	91
5.40.2.1 addedLinkedNotebooks()	91
5.40.2.2 addedNotebooks()	91
5.40.2.3 addedSavedSearches()	91
5.40.2.4 addedTags()	92
5.40.2.5 expungedLinkedNotebooks()	92
5.40.2.6 expungedNotebooks()	92
5.40.2.7 expungedSavedSearches()	92
5.40.2.8 expungedTags()	92
5.40.2.9 totalExpungedLinkedNotebooks()	92
5.40.2.10 totalExpungedNotebooks()	92
5.40.2.11 totalExpungedSavedSearches()	93
5.40.2.12 totalExpungedTags()	93
5.40.2.13 totalLinkedNotebooks()	93
5.40.2.14 totalNotebooks()	93
5.40.2.15 totalSavedSearches()	93
5.40.2.16 totalTags()	93
5.40.2.17 updatedLinkedNotebooks()	93
5.40.2.18 updatedNotebooks()	94
5.40.2.19 updatedSavedSearches()	94
5.40.2.20 updatedTags()	94
5.41 quantier::synchronization::ISyncConflictResolver Class Reference	94
5.41.1 Detailed Description	95
5.42 quantier::synchronization::ISyncEventsNotifier Class Reference	95
5.42.1 Member Function Documentation	96
5.42.1.1 downloadFinished	96
5.42.1.2 linkedNotebookNotesDownloadProgress	97
5.42.1.3 linkedNotebookResourcesDownloadProgress	97
5.42.1.4 linkedNotebookSendStatusUpdate	97
5.42.1.5 linkedNotebookSyncChunksDataProcessingProgress	98
5.42.1.6 linkedNotebookSyncChunksDownloaded	98
5.42.1.7 linkedNotebookSyncChunksDownloadProgress	98

5.42.1.8 notesDownloadProgress	99
5.42.1.9 resourcesDownloadProgress	99
5.42.1.10 startLinkedNotebooksDataDownloading	99
5.42.1.11 syncChunksDataProcessingProgress	100
5.42.1.12 syncChunksDownloaded	100
5.42.1.13 syncChunksDownloadProgress	100
5.42.1.14 userOwnSendStatusUpdate	100
5.43 quantier::synchronization::ISynchronizer Class Reference	101
5.44 quantier::synchronization::ISyncOptions Class Reference	101
5.44.1 Detailed Description	102
5.44.2 Member Function Documentation	102
5.44.2.1 downloadNoteThumbnails()	102
5.44.2.2 inkNoteImagesStorageDir()	103
5.44.2.3 maxConcurrentNoteDownloads()	103
5.44.2.4 maxConcurrentResourceDownloads()	103
5.44.2.5 requestContext()	103
5.44.2.6 retryPolicy()	103
5.45 quantier::synchronization::ISyncOptionsBuilder Class Reference	104
5.46 quantier::synchronization::ISyncResult Class Reference	104
5.47 quantier::synchronization::ISyncState Class Reference	105
5.47.1 Detailed Description	106
5.48 quantier::synchronization::ISyncStateBuilder Class Reference	106
5.49 quantier::synchronization::ISyncStateStorage Class Reference	107
5.49.1 Detailed Description	108
5.49.2 Member Function Documentation	108
5.49.2.1 notifySyncStateUpdated	108
5.50 quantier::ResourceRecognitionIndexItem::ITxtItem Struct Reference	108
5.51 quantier::synchronization::IUserStoreFactory Class Reference	108
5.52 quantier::local_storage::ILocalStorage::ListGuidsFilters Struct Reference	109
5.53 quantier::local_storage::ILocalStorage::ListLinkedNotebooksOptions Struct Reference	109
5.54 quantier::local_storage::ILocalStorage::ListNotebooksOptions Struct Reference	110
5.55 quantier::local_storage::ILocalStorage::ListNotesOptions Struct Reference	111
5.56 quantier::local_storage::ILocalStorage::ListObjectsFilters Struct Reference	112
5.57 quantier::local_storage::ILocalStorage::ListOptionsBase Struct Reference	113
5.58 quantier::local_storage::ILocalStorage::ListSavedSearchesOptions Struct Reference	114
5.59 quantier::local_storage::ILocalStorage::ListTagsOptions Struct Reference	115
5.60 quantier::local_storage::LocalStorageOpenException Class Reference	116
5.60.1 Detailed Description	117
5.60.2 Member Function Documentation	117
5.60.2.1 exceptionDisplayName()	117
5.61 quantier::local_storage::LocalStorageOperationException Class Reference	117
5.61.1 Detailed Description	118

5.61.2 Member Function Documentation	118
5.61.2.1 exceptionDisplayName()	118
5.62 quantier::utility::LRUCache< Key, Value, Allocator > Class Template Reference	119
5.63 quantier::utility::cancelers::ManualCanceler Class Reference	120
5.63.1 Detailed Description	120
5.63.2 Member Function Documentation	121
5.63.2.1 cancel()	121
5.63.2.2 isCanceled()	121
5.64 quantier::synchronization::tests::mocks::MockIAuthenticator Class Reference	121
5.65 quantier::utility::tests::mocks::MockIKeychainService Class Reference	122
5.66 quantier::local_storage::tests::mocks::MockILocalStorage Class Reference	123
5.67 quantier::synchronization::tests::mocks::MockINoteStoreFactory Class Reference	126
5.68 quantier::synchronization::tests::mocks::MockISyncConflictResolver Class Reference	127
5.69 quantier::synchronization::tests::mocks::MockISyncStateStorage Class Reference	128
5.70 quantier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T > Struct Template Reference	129
5.70.1 Detailed Description	129
5.70.2 Member Data Documentation	129
5.70.2.1 mine	129
5.71 quantier::NoteEditor Class Reference	130
5.71.1 Detailed Description	133
5.71.2 Member Function Documentation	133
5.71.2.1 backend()	133
5.71.2.2 clear()	134
5.71.2.3 convertToNote	134
5.71.2.4 currentNoteLocalId()	134
5.71.2.5 defaultFont()	134
5.71.2.6 defaultPalette()	134
5.71.2.7 idleTime()	134
5.71.2.8 inAppNoteLinkPasteRequested	135
5.71.2.9 initialize()	135
5.71.2.10 isEditorPageModified()	135
5.71.2.11 isModified()	136
5.71.2.12 isNoteLoaded()	136
5.71.2.13 saveNoteToLocalStorage	136
5.71.2.14 setAccount()	136
5.71.2.15 setBackend()	136
5.71.2.16 setCurrentNoteLocalId()	136
5.71.2.17 setDefaultFont	137
5.71.2.18 setDefaultPalette	137
5.71.2.19 setFocus()	137
5.71.2.20 setInitialPageHtml()	138

5.71.2.21 setNoteDeletedPageHtml()	138
5.71.2.22 setNoteLoadingPageHtml()	138
5.71.2.23 setNoteNotFoundPageHtml()	138
5.71.2.24 setNoteTitle	138
5.71.2.25 setTagIds	138
5.71.2.26 setUndoStack()	139
5.71.2.27 undoStack()	139
5.72 quantier::local_storage::NoteSearchQuery Class Reference	139
5.72.1 Member Function Documentation	142
5.72.1.1 notebookModifier()	142
5.72.1.2 print()	142
5.72.1.3 queryString()	142
5.73 quantier::OperationCanceled Class Reference	142
5.73.1 Member Function Documentation	143
5.73.1.1 exceptionDisplayName()	143
5.74 quantier::utility::Printable Class Reference	144
5.74.1 Detailed Description	145
5.75 QPromise< T > Class Template Reference	145
5.76 quantier::utility::QuantierApplication Class Reference	145
5.77 quantier::utility::QuantierUndoCommand Class Reference	146
5.77.1 Detailed Description	147
5.78 quantier::synchronization::RateLimitReachedError Struct Reference	148
5.78.1 Detailed Description	148
5.78.2 Member Data Documentation	148
5.78.2.1 rateLimitDurationSec	148
5.79 quantier::ResourceRecognitionIndexItem Class Reference	148
5.79.1 Member Function Documentation	150
5.79.1.1 print()	150
5.80 quantier::ResourceRecognitionIndices Class Reference	150
5.80.1 Member Function Documentation	151
5.80.1.1 print()	151
5.81 quantier::Result< ValueType, ErrorType, typename > Class Template Reference	152
5.81.1 Member Function Documentation	152
5.81.1.1 isValid()	152
5.82 quantier::threading::detail::ResultTypeHelper< F, Arg, Enable > Struct Template Reference	152
5.83 quantier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_↵ invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference	152
5.84 quantier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_↵ invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference	153
5.85 quantier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_↵ invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference	153
5.86 quantier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_↵ invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference	153

5.87	quentier::RuntimeError Class Reference	153
5.87.1	Member Function Documentation	154
5.87.1.1	exceptionDisplayName()	154
5.88	quentier::utility::ShortcutManager Class Reference	155
5.88.1	Member Function Documentation	156
5.88.1.1	defaultShortcut() [1/2]	156
5.88.1.2	defaultShortcut() [2/2]	157
5.88.1.3	shortcut() [1/2]	157
5.88.1.4	shortcut() [2/2]	157
5.88.1.5	userShortcut() [1/2]	157
5.88.1.6	userShortcut() [2/2]	158
5.89	quentier::SpellChecker Class Reference	158
5.90	quentier::utility::StringUtils Class Reference	159
5.91	quentier::utility::SysInfo Class Reference	159
5.92	quentier::threading::TrackedTask< LockableObject, Function > Class Template Reference	159
5.92.1	Detailed Description	160
5.93	quentier::utility::UidGenerator Class Reference	160
5.94	quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine Struct Reference	160
5.94.1	Detailed Description	160
5.95	quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs Struct Reference	160
5.95.1	Detailed Description	160
6	File Documentation	161
6.1	ISkipRule.h	161
6.2	ISkipRuleBuilder.h	162
6.3	MatchMode.h	162
6.4	enml/conversion_rules/Factory.h	163
6.5	enml/Factory.h	163
6.6	local_storage/Factory.h	164
6.7	synchronization/Factory.h	164
6.8	threading/Factory.h	165
6.9	utility/Factory.h	165
6.10	HtmlUtils.h	166
6.11	IConverter.h	166
6.12	IDecryptedTextCache.h	167
6.13	IENMLTagsConverter.h	168
6.14	IHtmlData.h	169
6.15	InvalidArgument.h	170
6.16	IQuentierException.h	170
6.17	OperationCanceled.h	171
6.18	RuntimeError.h	171
6.19	ILocalStorage.h	172

6.20 ILocalStorageNotifier.h	180
6.21 IPatch.h	181
6.22 LocalStorageOpenException.h	182
6.23 LocalStorageOperationException.h	182
6.24 NoteSearchQuery.h	183
6.25 MockILocalStorage.h	185
6.26 QuentierLogger.h	189
6.27 INoteEditorBackend.h	190
6.28 NoteEditor.h	193
6.29 SpellChecker.h	196
6.30 IAuthenticator.h	197
6.31 INoteStoreFactory.h	198
6.32 ISyncConflictResolver.h	198
6.33 ISyncEventsNotifier.h	200
6.34 ISynchronizer.h	201
6.35 ISyncStateStorage.h	202
6.36 IUserStoreFactory.h	202
6.37 MockIAuthenticator.h	203
6.38 MockINoteStoreFactory.h	203
6.39 MockISyncConflictResolver.h	204
6.40 MockISyncStateStorage.h	204
6.41 Errors.h	205
6.42 IAuthenticationInfo.h	206
6.43 IAuthenticationInfoBuilder.h	206
6.44 IDownloadNotesStatus.h	207
6.45 IDownloadResourcesStatus.h	208
6.46 ISendStatus.h	209
6.47 ISyncChunksDataCounters.h	210
6.48 ISyncOptions.h	211
6.49 ISyncOptionsBuilder.h	212
6.50 ISyncResult.h	213
6.51 ISyncState.h	214
6.52 ISyncStateBuilder.h	214
6.53 AuthenticationInfo.h	215
6.54 DownloadNotesStatus.h	215
6.55 DownloadResourcesStatus.h	216
6.56 SendStatus.h	216
6.57 SyncChunksDataCounters.h	217
6.58 SyncResult.h	217
6.59 SyncState.h	218
6.60 Future.h	218
6.61 Post.h	221

6.62 Qt5Promise.h	222
6.63 QtFutureContinuations.h	224
6.64 QtFutureHelpers.h	230
6.65 Runnable.h	233
6.66 TrackedTask.h	233
6.67 Account.h	234
6.68 ErrorString.h	236
6.69 enml/conversion_rules/Fwd.h	237
6.70 enml/Fwd.h	237
6.71 local_storage/Fwd.h	238
6.72 synchronization/Fwd.h	238
6.73 synchronization/types/Fwd.h	239
6.74 threading/Fwd.h	239
6.75 types/Fwd.h	240
6.76 utility/cancelers/Fwd.h	240
6.77 utility/Fwd.h	241
6.78 NoteUtils.h	241
6.79 RegisterMetatypes.h	242
6.80 ResourceRecognitionIndexItem.h	242
6.81 ResourceRecognitionIndices.h	244
6.82 ResourceUtils.h	245
6.83 Result.h	245
6.84 Validation.h	248
6.85 ApplicationSettings.h	248
6.86 AnyOfCanceler.h	250
6.87 FutureCanceler.h	251
6.88 ICanceler.h	251
6.89 ManualCanceler.h	252
6.90 Compat.h	252
6.91 DateTime.h	253
6.92 EventLoopWithExitStatus.h	253
6.93 FileIOProcessorAsync.h	254
6.94 FileSystem.h	255
6.95 FileSystemWatcher.h	255
6.96 IEncryptor.h	256
6.97 IKeychainService.h	257
6.98 Initialize.h	258
6.99 LRUCache.hpp	258
6.100 MessageBox.h	261
6.101 PlatformUtils.h	262
6.102 Printable.h	262
6.103 QuentierApplication.h	263

6.104 QuentierUndoCommand.h	264
6.105 ShortcutManager.h	265
6.106 Size.h	267
6.107 StandardPaths.h	267
6.108 StringUtils.h	268
6.109 SuppressWarnings.h	268
6.110 SysInfo.h	269
6.111 TagSortByParentChildRelations.h	270
6.112 MockIKeychainService.h	271
6.113 UidGenerator.h	271
6.114 Unreachable.h	272
Index	273

Chapter 1

libquentier

Set of Qt/C++ APIs for feature rich desktop clients for Evernote service

1.1 What's this

This library presents a set of Qt/C++ APIs useful for applications working as feature rich desktop clients for Evernote service. The most important and useful components of the library are the following:

- Local storage - persistence of data downloaded from Evernote service in a local SQLite database
- Synchronization - the logics of exchanging new and/or modified data with Evernote service
- Note editor - the UI component capable for notes displaying and editing

The library is based on the lower level functionality provided by `QEverCloud` library. It also serves as the functional core of `Quentier` application.

1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!

1.2 How to build/install

Please see the building/installation guide.

1.3 How to contribute

Please see the contribution guide for detailed info.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

quentier::utility::ApplicationSettings::ArrayCloser	31
quentier::synchronization::AuthenticationExpiredError	32
quentier::synchronization::ISyncConflictResolver::ConflictResolution	32
quentier::utility::ApplicationSettings::GroupCloser	43
quentier::synchronization::IAuthenticationInfoBuilder	46
quentier::synchronization::IAuthenticator	47
quentier::synchronization::tests::mocks::MockIAuthenticator	121
quentier::ResourceRecognitionIndexItem::IBarcodeItem	47
quentier::utility::cancelers::ICanceler	47
quentier::utility::cancelers::AnyOfCanceler	19
quentier::utility::cancelers::FutureCanceler< T >	42
quentier::utility::cancelers::ManualCanceler	120
quentier::enml::IConverter	48
quentier::enml::IDecryptedTextCache	54
quentier::utility::IEncryptor	57
quentier::enml::IENMLTagsConverter	58
quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine	61
quentier::utility::IKeychainService	63
quentier::utility::tests::mocks::MockIKeychainService	122
quentier::local_storage::ILocalStorage	66
quentier::local_storage::tests::mocks::MockILocalStorage	123
quentier::INoteEditorBackend	73
quentier::synchronization::INoteStoreFactory	76
quentier::synchronization::tests::mocks::MockINoteStoreFactory	126
quentier::ResourceRecognitionIndexItem::IObjectItem	77
quentier::local_storage::IPatch	78
quentier::ResourceRecognitionIndexItem::IShapeItem	86
quentier::enml::conversion_rules::ISkipRuleBuilder	89
quentier::synchronization::ISyncConflictResolver	94
quentier::synchronization::tests::mocks::MockISyncConflictResolver	127
quentier::synchronization::ISynchronizer	101
quentier::synchronization::ISyncOptionsBuilder	104
quentier::synchronization::ISyncStateBuilder	106

quentier::ResourceRecognitionIndexItem::ITextItem	108
quentier::synchronization::IUserStoreFactory	108
quentier::local_storage::ILocalStorage::ListGuidsFilters	109
quentier::local_storage::ILocalStorage::ListObjectsFilters	112
quentier::local_storage::ILocalStorage::ListOptionsBase	113
quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions	109
quentier::local_storage::ILocalStorage::ListNotebooksOptions	110
quentier::local_storage::ILocalStorage::ListNotesOptions	111
quentier::local_storage::ILocalStorage::ListSavedSearchesOptions	114
quentier::local_storage::ILocalStorage::ListTagsOptions	115
quentier::utility::LRUCache< Key, Value, Allocator >	119
quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >	129
quentier::utility::Printable	144
quentier::Account	15
quentier::ErrorString	33
quentier::IQuentierException	80
quentier::InvalidArgument	76
quentier::OperationCanceled	142
quentier::RuntimeError	153
quentier::local_storage::LocalStorageOpenException	116
quentier::local_storage::LocalStorageOperationException	117
quentier::utility::IKeychainService::Exception	36
quentier::ResourceRecognitionIndexItem	148
quentier::ResourceRecognitionIndices	150
quentier::enml::IHtmlData	61
quentier::enml::conversion_rules::ISkipRule	87
quentier::local_storage::NoteSearchQuery	139
quentier::synchronization::IAuthenticationInfo	44
quentier::synchronization::IDownloadNotesStatus	54
quentier::synchronization::IDownloadResourcesStatus	56
quentier::synchronization::ISendStatus	82
quentier::synchronization::ISyncChunksDataCounters	90
quentier::synchronization::ISyncOptions	101
quentier::synchronization::ISyncResult	104
quentier::synchronization::ISyncState	105
quentier::utility::ApplicationSettings	20
QApplication	
quentier::utility::QuentierApplication	145
QEventLoop	
quentier::utility::EventLoopWithExitStatus	35
QException	
quentier::IQuentierException	80
QObject	
quentier::SpellChecker	158
quentier::local_storage::ILocalStorageNotifier	71
quentier::synchronization::ISyncEventsNotifier	95
quentier::synchronization::ISyncStateStorage	107
quentier::synchronization::tests::mocks::MockISyncStateStorage	128
quentier::utility::FileIOProcessorAsync	37
quentier::utility::FileSystemWatcher	41
quentier::utility::QuentierUndoCommand	146
quentier::utility::ShortcutManager	155
QPromise< T >	145
QSettings	
quentier::utility::ApplicationSettings	20
QUndoCommand	
quentier::utility::QuentierUndoCommand	146
QWidget	

quentier::NoteEditor	130
quentier::synchronization::RateLimitReachedError	148
quentier::Result< ValueType, ErrorType, typename >	152
quentier::threading::detail::ResultTypeHelper< F, Arg, Enable >	152
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > >	152
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > >	153
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >	153
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >	153
quentier::utility::StringUtils	159
quentier::utility::SysInfo	159
quentier::threading::TrackedTask< LockableObject, Function >	159
quentier::utility::UidGenerator	160
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine	160
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs	160

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

quentier::Account	Encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc	15
quentier::utility::cancelers::AnyOfCanceler		19
quentier::utility::ApplicationSettings	Enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings	20
quentier::utility::ApplicationSettings::ArrayCloser		31
quentier::synchronization::AuthenticationExpiredError		32
quentier::synchronization::ISyncConflictResolver::ConflictResolution	The ConflictResolution struct is a namespace inside which several other structs determining actual conflict resolutions	32
quentier::ErrorString	Encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description	33
quentier::utility::EventLoopWithExitStatus		35
quentier::utility::IKeychainService::Exception	The IKeychainService::Exception class is the base class for exceptions returned inside QFutures from methods of IKeychainService	36
quentier::utility::FileIOProcessorAsync	Wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO . . .	37
quentier::utility::FileSystemWatcher		41
quentier::utility::cancelers::FutureCanceler< T >		42
quentier::utility::ApplicationSettings::GroupCloser		43
quentier::synchronization::IAuthenticationInfo	The IAuthenticationInfo interface represents the information obtained through OAuth and necessary to access Evernote API	44
quentier::synchronization::IAuthenticationInfoBuilder		46
quentier::synchronization::IAuthenticator		47
quentier::ResourceRecognitionIndexItem::IBarcodeItem		47
quentier::utility::cancelers::ICanceler	The ICanceler interface provides isCanceled method which can be used to check whether some processing can be skipped because it was canceled	47
quentier::enml::IConverter	The IConverter interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML	48

quentier::enml::IDecryptedTextCache	54
quentier::synchronization::IDownloadNotesStatus	
The IDownloadNotesStatus interface presents information about the status of notes downloading process	54
quentier::synchronization::IDownloadResourcesStatus	56
quentier::utility::IEncryptor	
The IEncryptor interface provides encryption and decryption functionality which is compatible with that used by Evernote service	57
quentier::enml::IENMLTagsConverter	
The IENMLTagsConverter interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content	58
quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine	
The IgnoreMine conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version"	61
quentier::enml::IHtmlData	
The IHtmlData represents the result of ENML to HTML conversion: HTML itself plus some meta-data	61
quentier::utility::IKeychainService	
The IKeychainService interface provides the ability to interact with the storage of sensitive data - read, write and delete it	63
quentier::local_storage::ILocalStorage	66
quentier::local_storage::ILocalStorageNotifier	71
quentier::INoteEditorBackend	73
quentier::synchronization::INoteStoreFactory	76
quentier::InvalidArgument	76
quentier::ResourceRecognitionIndexItem::IObjectItem	77
quentier::local_storage::IPatch	
The IPatch interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it	78
quentier::IQuentierException	
Interface for exceptions specific to libquentier and applications based on it	80
quentier::synchronization::ISendStatus	
The ISendStatus interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote	82
quentier::ResourceRecognitionIndexItem::IShapeItem	86
quentier::enml::conversion_rules::ISkipRule	
The ISkipRule interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion	87
quentier::enml::conversion_rules::ISkipRuleBuilder	89
quentier::synchronization::ISyncChunksDataCounters	
The ISyncChunksDataCounters interface provides integer counters representing the current progress on processing the data from downloaded sync chunks	90
quentier::synchronization::ISyncConflictResolver	
The ISyncConflictResolver interface provides methods used to resolve conflicts between local and remote versions of the same data item	94
quentier::synchronization::ISyncEventsNotifier	95
quentier::synchronization::ISynchronizer	101
quentier::synchronization::ISyncOptions	
Options for synchronization process	101
quentier::synchronization::ISyncOptionsBuilder	104
quentier::synchronization::ISyncResult	104
quentier::synchronization::ISyncState	
The ISyncState interface provides accessory methods to determine the sync state for the account	105
quentier::synchronization::ISyncStateBuilder	106

quentier::synchronization::ISyncStateStorage	
The ISyncStateStorage interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states	107
quentier::ResourceRecognitionIndexItem::ITextItem	108
quentier::synchronization::IUserStoreFactory	108
quentier::local_storage::ILocalStorage::ListGuidsFilters	109
quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions	109
quentier::local_storage::ILocalStorage::ListNotebooksOptions	110
quentier::local_storage::ILocalStorage::ListNotesOptions	111
quentier::local_storage::ILocalStorage::ListObjectsFilters	112
quentier::local_storage::ILocalStorage::ListOptionsBase	113
quentier::local_storage::ILocalStorage::ListSavedSearchesOptions	114
quentier::local_storage::ILocalStorage::ListTagsOptions	115
quentier::local_storage::LocalStorageOpenException	
The LocalStorageOpenException is thrown on failure to open the local storage database	116
quentier::local_storage::LocalStorageOperationException	
The LocalStorageOperationException is thrown when the local storage encounters some internal error during the attempt to process some operation	117
quentier::utility::LRUCache< Key, Value, Allocator >	119
quentier::utility::cancelers::ManualCanceler	120
quentier::synchronization::tests::mocks::MockIAuthenticator	121
quentier::utility::tests::mocks::MockIKeychainService	122
quentier::local_storage::tests::mocks::MockILocalStorage	123
quentier::synchronization::tests::mocks::MockINoteStoreFactory	126
quentier::synchronization::tests::mocks::MockISyncConflictResolver	127
quentier::synchronization::tests::mocks::MockISyncStateStorage	128
quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >	
The MoveMine conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one	129
quentier::NoteEditor	
Widget encapsulating all the functionality necessary for showing and editing notes	130
quentier::local_storage::NoteSearchQuery	139
quentier::OperationCanceled	142
quentier::utility::Printable	
Interface for Quentier's internal classes which should be able to write themselves into QText←Stream and/or convert to QString	144
QPromise< T >	145
quentier::utility::QuentierApplication	145
quentier::utility::QuentierUndoCommand	
Has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that	146
quentier::synchronization::RateLimitReachedError	148
quentier::ResourceRecognitionIndexItem	148
quentier::ResourceRecognitionIndices	150
quentier::Result< ValueType, ErrorType, typename >	152
quentier::threading::detail::ResultTypeHelper< F, Arg, Enable >	152
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFu	152
152	
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFu	153
153	
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFu	153
153	
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFu	153
153	

quentier::RuntimeError	153
quentier::utility::ShortcutManager	155
quentier::SpellChecker	158
quentier::utility::StringUtils	159
quentier::utility::SysInfo	159
quentier::threading::TrackedTask< LockableObject, Function >	159
quentier::utility::UidGenerator	160
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine	
The UseMine conflict resolution means "override theirs version with mine version"	160
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs	
The UseTheirs conflict resolution means "override mine version with theirs version"	160

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

ISkipRule.h	161
ISkipRuleBuilder.h	162
MatchMode.h	162
enml/conversion_rules/Factory.h	163
enml/Factory.h	163
local_storage/Factory.h	164
synchronization/Factory.h	164
threading/Factory.h	165
utility/Factory.h	165
HtmlUtils.h	166
IConverter.h	166
IDecryptedTextCache.h	167
IENMLTagsConverter.h	168
IHtmlData.h	169
InvalidArgument.h	170
IQuentierException.h	170
OperationCanceled.h	171
RuntimeError.h	171
ILocalStorage.h	172
ILocalStorageNotifier.h	180
IPatch.h	181
LocalStorageOpenException.h	182
LocalStorageOperationException.h	182
NoteSearchQuery.h	183
MockILocalStorage.h	185
QuentierLogger.h	189
INoteEditorBackend.h	190
NoteEditor.h	193
SpellChecker.h	196
IAuthenticator.h	197
INoteStoreFactory.h	198
ISyncConflictResolver.h	198
ISyncEventsNotifier.h	200
ISynchronizer.h	201
ISyncStateStorage.h	202

IUserStoreFactory.h	202
MockIAuthenticator.h	203
MockINoteStoreFactory.h	203
MockISyncConflictResolver.h	204
MockISyncStateStorage.h	204
Errors.h	205
IAuthenticationInfo.h	206
IAuthenticationInfoBuilder.h	206
IDownloadNotesStatus.h	207
IDownloadResourcesStatus.h	208
ISendStatus.h	209
ISyncChunksDataCounters.h	210
ISyncOptions.h	211
ISyncOptionsBuilder.h	212
ISyncResult.h	213
ISyncState.h	214
ISyncStateBuilder.h	214
AuthenticationInfo.h	215
DownloadNotesStatus.h	215
DownloadResourcesStatus.h	216
SendStatus.h	216
SyncChunksDataCounters.h	217
SyncResult.h	217
SyncState.h	218
Future.h	218
Post.h	221
Qt5Promise.h	222
QtFutureContinuations.h	224
QtFutureHelpers.h	230
Runnable.h	233
TrackedTask.h	233
Account.h	234
ErrorString.h	236
enml/conversion_rules/Fwd.h	237
enml/Fwd.h	237
local_storage/Fwd.h	238
synchronization/Fwd.h	238
synchronization/types/Fwd.h	239
threading/Fwd.h	239
types/Fwd.h	240
utility/cancelers/Fwd.h	240
utility/Fwd.h	241
NoteUtils.h	241
RegisterMetatypes.h	242
ResourceRecognitionIndexItem.h	242
ResourceRecognitionIndices.h	244
ResourceUtils.h	245
Result.h	245
Validation.h	248
ApplicationSettings.h	248
AnyOfCanceler.h	250
FutureCanceler.h	251
ICanceler.h	251
ManualCanceler.h	252
Compat.h	252
DateTime.h	253
EventLoopWithExitStatus.h	253
FileIOProcessorAsync.h	254

FileSystem.h	255
FileSystemWatcher.h	255
IEncryptor.h	256
IKeychainService.h	257
Initialize.h	258
LRUCache.hpp	258
MessageBox.h	261
PlatformUtils.h	262
Printable.h	262
QuentierApplication.h	263
QuentierUndoCommand.h	264
ShortcutManager.h	265
Size.h	267
StandardPaths.h	267
StringUtils.h	268
SuppressWarnings.h	268
SysInfo.h	269
TagSortByParentChildRelations.h	270
MockIKeychainService.h	271
UidGenerator.h	271
Unreachable.h	272

Chapter 5

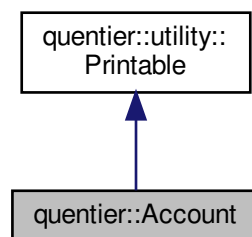
Class Documentation

5.1 `quentier::Account` Class Reference

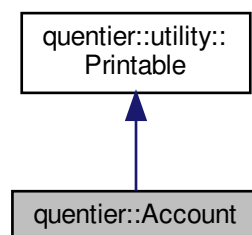
The [Account](#) class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

```
#include <Account.h>
```

Inheritance diagram for `quentier::Account`:



Collaboration diagram for `quentier::Account`:



Public Types

- enum class **Type** { **Local** , **Evernote** }
- enum class **EvernoteAccountType** { **Free** , **Plus** , **Premium** , **Business** }

Public Member Functions

- **Account** (QString **name**, Type **type**, qevercloud::UserID **userId**=-1, EvernoteAccountType **evernoteAccountType**=EvernoteAccountType::Free, QString **evernoteHost**={}, QString **shardId**={})
- **Account** (const **Account** &other)
- **Account** (**Account** &&other) noexcept
- **Account** & **operator=** (const **Account** &other)
- **Account** & **operator=** (**Account** &&other) noexcept
- bool **operator==** (const **Account** &other) const noexcept
- bool **operator!=** (const **Account** &other) const noexcept
- bool **isEmpty** () const
- QString **name** () const
- void **setName** (QString **name**)
setName sets the username to the account
- QString **displayName** () const
- void **setDisplayName** (QString **displayName**)
- Type **type** () const
- qevercloud::UserID **id** () const
- EvernoteAccountType **evernoteAccountType** () const
- QString **evernoteHost** () const
- QString **shardId** () const
- void **setEvernoteAccountType** (EvernoteAccountType **evernoteAccountType**)
- void **setEvernoteHost** (QString **evernoteHost**)
- void **setShardId** (QString **shardId**)
- qint32 **mailLimitDaily** () const
- qint64 **noteSizeMax** () const
- qint64 **resourceSizeMax** () const
- qint32 **linkedNotebookMax** () const
- qint32 **noteCountMax** () const
- qint32 **notebookCountMax** () const
- qint32 **tagCountMax** () const
- qint32 **noteTagCountMax** () const
- qint32 **savedSearchCountMax** () const
- qint32 **noteResourceCountMax** () const
- void **setEvernoteAccountLimits** (const qevercloud::AccountLimits &limits)
- QTextStream & **print** (QTextStream &strm) const override

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, Type **type**)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, Type **type**)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, EvernoteAccountType **type**)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, EvernoteAccountType **type**)

5.1.1 Detailed Description

The **Account** class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

5.1.2 Member Function Documentation

5.1.2.1 displayName()

```
QString quentier::Account::displayName ( ) const
```

Returns

Displayable user's name which is not used to uniquely identify the account, so this name may repeat across different local and Evernote accounts

5.1.2.2 evernoteAccountType()

```
EvernoteAccountType quentier::Account::evernoteAccountType ( ) const
```

Returns

The type of the Evernote account; if applied to free account, returns "Free"

5.1.2.3 evernoteHost()

```
QString quentier::Account::evernoteHost ( ) const
```

Returns

The Evernote server host with which the account is associated

5.1.2.4 id()

```
qevercloud::UserID quentier::Account::id ( ) const
```

Returns

User id for Evernote accounts, -1 for local accounts (as the concept of user id is not defined for local accounts)

5.1.2.5 isEmpty()

```
bool quentier::Account::isEmpty ( ) const
```

Returns

True if either the account is local but the name is empty or if the account is Evernote but user id is negative; in all other cases return false

5.1.2.6 name()

```
QString quentier::Account::name ( ) const
```

Returns

Username for either local or Evernote account

5.1.2.7 print()

```
QTextStream & quentier::Account::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

5.1.2.8 setDisplayName()

```
void quentier::Account::setDisplayName (
    QString displayName )
```

Set the printable name of the account

5.1.2.9 shardId()

```
QString quentier::Account::shardId ( ) const
```

Returns

Shard id for Evernote accounts, empty string for local accounts (as the concept of shard id is not defined for local accounts)

5.1.2.10 type()

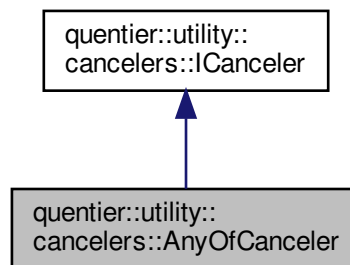
```
Type quantier::Account::type ( ) const
```

Returns

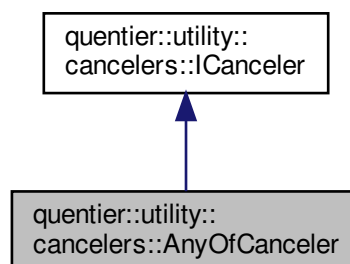
The type of the account: either local of Evernote

5.2 quantier::utility::cancelers::AnyOfCanceler Class Reference

Inheritance diagram for quantier::utility::cancelers::AnyOfCanceler:



Collaboration diagram for quantier::utility::cancelers::AnyOfCanceler:



Public Member Functions

- **AnyOfCanceler** (QList< ICancelerPtr > cancelers)
- **AnyOfCanceler** ([AnyOfCanceler](#) &&other) noexcept
- [AnyOfCanceler](#) & **operator=** ([AnyOfCanceler](#) &&other) noexcept
- bool [isCanceled](#) () const noexcept override

5.2.1 Member Function Documentation

5.2.1.1 isCanceled()

```
bool quantier::utility::cancelers::AnyOfCanceler::isCanceled ( ) const [override], [virtual],  
[noexcept]
```

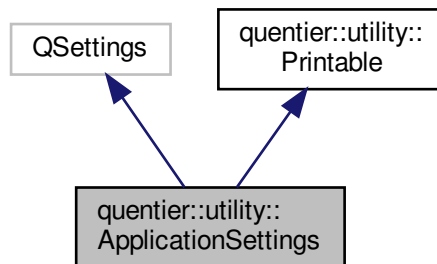
Implements [quantier::utility::cancelers::ICanceler](#).

5.3 quantier::utility::ApplicationSettings Class Reference

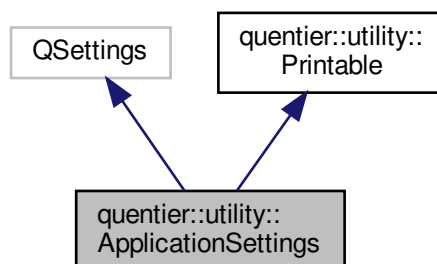
The [ApplicationSettings](#) class enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings.

```
#include <ApplicationSettings.h>
```

Inheritance diagram for quantier::utility::ApplicationSettings:



Collaboration diagram for quantier::utility::ApplicationSettings:



Classes

- struct [ArrayCloser](#)
- struct [GroupCloser](#)

Public Member Functions

- [ApplicationSettings](#) (const QString &settingsName={})
- [ApplicationSettings](#) (const [Account](#) &account, const QString &settingsName={})
- [ApplicationSettings](#) (const [Account](#) &account, const char *settingsName, int settingsNameSize=-1)
- [ApplicationSettings](#) (const [Account](#) &account, std::string_view settingsName)
- [~ApplicationSettings](#) () override
- void [beginGroup](#) (const QString &prefix)
- void [beginGroup](#) (const char *prefix, int size=-1)
- void [beginGroup](#) (std::string_view prefix)
- int [beginReadArray](#) (const QString &prefix)
- int [beginReadArray](#) (const char *prefix, int size=-1)
- int [beginReadArray](#) (std::string_view prefix)
- void [beginWriteArray](#) (const QString &prefix, int arraySize=-1)
- void [beginWriteArray](#) (const char *prefix, int arraySize=-1, int prefixSize=-1)
- void [beginWriteArray](#) (std::string_view prefix, int arraySize=-1)
- bool [contains](#) (const QString &key) const
- bool [contains](#) (const char *key, int size=-1) const
- bool [contains](#) (std::string_view key) const
- void [remove](#) (const QString &key)
- void [remove](#) (const char *key, int size=-1)
- void [remove](#) (std::string_view key)
- void [setValue](#) (const QString &key, const QVariant &value)
- void [setValue](#) (const char *key, const QVariant &value, int keySize=-1)
- void [setValue](#) (std::string_view key, const QVariant &value)
- QVariant [value](#) (const QString &key, const QVariant &defaultValue={}) const
- QVariant [value](#) (const char *key, const QVariant &defaultValue={}, int keySize=-1) const
- QVariant [value](#) (std::string_view key, const QVariant &defaultValue={}) const
- QTextStream & [print](#) (QTextStream &strm) const override

5.3.1 Detailed Description

The [ApplicationSettings](#) class enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ApplicationSettings() [1/4]

```
quantier::utility::ApplicationSettings::ApplicationSettings (
    const QString & settingsName = {} ) [explicit]
```

Constructor for application settings not being account-specific

Parameters

<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the common settings storage; otherwise they would be stored in the default settings file for the account
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.3.2.2 ApplicationSettings() [2/4]

```
quentier::utility::ApplicationSettings::ApplicationSettings (
    const Account & account,
    const QString & settingsName = {} ) [explicit]
```

Constructor for application settings specific to the account

Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account

5.3.2.3 ApplicationSettings() [3/4]

```
quentier::utility::ApplicationSettings::ApplicationSettings (
    const Account & account,
    const char * settingsName,
    int settingsNameSize = -1 )
```

Constructor for application settings specific to the account

Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not nullptr, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>settingsNameSize</i>	Size of the settingsName string. If negative (the default), the settingsName size is taken to be strlen(settingsName)

5.3.2.4 ApplicationSettings() [4/4]

```
quentier::utility::ApplicationSettings::ApplicationSettings (
```

```
const Account & account,
std::string_view settingsName )
```

Constructor for application settings specific to the account

Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8

5.3.2.5 ~ApplicationSettings()

```
quentier::utility::ApplicationSettings::~~ApplicationSettings ( ) [override]
```

Destructor

5.3.3 Member Function Documentation

5.3.3.1 beginGroup() [1/3]

```
void quentier::utility::ApplicationSettings::beginGroup (
    const char * prefix,
    int size = -1 )
```

Appends prefix to the current group. Overload of beginGroup accepting const char * and optionally the size of the string

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix).

5.3.3.2 beginGroup() [2/3]

```
void quentier::utility::ApplicationSettings::beginGroup (
    const QString & prefix )
```

Appends prefix to the current group. The call is redirected to QSettings::beginGroup. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>prefix</i>	String containing the prefix name
---------------	-----------------------------------

5.3.3.3 beginGroup() [3/3]

```
void quantier::utility::ApplicationSettings::beginGroup (
    std::string_view prefix )
```

Appends prefix to the current group. Overload of beginGroup accepting std::string_view

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix).

5.3.3.4 beginReadArray() [1/3]

```
int quantier::utility::ApplicationSettings::beginReadArray (
    const char * prefix,
    int size = -1 )
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting const char * and optionally the size of the string

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix)

5.3.3.5 beginReadArray() [2/3]

```
int quantier::utility::ApplicationSettings::beginReadArray (
    const QString & prefix )
```

Adds prefix to the current group and starts reading from an array. The call is redirected to QSettings::beginReadArray. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>prefix</i>	String containing the prefix name
---------------	-----------------------------------

Returns

The size of the array

5.3.3.6 beginReadArray() [3/3]

```
int quotientier::utility::ApplicationSettings::beginReadArray (
    std::string_view prefix )
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting std::string_view

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
---------------	-------------------------------------------------------------------------------------------------------------------------

5.3.3.7 beginWriteArray() [1/3]

```
void quotientier::utility::ApplicationSettings::beginWriteArray (
    const char * prefix,
    int arraySize = -1,
    int prefixSize = -1 )
```

Adds prefix to the current group and starts writing an array of size arraySize. Overload of beginWriteArray accepting const char * and optionally the size of the string

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.
<i>prefixSize</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix)

5.3.3.8 beginWriteArray() [2/3]

```
void quotientier::utility::ApplicationSettings::beginWriteArray (
```

```
const QString & prefix,
int arraySize = -1 )
```

Adds prefix to the current group and starts writing an array of size arraySize. The call is redirected to QSettings::beginWriteArray. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>prefix</i>	String containing the prefix name
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

5.3.3.9 beginWriteArray() [3/3]

```
void quentier::utility::ApplicationSettings::beginWriteArray (
    std::string_view prefix,
    int arraySize = -1 )
```

Adds prefix to the current group and starts writing an array of size arraySize. Overload of beginWriteArray accepting std::string_view

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

5.3.3.10 contains() [1/3]

```
bool quentier::utility::ApplicationSettings::contains (
    const char * key,
    int size = -1 ) const
```

Overload of contains accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key)

Returns

True if there exists a setting called key; false otherwise

5.3.3.11 contains() [2/3]

```
bool quentier::utility::ApplicationSettings::contains (
    const QString & key ) const
```

The call is redirected to QSettings::contains. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	The key being checked for presence
------------	------------------------------------

Returns

True if there exists a setting called key; false otherwise

5.3.3.12 contains() [3/3]

```
bool quentier::utility::ApplicationSettings::contains (
    std::string_view key ) const
```

Overload of contains accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
------------	--------------------------------------------------------------------------------------------------------------------------

Returns

True if there exists a setting called key; false otherwise

5.3.3.13 print()

```
QTextStream & quentier::utility::ApplicationSettings::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

5.3.3.14 remove() [1/3]

```
void quentier::utility::ApplicationSettings::remove (
    const char * key,
    int size = -1 )
```

Removes the setting key and any sub-settings of key. Overload of remove accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key).

5.3.3.15 remove() [2/3]

```
void quantier::utility::ApplicationSettings::remove (
    const QString & key )
```

Removes the setting key and any sub-settings of key. The call is redirected to QSettings::remove. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	String containing the setting name
------------	------------------------------------

5.3.3.16 remove() [3/3]

```
void quantier::utility::ApplicationSettings::remove (
    std::string_view key )
```

Removes the setting key and any sub-settings of key. Overload of remove accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
------------	--------------------------------------------------------------------------------------------------------------------------

5.3.3.17 setValue() [1/3]

```
void quantier::utility::ApplicationSettings::setValue (
    const char * key,
    const QVariant & value,
    int keySize = -1 )
```

Sets the value of setting. Overload of setValue accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>value</i>	Value for setting key
<i>keySize</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key).

5.3.3.18 setValue() [2/3]

```
void quantier::utility::ApplicationSettings::setValue (
    const QString & key,
    const QVariant & value )
```

Sets the value of setting. The call is redirected to QSettings::setValue. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	String containing the setting name
<i>value</i>	Value for setting key

5.3.3.19 setValue() [3/3]

```
void quantier::utility::ApplicationSettings::setValue (
    std::string_view key,
    const QVariant & value )
```

Sets the value of setting. Overload of setValue accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>value</i>	Value for setting key

5.3.3.20 value() [1/3]

```
QVariant quantier::utility::ApplicationSettings::value (
    const char * key,
    const QVariant & defaultValue = {},
    int keySize = -1 ) const
```

Fetches the value of setting. Overload of value accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>defaultValue</i>	Default value returned if the setting doesn't exist
<i>keySize</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key)

Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.3.3.21 value() [2/3]

```
QVariant quentier::utility::ApplicationSettings::value (
    const QString & key,
    const QVariant & defaultValue = {} ) const
```

Fetches the value of setting. The call is redirected to QSettings::value. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	String containing the setting name
<i>defaultValue</i>	Default value returned if the setting doesn't exist

Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.3.3.22 value() [3/3]

```
QVariant quentier::utility::ApplicationSettings::value (
    std::string_view key,
    const QVariant & defaultValue = {} ) const
```

Fetches the value of setting. Overload of value accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>defaultValue</i>	Default value returned if the setting doesn't exist

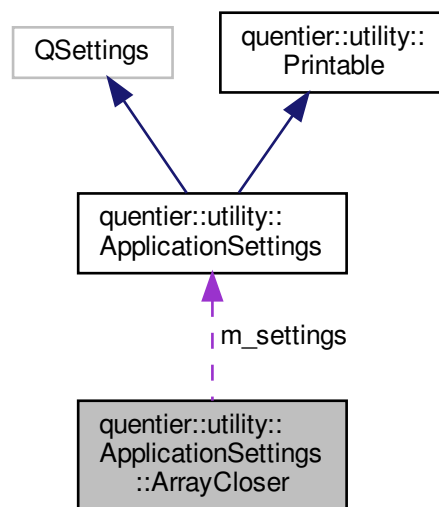
Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.4 quantier::utility::ApplicationSettings::ArrayCloser Struct Reference

```
#include <ApplicationSettings.h>
```

Collaboration diagram for quantier::utility::ApplicationSettings::ArrayCloser:

**Public Member Functions**

- **ArrayCloser** ([ApplicationSettings](#) &settings)

Public Attributes

- [ApplicationSettings](#) & **m_settings**

5.4.1 Detailed Description

Helper struct for RAII style of ensuring the array once began would be closed even if exception is thrown after beginning the array

5.5 `quentier::synchronization::AuthenticationExpiredError` Struct Reference

```
#include <Errors.h>
```

5.5.1 Detailed Description

Authentication expired error indicates that used authentication token has expired so authentication should be repeated before the next attempt to run synchronization.

5.6 `quentier::synchronization::ISyncConflictResolver::Conflict` Resolution Struct Reference

The [ConflictResolution](#) struct is a namespace inside which several other structs determining actual conflict resolutions.

```
#include <ISyncConflictResolver.h>
```

Classes

- struct [IgnoreMine](#)
The [IgnoreMine](#) conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".
- struct [MoveMine](#)
The [MoveMine](#) conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.
- struct [UseMine](#)
The [UseMine](#) conflict resolution means "override theirs version with mine version".
- struct [UseTheirs](#)
The [UseTheirs](#) conflict resolution means "override mine version with theirs version".

5.6.1 Detailed Description

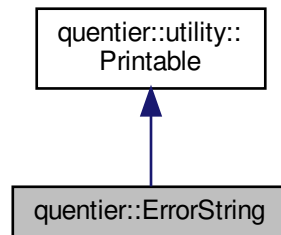
The [ConflictResolution](#) struct is a namespace inside which several other structs determining actual conflict resolutions.

5.7 quentier::ErrorString Class Reference

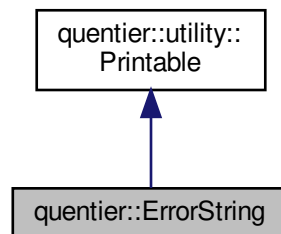
The [ErrorString](#) class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

```
#include <ErrorString.h>
```

Inheritance diagram for quentier::ErrorString:



Collaboration diagram for quentier::ErrorString:



Public Member Functions

- **ErrorString** (const char *error=nullptr)
- **ErrorString** (const QString &error)
- **ErrorString** (const [ErrorString](#) &other)
- **ErrorString** ([ErrorString](#) &&other) noexcept
- [ErrorString](#) & **operator=** (const [ErrorString](#) &other)
- [ErrorString](#) & **operator=** ([ErrorString](#) &&other) noexcept
- const QString & **base** () const noexcept
- QString & **base** ()
- const QStringList & **additionalBases** () const noexcept
- QStringList & **additionalBases** ()

- `const QString & details () const noexcept`
- `QString & details ()`
- `void setBase (QString error)`
- `void setBase (const char *error)`
- `void appendBase (const QString &error)`
- `void appendBase (const QStringList &errors)`
- `void appendBase (const char *error)`
- `void setDetails (const QString &error)`
- `void setDetails (const char *error)`
- `bool isEmpty () const`
- `void clear ()`
- `QString localizedString () const`
- `QString nonLocalizedString () const`
- `QTextStream & print (QTextStream &strm) const override`

5.7.1 Detailed Description

The [ErrorString](#) class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

1. `base()` methods return const and non-const links to the primary translatable string
2. `details()` methods return const and non-const links to non-translatable string (coming from some third party library etc)
3. `additionalBases()` methods return const and non-const links to additional translatable strings; one translatable string is not always enough because the error message might be composed from different parts

5.7.2 Member Function Documentation

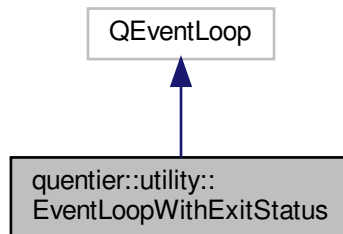
5.7.2.1 `print()`

```
QTextStream & quentier::ErrorString::print (
    QTextStream & strm ) const [override], [virtual]
```

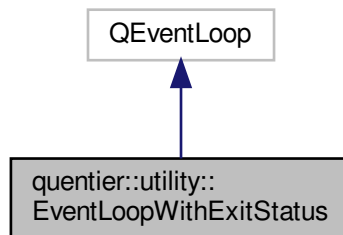
Implements [quentier::utility::Printable](#).

5.8 quantier::utility::EventLoopWithExitStatus Class Reference

Inheritance diagram for quantier::utility::EventLoopWithExitStatus:



Collaboration diagram for quantier::utility::EventLoopWithExitStatus:



Public Types

- enum class **ExitStatus** { **Success** , **Failure** , **Timeout** }

Public Slots

- void **exitAsSuccess** ()
- void **exitAsFailure** ()
- void **exitAsFailureWithError** (QString errorDescription)
- void **exitAsFailureWithErrorString** ([ErrorString](#) errorDescription)
- void **exitAsTimeout** ()

Public Member Functions

- **EventLoopWithExitStatus** (QObject *parent=nullptr)
- ExitStatus **exitStatus** () const
- const [ErrorString](#) & **errorDescription** () const

Friends

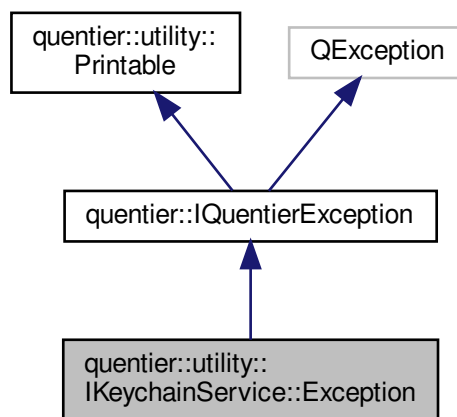
- QDebug & **operator**<< (QDebug &dbg, ExitStatus status)
- QTextStream & **operator**<< (QTextStream &strm, ExitStatus status)

5.9 quantier::utility::IKeychainService::Exception Class Reference

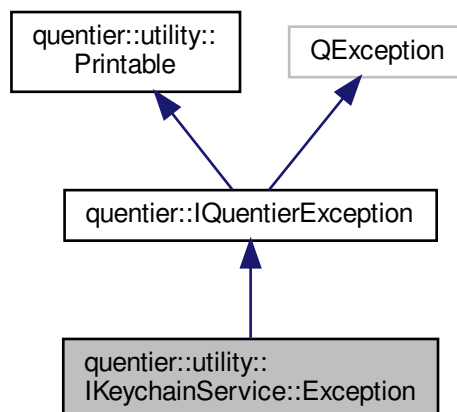
The [IKeychainService::Exception](#) class is the base class for exceptions returned inside QFutures from methods of [IKeychainService](#).

```
#include <IKeychainService.h>
```

Inheritance diagram for quantier::utility::IKeychainService::Exception:



Collaboration diagram for quantier::utility::IKeychainService::Exception:



Public Member Functions

- **Exception** ([ErrorCode](#) errorCode) noexcept
- **Exception** ([ErrorCode](#) errorCode, [ErrorString](#) errorDescription) noexcept
- [ErrorCode](#) **errorCode** () const noexcept
- QString [exceptionDisplayName](#) () const override
- void **raise** () const override
- [Exception](#) * **clone** () const override

Additional Inherited Members

5.9.1 Detailed Description

The [IKeychainService::Exception](#) class is the base class for exceptions returned inside QFutures from methods of [IKeychainService](#).

5.9.2 Member Function Documentation

5.9.2.1 exceptionDisplayName()

```
QString quantier::utility::IKeychainService::Exception::exceptionDisplayName ( ) const [override],
[virtual]
```

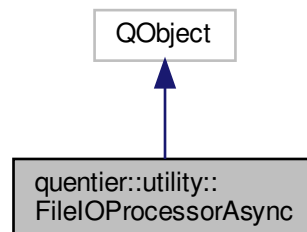
Implements [quantier::IQuantierException](#).

5.10 quantier::utility::FileIOProcessorAsync Class Reference

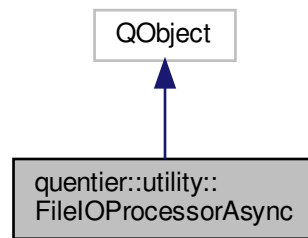
The [FileIOProcessorAsync](#) class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

```
#include <FileIOProcessorAsync.h>
```

Inheritance diagram for quantier::utility::FileIOProcessorAsync:



Collaboration diagram for `quentier::utility::FileIOProcessorAsync`:



Public Slots

- void [onWriteFileRequest](#) (QString absoluteFilePath, QByteArray data, QUuid requestId, bool append)
onWriteFileRequest slot processes file write requests with given request ids
- void [onReadFileRequest](#) (QString absoluteFilePath, QUuid requestId)
onReadFileRequest slot processes file read requests with given request ids

Signals

- void **readyForIO** ()
readyForIO signal is emitted when the queue for file IO is empty for some time (30 seconds by default, can also be configured via `setIdleTimePeriod` method) after the last IO event to signal listeners that they can perform some IO via the [FileIOProcessorAsync](#)
- void [writeFileRequestProcessed](#) (bool success, [ErrorString](#) errorDescription, QUuid requestId)
writeFileRequestProcessed signal is emitted when the file write request with given id is finished
- void [readFileRequestProcessed](#) (bool success, [ErrorString](#) errorDescription, QByteArray data, QUuid requestId)
readFileRequestProcessed signal is emitted when the file read request with given id is finished

Public Member Functions

- **FileIOProcessorAsync** (QObject *parent=nullptr)
- void [setIdleTimePeriod](#) (qint32 seconds)
setIdleTimePeriod sets time period defining the idle state of [FileIOProcessorAsync](#): once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

5.10.1 Detailed Description

The [FileIOProcessorAsync](#) class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

5.10.2 Member Function Documentation

5.10.2.1 onReadFileRequest

```
void quotientier::utility::FileIOProcessorAsync::onReadFileRequest (
    QString absoluteFilePath,
    QUuid requestId ) [slot]
```

onReadFileRequest slot processes file read requests with given request ids

Parameters

<i>absoluteFilePath</i>	Absolute file path to be read
<i>requestId</i>	Unique identifier of the file read request

5.10.2.2 onWriteFileRequest

```
void quotientier::utility::FileIOProcessorAsync::onWriteFileRequest (
    QString absoluteFilePath,
    QByteArray data,
    QUuid requestId,
    bool append ) [slot]
```

onWriteFileRequest slot processes file write requests with given request ids

Parameters

<i>absoluteFilePath</i>	Absolute file path to be written
<i>data</i>	Data to be written to the file
<i>requestId</i>	Unique identifier of the file write request
<i>append</i>	If true, the data would be appended to file, otherwise the entire file would be erased before with the data is written

5.10.2.3 readFileRequestProcessed

```
void quotientier::utility::FileIOProcessorAsync::readFileRequestProcessed (
    bool success,
    QString errorDescription,
    QByteArray data,
    QUuid requestId ) [signal]
```

readFileRequestProcessed signal is emitted when the file read request with given id is finished

Parameters

<i>success</i>	True if read operation was successful, false otherwise
<i>errorDescription</i>	Textual description of the error
<i>data</i>	Data read from file
<i>requestId</i>	Unique identifier of the file read request

5.10.2.4 setIdleTimePeriod()

```
void quentier::utility::FileIOProcessorAsync::setIdleTimePeriod (
    qint32 seconds )
```

`setIdleTimePeriod` sets time period defining the idle state of [FileIOProcessorAsync](#): once the time measured since the last IO operation is over the specified number of seconds, the class emits `readyForIO` signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

Parameters

<i>seconds</i>	Number of seconds for idle time period
----------------	----------------------------------------

5.10.2.5 writeFileRequestProcessed

```
void quentier::utility::FileIOProcessorAsync::writeFileRequestProcessed (
    bool success,
    ErrorString errorDescription,
    QUuid requestId ) [signal]
```

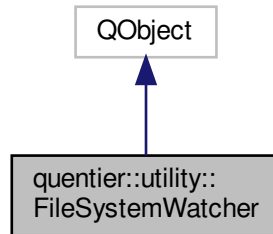
`writeFileRequestProcessed` signal is emitted when the file write request with given id is finished

Parameters

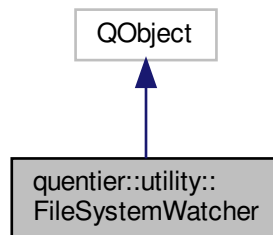
<i>success</i>	True if write operation was successful, false otherwise
<i>errorDescription</i>	Textual description of the error
<i>requestId</i>	Unique identifier of the file write request

5.11 quantier::utility::FileSystemWatcher Class Reference

Inheritance diagram for quantier::utility::FileSystemWatcher:



Collaboration diagram for quantier::utility::FileSystemWatcher:



Signals

- void **directoryChanged** (const QString &path)
- void **directoryRemoved** (const QString &path)
- void **fileChanged** (const QString &path)
- void **fileRemoved** (const QString &path)

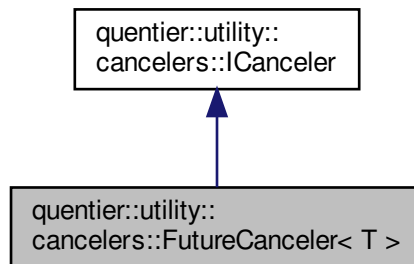
Public Member Functions

- **FileSystemWatcher** (int removalTimeoutMSec=500, QObject *parent=nullptr)
- **FileSystemWatcher** (const QStringList &paths, int removalTimeoutMSec=500, QObject *parent=nullptr)
- void **addPath** (const QString &path)
- void **addPaths** (const QStringList &paths)
- QStringList **directories** () const
- QStringList **files** () const
- void **removePath** (const QString &path)
- void **removePaths** (const QStringList &paths)

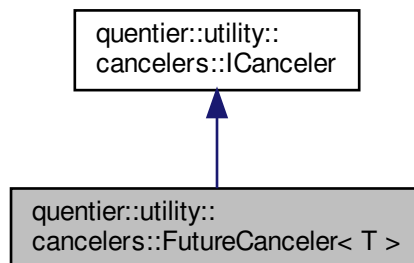
5.12 `quentier::utility::cancelers::FutureCanceler< T >` Class Template Reference

```
#include <FutureCanceler.h>
```

Inheritance diagram for `quentier::utility::cancelers::FutureCanceler< T >`:



Collaboration diagram for `quentier::utility::cancelers::FutureCanceler< T >`:



Public Member Functions

- **FutureCanceler** (`QFuture< T > future`)
- `bool isCanceled ()` `const noexcept` override

5.12.1 Detailed Description

```
template<class T>
class quentier::utility::cancelers::FutureCanceler< T >
```

[ICanceler](#) implementation which tracks the canceled status of a future.

5.12.2 Member Function Documentation

5.12.2.1 `isCanceled()`

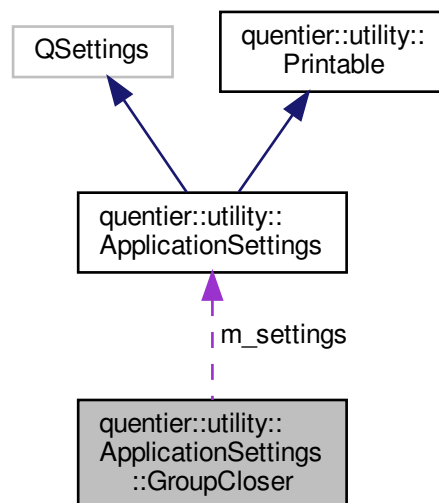
```
template<class T >
bool quentier::utility::cancelers::FutureCanceler< T >::isCanceled ( ) const [inline], [override],
[virtual], [noexcept]
```

Implements [quentier::utility::cancelers::ICanceler](#).

5.13 `quentier::utility::ApplicationSettings::GroupCloser` Struct Reference

```
#include <ApplicationSettings.h>
```

Collaboration diagram for `quentier::utility::ApplicationSettings::GroupCloser`:



Public Member Functions

- `GroupCloser` ([ApplicationSettings](#) &settings)

Public Attributes

- [ApplicationSettings](#) & `m_settings`

5.13.1 Detailed Description

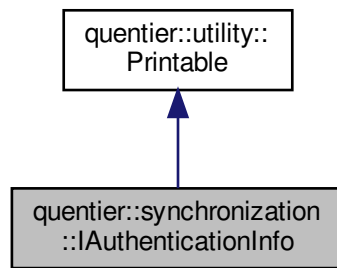
Helper struct for RAll style of ensuring the group once opened would be closed even if exception is thrown after beginning the group

5.14 `quentier::synchronization::IAuthenticationInfo` Class Reference

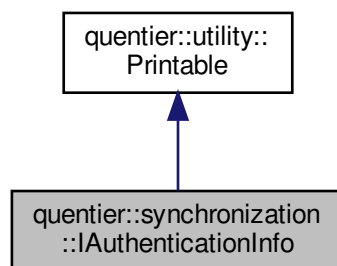
The [IAuthenticationInfo](#) interface represents the information obtained through OAuth and necessary to access Evernote API.

```
#include <IAuthenticationInfo.h>
```

Inheritance diagram for `quentier::synchronization::IAuthenticationInfo`:



Collaboration diagram for `quentier::synchronization::IAuthenticationInfo`:



Public Member Functions

- virtual qevercloud::UserID [userId](#) () const =0
- virtual QString [authToken](#) () const =0
- virtual qevercloud::Timestamp [authTokenExpirationTime](#) () const =0
- virtual qevercloud::Timestamp [authenticationTime](#) () const =0
- virtual QString [shardId](#) () const =0
- virtual QString [noteStoreUrl](#) () const =0
- virtual QString [webApiUrlPrefix](#) () const =0
- virtual QList< QNetworkCookie > [userStoreCookies](#) () const =0

5.14.1 Detailed Description

The [IAuthenticationInfo](#) interface represents the information obtained through OAuth and necessary to access Evernote API.

5.14.2 Member Function Documentation

5.14.2.1 [authenticationTime\(\)](#)

```
virtual qevercloud::Timestamp quantier::synchronization::IAuthenticationInfo::authenticationTime ( ) const [pure virtual]
```

Timestamp at which authentication info was received from Evernote

5.14.2.2 [authToken\(\)](#)

```
virtual QString quantier::synchronization::IAuthenticationInfo::authToken ( ) const [pure virtual]
```

Authentication token which needs to be used for access to Evernote API

5.14.2.3 [authTokenExpirationTime\(\)](#)

```
virtual qevercloud::Timestamp quantier::synchronization::IAuthenticationInfo::authTokenExpirationTime ( ) const [pure virtual]
```

Expiration timestamp for the authentication token

5.14.2.4 [noteStoreUrl\(\)](#)

```
virtual QString quantier::synchronization::IAuthenticationInfo::noteStoreUrl ( ) const [pure virtual]
```

Url of the note store service for this user

5.14.2.5 shardId()

```
virtual QString quentier::synchronization::IAuthenticationInfo::shardId ( ) const [pure virtual]
```

Shard identifier which needs to be used for access to Evernote API along with the authentication token

5.14.2.6 userId()

```
virtual qevercloud::UserID quentier::synchronization::IAuthenticationInfo::userId ( ) const [pure virtual]
```

Identifier of the authenticated user

5.14.2.7 userStoreCookies()

```
virtual QList< QNetworkCookie > quentier::synchronization::IAuthenticationInfo::userStore↵
Cookies ( ) const [pure virtual]
```

The list of network cookies received during OAuth procedure. Although is is not mentioned anywhere in Evernote docs, these cookies might have to be used for access to user store. See this discussion for reference: <https://discussion.evernote.com/forums/topic/124257-calls-to-userstore-from-evernote-api-stop>

5.14.2.8 webApiUrlPrefix()

```
virtual QString quentier::synchronization::IAuthenticationInfo::webApiUrlPrefix ( ) const [pure virtual]
```

Url prefix for Evernote Web API.

See also

qevercloud::PublicUserInfo::webApiUrlPrefix

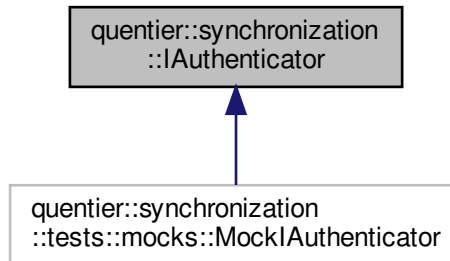
5.15 quentier::synchronization::IAuthenticationInfoBuilder Class Reference

Public Member Functions

- virtual [IAuthenticationInfoBuilder](#) & **setUserId** (qevercloud::UserID userId)=0
- virtual [IAuthenticationInfoBuilder](#) & **setAuthToken** (QString token)=0
- virtual [IAuthenticationInfoBuilder](#) & **setAuthTokenExpirationTime** (qevercloud::Timestamp expiration↵Time)=0
- virtual [IAuthenticationInfoBuilder](#) & **setAuthenticationTime** (qevercloud::Timestamp authenticationTime)=0
- virtual [IAuthenticationInfoBuilder](#) & **setShardId** (QString shardId)=0
- virtual [IAuthenticationInfoBuilder](#) & **setNoteStoreUrl** (QString noteStoreUrl)=0
- virtual [IAuthenticationInfoBuilder](#) & **setWebApiUrlPrefix** (QString webApiUrlPrefix)=0
- virtual [IAuthenticationInfoBuilder](#) & **setUserStoreCookies** (QList< QNetworkCookie > cookies)=0
- virtual [IAuthenticationInfoPtr](#) **build** ()=0

5.16 quantier::synchronization::IAuthenticator Class Reference

Inheritance diagram for quantier::synchronization::IAuthenticator:



Public Member Functions

- virtual QFuture< IAuthenticationInfoPtr > **authenticateNewAccount** ()=0
- virtual QFuture< IAuthenticationInfoPtr > **authenticateAccount** ([Account](#) account)=0

5.17 quantier::ResourceRecognitionIndexItem::IBarcodeItem Struct Reference

Public Member Functions

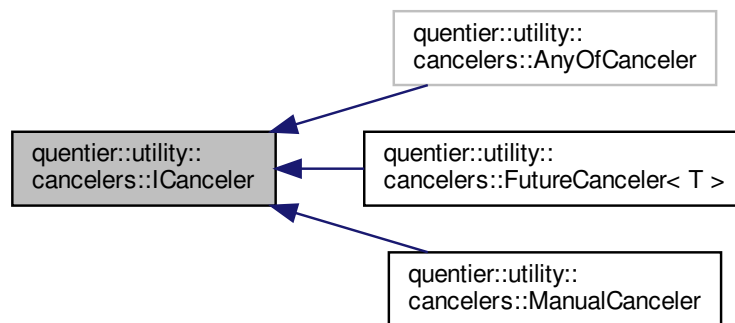
- virtual QString **barcode** () const =0
- virtual int **weight** () const =0

5.18 quantier::utility::cancelers::ICanceler Class Reference

The [ICanceler](#) interface provides isCanceled method which can be used to check whether some processing can be skipped because it was canceled.

```
#include <ICanceler.h>
```

Inheritance diagram for `quentier::utility::cancelers::ICanceler`:



Public Member Functions

- virtual bool **isCanceled** () const =0

5.18.1 Detailed Description

The [ICanceler](#) interface provides `isCanceled` method which can be used to check whether some processing can be skipped because it was canceled.

5.19 quentier::enml::IConverter Class Reference

The [IConverter](#) interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML.

```
#include <IConverter.h>
```

Public Types

- enum class [EnexExportTags](#) { **Yes** = 0 , **No** }

The `EnexExportTags` enum allows to specify whether export of note(s) to ENEX should include the names of note's tags or not.

Public Member Functions

- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertHtmlToEnml](#) (const [QString](#) &html, [IDecryptedTextCache](#) &decryptedTextCache, const [QList](#)< [conversion_rules::ISkipRulePtr](#) > &skipRules={}) const =0
- virtual [Result](#)< void, [ErrorString](#) > [convertHtmlToDoc](#) (const [QString](#) &html, [QTextDocument](#) &doc, const [QList](#)< [conversion_rules::ISkipRulePtr](#) > &skipRules={}) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertHtmlToXml](#) (const [QString](#) &html) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertHtmlToXhtml](#) (const [QString](#) &html) const =0
- virtual [Result](#)< [IHtmlDataPtr](#), [ErrorString](#) > [convertEnmlToHtml](#) (const [QString](#) &enml, [IDecryptedTextCache](#) &decryptedTextCache) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertEnmlToPlainText](#) (const [QString](#) &enml) const =0
- virtual [Result](#)< [QStringList](#), [ErrorString](#) > [convertEnmlToWordsList](#) (const [QString](#) &enml) const =0
- virtual [QStringList](#) [convertPlainTextToWordsList](#) (const [QString](#) &plainText) const =0
- virtual [Result](#)< void, [ErrorString](#) > [validateEnml](#) (const [QString](#) &enml) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [validateAndFixupEnml](#) (const [QString](#) &enml) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [exportNotesToEnex](#) (const [QList](#)< [qevercloud::Note](#) > ¬es, const [QHash](#)< [QString](#), [QString](#) > &tagNamesByTagLocalIds, [EnexExportTags](#) exportTagsOption, const [QString](#) &version={}) const =0
- virtual [Result](#)< [QList](#)< [qevercloud::Note](#) >, [ErrorString](#) > [importEnex](#) (const [QString](#) &enex) const =0

5.19.1 Detailed Description

The [IConverter](#) interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML.

5.19.2 Member Function Documentation

5.19.2.1 [convertEnmlToHtml\(\)](#)

```
virtual Result< IHtmlDataPtr, ErrorString > quotientier::enml::IConverter::convertEnmlToHtml (
    const QString & enml,
    IDecryptedTextCache & decryptedTextCache ) const    [pure virtual]
```

Converts ENML into HTML representation of note content

Parameters

<i>enml</i>	ENML representation of note content
<i>decryptedTextCache</i>	cache of decrypted text fragments

Returns

[Result](#) with HTML data in case of success or error string in case of failure

5.19.2.2 convertEnmlToPlainText()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertEnmlToPlainText (
    const QString & enml ) const [pure virtual]
```

Converts ENML into plain text representation of note content

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with plain text representation of note content in case of success or error string in case of failure

5.19.2.3 convertEnmlToWordsList()

```
virtual Result< QStringList, ErrorString > quantier::enml::IConverter::convertEnmlToWordsList (
    const QString & enml ) const [pure virtual]
```

Converts ENML into a list of words

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with list of words in case of success or error string in case of failure

5.19.2.4 convertHtmlToDoc()

```
virtual Result< void, ErrorString > quantier::enml::IConverter::convertHtmlToDoc (
    const QString & html,
    QTextDocument & doc,
    const QList< conversion_rules::ISkipRulePtr > & skipRules = {} ) const [pure
virtual]
```

Convert HTML representation of note content into QTextDocument

Parameters

<i>html</i>	HTML representation of note content
<i>doc</i>	QTextDocument into which the converted note content is put
<i>skipRules</i>	skip rules to be used during the conversion

Returns

Valid result in case of success or error string in case of failure

5.19.2.5 convertHtmlToEnml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertHtmlToEnml (
    const QString & html,
    IDecryptedTextCache & decryptedTextCache,
    const QList< conversion_rules::ISkipRulePtr > & skipRules = {} ) const [pure
virtual]
```

Converts HTML representation of note content into ENML

Parameters

<i>html</i>	HTML representation of note content
<i>decryptedTextCache</i>	cache of decrypted text fragments
<i>skipRules</i>	skip rules to be used during the conversion

Returns

[Result](#) with ENML in case of success or error string in case of failure

5.19.2.6 convertHtmlToXhtml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertHtmlToXhtml (
    const QString & html ) const [pure virtual]
```

Convert HTML representation of note content into a valid XHTML document

Parameters

<i>html</i>	HTML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with XHTML in case of success or error string in case of failure

5.19.2.7 convertHtmlToXml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertHtmlToXml (
    const QString & html ) const [pure virtual]
```

Convert HTML representation of note content into a valid XML document

Parameters

<i>html</i>	HTML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with XML in case of success or error string in case of failure

5.19.2.8 convertPlainTextToWordsList()

```
virtual QStringList quantier::enml::IConverter::convertPlainTextToWordsList (
    const QString & plainText ) const [pure virtual]
```

Converts plain text into a list of words

Parameters

<i>plainText</i>	plain text representation of note content
------------------	-------------------------------------------

Returns

list of words

5.19.2.9 exportNotesToEnex()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::exportNotesToEnex (
    const QList< qevercloud::Note > & notes,
    const QHash< QString, QString > & tagNamesByTagLocalIds,
    EnexExportTags exportTagsOption,
    const QString & version = {} ) const [pure virtual]
```

Exports a list of notes into ENEX

Parameters

<i>notes</i>	notes to be exported into ENEX
<i>tagNamesByTagLocalIds</i>	mapper from tag local ids into tag names
<i>exportTagsOption</i>	option controlling the export of tag names
<i>version</i>	optional version tag for ENEX, omitted if not set

Returns

[Result](#) with ENEX in case of success or error string in case of failure

5.19.2.10 importEnex()

```
virtual Result< QList< qevercloud::Note >, ErrorString > quantier::enml::IConverter::importEnex (
    const QString & enex ) const [pure virtual]
```

Import notes from ENEX

Parameters

<i>enex</i>	ENEX to be used for import
-------------	----------------------------

Returns

[Result](#) with list of notes in case of success or error string in case of failure

Note

if tag names are present in ENEX, corresponding notes would have their tagName field filled

5.19.2.11 validateAndFixupEnml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::validateAndFixupEnml (
    const QString & enml ) const [pure virtual]
```

Validates ENML and attempts to fix it automatically if it's not valid

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with either unchanged or fixed up ENML in case of success or error string in case of failure

5.19.2.12 validateEnml()

```
virtual Result< void, ErrorString > quantier::enml::IConverter::validateEnml (
    const QString & enml ) const [pure virtual]
```

Validates ENML against rules

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

valid [Result](#) in case of success or error string in case of failure

5.20 quantier::enml::IDecryptedTextCache Class Reference

Public Types

- enum class **RememberForSession** { **Yes** , **No** }

Public Member Functions

- virtual void **addDecryptexTextInfo** (const QString &encryptedText, const QString &decryptedText, const QString &passphrase, [utility::IEncryptor::Cipher](#) cipher, RememberForSession rememberForSession)=0
- virtual std::optional< std::pair< QString, RememberForSession > > **findDecryptedTextInfo** (const QString &encryptedText) const =0
- virtual std::optional< QString > **updateDecryptedTextInfo** (const QString &originalEncryptedText, const QString &newDecryptedText)=0
- virtual bool **containsRememberedForSessionEntries** () const =0
- virtual void **removeDecryptedTextInfo** (const QString &encryptedText)=0
- virtual void **clearNonRememberedForSessionEntries** ()=0

Friends

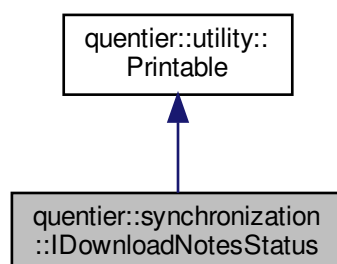
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, RememberForSession rememberForSession)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, RememberForSession rememberForSession)

5.21 quantier::synchronization::IDownloadNotesStatus Class Reference

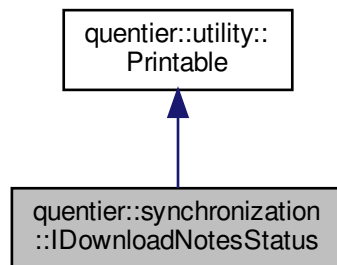
The [IDownloadNotesStatus](#) interface presents information about the status of notes downloading process.

```
#include <IDownloadNotesStatus.h>
```

Inheritance diagram for quantier::synchronization::IDownloadNotesStatus:



Collaboration diagram for quantier::synchronization::IDownloadNotesStatus:



Public Types

- using **QExceptionPtr** = `std::shared_ptr< QException >`
- using **NoteWithException** = `std::pair< qevercloud::Note, QExceptionPtr >`
- using **GuidWithException** = `std::pair< qevercloud::Guid, QExceptionPtr >`
- using **UpdateSequenceNumbersByGuid** = `QHash< qevercloud::Guid, qint32 >`

Public Member Functions

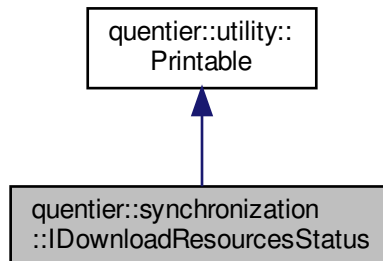
- virtual quint64 **totalNewNotes** () const =0
- virtual quint64 **totalUpdatedNotes** () const =0
- virtual quint64 **totalExpungedNotes** () const =0
- virtual QList< NoteWithException > **notesWhichFailedToDownload** () const =0
- virtual QList< NoteWithException > **notesWhichFailedToProcess** () const =0
- virtual QList< GuidWithException > **noteGuidsWhichFailedToExpunge** () const =0
- virtual UpdateSequenceNumbersByGuid **processedNoteGuidsAndUsns** () const =0
- virtual UpdateSequenceNumbersByGuid **cancelledNoteGuidsAndUsns** () const =0
- virtual QList< qevercloud::Guid > **expungedNoteGuids** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

5.21.1 Detailed Description

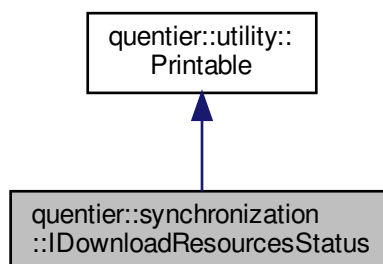
The [IDownloadNotesStatus](#) interface presents information about the status of notes downloading process.

5.22 quentier::synchronization::IDownloadResourcesStatus Class Reference

Inheritance diagram for quentier::synchronization::IDownloadResourcesStatus:



Collaboration diagram for quentier::synchronization::IDownloadResourcesStatus:



Public Types

- using **QExceptionPtr** = `std::shared_ptr< QException >`
- using **ResourceWithException** = `std::pair< qevercloud::Resource, QExceptionPtr >`
- using **UpdateSequenceNumbersByGuid** = `QHash< qevercloud::Guid, qint32 >`

Public Member Functions

- virtual quint64 **totalNewResources** () const =0
- virtual quint64 **totalUpdatedResources** () const =0
- virtual QList< ResourceWithException > **resourcesWhichFailedToDownload** () const =0
- virtual QList< ResourceWithException > **resourcesWhichFailedToProcess** () const =0
- virtual UpdateSequenceNumbersByGuid **processedResourceGuidsAndUsns** () const =0
- virtual UpdateSequenceNumbersByGuid **cancelledResourceGuidsAndUsns** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

5.23 quantier::utility::IEncryptor Struct Reference

The [IEncryptor](#) interface provides encryption and decryption functionality which is compatible with that used by Evernote service.

```
#include <IEncryptor.h>
```

Public Types

- enum class [Cipher](#) { [RC2](#) , [AES](#) }

Public Member Functions

- virtual [Result](#)< [QString](#), [ErrorString](#) > [encrypt](#) (const [QString](#) &text, const [QString](#) &passphrase)=0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [decrypt](#) (const [QString](#) &encryptedText, const [QString](#) &passphrase, [Cipher](#) cipher)=0

Friends

- QUENTIER_EXPORT [QDebug](#) & [operator](#)<< ([QDebug](#) &dbg, [Cipher](#) cipher)
- QUENTIER_EXPORT [QTextStream](#) & [operator](#)<< ([QTextStream](#) &strm, [Cipher](#) cipher)

5.23.1 Detailed Description

The [IEncryptor](#) interface provides encryption and decryption functionality which is compatible with that used by Evernote service.

5.23.2 Member Enumeration Documentation

5.23.2.1 Cipher

```
enum class quantier::utility::IEncryptor::Cipher [strong]
```

Cipher used for encryption/decryption

Enumerator

RC2	RC2 64 bit block cipher
AES	AES 128 bit block cipher

5.23.3 Member Function Documentation

5.23.3.1 decrypt()

```
virtual Result< QString, ErrorString > quantier::utility::IEncryptor::decrypt (
    const QString & encryptedText,
    const QString & passphrase,
    Cipher cipher ) [pure virtual]
```

Decrypt previously encrypted text fragment

Parameters

<i>encryptedText</i>	Encrypted text to decrypt
<i>passphrase</i>	Passhprase used to encrypt text
<i>cipher</i>	Cipher used to encrypt text

Returns

[Result](#) with either decrypted text or error message

5.23.3.2 encrypt()

```
virtual Result< QString, ErrorString > quantier::utility::IEncryptor::encrypt (
    const QString & text,
    const QString & passphrase ) [pure virtual]
```

Encrypt text fragment using AES cipher (RC2 cipher is only used for decryption)

Parameters

<i>text</i>	Text to encrypt
<i>passphrase</i>	Passphrase which can be used to decrypt the text

Returns

[Result](#) with either encrypted text or error message

5.24 quantier::enml::IENMLTagsConverter Class Reference

The [IENMLTagsConverter](#) interfaces provides methods which convert Evernote-specific markup tags such as encrypt, en-media etc. into their counterparts which should be used in the HTML representation of note content.

```
#include <IENMLTagsConverter.h>
```

Public Member Functions

- virtual QString [convertEnToDo](#) (bool checked, quint32 index) const =0
- virtual QString [convertEncryptedText](#) (const QString &encryptedText, const QString &hint, [utility::IEncryptor::Cipher](#) cipher, quint32 index) const =0
- virtual QString [convertDecryptedText](#) (const QString &decryptedText, const QString &encryptedText, const QString &hint, [utility::IEncryptor::Cipher](#) cipher, quint32 index) const =0
- virtual [Result](#)< QString, [ErrorString](#) > [convertResource](#) (const qevercloud::Resource &resource) const =0

5.24.1 Detailed Description

The [IENMLTagsConverter](#) interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content.

5.24.2 Member Function Documentation

5.24.2.1 convertDecryptedText()

```
virtual QString quantier::enml::IENMLTagsConverter::convertDecryptedText (
    const QString & decryptedText,
    const QString & encryptedText,
    const QString & hint,
    utility::IEncryptor::Cipher cipher,
    quint32 index ) const [pure virtual]
```

Converts already decrypted en-crypt tag into its HTML counterpart

Parameters

<i>decryptedText</i>	decrypted text from en-crypt tag
<i>encryptedText</i>	encrypted text contained within en-crypt tag
<i>hint</i>	hint to be displayed when user tries to decrypt the text
<i>cipher</i>	cipher used to ecrypt the text
<i>index</i>	index of particular en-crypt tag within the note content so that different en-crypt tags can be differentiated

Returns

HTML representation of decrypted en-crypt tag

5.24.2.2 convertEncryptedText()

```
virtual QString quantier::enml::IENMLTagsConverter::convertEncryptedText (
    const QString & encryptedText,
```

```
const QString & hint,
utility::IEncryptor::Cipher cipher,
quint32 index ) const [pure virtual]
```

Converts en-crypt tag into its HTML counterpart

Parameters

<i>encryptedText</i>	encrypted text contained within en-crypt tag
<i>hint</i>	hint to be displayed when user tries to decrypt the text
<i>cipher</i>	cipher used to encrypt the text
<i>index</i>	index of particular en-crypt tag within the note content so that different en-crypt tags can be differentiated

Returns

HTML representation of en-crypt tag

5.24.2.3 convertEnToDo()

```
virtual QString quentier::enml::IENMLTagsConverter::convertEnToDo (
    bool checked,
    quint32 index ) const [pure virtual]
```

Converts en-todo tag into its HTML counterpart

Parameters

<i>checked</i>	indicates whether this todo is checked or not
<i>index</i>	index of particular en-todo tag within the note content so that different todo tags can be differentiated

Returns

HTML representation of en-todo tag

5.24.2.4 convertResource()

```
virtual Result< QString, ErrorString > quentier::enml::IENMLTagsConverter::convertResource (
    const qevercloud::Resource & resource ) const [pure virtual]
```

Converts en-media tag representing a resource into its HTML counterpart

Parameters

<i>resource</i>	resource corresponding to en-media tag
-----------------	----------------------------------------

Returns

[Result](#) with valid HTML representing the resource/en-media tag in case of success or error string in case of failure

5.25 quantier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine Struct Reference

The [IgnoreMine](#) conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".

```
#include <ISyncConflictResolver.h>
```

5.25.1 Detailed Description

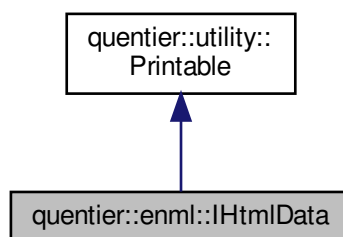
The [IgnoreMine](#) conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".

5.26 quantier::enml::IHtmlData Struct Reference

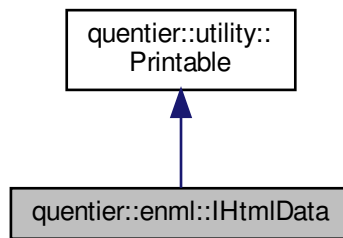
The [IHtmlData](#) represents the result of ENML to HTML conversion: HTML itself plus some metadata.

```
#include <IHtmlData.h>
```

Inheritance diagram for quantier::enml::IHtmlData:



Collaboration diagram for `quentier::enml::IHtmlData`:



Public Member Functions

- virtual QString [html](#) () const =0
- virtual quint32 [numEnToDoNodes](#) () const =0
- virtual quint32 [numHyperlinkNodes](#) () const =0
- virtual quint32 [numEnCryptNodes](#) () const =0
- virtual quint32 [numEnDecryptedNodes](#) () const =0
- QTextStream & [print](#) (QTextStream &strm) const override

5.26.1 Detailed Description

The [IHtmlData](#) represents the result of ENML to HTML conversion: HTML itself plus some metadata.

5.26.2 Member Function Documentation

5.26.2.1 [html\(\)](#)

```
virtual QString quentier::enml::IHtmlData::html ( ) const [pure virtual]
```

HTML representation of note content

5.26.2.2 [numEnCryptNodes\(\)](#)

```
virtual quint32 quentier::enml::IHtmlData::numEnCryptNodes ( ) const [pure virtual]
```

Number of en-crypt nodes within note HTML

5.26.2.3 numEnDecryptedNodes()

```
virtual quint32 quantier::enml::IHtmlData::numEnDecryptedNodes ( ) const [pure virtual]
```

Number of decrypted en-crypt nodes within note HTML

5.26.2.4 numEnToDoNodes()

```
virtual quint32 quantier::enml::IHtmlData::numEnToDoNodes ( ) const [pure virtual]
```

Number of ToDo nodes within note HTML

5.26.2.5 numHyperlinkNodes()

```
virtual quint32 quantier::enml::IHtmlData::numHyperlinkNodes ( ) const [pure virtual]
```

Number of hyperlink nodes within note HTML

5.26.2.6 print()

```
QTextStream & quantier::enml::IHtmlData::print (
    QTextStream & strm ) const [override], [virtual]
```

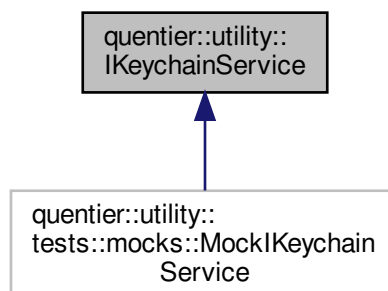
Implements [quantier::utility::Printable](#).

5.27 quantier::utility::IKeychainService Class Reference

The [IKeychainService](#) interface provides the ability to interact with the storage of sensitive data - read, write and delete it.

```
#include <IKeychainService.h>
```

Inheritance diagram for quantier::utility::IKeychainService:



Classes

- class [Exception](#)

The [IKeychainService::Exception](#) class is the base class for exceptions returned inside QFutures from methods of [IKeychainService](#).

Public Types

- enum class [ErrorCode](#) {
[NoError](#) , [EntryNotFound](#) , [CouldNotDeleteEntry](#) , [AccessDeniedByUser](#) ,
[AccessDenied](#) , [NoBackendAvailable](#) , [NotImplemented](#) , [OtherError](#) }

Public Member Functions

- virtual QFuture< void > [writePassword](#) (QString service, QString key, QString password)=0
- virtual QFuture< QString > [readPassword](#) (QString service, QString key) const =0
- virtual QFuture< void > [deletePassword](#) (QString service, QString key)=0

Friends

- QUINTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, [ErrorCode](#) errorCode)
- QUINTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, [ErrorCode](#) errorCode)

5.27.1 Detailed Description

The [IKeychainService](#) interface provides the ability to interact with the storage of sensitive data - read, write and delete it.

5.27.2 Member Enumeration Documentation

5.27.2.1 ErrorCode

```
enum class quentier::utility::IKeychainService::ErrorCode [strong]
```

Error codes for results of operations with the keychain service

Enumerator

NoError	No error occurred, operation was successful
EntryNotFound	For the given key no data was found
CouldNotDeleteEntry	Could not delete existing secret data
AccessDeniedByUser	User denied access to keychain
AccessDenied	Access denied for some reason
NoBackendAvailable	No platform-specific keychain service available
NotImplemented	Not implemented on platform
OtherError	Something else went wrong, the error description specifies what

5.27.3 Member Function Documentation

5.27.3.1 deletePassword()

```
virtual QFuture< void > quantier::utility::IKeychainService::deletePassword (
    QString service,
    QString key ) [pure virtual]
```

deletePassword potentially asynchronously deletes password from the keychain.

Parameters

<i>service</i>	Name of service within the keychain
<i>key</i>	Key under which the password is stored

Returns

Future which becomes finished when the operation is complete. If the operation fails, the future would contain an exception.

5.27.3.2 readPassword()

```
virtual QFuture< QString > quantier::utility::IKeychainService::readPassword (
    QString service,
    QString key ) const [pure virtual]
```

readPassword method potentially asynchronously reads password from the keychain.

Parameters

<i>service</i>	Name of service within the keychain
<i>key</i>	Key under which the password is stored

Returns

Future which becomes finished when the operation is complete. The value inside the future would be the read password. If the operation fails, the future would contain an exception.

5.27.3.3 writePassword()

```
virtual QFuture< void > quantier::utility::IKeychainService::writePassword (
    QString service,
```

```
QString key,
QString password ) [pure virtual]
```

writePassword method potentially asynchronously writes password to the keychain.

Parameters

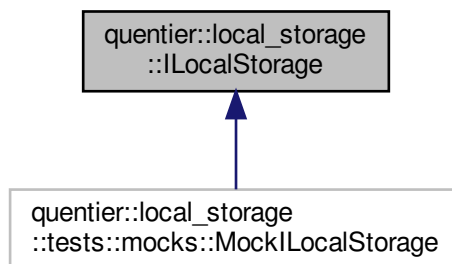
<i>service</i>	Name of service within the keychain
<i>key</i>	Key to store the password under
<i>password</i>	Password to store in the keychain

Returns

Future which becomes finished when the operation is complete. If the operation fails, the future would contain an exception.

5.28 quantier::local_storage::ILocalStorage Class Reference

Inheritance diagram for quantier::local_storage::ILocalStorage:



Classes

- struct [ListGuidsFilters](#)
- struct [ListLinkedNotebooksOptions](#)
- struct [ListNotebooksOptions](#)
- struct [ListNotesOptions](#)
- struct [ListObjectsFilters](#)
- struct [ListOptionsBase](#)
- struct [ListSavedSearchesOptions](#)
- struct [ListTagsOptions](#)

Public Types

- enum class **StartupOption** { **ClearDatabase** = 1 << 1 , **OverrideLock** = 1 << 2 }
- enum class **ListObjectsFilter** { **Include** , **Exclude** }
- enum class **OrderDirection** { **Ascending** , **Descending** }
- enum class **ListNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByNotebookName** , **ByCreationTimestamp** , **ByModificationTimestamp** }
- enum class **ListLinkedNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByShareName** , **ByUsername** }
- enum class **ListTagsOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** }
- enum class **ListNotesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByTitle** , **ByCreationTimestamp** , **ByModificationTimestamp** , **ByDeletionTimestamp** , **ByAuthor** , **BySource** , **BySourceApplication** , **ByReminderTime** , **ByPlaceName** }
- enum class **ListSavedSearchesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** , **ByFormat** }
- enum class **Affiliation** { **Any** , **User** , **AnyLinkedNotebook** , **ParticularLinkedNotebooks** }
- enum class **TagNotesRelation** { **Any** , **WithNotes** , **WithoutNotes** }
- enum class **NoteCountOption** { **IncludeNonDeletedNotes** = 1 << 1 , **IncludeDeletedNotes** = 1 << 2 }
- enum class **UpdateNoteOption** { **UpdateResourceMetadata** = 1 << 1 , **UpdateResourceBinaryData** = 1 << 2 , **UpdateTags** = 1 << 3 }
- enum class **FetchNoteOption** { **WithResourceMetadata** = 1 << 1 , **WithResourceBinaryData** = 1 << 2 }
- enum class **FetchResourceOption** { **WithBinaryData** = 1 << 1 }
- enum class **HighestUsnOption** { **WithinUserOwnContent** , **WithinUserOwnContentAndLinkedNotebooks** }

Public Member Functions

- **Q_DECLARE_FLAGS** (StartupOptions, StartupOption)
- virtual QFuture< bool > **isVersionTooHigh** () const =0
- virtual QFuture< bool > **requiresUpgrade** () const =0
- virtual QFuture< QList< IPatchPtr > > **requiredPatches** () const =0
- virtual QFuture< qint32 > **version** () const =0
- virtual QFuture< qint32 > **highestSupportedVersion** () const =0
- virtual QFuture< quint32 > **userCount** () const =0
- virtual QFuture< void > **putUser** (qevercloud::User user)=0
- virtual QFuture< std::optional< qevercloud::User > > **findUserById** (qevercloud::UserID userId) const =0
- virtual QFuture< void > **expungeUserById** (qevercloud::UserID userId)=0
- virtual QFuture< quint32 > **notebookCount** () const =0
- virtual QFuture< void > **putNotebook** (qevercloud::Notebook notebook)=0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByLocalId** (QString notebookLocalId) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByName** (QString notebookName, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findDefaultNotebook** () const =0
- virtual QFuture< void > **expungeNotebookByLocalId** (QString notebookLocalId)=0
- virtual QFuture< void > **expungeNotebookByGuid** (qevercloud::Guid notebookGuid)=0
- virtual QFuture< void > **expungeNotebookByName** (QString name, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt)=0
- virtual QFuture< QList< qevercloud::Notebook > > **listNotebooks** ([ListNotebooksOptions](#) options={}) const =0
- virtual QFuture< QList< qevercloud::SharedNotebook > > **listSharedNotebooks** (qevercloud::Guid notebookGuid={}) const =0

- virtual QFuture< QSet< qevercloud::Guid > > **listNotebookGuids** ([ListGuidsFilters](#) filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< quint32 > **linkedNotebookCount** () const =0
- virtual QFuture< void > **putLinkedNotebook** (qevercloud::LinkedNotebook linkedNotebook)=0
- virtual QFuture< std::optional< qevercloud::LinkedNotebook > > **findLinkedNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< void > **expungeLinkedNotebookByGuid** (qevercloud::Guid guid)=0
- virtual QFuture< QList< qevercloud::LinkedNotebook > > **listLinkedNotebooks** ([ListLinkedNotebooksOptions](#) options={}) const =0
- virtual QFuture< quint32 > **noteCount** (NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerNotebookLocalId** (QString notebookLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerTagLocalId** (QString tagLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< QHash< QString, quint32 > > **noteCountsPerTags** ([ListTagsOptions](#) listTagsOptions={}, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerNotebookAndTagLocalIds** (QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< void > **putNote** (qevercloud::Note note)=0
- virtual QFuture< void > **updateNote** (qevercloud::Note note, UpdateNoteOptions options)=0
- virtual QFuture< std::optional< qevercloud::Note > > **findNoteByLocalId** (QString noteLocalId, FetchNoteOptions options) const =0
- virtual QFuture< std::optional< qevercloud::Note > > **findNoteByGuid** (qevercloud::Guid noteGuid, FetchNoteOptions options) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotes** (FetchNoteOptions fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesPerNotebookLocalId** (QString notebookLocalId, FetchNoteOptions fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesPerTagLocalId** (QString tagLocalId, FetchNoteOptions fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesPerNotebookAndTagLocalIds** (QStringList notebookLocalIds, QStringList tagLocalIds, FetchNoteOptions fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesByLocalIds** (QStringList noteLocalIds, FetchNoteOptions fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listNoteGuids** ([ListGuidsFilters](#) filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **queryNotes** ([NoteSearchQuery](#) query, FetchNoteOptions fetchOptions) const =0
- virtual QFuture< QStringList > **queryNoteLocalIds** ([NoteSearchQuery](#) query) const =0
- virtual QFuture< void > **expungeNoteByLocalId** (QString noteLocalId)=0
- virtual QFuture< void > **expungeNoteByGuid** (qevercloud::Guid noteGuid)=0
- virtual QFuture< quint32 > **tagCount** () const =0
- virtual QFuture< void > **putTag** (qevercloud::Tag tag)=0
- virtual QFuture< std::optional< qevercloud::Tag > > **findTagByLocalId** (QString tagLocalId) const =0
- virtual QFuture< std::optional< qevercloud::Tag > > **findTagByGuid** (qevercloud::Guid tagGuid) const =0
- virtual QFuture< std::optional< qevercloud::Tag > > **findTagByName** (QString tagName, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt) const =0
- virtual QFuture< QList< qevercloud::Tag > > **listTags** ([ListTagsOptions](#) options={}) const =0
- virtual QFuture< QList< qevercloud::Tag > > **listTagsPerNoteLocalId** (QString noteLocalId, [ListTagsOptions](#) options={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listTagGuids** ([ListGuidsFilters](#) filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< void > **expungeTagByLocalId** (QString tagLocalId)=0

- virtual QFuture< void > **expungeTagByGuid** (qevercloud::Guid tagGuid)=0
- virtual QFuture< void > **expungeTagByName** (QString name, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt)=0
- virtual QFuture< quint32 > **resourceCount** (NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **resourceCountPerNoteLocalId** (QString noteLocalId) const =0
- virtual QFuture< void > **putResource** (qevercloud::Resource resource)=0
- virtual QFuture< std::optional< qevercloud::Resource > > **findResourceByLocalId** (QString resourceLocalId, FetchResourceOptions options={}) const =0
- virtual QFuture< std::optional< qevercloud::Resource > > **findResourceByGuid** (qevercloud::Guid resourceGuid, FetchResourceOptions options={}) const =0
- virtual QFuture< void > **expungeResourceByLocalId** (QString resourceLocalId)=0
- virtual QFuture< void > **expungeResourceByGuid** (qevercloud::Guid resourceGuid)=0
- virtual QFuture< quint32 > **savedSearchCount** () const =0
- virtual QFuture< void > **putSavedSearch** (qevercloud::SavedSearch search)=0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByLocalId** (QString savedSearchLocalId) const =0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByName** (QString name) const =0
- virtual QFuture< QList< qevercloud::SavedSearch > > **listSavedSearches** ([ListSavedSearchesOptions](#) options={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listSavedSearchGuids** ([ListGuidsFilters](#) filters) const =0
- virtual QFuture< void > **expungeSavedSearchByLocalId** (QString savedSearchLocalId)=0
- virtual QFuture< void > **expungeSavedSearchByGuid** (qevercloud::Guid guid)=0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (HighestUsnOption option) const =0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (qevercloud::Guid linkedNotebookGuid) const =0
- virtual [ILocalStorageNotifier](#) * **notifier** () const =0

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, StartupOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, StartupOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, StartupOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, StartupOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListObjectsFilter filter)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListObjectsFilter filter)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListObjectsFilters](#) &filters)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListObjectsFilters](#) &filters)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListGuidsFilters](#) &filters)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListGuidsFilters](#) &filters)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, OrderDirection orderDirection)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, OrderDirection orderDirection)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListNotebooksOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListNotebooksOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListLinkedNotebooksOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListLinkedNotebooksOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListTagsOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListTagsOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListNotesOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListNotesOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListSavedSearchesOrder order)

- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListSavedSearchesOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, [Affiliation](#) affiliation)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, [Affiliation](#) affiliation)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListNotebooksOptions](#) &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListNotebooksOptions](#) &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListLinkedNotebooksOptions](#) &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListLinkedNotebooksOptions](#) &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListSavedSearchesOptions](#) &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListSavedSearchesOptions](#) &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListNotesOptions](#) &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListNotesOptions](#) &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListTagsOptions](#) &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListTagsOptions](#) &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, NoteCountOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, NoteCountOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, NoteCountOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, NoteCountOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, UpdateNoteOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, UpdateNoteOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, UpdateNoteOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, UpdateNoteOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchNoteOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchNoteOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchNoteOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchNoteOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchResourceOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchResourceOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchResourceOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchResourceOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, HighestUsnOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, HighestUsnOption option)

5.28.1 Member Enumeration Documentation

5.28.1.1 Affiliation

```
enum class quentier::local\_storage::ILocalStorage::Affiliation [strong]
```

Denotes whether some data item belongs to user's own account, any of linked notebooks or particular linked notebooks

5.28.1.2 TagNotesRelation

```
enum class quentier::local\_storage::ILocalStorage::TagNotesRelation [strong]
```

Denotes the relation between tag and notes - whether any note is using the given tag

Enumerator

Any	The tag might be used by some notes or it might not be.
WithNotes	The tag is used by some notes.
WithoutNotes	The tag is not used by any note.

5.28.2 Member Function Documentation

5.28.2.1 notifier()

```
virtual ILocalStorageNotifier * quantier::local_storage::ILocalStorage::notifier ( ) const
[pure virtual]
```

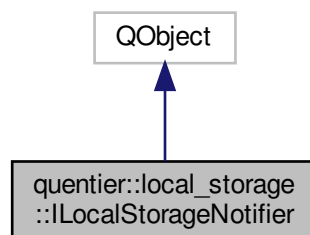
Notifications about the events occurring with the local storage are done via signals emitted by [ILocalStorageNotifier](#).

Returns

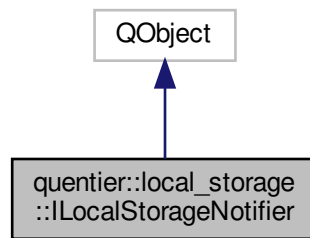
the pointer to [ILocalStorageNotifier](#) object which would be valid for at least as long as [ILocalStorage](#) object that returned it is alive.

5.29 quantier::local_storage::ILocalStorageNotifier Class Reference

Inheritance diagram for quantier::local_storage::ILocalStorageNotifier:



Collaboration diagram for `quentier::local_storage::ILocalStorageNotifier`:



Signals

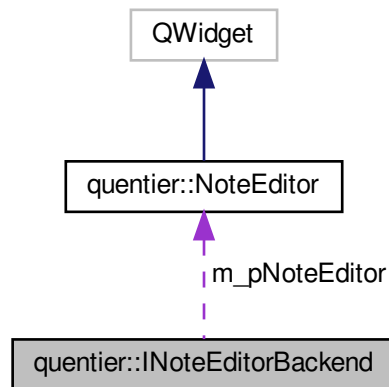
- void **userPut** (`qevercloud::User user`)
- void **userExpunged** (`qevercloud::UserID userId`)
- void **notebookPut** (`qevercloud::Notebook notebook`)
- void **notebookExpunged** (`QString notebookLocalId`)
- void **linkedNotebookPut** (`qevercloud::LinkedNotebook linkedNotebook`)
- void **linkedNotebookExpunged** (`qevercloud::Guid linkedNotebookGuid`)
- void **notePut** (`qevercloud::Note note`)
- void **noteUpdated** (`qevercloud::Note note`, `ILocalStorage::UpdateNoteOptions options`)
- void **noteExpunged** (`QString noteLocalId`)
- void **tagPut** (`qevercloud::Tag tag`)
- void **tagExpunged** (`QString tagLocalId`, `QStringList expungedChildTagLocalIds`)
- void **resourcePut** (`qevercloud::Resource resource`)
- void **resourceMetadataPut** (`qevercloud::Resource resource`)
- void **resourceExpunged** (`QString resourceLocalId`)
- void **savedSearchPut** (`qevercloud::SavedSearch savedSearch`)
- void **savedSearchExpunged** (`QString savedSearchLocalId`)

Protected Member Functions

- **ILocalStorageNotifier** (`QObject *parent=nullptr`)

5.30 quantier::INoteEditorBackend Class Reference

Collaboration diagram for quantier::INoteEditorBackend:



Public Types

- enum class **Rotation** { **Clockwise** , **Counterclockwise** }

Public Member Functions

- virtual void **initialize** (local_storage::ILocalStoragePtr localStorage, [SpellChecker](#) &spellChecker, const [Account](#) &account, QThread *backgroundJobsThread, enml::IDecryptedTextCachePtr decryptedTextCache)=0
- virtual QObject * **object** ()=0
- virtual QWidget * **widget** ()=0
- virtual void **setAccount** (const [Account](#) &account)=0
- virtual void **setUndoStack** (QUndoStack *pUndoStack)=0
- virtual void **setInitialPageHtml** (const QString &html)=0
- virtual void **setNoteNotFoundPageHtml** (const QString &html)=0
- virtual void **setNoteDeletedPageHtml** (const QString &html)=0
- virtual void **setNoteLoadingPageHtml** (const QString &html)=0
- virtual bool **isNoteLoaded** () const =0
- virtual qint64 **idleTime** () const =0
- virtual void **convertToNote** ()=0
- virtual void **saveNoteToLocalStorage** ()=0
- virtual void **setNoteTitle** (const QString ¬eTitle)=0
- virtual void **setTagIds** (const QStringList &tagLocalUids, const QStringList &tagGuids)=0
- virtual void **undo** ()=0
- virtual void **redo** ()=0
- virtual void **cut** ()=0
- virtual void **copy** ()=0
- virtual void **paste** ()=0
- virtual void **pasteUnformatted** ()=0

- virtual void **selectAll** ()=0
- virtual void **formatSelectionAsSourceCode** ()=0
- virtual void **fontMenu** ()=0
- virtual void **textBold** ()=0
- virtual void **textItalic** ()=0
- virtual void **textUnderline** ()=0
- virtual void **textStrikethrough** ()=0
- virtual void **textHighlight** ()=0
- virtual void **alignLeft** ()=0
- virtual void **alignCenter** ()=0
- virtual void **alignRight** ()=0
- virtual void **alignFull** ()=0
- virtual QString **selectedText** () const =0
- virtual bool **hasSelection** () const =0
- virtual void **findNext** (const QString &text, bool matchCase) const =0
- virtual void **findPrevious** (const QString &text, bool matchCase) const =0
- virtual void **replace** (const QString &textToReplace, const QString &replacementText, bool matchCase)=0
- virtual void **replaceAll** (const QString &textToReplace, const QString &replacementText, bool matchCase)=0
- virtual void **insertToDoCheckbox** ()=0
- virtual void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString ¬eGuid, const QString &linkText)=0
- virtual void **setSpellcheck** (bool enabled)=0
- virtual bool **spellCheckEnabled** () const =0
- virtual void **setFont** (const QFont &font)=0
- virtual void **setFontHeight** (int height)=0
- virtual void **setFontColor** (const QColor &color)=0
- virtual void **setBackgroundColor** (const QColor &color)=0
- virtual QPalette **defaultPalette** () const =0
- virtual void **setDefaultPalette** (const QPalette &pal)=0
- virtual const QFont * **defaultFont** () const =0
- virtual void **setDefaultFont** (const QFont &font)=0
- virtual void **insertHorizontalLine** ()=0
- virtual void **increaseFontSize** ()=0
- virtual void **decreaseFontSize** ()=0
- virtual void **increaseIndentation** ()=0
- virtual void **decreaseIndentation** ()=0
- virtual void **insertBulletedList** ()=0
- virtual void **insertNumberedList** ()=0
- virtual void **insertTableDialog** ()=0
- virtual void **insertFixedWidthTable** (int rows, int columns, int widthInPixels)=0
- virtual void **insertRelativeWidthTable** (int rows, int columns, double relativeWidth)=0
- virtual void **insertTableRow** ()=0
- virtual void **insertTableColumn** ()=0
- virtual void **removeTableRow** ()=0
- virtual void **removeTableColumn** ()=0
- virtual void **addAttachmentDialog** ()=0
- virtual void **saveAttachmentDialog** (const QByteArray &resourceHash)=0
- virtual void **saveAttachmentUnderCursor** ()=0
- virtual void **openAttachment** (const QByteArray &resourceHash)=0
- virtual void **openAttachmentUnderCursor** ()=0
- virtual void **copyAttachment** (const QByteArray &resourceHash)=0
- virtual void **copyAttachmentUnderCursor** ()=0
- virtual void **removeAttachment** (const QByteArray &resourceHash)=0
- virtual void **removeAttachmentUnderCursor** ()=0
- virtual void **renameAttachment** (const QByteArray &resourceHash)=0

- virtual void **renameAttachmentUnderCursor** ()=0
- virtual void **rotateImageAttachment** (const QByteArray &resourceHash, Rotation rotationDirection)=0
- virtual void **rotateImageAttachmentUnderCursor** (Rotation rotationDirection)=0
- virtual void **encryptSelectedText** ()=0
- virtual void **decryptEncryptedTextUnderCursor** ()=0
- virtual void **decryptEncryptedText** (QString encryptedText, QString cipher, QString hint, QString encryptIndex)=0
- virtual void **hideDecryptedTextUnderCursor** ()=0
- virtual void **hideDecryptedText** (QString encryptedText, QString decryptedText, QString cipher, QString hint, QString encryptIndex)=0
- virtual void **editHyperlinkDialog** ()=0
- virtual void **copyHyperlink** ()=0
- virtual void **removeHyperlink** ()=0
- virtual void **onNoteLoadCancelled** ()=0
- virtual bool **print** (QPrinter &printer, [ErrorString](#) &errorDescription)=0
- virtual bool **exportToPdf** (const QString &absoluteFilePath, [ErrorString](#) &errorDescription)=0
- virtual bool **exportToEnex** (const QStringList &tagNames, QString &enex, [ErrorString](#) &errorDescription)=0
- virtual QString **currentNoteLocalId** () const =0
- virtual void **setCurrentNoteLocalId** (const QString ¬eLocalUid)=0
- virtual void **clear** ()=0
- virtual bool **isModified** () const =0
- virtual bool **isEditorPageModified** () const =0
- virtual void **setFocusToEditor** ()=0

Protected Member Functions

- **INoteEditorBackend** ([NoteEditor](#) *parent)

Protected Attributes

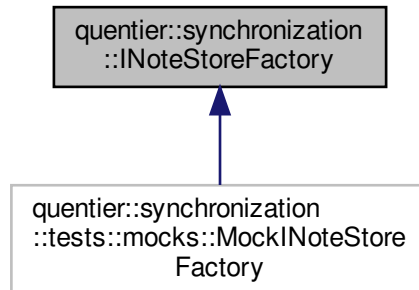
- [NoteEditor](#) * **m_pNoteEditor**

Friends

- QUENTIER_EXPORT QTextStream & **operator<<** (QTextStream &strm, Rotation rotation)
- QUENTIER_EXPORT QDebug & **operator<<** (QDebug &dbg, Rotation rotation)

5.31 quantier::synchronization::INoteStoreFactory Class Reference

Inheritance diagram for quantier::synchronization::INoteStoreFactory:

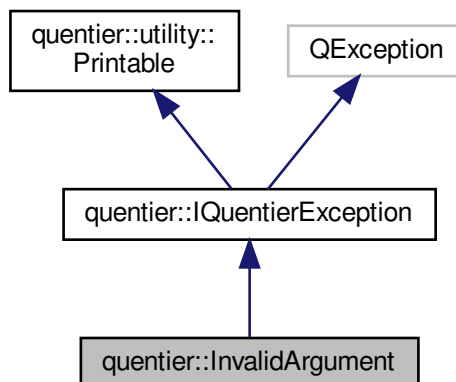


Public Member Functions

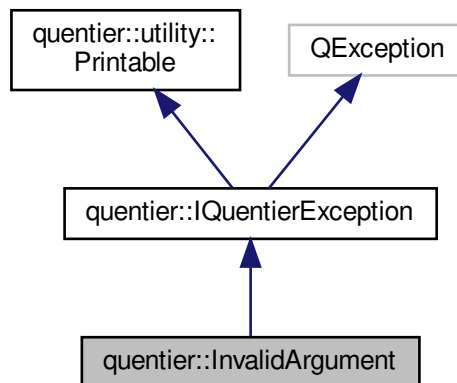
- virtual qevercloud::INoteStorePtr **createNoteStore** (QString noteStoreUrl={}, std::optional< qevercloud::↵ Guid > linkedNotebookGuid={}, qevercloud::IRequestContextPtr ctx={}, qevercloud::IRetryPolicyPtr retry↵ Policy={})=0

5.32 quantier::InvalidArgument Class Reference

Inheritance diagram for quantier::InvalidArgument:



Collaboration diagram for `quentier::InvalidArgument`:



Public Member Functions

- `InvalidArgument` (`ErrorString` message)
- `InvalidArgument * clone` () const override
- `void raise` () const override

Protected Member Functions

- `QString exceptionDisplayName` () const override

5.32.1 Member Function Documentation

5.32.1.1 `exceptionDisplayName()`

```
QString quentier::InvalidArgument::exceptionDisplayName ( ) const [override], [protected], [virtual]
```

Implements `quentier::IQuentierException`.

5.33 `quentier::ResourceRecognitionIndexItem::IObjectItem` Struct Reference

Public Member Functions

- virtual `QString objectType` () const =0
- virtual `int weight` () const =0

5.34 quotient::local_storage::IPatch Class Reference

The [IPatch](#) interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it.

```
#include <IPatch.h>
```

Public Member Functions

- virtual int [fromVersion](#) () const noexcept=0
- virtual int [toVersion](#) () const noexcept=0
- virtual QString [patchShortDescription](#) () const =0
- virtual QString [patchLongDescription](#) () const =0
- virtual QFuture< void > [backupLocalStorage](#) ()=0
- virtual QFuture< void > [restoreLocalStorageFromBackup](#) ()=0
- virtual QFuture< void > [removeLocalStorageBackup](#) ()=0
- virtual QFuture< void > [apply](#) ()=0

5.34.1 Detailed Description

The [IPatch](#) interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it.

5.34.2 Member Function Documentation

5.34.2.1 [apply\(\)](#)

```
virtual QFuture< void > quotient::local_storage::IPatch::apply ( ) [pure virtual]
```

Apply the patch to local storage

Returns

Future which can be awaited for patch application. Contains exception if patch application fails.

5.34.2.2 [backupLocalStorage\(\)](#)

```
virtual QFuture< void > quotient::local_storage::IPatch::backupLocalStorage ( ) [pure virtual]
```

Backup either the entire local storage or its parts affected by the particular patch, should be called before applying the patch (but can be skipped if not desired).

Returns

Future which can be awaited for the backup completion. Contains exception if backup fails.

5.34.2.3 fromVersion()

```
virtual int quantier::local_storage::IPatch::fromVersion ( ) const [pure virtual], [noexcept]
```

Returns

Version of local storage to which the patch needs to be applied

5.34.2.4 patchLongDescription()

```
virtual QString quantier::local_storage::IPatch::patchLongDescription ( ) const [pure virtual]
```

Returns

Long i.e. detailed description of the patch

5.34.2.5 patchShortDescription()

```
virtual QString quantier::local_storage::IPatch::patchShortDescription ( ) const [pure virtual]
```

Returns

Short description of the patch

5.34.2.6 removeLocalStorageBackup()

```
virtual QFuture< void > quantier::local_storage::IPatch::removeLocalStorageBackup ( ) [pure virtual]
```

Remove the previously made backup of local storage, presumably after successful application of the patch so the backup is no longer needed. It won't work if no backup was made before applying a patch, obviously.

Returns

Future which can be awaited for local storage backup removal. Contains exception if backup removal fails.

5.34.2.7 restoreLocalStorageFromBackup()

```
virtual QFuture< void > quentier::local_storage::IPatch::restoreLocalStorageFromBackup ( )
[pure virtual]
```

Restore local storage from previously made backup, presumably after the failed attempt to apply a patch. Won't work if no backup was made before applying a patch, obviously.

Returns

Future which can be awaited for the backup restoration completion. Contains exception if backup restoration fails.

5.34.2.8 toVersion()

```
virtual int quentier::local_storage::IPatch::toVersion ( ) const [pure virtual], [noexcept]
```

Returns

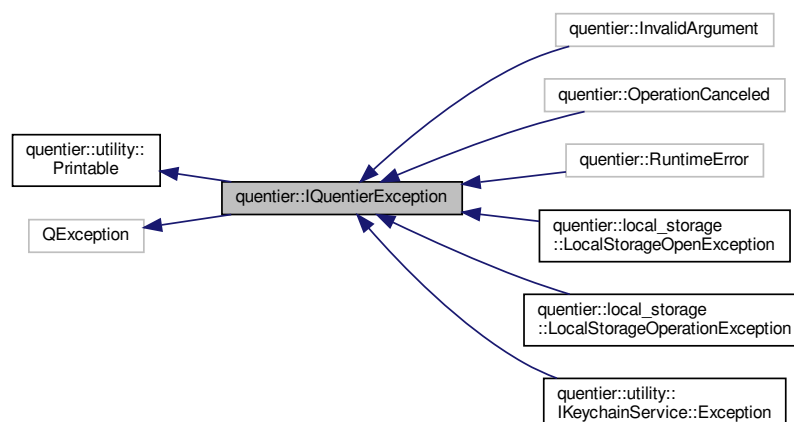
Version of local storage to which the patch would upgrade the local storage

5.35 quentier::IQuentierException Class Reference

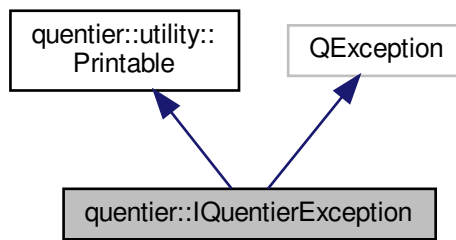
The [IQuentierException](#) class represents the interface for exceptions specific to libquentier and applications based on it.

```
#include <IQuentierException.h>
```

Inheritance diagram for quentier::IQuentierException:



Collaboration diagram for `quentier::IQuentierException`:



Public Member Functions

- [ErrorString](#) **errorMessage** () const
- QString **localizedErrorMessage** () const
- QString **nonLocalizedErrorMessage** () const
- const char * **what** () const noexcept override
- QTextStream & [print](#) (QTextStream &strm) const override

Protected Member Functions

- **IQuentierException** ([ErrorString](#) message)
- **IQuentierException** (const [IQuentierException](#) &other)
- [IQuentierException](#) & **operator=** (const [IQuentierException](#) &other)
- virtual QString **exceptionDisplayName** () const =0

5.35.1 Detailed Description

The [IQuentierException](#) class represents the interface for exceptions specific to libquentier and applications based on it.

In addition to standard exception features inherited from `std::exception`, [IQuentierException](#) based exceptions can provide both localized and non-localized error messages.

5.35.2 Member Function Documentation

5.35.2.1 `print()`

```

QTextStream & quentier::IQuentierException::print (
    QTextStream & strm ) const [override], [virtual]
  
```

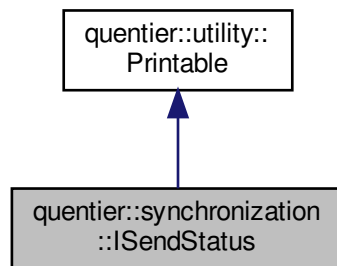
Implements [quentier::utility::Printable](#).

5.36 quantier::synchronization::ISendStatus Class Reference

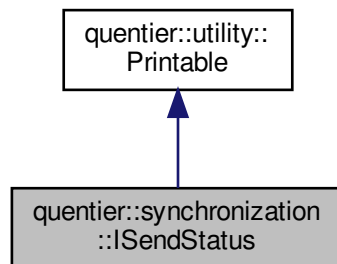
The [ISendStatus](#) interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote.

```
#include <ISendStatus.h>
```

Inheritance diagram for quantier::synchronization::ISendStatus:



Collaboration diagram for quantier::synchronization::ISendStatus:



Public Types

- using **QExceptionPtr** = `std::shared_ptr< QException >`
- using **NoteWithException** = `std::pair< qevercloud::Note, QExceptionPtr >`
- using **NotebookWithException** = `std::pair< qevercloud::Notebook, QExceptionPtr >`
- using **SavedSearchWithException** = `std::pair< qevercloud::SavedSearch, QExceptionPtr >`
- using **TagWithException** = `std::pair< qevercloud::Tag, QExceptionPtr >`

Public Member Functions

- virtual quint64 [totalAttemptedToSendNotes](#) () const =0
- virtual quint64 [totalAttemptedToSendNotebooks](#) () const =0
- virtual quint64 [totalAttemptedToSendSavedSearches](#) () const =0
- virtual quint64 [totalAttemptedToSendTags](#) () const =0
- virtual quint64 [totalSuccessfullySentNotes](#) () const =0
- virtual QList< NoteWithException > [failedToSendNotes](#) () const =0
- virtual quint64 [totalSuccessfullySentNotebooks](#) () const =0
- virtual QList< NotebookWithException > [failedToSendNotebooks](#) () const =0
- virtual quint64 [totalSuccessfullySentSavedSearches](#) () const =0
- virtual QList< SavedSearchWithException > [failedToSendSavedSearches](#) () const =0
- virtual quint64 [totalSuccessfullySentTags](#) () const =0
- virtual QList< TagWithException > [failedToSendTags](#) () const =0
- virtual StopSynchronizationError [stopSynchronizationError](#) () const =0
- virtual bool [needToRepeatIncrementalSync](#) () const =0

5.36.1 Detailed Description

The [ISendStatus](#) interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote.

5.36.2 Member Function Documentation

5.36.2.1 failedToSendNotebooks()

```
virtual QList< NotebookWithException > quantier::synchronization::ISendStatus::failedToSendNotebooks ( ) const [pure virtual]
```

Returns

list with notebooks and exceptions representing failures to send these notebooks to Evernote

5.36.2.2 failedToSendNotes()

```
virtual QList< NoteWithException > quantier::synchronization::ISendStatus::failedToSendNotes ( ) const [pure virtual]
```

Returns

list with notes and exceptions representing failures to send these notes to Evernote

5.36.2.3 failedToSendSavedSearches()

```
virtual QList< SavedSearchWithException > quantier::synchronization::ISendStatus::failedToSendSavedSearches ( ) const [pure virtual]
```

Returns

list with saved searches and exceptions representing failures to send these saved searches to Evernote

5.36.2.4 failedToSendTags()

```
virtual QList< TagWithException > quantier::synchronization::ISendStatus::failedToSendTags ( ) const [pure virtual]
```

Returns

list with tags and exceptions representing failures to send these tags to Evernote

5.36.2.5 needToRepeatIncrementalSync()

```
virtual bool quantier::synchronization::ISendStatus::needToRepeatIncrementalSync ( ) const [pure virtual]
```

If during the send step of synchronization it was found out that Evernote service's state of account has been updated since the last download step, returns true meaning that incremental download step should be repeated. Otherwise returns false.

5.36.2.6 stopSynchronizationError()

```
virtual StopSynchronizationError quantier::synchronization::ISendStatus::stopSynchronizationError ( ) const [pure virtual]
```

Returns

error which might have occurred during sending the data to Evernote which has prevented further attempts to send anything to Evernote or std::monostate if no such error has occurred

5.36.2.7 totalAttemptedToSendNotebooks()

```
virtual quint64 quantier::synchronization::ISendStatus::totalAttemptedToSendNotebooks ( ) const [pure virtual]
```

Returns

total number of notebooks attempted to be sent to Evernote

5.36.2.8 totalAttemptedToSendNotes()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendNotes ( ) const  
[pure virtual]
```

Returns

total number of notes attempted to be sent to Evernote

5.36.2.9 totalAttemptedToSendSavedSearches()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendSavedSearches ( )  
const [pure virtual]
```

Returns

total number of saved searches attempted to be sent to Evernote

5.36.2.10 totalAttemptedToSendTags()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendTags ( ) const  
[pure virtual]
```

Returns

total number of tags attempted to be sent to Evernote

5.36.2.11 totalSuccessfullySentNotebooks()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentNotebooks ( )  
const [pure virtual]
```

Returns

number of notebooks which were successfully sent to Evernote

5.36.2.12 totalSuccessfullySentNotes()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentNotes ( ) const  
[pure virtual]
```

Returns

number of notes which were successfully sent to Evernote

5.36.2.13 totalSuccessfullySentSavedSearches()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentSavedSearches ( )  
const [pure virtual]
```

Returns

number of saved searches which were successfully sent to Evernote

5.36.2.14 totalSuccessfullySentTags()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentTags ( ) const  
[pure virtual]
```

Returns

number of tags which were successfully sent to Evernote

5.37 quentier::ResourceRecognitionIndexItem::IShapeItem Struct Reference

Public Member Functions

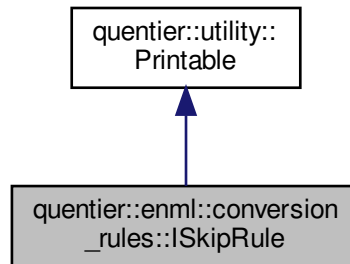
- virtual QString **shape** () const =0
- virtual int **weight** () const =0

5.38 quantier::enml::conversion_rules::ISkipRule Class Reference

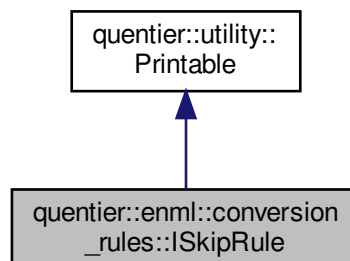
The [ISkipRule](#) interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion.

```
#include <ISkipRule.h>
```

Inheritance diagram for quantier::enml::conversion_rules::ISkipRule:



Collaboration diagram for quantier::enml::conversion_rules::ISkipRule:



Public Types

- enum class [Target](#) { [Element](#) , [AttributeName](#) , [AttributeValue](#) }

Public Member Functions

- virtual [Target](#) [target](#) () const =0
- virtual QString [value](#) () const =0
- virtual MatchMode [matchMode](#) () const =0
- virtual bool [includeContents](#) () const =0
- virtual Qt::CaseSensitivity [caseSensitivity](#) () const =0
- QTextStream & [print](#) (QTextStream &strm) const override

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, [Target target](#))
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, [Target target](#))

5.38.1 Detailed Description

The [ISkipRule](#) interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion.

ENML format prohibits the use of certain HTML tags and attributes. This interface facilitates skipping these tags and attributes in the process of conversion from HTML to ENML

5.38.2 Member Enumeration Documentation

5.38.2.1 Target

```
enum class quentier::enml::conversion_rules::ISkipRule::Target [strong]
```

Target to be affected by the skip rule

Enumerator

Element	HTML element
AttributeName	HTML attribute with specified name
AttributeValue	HTML attribute with specified value

5.38.3 Member Function Documentation

5.38.3.1 caseSensitivity()

```
virtual Qt::CaseSensitivity quentier::enml::conversion_rules::ISkipRule::caseSensitivity ( )
const [pure virtual]
```

Case sensitivity for target name/value check

5.38.3.2 includeContents()

```
virtual bool quentier::enml::conversion_rules::ISkipRule::includeContents ( ) const [pure
virtual]
```

Specifies whether the element contents should be included without the element itself if it needs to be skipped or not

5.38.3.3 matchMode()

```
virtual MatchMode quantier::enml::conversion_rules::ISkipRule::matchMode ( ) const [pure virtual]
```

Match mode for name or value of the target

5.38.3.4 print()

```
QTextStream & quantier::enml::conversion_rules::ISkipRule::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::utility::Printable](#).

5.38.3.5 target()

```
virtual Target quantier::enml::conversion_rules::ISkipRule::target ( ) const [pure virtual]
```

Target to be affected by the skip rule

5.38.3.6 value()

```
virtual QString quantier::enml::conversion_rules::ISkipRule::value ( ) const [pure virtual]
```

Name or value of the target

5.39 quantier::enml::conversion_rules::ISkipRuleBuilder Class Reference

Public Member Functions

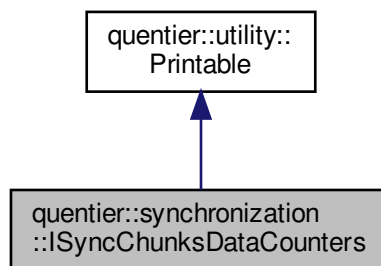
- virtual [ISkipRuleBuilder](#) & **setTarget** ([ISkipRule::Target](#) target)=0
- virtual [ISkipRuleBuilder](#) & **setValue** (QString value)=0
- virtual [ISkipRuleBuilder](#) & **setMatchMode** (MatchMode matchMode)=0
- virtual [ISkipRuleBuilder](#) & **setIncludeContents** (bool includeContents)=0
- virtual [ISkipRuleBuilder](#) & **setCaseSensitivity** (Qt::CaseSensitivity caseSensitivity)=0
- virtual [ISkipRulePtr](#) **build** ()=0

5.40 quantier::synchronization::ISyncChunksDataCounters Struct Reference

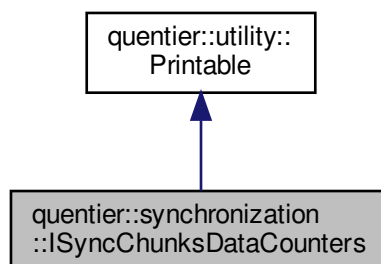
The [ISyncChunksDataCounters](#) interface provides integer counters representing the current progress on processing the data from downloaded sync chunks.

```
#include <ISyncChunksDataCounters.h>
```

Inheritance diagram for quantier::synchronization::ISyncChunksDataCounters:



Collaboration diagram for quantier::synchronization::ISyncChunksDataCounters:



Public Member Functions

- virtual quint64 [totalSavedSearches](#) () const noexcept=0
- virtual quint64 [totalExpungedSavedSearches](#) () const noexcept=0
- virtual quint64 [addedSavedSearches](#) () const noexcept=0
- virtual quint64 [updatedSavedSearches](#) () const noexcept=0
- virtual quint64 [expungedSavedSearches](#) () const noexcept=0
- virtual quint64 [totalTags](#) () const noexcept=0
- virtual quint64 [totalExpungedTags](#) () const noexcept=0

- virtual quint64 [addedTags](#) () const noexcept=0
- virtual quint64 [updatedTags](#) () const noexcept=0
- virtual quint64 [expungedTags](#) () const noexcept=0
- virtual quint64 [totalLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [totalExpungedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [addedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [updatedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [expungedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [totalNotebooks](#) () const noexcept=0
- virtual quint64 [totalExpungedNotebooks](#) () const noexcept=0
- virtual quint64 [addedNotebooks](#) () const noexcept=0
- virtual quint64 [updatedNotebooks](#) () const noexcept=0
- virtual quint64 [expungedNotebooks](#) () const noexcept=0

5.40.1 Detailed Description

The [ISyncChunksDataCounters](#) interface provides integer counters representing the current progress on processing the data from downloaded sync chunks.

5.40.2 Member Function Documentation

5.40.2.1 [addedLinkedNotebooks\(\)](#)

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedLinkedNotebooks ( )  
const [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks added to the local storage so far

5.40.2.2 [addedNotebooks\(\)](#)

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedNotebooks ( ) const  
[pure virtual], [noexcept]
```

Number of notebooks from sync chunks added to the local storage so far

5.40.2.3 [addedSavedSearches\(\)](#)

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedSavedSearches ( )  
const [pure virtual], [noexcept]
```

Number of saved searches from sync chunks added to the local storage so far

5.40.2.4 addedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedTags ( ) const [pure virtual], [noexcept]
```

Number of tags from sync chunks added to the local storage so far

5.40.2.5 expungedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedLinkedNotebooks ( ) const [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks expunged from the local storage so far

5.40.2.6 expungedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedNotebooks ( ) const [pure virtual], [noexcept]
```

Number of notebooks from sync chunks expunged from the local storage so far

5.40.2.7 expungedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedSavedSearches ( ) const [pure virtual], [noexcept]
```

Number of saved searches from sync chunks expunged from the local storage so far

5.40.2.8 expungedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedTags ( ) const [pure virtual], [noexcept]
```

Number of tags from sync chunks expunged from the local storage so far

5.40.2.9 totalExpungedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedLinkedNotebooks ( ) const [pure virtual], [noexcept]
```

Total number of expunged saved searches in downloaded sync chunks

5.40.2.10 totalExpungedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedNotebooks ( ) const [pure virtual], [noexcept]
```

Total number of expunged notebooks in downloaded sync chunks

5.40.2.11 totalExpungedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedSavedSearches  
( ) const [pure virtual], [noexcept]
```

Total number of expunged saved searches in downloaded sync chunks

5.40.2.12 totalExpungedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedTags ( )  
const [pure virtual], [noexcept]
```

Total number of expunged tags in downloaded sync chunks

5.40.2.13 totalLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalLinkedNotebooks ( )  
const [pure virtual], [noexcept]
```

Total number of new or updated linked notebooks in downloaded sync chunks

5.40.2.14 totalNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalNotebooks ( ) const  
[pure virtual], [noexcept]
```

Total number of new or updated notebooks in downloaded sync chunks

5.40.2.15 totalSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalSavedSearches ( )  
const [pure virtual], [noexcept]
```

Total number of new or updated saved searches in downloaded sync chunks

5.40.2.16 totalTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalTags ( ) const [pure  
virtual], [noexcept]
```

Total number of new or updated tags in downloaded sync chunks

5.40.2.17 updatedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::updatedLinkedNotebooks ( )  
const [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks updated in the local storage so far

5.40.2.18 updatedNotebooks()

```
virtual quint64 quantier::synchronization::ISyncChunksDataCounters::updatedNotebooks ( ) const
[pure virtual], [noexcept]
```

Number of notebooks from sync chunks updated in the local storage so far

5.40.2.19 updatedSavedSearches()

```
virtual quint64 quantier::synchronization::ISyncChunksDataCounters::updatedSavedSearches ( )
const [pure virtual], [noexcept]
```

Number of saved searches from sync chunks updated in the local storage so far

5.40.2.20 updatedTags()

```
virtual quint64 quantier::synchronization::ISyncChunksDataCounters::updatedTags ( ) const
[pure virtual], [noexcept]
```

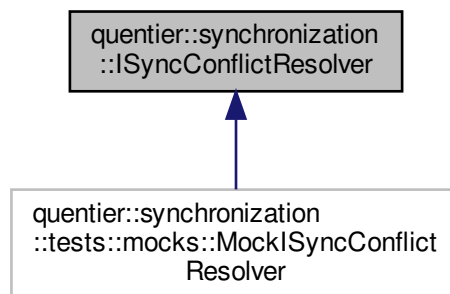
Number of tags from sync chunks updated in the local storage so far

5.41 quantier::synchronization::ISyncConflictResolver Class Reference

The [ISyncConflictResolver](#) interface provides methods used to resolve conflicts between local and remote versions of the same data item.

```
#include <ISyncConflictResolver.h>
```

Inheritance diagram for quantier::synchronization::ISyncConflictResolver:



Classes

- struct [ConflictResolution](#)

The [ConflictResolution](#) struct is a namespace inside which several other structs determining actual conflict resolutions.

Public Types

- using **NotebookConflictResolution** = std::variant< [ConflictResolution::UseTheirs](#), [ConflictResolution::UseMine](#), [ConflictResolution::IgnoreMine](#), [ConflictResolution::MoveMine](#)< qevercloud::Notebook > >
- using **NoteConflictResolution** = std::variant< [ConflictResolution::UseTheirs](#), [ConflictResolution::UseMine](#), [ConflictResolution::IgnoreMine](#), [ConflictResolution::MoveMine](#)< qevercloud::Note > >
- using **SavedSearchConflictResolution** = std::variant< [ConflictResolution::UseTheirs](#), [ConflictResolution::UseMine](#), [ConflictResolution::IgnoreMine](#), [ConflictResolution::MoveMine](#)< qevercloud::SavedSearch > >
- using **TagConflictResolution** = std::variant< [ConflictResolution::IgnoreMine](#), [ConflictResolution::UseTheirs](#), [ConflictResolution::UseMine](#), [ConflictResolution::MoveMine](#)< qevercloud::Tag > >

Public Member Functions

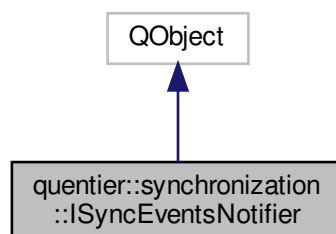
- virtual QFuture< NotebookConflictResolution > **resolveNotebookConflict** (qevercloud::Notebook theirs, qevercloud::Notebook mine)=0
- virtual QFuture< NoteConflictResolution > **resolveNoteConflict** (qevercloud::Note theirs, qevercloud::Note mine)=0
- virtual QFuture< SavedSearchConflictResolution > **resolveSavedSearchConflict** (qevercloud::SavedSearch theirs, qevercloud::SavedSearch mine)=0
- virtual QFuture< TagConflictResolution > **resolveTagConflict** (qevercloud::Tag theirs, qevercloud::Tag mine)=0

5.41.1 Detailed Description

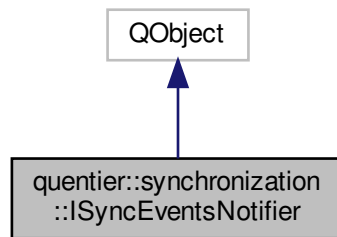
The [ISyncConflictResolver](#) interface provides methods used to resolve conflicts between local and remote versions of the same data item.

5.42 quantier::synchronization::ISyncEventsNotifier Class Reference

Inheritance diagram for quantier::synchronization::ISyncEventsNotifier:



Collaboration diagram for `quentier::synchronization::ISyncEventsNotifier`:



Signals

- void [syncChunksDownloadProgress](#) (qint32 highestDownloadedUsn, qint32 highestServerUsn, qint32 lastPreviousUsn)
- void [syncChunksDownloaded](#) ()
- void [syncChunksDataProcessingProgress](#) (ISyncChunksDataCountersPtr counters)
- void [startLinkedNotebooksDataDownloading](#) (const QList< qevercloud::LinkedNotebook > &linkedNotebooks)
- void [linkedNotebookSyncChunksDownloadProgress](#) (qint32 highestDownloadedUsn, qint32 highestServerUsn, qint32 lastPreviousUsn, const qevercloud::LinkedNotebook &linkedNotebook)
- void [linkedNotebookSyncChunksDownloaded](#) (const qevercloud::LinkedNotebook &linkedNotebook)
- void [linkedNotebookSyncChunksDataProcessingProgress](#) (ISyncChunksDataCountersPtr counters, const qevercloud::LinkedNotebook &linkedNotebook)
- void [notesDownloadProgress](#) (quint32 notesDownloaded, quint32 totalNotesToDownload)
- void [linkedNotebookNotesDownloadProgress](#) (quint32 notesDownloaded, quint32 totalNotesToDownload, const qevercloud::LinkedNotebook &linkedNotebook)
- void [resourcesDownloadProgress](#) (quint32 resourcesDownloaded, quint32 totalResourcesToDownload)
- void [linkedNotebookResourcesDownloadProgress](#) (quint32 resourcesDownloaded, quint32 totalResourcesToDownload, const qevercloud::LinkedNotebook &linkedNotebook)
- void [downloadFinished](#) (bool dataDownloaded)
- void [userOwnSendStatusUpdate](#) (ISendStatusPtr sendStatus)
- void [linkedNotebookSendStatusUpdate](#) (const qevercloud::Guid &linkedNotebookGuid, ISendStatusPtr sendStatus)

Protected Member Functions

- **ISyncEventsNotifier** (QObject *parent=nullptr)

5.42.1 Member Function Documentation

5.42.1.1 downloadFinished

```
void quentier::synchronization::ISyncEventsNotifier::downloadFinished (
    bool dataDownloaded ) [signal]
```

This signal is emitted when the initial download step is finished.

Parameters

<i>dataDownloaded</i>	True if some data was actually downloaded during this step i.e. there were some updates on Evernote servers compared to local state, false otherwise.
-----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

5.42.1.2 linkedNotebookNotesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookNotesDownloadProgress (
    quint32 notesDownloaded,
    quint32 totalNotesToDownload,
    const qevercloud::LinkedNotebook & linkedNotebook ) [signal]
```

This signal is emitted on each successful download of full note data from some linked notebook.

Parameters

<i>notesDownloaded</i>	The number of notes downloaded by the moment
<i>totalNotesToDownload</i>	The total number of notes that need to be downloaded
<i>linkedNotebook</i>	The linked notebook which notes download progress is being reported

5.42.1.3 linkedNotebookResourcesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookResourcesDownloadProgress (
    quint32 resourcesDownloaded,
    quint32 totalResourcesToDownload,
    const qevercloud::LinkedNotebook & linkedNotebook ) [signal]
```

This signal is emitted on each successful download of full resource data from linked notebooks during the incremental sync (as individual resources are downloaded along with their notes during full sync).

Parameters

<i>resourcesDownloaded</i>	The number of resources downloaded by the moment
<i>totalResourcesToDownload</i>	The total number of resources that need to be downloaded
<i>linkedNotebook</i>	The linked notebook which resources download progress is being reported

5.42.1.4 linkedNotebookSendStatusUpdate

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSendStatusUpdate (
    const qevercloud::Guid & linkedNotebookGuid,
    ISendStatusPtr sendStatus ) [signal]
```

This signal is emitted on each successful or unsuccessful attempt to send some new or locally modified data item from some linked notebook to Evernote.

Parameters

<i>linkedNotebookGuid</i>	Guid of the linked notebook for which the send status was updated
<i>sendStatus</i>	The updated send status

5.42.1.5 linkedNotebookSyncChunksDataProcessingProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDataProcessingProgress (
    ISyncChunksDataCountersPtr counters,
    const qevercloud::LinkedNotebook & linkedNotebook ) [signal]
```

This signal is emitted during some linked notebook's downloaded sync chunks contents processing and denotes the progress on that step.

Parameters

<i>counters</i>	Updated sync chunks data counters
<i>linkedNotebook</i>	The linked notebook which sync chunks data processing progress is being reported

5.42.1.6 linkedNotebookSyncChunksDownloaded

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDownloaded (
    const qevercloud::LinkedNotebook & linkedNotebook ) [signal]
```

This signal is emitted when the sync chunks for data from some linked notebook are downloaded during "remote to local" synchronization step

Parameters

<i>linkedNotebook</i>	The linked notebook which sync chunks were downloaded
-----------------------	-------------------------------------------------------

5.42.1.7 linkedNotebookSyncChunksDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDownloadProgress (
    quint32 highestDownloadedUsn,
    quint32 highestServerUsn,
    quint32 lastPreviousUsn,
    const qevercloud::LinkedNotebook & linkedNotebook ) [signal]
```

This signal is emitted during linked notebooks sync chunks downloading and denotes the progress of that step, individually for each linked notebook. The percentage of completeness can be computed roughly as $(\text{highestDownloadedUsn} - \text{lastPreviousUsn}) / (\text{highestServerUsn} - \text{lastPreviousUsn}) * 100\%$.

Parameters

<i>highestDownloadedUsn</i>	The highest update sequence number within data items from linked notebook sync chunks downloaded so far
<i>highestServerUsn</i>	The current highest update sequence number within the linked notebook
<i>lastPreviousUsn</i>	The last update sequence number from previous sync of the given linked notebook; if current sync is the first one, this value is zero
<i>linkedNotebook</i>	The linked notebook which sync chunks download progress is reported

5.42.1.8 notesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::notesDownloadProgress (
    quint32 notesDownloaded,
    quint32 totalNotesToDownload ) [signal]
```

This signal is emitted on each successful download of full note data from user's own account.

Parameters

<i>notesDownloaded</i>	The number of notes downloaded by the moment
<i>totalNotesToDownload</i>	The total number of notes that need to be downloaded

5.42.1.9 resourcesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::resourcesDownloadProgress (
    quint32 resourcesDownloaded,
    quint32 totalResourcesToDownload ) [signal]
```

This signal is emitted on each successful download of full resource data from user's own account during the incremental sync (as individual resources are downloaded along with their notes during full sync).

Parameters

<i>resourcesDownloaded</i>	The number of resources downloaded by the moment
<i>totalResourcesToDownload</i>	The total number of resources that need to be downloaded

5.42.1.10 startLinkedNotebooksDataDownloading

```
void quantier::synchronization::ISyncEventsNotifier::startLinkedNotebooksDataDownloading (
    const QList< qevercloud::LinkedNotebook > & linkedNotebooks ) [signal]
```

This signal is emitted before the downloading of data corresponding to linked notebooks starts.

Parameters

<i>linkedNotebooks</i>	Linked notebooks the data from which will start being downloaded after the execution of this callback
------------------------	-------------------------------------------------------------------------------------------------------

5.42.1.11 syncChunksDataProcessingProgress

```
void quantier::synchronization::ISyncEventsNotifier::syncChunksDataProcessingProgress (
    ISyncChunksDataCountersPtr counters ) [signal]
```

This signal is emitted during user own account's downloaded sync chunks contents processing and denotes the progress on that step.

5.42.1.12 syncChunksDownloaded

```
void quantier::synchronization::ISyncEventsNotifier::syncChunksDownloaded ( ) [signal]
```

This signal is emitted when the sync chunks for data from user's own account are downloaded during the download synchronization step.

5.42.1.13 syncChunksDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::syncChunksDownloadProgress (
    quint32 highestDownloadedUsn,
    quint32 highestServerUsn,
    quint32 lastPreviousUsn ) [signal]
```

This signal is emitted during user own account's sync chunks downloading and denotes the progress of that step. The percentage of completeness can be computed roughly as $(\text{highestDownloadedUsn} - \text{lastPreviousUsn}) / (\text{highestServerUsn} - \text{lastPreviousUsn}) * 100\%$.

Parameters

<i>highestDownloadedUsn</i>	The highest update sequence number within data items from sync chunks downloaded so far
<i>highestServerUsn</i>	The current highest update sequence number within the account
<i>lastPreviousUsn</i>	The last update sequence number from previous sync; if current sync is the first one, this value is zero

5.42.1.14 userOwnSendStatusUpdate

```
void quantier::synchronization::ISyncEventsNotifier::userOwnSendStatusUpdate (
    ISendStatusPtr sendStatus ) [signal]
```

This signal is emitted on each successful or unsuccessful attempt to send some new or locally modified data item from user's own account to Evernote.

Parameters

<i>sendStatus</i>	The updated send status
-------------------	-------------------------

5.43 quantier::synchronization::ISynchronizer Class Reference

Public Types

- using **SyncResult** = std::pair< QFuture< ISyncResultPtr >, ISyncEventsNotifier * >

Public Member Functions

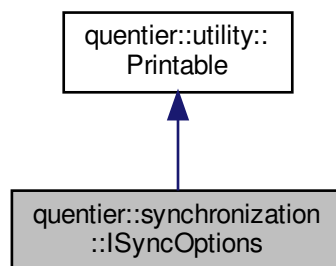
- virtual QFuture< std::pair< [Account](#), IAuthenticationInfoPtr > > **authenticateNewAccount** ()=0
- virtual QFuture< IAuthenticationInfoPtr > **authenticateAccount** ([Account](#) account)=0
- virtual SyncResult **synchronizeAccount** ([Account](#) account, local_storage::ILocalStoragePtr localStorage, utility::cancelers::ICancelerPtr canceler, ISyncOptionsPtr options=nullptr, ISyncConflictResolverPtr syncConflictResolver=nullptr)=0
- virtual void **revokeAuthentication** (qevercloud::UserID userId)=0

5.44 quantier::synchronization::ISyncOptions Class Reference

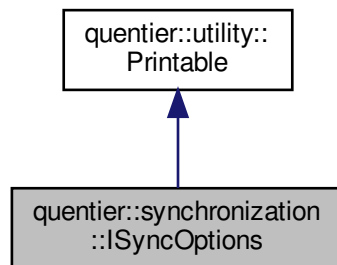
Options for synchronization process.

```
#include <ISyncOptions.h>
```

Inheritance diagram for quantier::synchronization::ISyncOptions:



Collaboration diagram for `quentier::synchronization::ISyncOptions`:



Public Member Functions

- virtual bool [downloadNoteThumbnails](#) () const =0
- virtual std::optional< QDir > [inkNoteImagesStorageDir](#) () const =0
- virtual qevercloud::IRequestContextPtr [requestContext](#) () const =0
- virtual qevercloud::IRetryPolicyPtr [retryPolicy](#) () const =0
- virtual std::optional< quint32 > [maxConcurrentNoteDownloads](#) () const =0
- virtual std::optional< quint32 > [maxConcurrentResourceDownloads](#) () const =0

5.44.1 Detailed Description

Options for synchronization process.

5.44.2 Member Function Documentation

5.44.2.1 downloadNoteThumbnails()

```
virtual bool quentier::synchronization::ISyncOptions::downloadNoteThumbnails ( ) const [pure virtual]
```

Flag to enable or disable downloading of note thumbnails during the sync. Note thumbnails are stored inside the local storage along with other note data.

5.44.2.2 inkNoteImagesStorageDir()

```
virtual std::optional< QDir > quentier::synchronization::ISyncOptions::inkNoteImagesStorageDir  
( ) const [pure virtual]
```

Directory to store the downloaded ink note images. If this method returns `std::nullopt`, ink note images would not be downloaded during the sync.

Ink notes images data is stored inside note's resources but the format of the data is not documented, which makes it quite hard to implement note editor able to fully handle ink notes. An easier option is to visualize a static image corresponding to the last revision of the ink note. Such images need to be downloaded separately during the sync if they are required.

Ink note images are stored right in this directory without any subdirectories, file names correspond to pattern `<resource guid>.png`.

5.44.2.3 maxConcurrentNoteDownloads()

```
virtual std::optional< quint32 > quentier::synchronization::ISyncOptions::maxConcurrentNote↵  
Downloads ( ) const [pure virtual]
```

Maximal number of concurrent note downloads. Allowing unlimited concurrent note downloads can lead to errors due to reaching the platform's limit on the number of open files. If `std::nullopt` is returned, the default limit is used.

5.44.2.4 maxConcurrentResourceDownloads()

```
virtual std::optional< quint32 > quentier::synchronization::ISyncOptions::maxConcurrent↵  
ResourceDownloads ( ) const [pure virtual]
```

Maximal number of concurrent resource downloads. Allowing unlimited concurrent resource downloads can lead to errors due to reaching the platform's limit on the number of open files. If `std::nullopt` is returned, the default limit is used.

5.44.2.5 requestContext()

```
virtual gevercloud::IRequestContextPtr quentier::synchronization::ISyncOptions::requestContext  
( ) const [pure virtual]
```

Request context with settings which should be used during the sync. If `nullptr` then request context with default settings would be used.

5.44.2.6 retryPolicy()

```
virtual gevercloud::IRetryPolicyPtr quentier::synchronization::ISyncOptions::retryPolicy ( )  
const [pure virtual]
```

Retry policy which should be used during the sync. If `nullptr` then the default retry policy would be used.

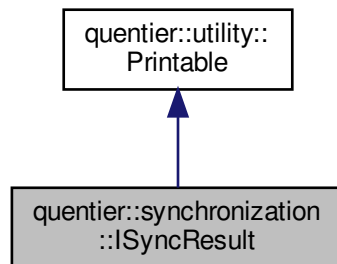
5.45 quantier::synchronization::ISyncOptionsBuilder Class Reference

Public Member Functions

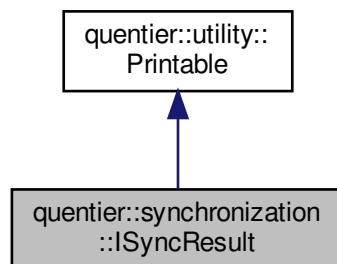
- virtual [ISyncOptionsBuilder](#) & **setDownloadNoteThumbnails** (bool value)=0
- virtual [ISyncOptionsBuilder](#) & **setLinkNoteImagesStorageDir** (std::optional< QDir > dir)=0
- virtual [ISyncOptionsBuilder](#) & **setRequestContext** (qevercloud::IRequestContextPtr ctx)=0
- virtual [ISyncOptionsBuilder](#) & **setRetryPolicy** (qevercloud::IRetryPolicyPtr retryPolicy)=0
- virtual [ISyncOptionsBuilder](#) & **setMaxConcurrentNoteDownloads** (std::optional< quint32 > max↔ ConcurrentNoteDownloads)=0
- virtual [ISyncOptionsBuilder](#) & **setMaxConcurrentResourceDownloads** (std::optional< quint32 > max↔ ConcurrentResourceDownloads)=0
- virtual ISyncOptionsPtr **build** ()=0

5.46 quantier::synchronization::ISyncResult Class Reference

Inheritance diagram for quantier::synchronization::ISyncResult:



Collaboration diagram for quantier::synchronization::ISyncResult:



Public Member Functions

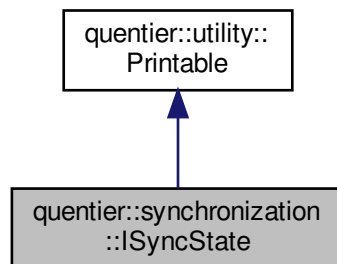
- virtual ISyncStatePtr **syncState** () const =0
- virtual ISyncChunksDataCountersPtr **userAccountSyncChunksDataCounters** () const =0
- virtual QHash< qevercloud::Guid, ISyncChunksDataCountersPtr > **linkedNotebookSyncChunksDataCounters** () const =0
- virtual bool **userAccountSyncChunksDownloaded** () const =0
- virtual QSet< qevercloud::Guid > **linkedNotebookGuidsWithSyncChunksDownloaded** () const =0
- virtual IDownloadNotesStatusPtr **userAccountDownloadNotesStatus** () const =0
- virtual QHash< qevercloud::Guid, IDownloadNotesStatusPtr > **linkedNotebookDownloadNotesStatuses** () const =0
- virtual IDownloadResourcesStatusPtr **userAccountDownloadResourcesStatus** () const =0
- virtual QHash< qevercloud::Guid, IDownloadResourcesStatusPtr > **linkedNotebookDownloadResourcesStatuses** () const =0
- virtual ISendStatusPtr **userAccountSendStatus** () const =0
- virtual QHash< qevercloud::Guid, ISendStatusPtr > **linkedNotebookSendStatuses** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

5.47 quantier::synchronization::ISyncState Class Reference

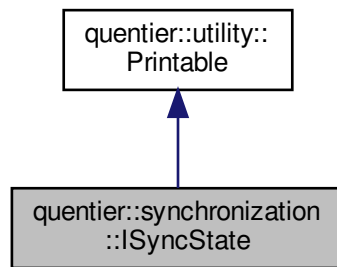
The [ISyncState](#) interface provides accessory methods to determine the sync state for the account.

```
#include <ISyncState.h>
```

Inheritance diagram for quantier::synchronization::ISyncState:



Collaboration diagram for `quentier::synchronization::ISyncState`:



Public Member Functions

- virtual `qint32` **userDataUpdateCount** () const =0
- virtual `qevercloud::Timestamp` **userDataLastSyncTime** () const =0
- virtual `QHash< qevercloud::Guid, qint32 >` **linkedNotebookUpdateCounts** () const =0
- virtual `QHash< qevercloud::Guid, qevercloud::Timestamp >` **linkedNotebookLastSyncTimes** () const =0

5.47.1 Detailed Description

The [ISyncState](#) interface provides accessory methods to determine the sync state for the account.

5.48 quentier::synchronization::ISyncStateBuilder Class Reference

Public Member Functions

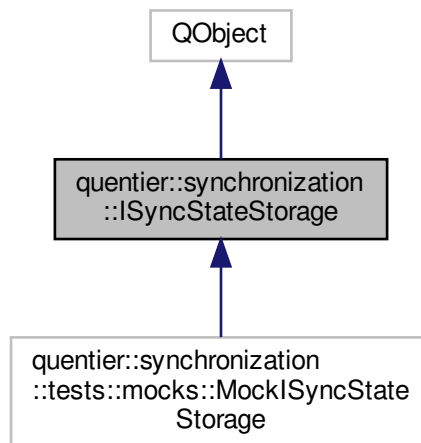
- virtual [ISyncStateBuilder](#) & **setUserDataUpdateCount** (`qint32` updateCount)=0
- virtual [ISyncStateBuilder](#) & **setUserDataLastSyncTime** (`qevercloud::Timestamp` lastSyncTime)=0
- virtual [ISyncStateBuilder](#) & **setLinkedNotebookUpdateCounts** (`QHash< qevercloud::Guid, qint32 >` updateCounts)=0
- virtual [ISyncStateBuilder](#) & **setLinkedNotebookLastSyncTimes** (`QHash< qevercloud::Guid, qevercloud::Timestamp >` lastSyncTimes)=0
- virtual `ISyncStatePtr` **build** ()=0

5.49 `quentier::synchronization::ISyncStateStorage` Class Reference

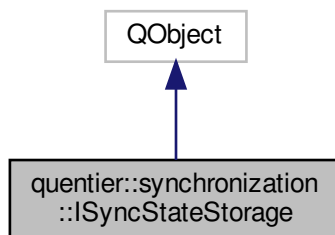
The [ISyncStateStorage](#) interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

```
#include <ISyncStateStorage.h>
```

Inheritance diagram for `quentier::synchronization::ISyncStateStorage`:



Collaboration diagram for `quentier::synchronization::ISyncStateStorage`:



Signals

- void [notifySyncStateUpdated](#) ([Account](#) account, [ISyncStatePtr](#) syncState)

Public Member Functions

- virtual ISyncStatePtr **getSyncState** (const Account &account)=0
- virtual void **setSyncState** (const Account &account, ISyncStatePtr syncState)=0

Protected Member Functions

- **ISyncStateStorage** (QObject *parent=nullptr)

5.49.1 Detailed Description

The [ISyncStateStorage](#) interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

5.49.2 Member Function Documentation

5.49.2.1 notifySyncStateUpdated

```
void quantier::synchronization::ISyncStateStorage::notifySyncStateUpdated (
    Account account,
    ISyncStatePtr syncState ) [signal]
```

Classes implementing [ISyncStateStorage](#) interface are expected to emit notifySyncStateUpdated signal each time when sync state for the corresponding account is updated

5.50 quantier::ResourceRecognitionIndexItem::ITextItem Struct Reference

Public Member Functions

- virtual QString **text** () const =0
- virtual int **weight** () const =0

5.51 quantier::synchronization::IUserStoreFactory Class Reference

Public Member Functions

- virtual qevercloud::IUserStorePtr **createUserStore** (QString userStoreUrl={}, qevercloud::IRequest↵ ContextPtr ctx={}, qevercloud::IRetryPolicyPtr retryPolicy={})=0

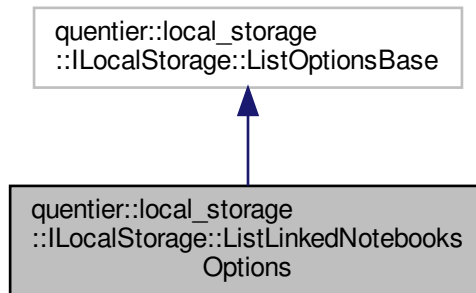
5.52 quantier::local_storage::ILocalStorage::ListGuidsFilters Struct Reference

Public Attributes

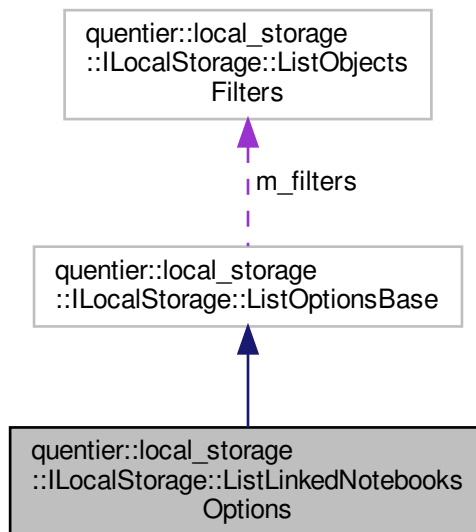
- std::optional< ListObjectsFilter > **m_locallyModifiedFilter**
- std::optional< ListObjectsFilter > **m_locallyFavoritedFilter**

5.53 quantier::local_storage::ILocalStorage::ListLinkedNotebooksOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListLinkedNotebooksOptions:



Collaboration diagram for `quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions`:

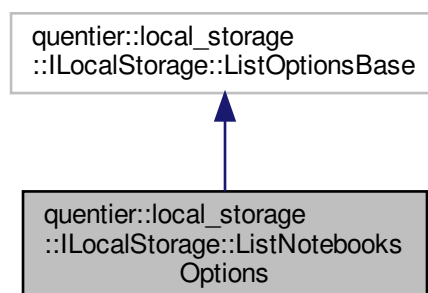


Public Attributes

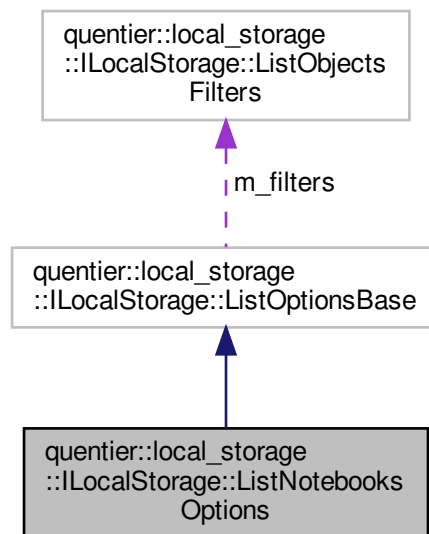
- `ListLinkedNotebooksOrder m_order = ListLinkedNotebooksOrder::NoOrder`

5.54 quentier::local_storage::ILocalStorage::ListNotebooksOptions Struct Reference

Inheritance diagram for `quentier::local_storage::ILocalStorage::ListNotebooksOptions`:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListNotebooksOptions:

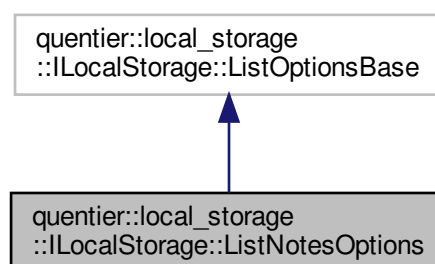


Public Attributes

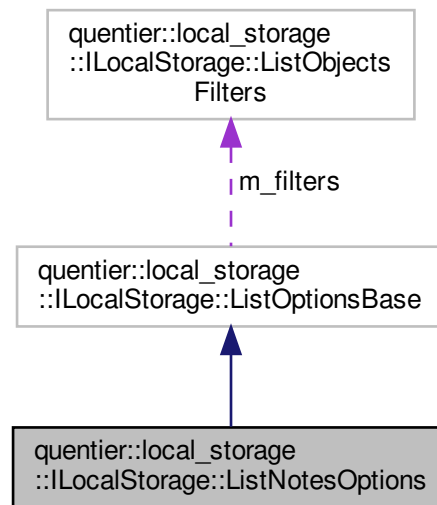
- ListNotebooksOrder **m_order** = ListNotebooksOrder::NoOrder
- [Affiliation](#) **m_affiliation** = Affiliation::Any
- `QList< qevercloud::Guid >` **m_linkedNotebookGuids**

5.55 quantier::local_storage::ILocalStorage::ListNotesOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListNotesOptions:



Collaboration diagram for `quentier::local_storage::ILocalStorage::ListNotesOptions`:



Public Attributes

- ListNotesOrder **m_order** = ListNotesOrder::NoOrder

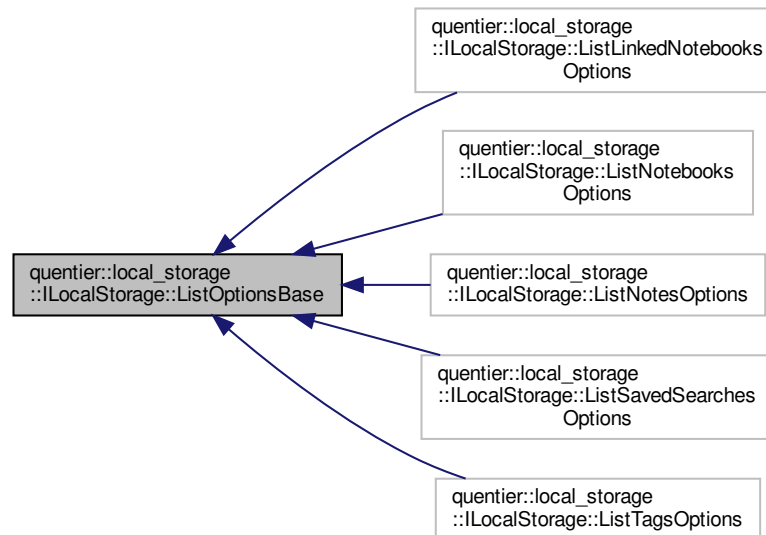
5.56 quentier::local_storage::ILocalStorage::ListObjectsFilters Struct Reference

Public Attributes

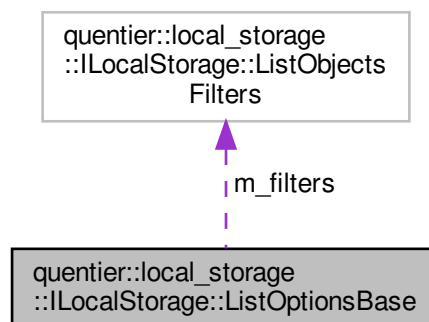
- `std::optional< ListObjectsFilter >` **m_locallyModifiedFilter**
- `std::optional< ListObjectsFilter >` **m_withGuidFilter**
- `std::optional< ListObjectsFilter >` **m_localOnlyFilter**
- `std::optional< ListObjectsFilter >` **m_locallyFavoritedFilter**

5.57 quantier::local_storage::ILocalStorage::ListOptionsBase Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListOptionsBase:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListOptionsBase:

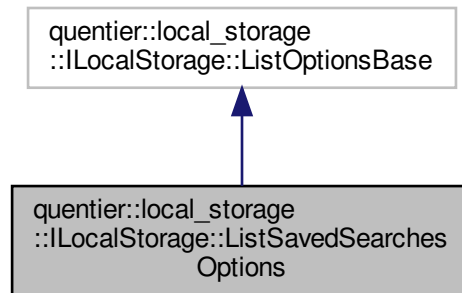


Public Attributes

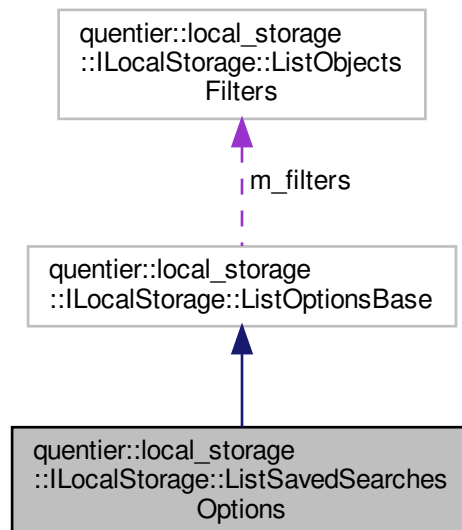
- [ListObjectsFilters](#) `m_filters` = {}
- quint64 `m_limit` = 0UL
- quint64 `m_offset` = 0UL
- OrderDirection `m_direction` = OrderDirection::Ascending

5.58 quantier::local_storage::ILocalStorage::ListSavedSearchesOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListSavedSearchesOptions:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListSavedSearchesOptions:

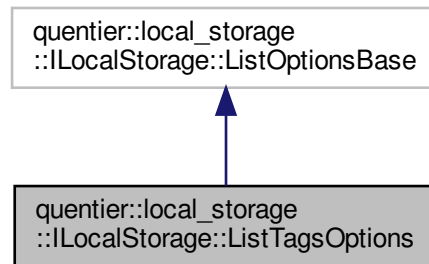


Public Attributes

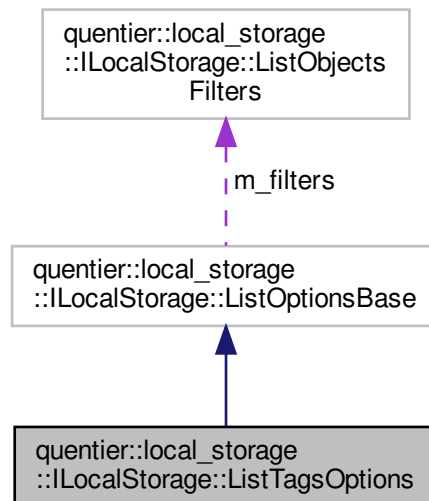
- ListSavedSearchesOrder **m_order** = ListSavedSearchesOrder::NoOrder

5.59 quantier::local_storage::ILocalStorage::ListTagsOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListTagsOptions:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListTagsOptions:



Public Attributes

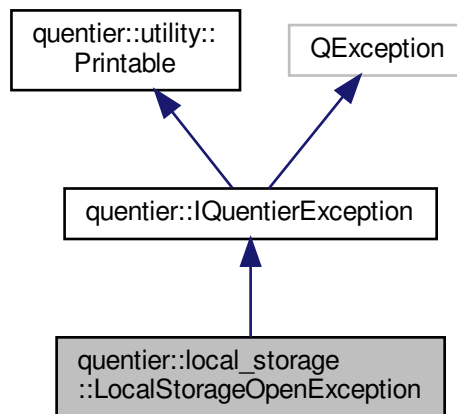
- ListTagsOrder **m_order** = ListTagsOrder::NoOrder
- [Affiliation](#) **m_affiliation** = Affiliation::Any
- QList< qevercloud::Guid > **m_linkedNotebookGuids**
- [TagNotesRelation](#) **m_tagNotesRelation** = [TagNotesRelation::Any](#)

5.60 quantier::local_storage::LocalStorageOpenException Class Reference

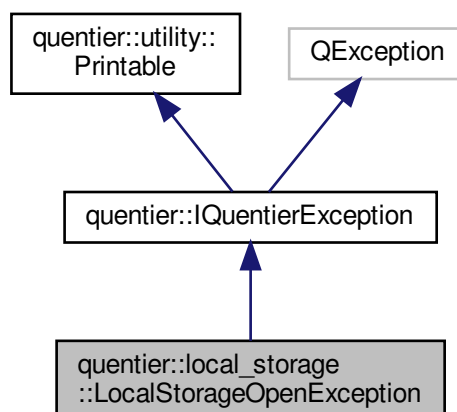
The [LocalStorageOpenException](#) is thrown on failure to open the local storage database.

```
#include <LocalStorageOpenException.h>
```

Inheritance diagram for quantier::local_storage::LocalStorageOpenException:



Collaboration diagram for quantier::local_storage::LocalStorageOpenException:



Public Member Functions

- **LocalStorageOperationException** (const [ErrorString](#) &message)
- [LocalStorageOperationException](#) * **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString [exceptionDisplayName](#) () const override

5.60.1 Detailed Description

The [LocalStorageOperationException](#) is thrown on failure to open the local storage database.

5.60.2 Member Function Documentation

5.60.2.1 exceptionDisplayName()

```
QString quantier::local_storage::LocalStorageOperationException::exceptionDisplayName ( ) const
[override], [protected], [virtual]
```

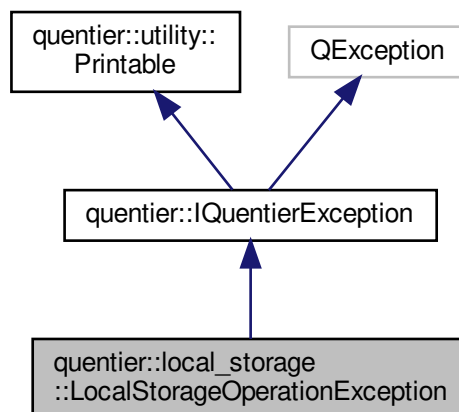
Implements [quantier::IQuantierException](#).

5.61 quantier::local_storage::LocalStorageOperationException Class Reference

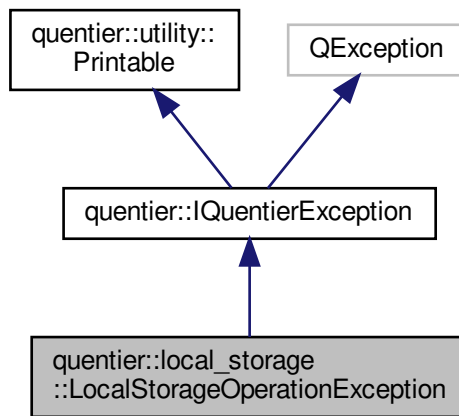
The [LocalStorageOperationException](#) is thrown when the local storage encounters some internal error during the attempt to process some operation.

```
#include <LocalStorageOperationException.h>
```

Inheritance diagram for quantier::local_storage::LocalStorageOperationException:



Collaboration diagram for `quentier::local_storage::LocalStorageOperationException`:



Public Member Functions

- `LocalStorageOperationException` ([ErrorString](#) message)
- `LocalStorageOperationException * clone ()` const override
- `void raise ()` const override

Protected Member Functions

- `QString exceptionDisplayName ()` const override

5.61.1 Detailed Description

The [LocalStorageOperationException](#) is thrown when the local storage encounters some internal error during the attempt to process some operation.

5.61.2 Member Function Documentation

5.61.2.1 exceptionDisplayName()

```

QString quentier::local_storage::LocalStorageOperationException::exceptionDisplayName ( )
const [override], [protected], [virtual]

```

Implements [quentier::IQuentierException](#).

5.62 quotient::utility::LRUCache< Key, Value, Allocator > Class Template Reference

Public Types

- using **key_type** = Key
- using **mapped_type** = Value
- using **allocator_type** = Allocator
- using **value_type** = std::pair< key_type, mapped_type >
- using **container_type** = std::list< value_type, allocator_type >
- using **size_type** = typename container_type::size_type
- using **difference_type** = typename container_type::difference_type
- using **iterator** = typename container_type::iterator
- using **const_iterator** = typename container_type::const_iterator
- using **reverse_iterator** = std::reverse_iterator< iterator >
- using **const_reverse_iterator** = std::reverse_iterator< const_iterator >
- using **reference** = value_type &
- using **const_reference** = const value_type &
- using **pointer** = typename std::allocator_traits< allocator_type >::pointer
- using **const_pointer** = typename std::allocator_traits< allocator_type >::const_pointer

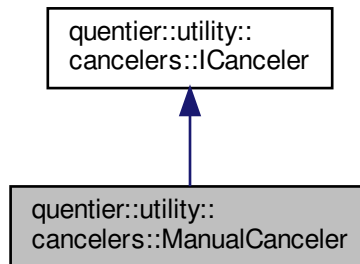
Public Member Functions

- **LRUCache** (const std::size_t maxSize=100)
- iterator **begin** () noexcept
- const_iterator **begin** () const noexcept
- reverse_iterator **rbegin** () noexcept
- const_reverse_iterator **rbegin** () const noexcept
- iterator **end** () noexcept
- const_iterator **end** () const noexcept
- reverse_iterator **rend** () noexcept
- const_reverse_iterator **rend** () const noexcept
- bool **empty** () const noexcept
- std::size_t **size** () const noexcept
- std::size_t **max_size** () const noexcept
- void **clear** ()
- void **put** (const key_type &key, const mapped_type &value)
- const mapped_type * **get** (const key_type &key) const noexcept
- bool **exists** (const key_type &key) const noexcept
- bool **remove** (const key_type &key) noexcept
- void **setMaxSize** (const std::size_t maxSize)

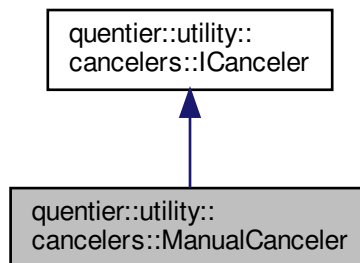
5.63 quantier::utility::cancelers::ManualCanceler Class Reference

```
#include <ManualCanceler.h>
```

Inheritance diagram for quantier::utility::cancelers::ManualCanceler:



Collaboration diagram for quantier::utility::cancelers::ManualCanceler:



Public Member Functions

- **ManualCanceler** ([ManualCanceler](#) &&other) noexcept
- **ManualCanceler** & **operator=** ([ManualCanceler](#) &&other) noexcept
- void **cancel** () noexcept
- bool **isCanceled** () const noexcept override

5.63.1 Detailed Description

[ICanceler](#) which allows one to manually call cancel method to cancel some task

5.63.2 Member Function Documentation

5.63.2.1 `cancel()`

```
void quentier::utility::cancelers::ManualCanceler::cancel ( ) [noexcept]
```

Manually cancel a task

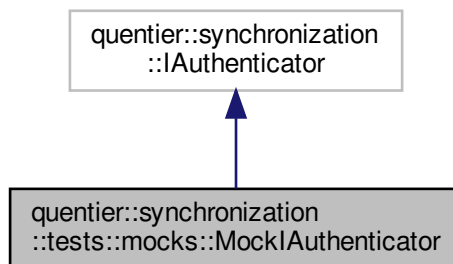
5.63.2.2 `isCanceled()`

```
bool quentier::utility::cancelers::ManualCanceler::isCanceled ( ) const [override], [virtual],  
[noexcept]
```

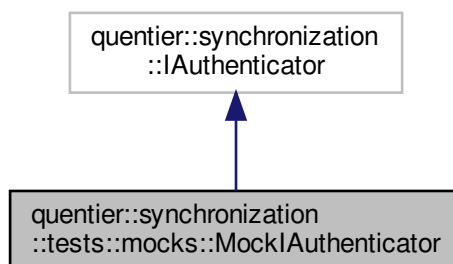
Implements [quentier::utility::cancelers::ICanceler](#).

5.64 `quentier::synchronization::tests::mocks::MockIAuthenticator` Class Reference

Inheritance diagram for `quentier::synchronization::tests::mocks::MockIAuthenticator`:



Collaboration diagram for `quentier::synchronization::tests::mocks::MockIAuthenticator`:

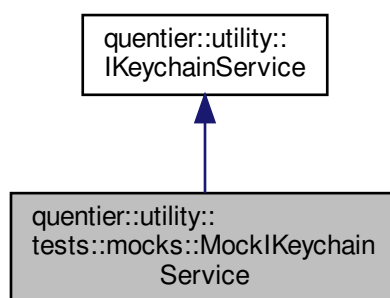


Public Member Functions

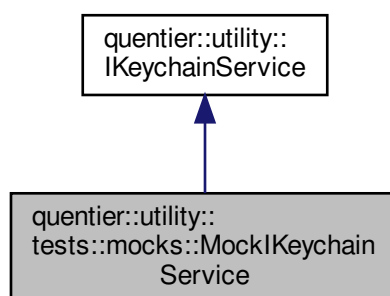
- **MOCK_METHOD** (QFuture< IAuthenticationInfoPtr >, authenticateNewAccount,(),(override))
- **MOCK_METHOD** (QFuture< IAuthenticationInfoPtr >, authenticateAccount,(Account account),(override))

5.65 quantier::utility::tests::mocks::MockIKeychainService Class Reference

Inheritance diagram for quantier::utility::tests::mocks::MockIKeychainService:



Collaboration diagram for quantier::utility::tests::mocks::MockIKeychainService:



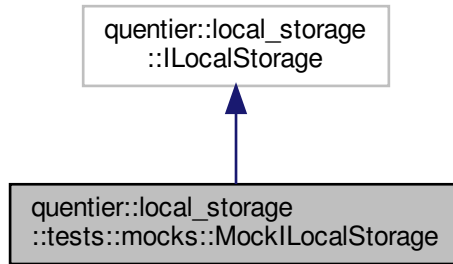
Public Member Functions

- **MOCK_METHOD** (QFuture< void >, writePassword,(QString service, QString key, QString password),(override))
- **MOCK_METHOD** (QFuture< QString >, readPassword,(QString service, QString key),(const, override))
- **MOCK_METHOD** (QFuture< void >, deletePassword,(QString service, QString key),(override))

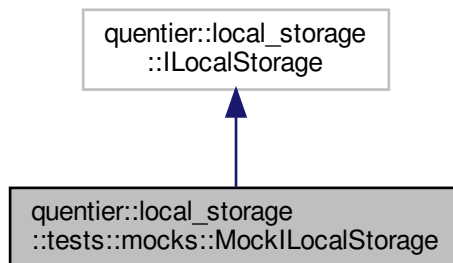
Additional Inherited Members

5.66 quantier::local_storage::tests::mocks::MockILocalStorage Class Reference

Inheritance diagram for quantier::local_storage::tests::mocks::MockILocalStorage:



Collaboration diagram for quantier::local_storage::tests::mocks::MockILocalStorage:



Public Member Functions

- **MOCK_METHOD** (QFuture< bool >, isVersionTooHigh,(),(const, override))
- **MOCK_METHOD** (QFuture< bool >, requiresUpgrade,(),(const, override))
- **MOCK_METHOD** (QFuture< QList< IPatchPtr > >, requiredPatches,(),(const, override))
- **MOCK_METHOD** (QFuture< qint32 >, version,(),(const, override))
- **MOCK_METHOD** (QFuture< qint32 >, highestSupportedVersion,(),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, userCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putUser,(qevercloud::User user),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::User > >, findUserById,(qevercloud::UserID userId),(const, override))

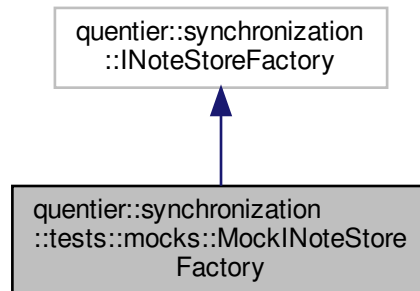
- **MOCK_METHOD** (QFuture< void >, expungeUserById,(qevercloud::UserID userId),(override))
- **MOCK_METHOD** (QFuture< quint32 >, notebookCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putNotebook,(qevercloud::Notebook notebook),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByLocalId,(QString localId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findDefaultNotebook,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Notebook > >, listNotebooks,(ListNotebooksOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::SharedNotebook > >, listSharedNotebooks,(qevercloud::Guid notebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listNotebookGuids,(ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, linkedNotebookCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putLinkedNotebook,(qevercloud::LinkedNotebook linkedNotebook),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::LinkedNotebook > >, findLinkedNotebookByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeLinkedNotebookByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::LinkedNotebook > >, listLinkedNotebooks,(ListLinkedNotebooksOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCount,(NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerNotebookLocalId,(QString notebookLocalId, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerTagLocalId,(QString tagLocalId, NoteCountOptions options),(const, override))
- **MOCK_METHOD** ((QFuture< QHash< QString, quint32 > >), noteCountsPerTags,(ListTagsOptions listTagsOptions, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerNotebookAndTagLocalIds,(QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, putNote,(qevercloud::Note note),(override))
- **MOCK_METHOD** (QFuture< void >, updateNote,(qevercloud::Note note, UpdateNoteOptions options),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Note > >, findNoteByLocalId,(QString localId, FetchNoteOptions options),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Note > >, findNoteByGuid,(qevercloud::Guid guid, FetchNoteOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeNoteByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNoteByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotes,(FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerNotebookLocalId,(QString notebookLocalId, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerTagLocalId,(QString tagLocalId, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerNotebookAndTagLocalIds,(QStringList notebookLocalIds, QStringList tagLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))

- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesByLocalIds,(QStringList noteLocalIds, FetchNoteOptions fetchOptions, [ListNotesOptions](#) options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listNoteGuids,([ListGuidsFilters](#) filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, queryNotes,([NoteSearchQuery](#) query, FetchNoteOptions fetchOptions),(const, override))
- **MOCK_METHOD** (QFuture< QStringList >, queryNoteLocalIds,([NoteSearchQuery](#) query),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, tagCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putTag,(qevercloud::Tag tag),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByLocalId,(QString tagLocalId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByGuid,(qevercloud::Guid tagGuid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByName,(QString tagName, std::optional< QString > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Tag > >, listTags,([ListTagsOptions](#) options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Tag > >, listTagsPerNoteLocalId,(QString noteLocalId, [ListTagsOptions](#) options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listTagGuids,([ListGuidsFilters](#) filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByLocalId,(QString tagLocalId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByGuid,(qevercloud::Guid tagGuid),(override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(override))
- **MOCK_METHOD** (QFuture< quint32 >, resourceCount,(NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, resourceCountPerNoteLocalId,(QString noteLocalId),(const, override))
- **MOCK_METHOD** (QFuture< void >, putResource,(qevercloud::Resource resource),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Resource > >, findResourceByLocalId,(QString resourceLocalId, FetchResourceOptions options),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Resource > >, findResourceByGuid,(qevercloud::Guid resourceGuid, FetchResourceOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeResourceByLocalId,(QString resourceLocalId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeResourceByGuid,(qevercloud::Guid resourceGuid),(override))
- **MOCK_METHOD** (QFuture< quint32 >, savedSearchCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putSavedSearch,(qevercloud::SavedSearch search),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByLocalId,(QString localId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByName,(QString name),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::SavedSearch > >, listSavedSearches,([ListSavedSearchesOptions](#) options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listSavedSearchGuids,([ListGuidsFilters](#) filters),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeSavedSearchByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeSavedSearchByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< qint32 >, highestUpdateSequenceNumber,(HighestUsnOption option),(const, override))
- **MOCK_METHOD** (QFuture< qint32 >, highestUpdateSequenceNumber,(qevercloud::Guid linkedNotebookGuid),(const, override))
- **MOCK_METHOD** ([ILocalStorageNotifier](#) *, [notifier](#),(),(const, override))

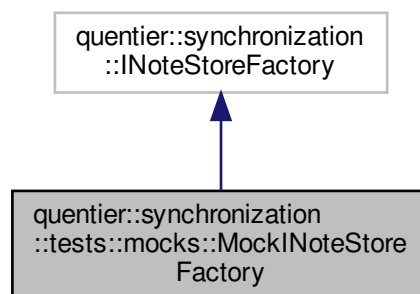
Additional Inherited Members

5.67 quantier::synchronization::tests::mocks::MockINoteStoreFactory Class Reference

Inheritance diagram for quantier::synchronization::tests::mocks::MockINoteStoreFactory:



Collaboration diagram for quantier::synchronization::tests::mocks::MockINoteStoreFactory:

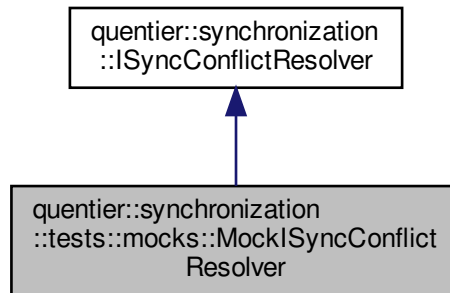


Public Member Functions

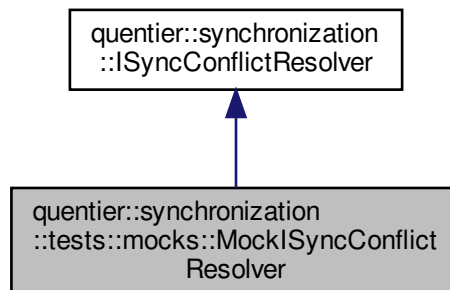
- **MOCK_METHOD** (::qevercloud::INoteStorePtr, createNoteStore,(QString noteStoreUrl, std::optional<↔
::qevercloud::Guid > linkedNotebookGuid, ::qevercloud::IRequestContextPtr ctx, ::qevercloud::IRetryPolicy↔
Ptr retryPolicy),(override))

5.68 quantier::synchronization::tests::mocks::MockISyncConflictResolver Class Reference

Inheritance diagram for quantier::synchronization::tests::mocks::MockISyncConflictResolver:



Collaboration diagram for quantier::synchronization::tests::mocks::MockISyncConflictResolver:



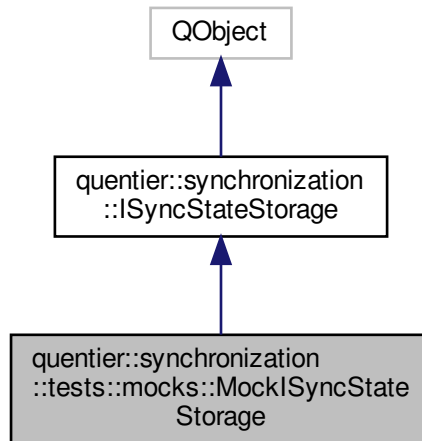
Public Member Functions

- **MOCK_METHOD** (QFuture< NotebookConflictResolution >, resolveNotebookConflict, (::qevercloud::Notebook theirs, ::qevercloud::Notebook mine), (override))
- **MOCK_METHOD** (QFuture< NoteConflictResolution >, resolveNoteConflict, (::qevercloud::Note theirs, ::qevercloud::Note mine), (override))
- **MOCK_METHOD** (QFuture< SavedSearchConflictResolution >, resolveSavedSearchConflict, (::qevercloud::SavedSearch theirs, ::qevercloud::SavedSearch mine), (override))
- **MOCK_METHOD** (QFuture< TagConflictResolution >, resolveTagConflict, (::qevercloud::Tag theirs, ::qevercloud::Tag mine), (override))

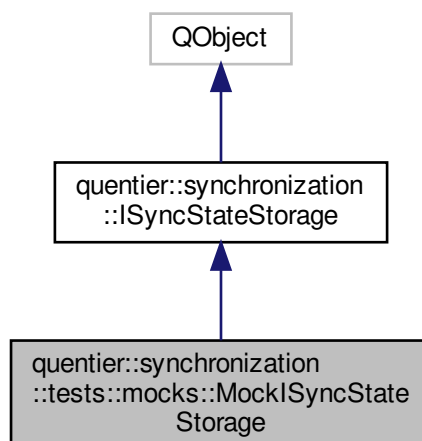
Additional Inherited Members

5.69 `quentier::synchronization::tests::mocks::MockISyncStateStorage` Class Reference

Inheritance diagram for `quentier::synchronization::tests::mocks::MockISyncStateStorage`:



Collaboration diagram for `quentier::synchronization::tests::mocks::MockISyncStateStorage`:



Public Member Functions

- **MOCK_METHOD** (ISyncStatePtr, getSyncState,(const Account &account),(override))
- **MOCK_METHOD** (void, setSyncState,(const Account &account, ISyncStatePtr syncState),(override))

Additional Inherited Members

5.70 quantier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T > Struct Template Reference

The [MoveMine](#) conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

```
#include <ISyncConflictResolver.h>
```

Public Types

- using **value_type** = T

Public Attributes

- T [mine](#)

5.70.1 Detailed Description

```
template<class T>
struct quantier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >
```

The [MoveMine](#) conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

5.70.2 Member Data Documentation

5.70.2.1 mine

```
template<class T >
T quantier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >::mine
```

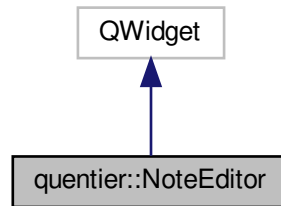
The changed value of mine data item.

5.71 quantier::NoteEditor Class Reference

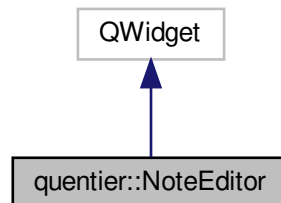
The [NoteEditor](#) class is a widget encapsulating all the functionality necessary for showing and editing notes.

```
#include <NoteEditor.h>
```

Inheritance diagram for quantier::NoteEditor:



Collaboration diagram for quantier::NoteEditor:



Public Slots

- void [convertToNote](#) ()
- void [saveNoteToLocalStorage](#) ()
- void [setNoteTitle](#) (const QString ¬eTitle)
- void [setTagIds](#) (const QStringList &tagLocalIds, const QStringList &tagGuids)
- void **undo** ()
- void **redo** ()
- void **cut** ()
- void **copy** ()
- void **paste** ()
- void **pasteUnformatted** ()
- void **selectAll** ()
- void **formatSelectionAsSourceCode** ()

- void **fontMenu** ()
- void **textBold** ()
- void **textItalic** ()
- void **textUnderline** ()
- void **textStrikethrough** ()
- void **textHighlight** ()
- void **alignLeft** ()
- void **alignCenter** ()
- void **alignRight** ()
- void **alignFull** ()
- void **findNext** (const QString &text, bool matchCase) const
- void **findPrevious** (const QString &text, bool matchCase) const
- void **replace** (const QString &textToReplace, const QString &replacementText, bool matchCase)
- void **replaceAll** (const QString &textToReplace, const QString &replacementText, bool matchCase)
- void **insertToDoCheckbox** ()
- void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString ¬eGuid, const QString &linkText)
- void **setSpellcheck** (bool enabled)
- void **setFont** (const QFont &font)
- void **setFontHeight** (int height)
- void **setFontColor** (const QColor &color)
- void **setBackgroundColor** (const QColor &color)
- void **setDefaultPalette** (const QPalette &pal)
- void **setDefaultFont** (const QFont &font)
- void **insertHorizontalLine** ()
- void **increaseFontSize** ()
- void **decreaseFontSize** ()
- void **increaseIndentation** ()
- void **decreaseIndentation** ()
- void **insertBulletedList** ()
- void **insertNumberedList** ()
- void **insertTableDialog** ()
- void **insertFixedWidthTable** (int rows, int columns, int widthInPixels)
- void **insertRelativeWidthTable** (int rows, int columns, double relativeWidth)
- void **insertTableRow** ()
- void **insertTableColumn** ()
- void **removeTableRow** ()
- void **removeTableColumn** ()
- void **addAttachmentDialog** ()
- void **saveAttachmentDialog** (const QByteArray &resourceHash)
- void **saveAttachmentUnderCursor** ()
- void **openAttachment** (const QByteArray &resourceHash)
- void **openAttachmentUnderCursor** ()
- void **copyAttachment** (const QByteArray &resourceHash)
- void **copyAttachmentUnderCursor** ()
- void **encryptSelectedText** ()
- void **decryptEncryptedTextUnderCursor** ()
- void **editHyperlinkDialog** ()
- void **copyHyperlink** ()
- void **removeHyperlink** ()
- void **onNoteLoadCancelled** ()

Signals

- void **contentChanged** ()
contentChanged signal is emitted when the note's content (text) gets modified via manual editing (i.e. not any action like paste or cut)
- void **noteAndNotebookFoundInLocalStorage** (qevercloud::Note note, qevercloud::Notebook notebook)
noteAndNotebookFoundInLocalStorage signal is emitted when note and its corresponding notebook were found within the local storage right before the note editor starts to load the note into the editor
- void **noteNotFound** (QString noteLocalId)
noteNotFound signal is emitted when the note could not be found within the local storage by the provided local id
- void **noteDeleted** (QString noteLocalId)
noteDeleted signal is emitted when the note displayed within the note editor is deleted. The note editor stops displaying the note in this case shortly after emitting this signal
- void **noteModified** ()
noteModified signal is emitted when the note's content within the editor gets modified via some way - either via manual editing or via some action (like paste or cut)
- void **notifyError** (ErrorString error)
notifyError signal is emitted when [NoteEditor](#) encounters some problem worth letting the user to know about
- void **inAppNoteLinkClicked** (QString userId, QString shardId, QString noteGuid)
inAppNoteLinkClicked signal is emitted when the in-app note link is clicked within the note editor
- void **inAppNoteLinkPasteRequested** (QString url, QString userId, QString shardId, QString noteGuid)
- void **convertedToNote** (qevercloud::Note note)
- void **cantConvertToNote** (ErrorString error)
- void **noteEditorHtmlUpdated** (QString html)
- void **currentNoteChanged** (qevercloud::Note note)
- void **spellCheckerNotReady** ()
- void **spellCheckerReady** ()
- void **noteLoaded** ()
- void **noteSavedToLocalStorage** (QString noteLocalId)
noteSavedToLocalStorage signal is emitted when the note has been saved within the local storage. [NoteEditor](#) doesn't do this on its own unless it's explicitly asked to do this via invoking its `saveNoteToLocalStorage` slot
- void **failedToSaveNoteToLocalStorage** (ErrorString errorDescription, QString noteLocalId)
failedToSaveNoteToLocalStorage signal is emitted in case of failure to save the note to local storage
- void **textBoldState** (bool state)
- void **textItalicState** (bool state)
- void **textUnderlineState** (bool state)
- void **textStrikethroughState** (bool state)
- void **textAlignLeftState** (bool state)
- void **textAlignCenterState** (bool state)
- void **textAlignRightState** (bool state)
- void **textAlignFullState** (bool state)
- void **textInsideOrderedListState** (bool state)
- void **textInsideUnorderedListState** (bool state)
- void **textInsideTableState** (bool state)
- void **textFontFamilyChanged** (QString fontFamily)
- void **textFontSizeChanged** (int fontSize)
- void **insertTableDialogRequested** ()

Public Member Functions

- **NoteEditor** (QWidget *parent=nullptr, Qt::WindowFlags flags={})
- void **initialize** (local_storage::ILocalStoragePtr localStorage, [SpellChecker](#) &spellChecker, const [Account](#) &account, QThread *backgroundJobsThread=nullptr, enml::IDecryptedTextCachePtr decryptedTextCache=nullptr)
- [INoteEditorBackend](#) * **backend** () noexcept
- void **setBackend** ([INoteEditorBackend](#) *backend)
- void **setAccount** (const [Account](#) &account)
- const QUndoStack * **undoStack** () const noexcept
- void **setUndoStack** (QUndoStack *pUndoStack)
- void **setInitialPageHtml** (const QString &html)
- void **setNoteNotFoundPageHtml** (const QString &html)
- void **setNoteDeletedPageHtml** (const QString &html)
- void **setNoteLoadingPageHtml** (const QString &html)
- QString **currentNoteLocalId** () const
- void **setCurrentNoteLocalId** (const QString ¬eLocalId)
- void **clear** ()
- bool **isModified** () const noexcept
- bool **isEditorPageModified** () const noexcept
- bool **isNoteLoaded** () const noexcept
- qint64 **idleTime** () const noexcept
- void **setFocus** ()
- QString **selectedText** () const noexcept
- bool **hasSelection** () const noexcept
- bool **spellCheckEnabled** () const noexcept
- bool **print** (QPrinter &printer, [ErrorString](#) &errorDescription)
- bool **exportToPdf** (const QString &absoluteFilePath, [ErrorString](#) &errorDescription)
- bool **exportToEnex** (const QStringList &tagNames, QString &enex, [ErrorString](#) &errorDescription)
- QPalette **defaultPalette** () const
- const QFont * **defaultFont** () const

Protected Member Functions

- void **dragMoveEvent** (QDragMoveEvent *pEvent) override
- void **dropEvent** (QDropEvent *pEvent) override

5.71.1 Detailed Description

The [NoteEditor](#) class is a widget encapsulating all the functionality necessary for showing and editing notes.

5.71.2 Member Function Documentation

5.71.2.1 backend()

```
INoteEditorBackend * quentier::NoteEditor::backend ( ) [noexcept]
```

Returns

the pointer to the note editor's backend

5.71.2.2 clear()

```
void quentier::NoteEditor::clear ( )
```

Clear the contents of the note editor

5.71.2.3 convertToNote

```
void quentier::NoteEditor::convertToNote ( ) [slot]
```

Invoke this slot to launch the asynchronous procedure of converting the current contents of the note editor to note; the convertedToNote signal would be emitted in response when the conversion is done

5.71.2.4 currentNoteLocalId()

```
QString quentier::NoteEditor::currentNoteLocalId ( ) const
```

Get the local id of the note currently set to the note editor

5.71.2.5 defaultFont()

```
const QFont * quentier::NoteEditor::defaultFont ( ) const
```

Returns

pointer to the default font used by the note editor; if no such font was set to the editor previously, returns null pointer

5.71.2.6 defaultPalette()

```
QPalette quentier::NoteEditor::defaultPalette ( ) const
```

Returns

palette containing default colors used by the editor; the palette is composed of colors from note editor widget's native palette but some of them might be overridden by colors from the palette specified previously via set↔DefaultPalette method: those colors from the specified palette which were valid

5.71.2.7 idleTime()

```
qint64 quentier::NoteEditor::idleTime ( ) const [noexcept]
```

Returns

the number of milliseconds since the last user's interaction with the note editor or -1 if there was no interaction or if no note is loaded at the moment

5.71.2.8 inAppNoteLinkPasteRequested

```
void quentier::NoteEditor::inAppNoteLinkPasteRequested (
    QString url,
    QString userId,
    QString shardId,
    QString noteGuid ) [signal]
```

inAppNoteLinkPasteRequested signal is emitted when the note editor detects the attempt to paste the in-app note link into the note editor; the link would not be inserted right away, instead this signal would be emitted. Whatever party managing the note editor is expected to connect some slot to this signal and provide the optionally amended link information to the note editor by sending the signal connected to its insertInAppNoteLink slot - this slot accepts both the URL of the link and the link text and performs the actual link insertion into the note. If the link text is empty, the URL itself is used as the link text.

5.71.2.9 initialize()

```
void quentier::NoteEditor::initialize (
    local_storage::ILocalStoragePtr localStorage,
    SpellChecker & spellChecker,
    const Account & account,
    QThread * backgroundJobsThread = nullptr,
    enml::IDecryptedTextCachePtr decryptedTextCache = nullptr )
```

[NoteEditor](#) requires [LocalStorageManagerAsync](#), [SpellChecker](#) and [Account](#) for its work but due to the particularities of Qt's .ui files processing these can't be passed right inside the constructor, hence here's a special initialization method

Parameters

<i>localStorage</i>	Local storage
<i>spellChecker</i>	Spell checker to be used by note editor
<i>account</i>	Current account
<i>backgroundJobsThread</i>	Pointer to the thread to be used for scheduling of background jobs of NoteEditor ; if null, NoteEditor 's background jobs would take place in GUI thread
<i>decryptedTextCache</i>	Decrypted text cache to be used for note rendering and text decryption. If null, a new instance would be created and used.

5.71.2.10 isEditorPageModified()

```
bool quentier::NoteEditor::isEditorPageModified ( ) const [noexcept]
```

Returns

true if there's content within the editor not yet converted to note, false otherwise

5.71.2.11 isModified()

```
bool quantier::NoteEditor::isModified ( ) const [noexcept]
```

Returns

true if there's content within the editor not yet converted to note or not saved to local storage, false otherwise

5.71.2.12 isNoteLoaded()

```
bool quantier::NoteEditor::isNoteLoaded ( ) const [noexcept]
```

Returns

true if the note last set to the editor has been fully loaded already, false otherwise

5.71.2.13 saveNoteToLocalStorage

```
void quantier::NoteEditor::saveNoteToLocalStorage ( ) [slot]
```

Invoke this slot to launch the asynchronous procedure of saving the modified current note back to the local storage. If no note is set to the editor or if the note is not modified, no action would be performed. Otherwise noteSaved↔ToLocalStorage signal would be emitted in case of successful saving or failedToSaveNoteToLocalStorage would be emitted otherwise

5.71.2.14 setAccount()

```
void quantier::NoteEditor::setAccount (
    const Account & account )
```

Set the current account to the note editor

5.71.2.15 setBackend()

```
void quantier::NoteEditor::setBackend (
    INoteEditorBackend * backend )
```

This method can be used to set the backend to the note editor; the note editor has the default backend so this method is not obligatory to be called

5.71.2.16 setCurrentNoteLocalId()

```
void quantier::NoteEditor::setCurrentNoteLocalId (
    const QString & noteLocalId )
```

Set note local id to the note editor. The note is being searched for within the local storage, in case of no note being found noteNotFound signal is emitted. Otherwise note editor page starts loading.

Parameters

<i>note</i> ↔ <i>LocalId</i>	The local id of note
---------------------------------	----------------------

5.71.2.17 setDefaultFont

```
void quantier::NoteEditor::setDefaultFont (
    const QFont & font ) [slot]
```

Sets the font which would be used by the editor by default

Parameters

<i>font</i>	The font to be used by the editor by default
-------------	----------------------------------------------

5.71.2.18 setDefaultPalette

```
void quantier::NoteEditor::setDefaultPalette (
    const QPalette & pal ) [slot]
```

Sets the palette with colors to be used by the editor. New colors are applied after the note is fully loaded. If no note is set to the editor, the palette is simply remembered for the next note to be loaded into it.

Colors within the palette and their usage:

1. WindowText - used as default font color
2. Base - used as default background color
3. HighlightedText - used as font color for selected text
4. Highlight - used as background color for selected text

Parameters

<i>pal</i>	The palette to be set. Invalid colors from it are substituted by colors from widget's palette by the editor
------------	-------------------------------------------------------------------------------------------------------------

5.71.2.19 setFocus()

```
void quantier::NoteEditor::setFocus ( )
```

Sets the focus to the backend note editor widget

5.71.2.20 setInitialPageHtml()

```
void quentier::NoteEditor::setInitialPageHtml (
    const QString & html )
```

Set the html to be displayed when the note is not set to the editor

5.71.2.21 setNoteDeletedPageHtml()

```
void quentier::NoteEditor::setNoteDeletedPageHtml (
    const QString & html )
```

Set the html to be displayed when the note set to the editor was deleted from the local storage (either marked as deleted or deleted permanently i.e. expunged)

5.71.2.22 setNoteLoadingPageHtml()

```
void quentier::NoteEditor::setNoteLoadingPageHtml (
    const QString & html )
```

Set the html to be displayed when the note set to the editor is being loaded into it

5.71.2.23 setNoteNotFoundPageHtml()

```
void quentier::NoteEditor::setNoteNotFoundPageHtml (
    const QString & html )
```

Set the html to be displayed when the note attempted to be set to the editor was not found within the local storage

5.71.2.24 setNoteTitle

```
void quentier::NoteEditor::setNoteTitle (
    const QString & noteTitle ) [slot]
```

Invoke this slot to set the title to the note displayed via the note editor. The note editor itself doesn't manage the note title in any way so any external code using the note editor can set the title to the note editor's note which would be considered modified if the title is new and then eventually the note would be saved to local storage

Parameters

<i>noteTitle</i>	The title of the note
------------------	-----------------------

5.71.2.25 setTagIds

```
void quentier::NoteEditor::setTagIds (
```

```
const QStringList & tagLocalIds,
const QStringList & tagGuids ) [slot]
```

Invoke this slot to set tag local ids and/or tag guides to the note displayed via the note editor. The note editor itself doesn't manage the note tags in any way so any external code using the note editor can set the tag ids to the note editor's internal note which would be considered modified if the tag ids are new and then eventually the note would be saved to local storage

Parameters

<i>tagLocalIds</i>	The list of tag local ids for the note
<i>tagGuids</i>	The list of tag guides for the note

5.71.2.26 setUndoStack()

```
void quantier::NoteEditor::setUndoStack (
    QUndoStack * pUndoStack )
```

Set the undo stack for the note editor to use

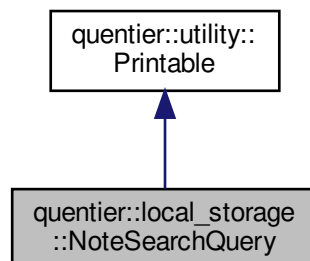
5.71.2.27 undoStack()

```
const QUndoStack * quantier::NoteEditor::undoStack ( ) const [noexcept]
```

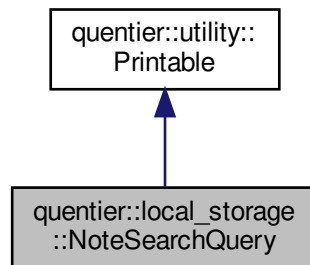
Get the undo stack serving to the note editor

5.72 quantier::local_storage::NoteSearchQuery Class Reference

Inheritance diagram for quantier::local_storage::NoteSearchQuery:



Collaboration diagram for `quentier::local_storage::NoteSearchQuery`:



Public Member Functions

- **NoteSearchQuery** (const [NoteSearchQuery](#) &other)
- **NoteSearchQuery** ([NoteSearchQuery](#) &&other) noexcept
- [NoteSearchQuery](#) & **operator=** (const [NoteSearchQuery](#) &other)
- [NoteSearchQuery](#) & **operator=** ([NoteSearchQuery](#) &&other) noexcept
- bool **isEmpty** () const
- void **clear** ()
- QString **queryString** () const
- bool **setQueryString** (const QString &[queryString](#), [ErrorString](#) &error)
- QString **notebookModifier** () const
- bool **hasAnyModifier** () const
- const QStringList & **tagNames** () const
- const QStringList & **negatedTagNames** () const
- bool **hasAnyTag** () const
- bool **hasNegatedAnyTag** () const
- const QStringList & **titleNames** () const
- const QStringList & **negatedTitleNames** () const
- bool **hasAnyTitleName** () const
- bool **hasNegatedAnyTitleName** () const
- const QList< qint64 > & **creationTimestamps** () const
- const QList< qint64 > & **negatedCreationTimestamps** () const
- bool **hasAnyCreationTimestamp** () const
- bool **hasNegatedAnyCreationTimestamp** () const
- const QList< qint64 > & **modificationTimestamps** () const
- const QList< qint64 > & **negatedModificationTimestamps** () const
- bool **hasAnyModificationTimestamp** () const
- bool **hasNegatedAnyModificationTimestamp** () const
- const QStringList & **resourceMimeTypes** () const
- const QStringList & **negatedResourceMimeTypes** () const
- bool **hasAnyResourceMimeType** () const
- bool **hasNegatedAnyResourceMimeType** () const
- const QList< qint64 > & **subjectDateTimestamps** () const
- const QList< qint64 > & **negatedSubjectDateTimestamps** () const
- bool **hasAnySubjectDateTimestamp** () const
- bool **hasNegatedAnySubjectDateTimestamp** () const

- const QList< double > & **latitudes** () const
- const QList< double > & **negatedLatitudes** () const
- bool **hasAnyLatitude** () const
- bool **hasNegatedAnyLatitude** () const
- const QList< double > & **longitudes** () const
- const QList< double > & **negatedLongitudes** () const
- bool **hasAnyLongitude** () const
- bool **hasNegatedAnyLongitude** () const
- const QList< double > & **altitudes** () const
- const QList< double > & **negatedAltitudes** () const
- bool **hasAnyAltitude** () const
- bool **hasNegatedAnyAltitude** () const
- const QStringList & **authors** () const
- const QStringList & **negatedAuthors** () const
- bool **hasAnyAuthor** () const
- bool **hasNegatedAnyAuthor** () const
- const QStringList & **sources** () const
- const QStringList & **negatedSources** () const
- bool **hasAnySource** () const
- bool **hasNegatedAnySource** () const
- const QStringList & **sourceApplications** () const
- const QStringList & **negatedSourceApplications** () const
- bool **hasAnySourceApplication** () const
- bool **hasNegatedAnySourceApplication** () const
- const QStringList & **contentClasses** () const
- const QStringList & **negatedContentClasses** () const
- bool **hasAnyContentClass** () const
- bool **hasNegatedAnyContentClass** () const
- const QStringList & **placeNames** () const
- const QStringList & **negatedPlaceNames** () const
- bool **hasAnyPlaceName** () const
- bool **hasNegatedAnyPlaceName** () const
- const QStringList & **applicationData** () const
- const QStringList & **negatedApplicationData** () const
- bool **hasAnyApplicationData** () const
- bool **hasNegatedAnyApplicationData** () const
- const QList< qint64 > & **reminderOrders** () const
- const QList< qint64 > & **negatedReminderOrders** () const
- bool **hasAnyReminderOrder** () const
- bool **hasNegatedAnyReminderOrder** () const
- const QList< qint64 > & **reminderTimes** () const
- const QList< qint64 > & **negatedReminderTimes** () const
- bool **hasAnyReminderTime** () const
- bool **hasNegatedAnyReminderTime** () const
- const QList< qint64 > & **reminderDoneTimes** () const
- const QList< qint64 > & **negatedReminderDoneTimes** () const
- bool **hasAnyReminderDoneTime** () const
- bool **hasNegatedAnyReminderDoneTime** () const
- bool **hasUnfinishedToDo** () const
- bool **hasNegatedUnfinishedToDo** () const
- bool **hasFinishedToDo** () const
- bool **hasNegatedFinishedToDo** () const
- bool **hasAnyToDo** () const
- bool **hasNegatedAnyToDo** () const
- bool **hasEncryption** () const

- bool **hasNegatedEncryption** () const
- const QStringList & **contentSearchTerms** () const
- const QStringList & **negatedContentSearchTerms** () const
- bool **hasAnyContentSearchTerms** () const
- bool **isMatcheable** () const
- QTextStream & **print** (QTextStream &strm) const override

5.72.1 Member Function Documentation

5.72.1.1 notebookModifier()

```
QString quentier::local_storage::NoteSearchQuery::notebookModifier ( ) const
```

If query string has "notebook:<notebook name>" scope modifier, this method returns the name of the notebook, otherwise it returns empty string

5.72.1.2 print()

```
QTextStream & quentier::local_storage::NoteSearchQuery::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

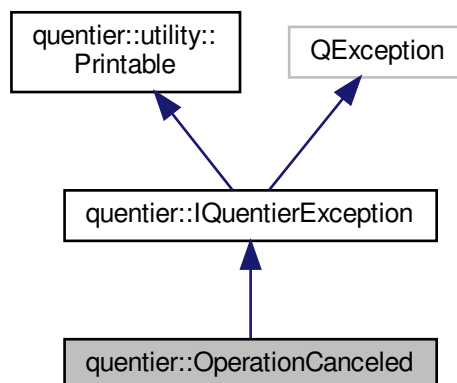
5.72.1.3 queryString()

```
QString quentier::local_storage::NoteSearchQuery::queryString ( ) const
```

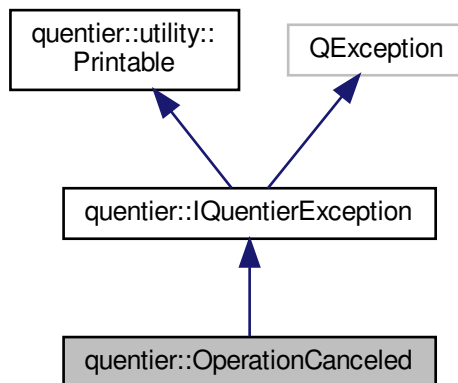
Returns the original non-parsed query string

5.73 quentier::OperationCanceled Class Reference

Inheritance diagram for quentier::OperationCanceled:



Collaboration diagram for `quentier::OperationCanceled`:



Public Member Functions

- `OperationCanceled * clone ()` const override
- `void raise ()` const override

Protected Member Functions

- `QString exceptionDisplayName ()` const override

5.73.1 Member Function Documentation

5.73.1.1 `exceptionDisplayName()`

```
QString quentier::OperationCanceled::exceptionDisplayName ( ) const [override], [protected],  
[virtual]
```

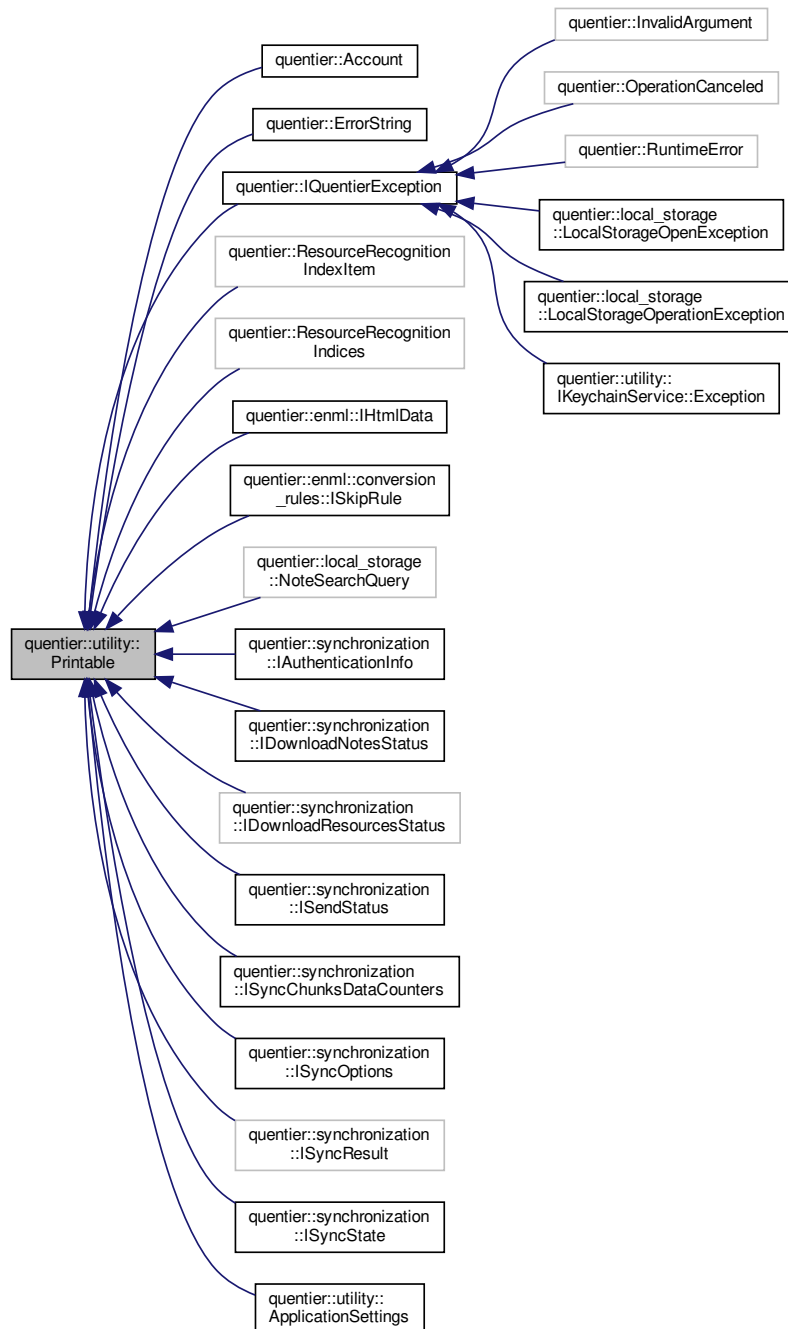
Implements `quentier::IQuentierException`.

5.74 quantier::utility::Printable Class Reference

The [Printable](#) class is the interface for Quantier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

```
#include <Printable.h>
```

Inheritance diagram for quantier::utility::Printable:



Public Member Functions

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const Printable &printable)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &debug, const Printable &printable)

5.74.1 Detailed Description

The [Printable](#) class is the interface for Quentier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

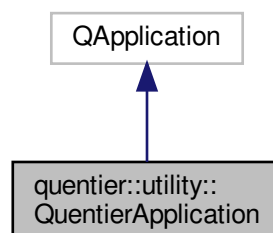
5.75 QPromise< T > Class Template Reference

Public Member Functions

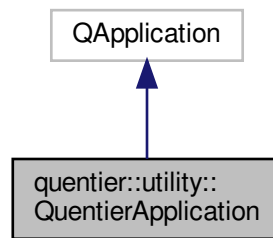
- **QPromise** (QPromise< T > &&other) noexcept
- **QPromise** (QFutureInterface< T > &other)
- **QPromise** & **operator=** (QPromise< T > &&other) noexcept
- QFuture< T > **future** () const
- template<typename U , typename = std::enable_if_t< std::is_same_v<U, T> || std::is_convertible_v<U, T>>> void **addResult** (U &&result, int index=-1)
- void **setException** (const QException &e)
- void **start** ()
- void **finish** ()
- void **suspendIfRequested** ()
- bool **isCanceled** () const
- void **setProgressRange** (int minimum, int maximum)
- void **setProgressValue** (int progressValue)
- void **setProgressValueAndText** (int progressValue, const QString &progressText)
- void **swap** (QPromise< T > &other) noexcept

5.76 quentier::utility::QuentierApplication Class Reference

Inheritance diagram for quentier::utility::QuentierApplication:



Collaboration diagram for `quentier::utility::QuentierApplication`:



Public Member Functions

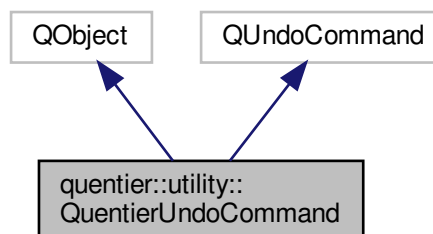
- **QuentierApplication** (int &argc, char *argv[])
- bool **notify** (QObject *object, QEvent *event) override
- bool **event** (QEvent *event) override

5.77 quentier::utility::QuentierUndoCommand Class Reference

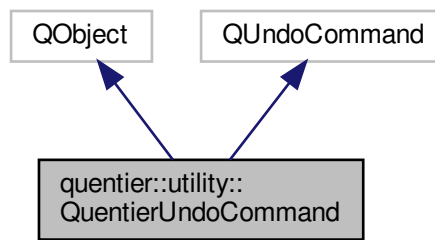
The [QuentierUndoCommand](#) class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push `QUndoCommand` to `QUndoStack`, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

```
#include <QuentierUndoCommand.h>
```

Inheritance diagram for `quentier::utility::QuentierUndoCommand`:



Collaboration diagram for `quentier::utility::QuentierUndoCommand`:



Signals

- void **notifyError** ([ErrorString](#) error)

Public Member Functions

- **QuentierUndoCommand** (`QUndoCommand *parent=nullptr`)
- **QuentierUndoCommand** (`const QString &text, QUndoCommand *parent=nullptr`)
- void **undo** () final
- void **redo** () final
- bool **onceUndoExecuted** () const noexcept

Protected Member Functions

- virtual void **undoImpl** ()=0
- virtual void **redoImpl** ()=0

5.77.1 Detailed Description

The [QuentierUndoCommand](#) class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push `QUndoCommand` to `QUndoStack`, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

The rationale behind the current behaviour seems to be the compliance with "command pattern behaviour" when you create the command to execute the action instead of just executing it immediately. This design is enforced by Qt's undo/redo framework, there's no option to choose not to call "redo" when pushing to the stack.

One thing which this design fails to see is the fact that the command may be already executed externally by the moment the `QUndoCommand` can be created. Suppose we can get the information about how to undo (and then again redo) that command. We create the corresponding `QUndoCommand`, set up the stuff for its undo/redo methods and push it to `QUndoStack` for future use... But at the same time `QUndoStack` calls "redo" method of the command. Really not the behaviour you'd like to have.

[QuentierUndoCommand](#) is also `QObject`, it is for error reporting via `notifyError` signal

5.78 `quentier::synchronization::RateLimitReachedError` Struct Reference

```
#include <Errors.h>
```

Public Attributes

- `std::optional< quint32 >` [`rateLimitDurationSec`](#)

5.78.1 Detailed Description

Information about "API rate limit reached" error which Evernote servers might return if too much of their API calls were made recently. In case of such error synchronization should be repeated later, after some time passes.

5.78.2 Member Data Documentation

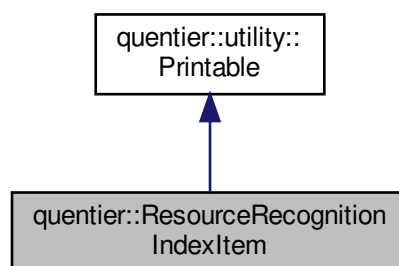
5.78.2.1 `rateLimitDurationSec`

```
std::optional<quint32> quentier::synchronization::RateLimitReachedError::rateLimitDurationSec
```

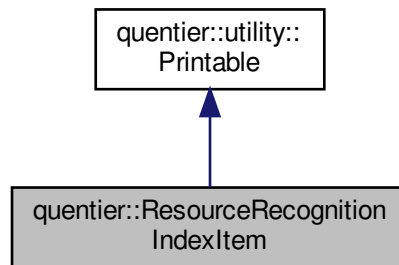
Number of seconds during which since the current moment during which any call to Evernote API would again result in "API rate limit reached" error i.e. the number of seconds to wait for before the next attempt to run synchronization

5.79 `quentier::ResourceRecognitionIndexItem` Class Reference

Inheritance diagram for `quentier::ResourceRecognitionIndexItem`:



Collaboration diagram for quantier::ResourceRecognitionIndexItem:



Classes

- struct [IBarcodeItem](#)
- struct [IObjectItem](#)
- struct [IShapeItem](#)
- struct [ITextItem](#)

Public Types

- using `ITextItemPtr` = `std::shared_ptr< ITextItem >`
- using `IObjectItemPtr` = `std::shared_ptr< IObjectItem >`
- using `IShapeItemPtr` = `std::shared_ptr< IShapeItem >`
- using `IBarcodeItemPtr` = `std::shared_ptr< IBarcodeItem >`

Public Member Functions

- `ResourceRecognitionIndexItem` (const [ResourceRecognitionIndexItem](#) &other)
- `ResourceRecognitionIndexItem` ([ResourceRecognitionIndexItem](#) &&other) noexcept
- [ResourceRecognitionIndexItem](#) & `operator=` (const [ResourceRecognitionIndexItem](#) &other)
- [ResourceRecognitionIndexItem](#) & `operator=` ([ResourceRecognitionIndexItem](#) &&other) noexcept
- bool `isValid` () const
- int `x` () const
- void `setX` (int x)
- int `y` () const
- void `setY` (int y)
- int `h` () const
- void `setH` (int h)
- int `w` () const
- void `setW` (int w)
- int `offset` () const
- void `setOffset` (int offset)
- int `duration` () const
- void `setDuration` (int duration)
- QList< int > `strokes` () const

- void **setStrokes** (QList< int > strokes)
- QList< ITextItemPtr > **textItems** () const
- void **setTextItems** (QList< ITextItemPtr > textItems)
- QList< IObjectItemPtr > **objectItems** () const
- void **setObjectItems** (QList< IObjectItemPtr > objectItems)
- QList< IShapeItemPtr > **shapeItems** () const
- void **setShapeItems** (QList< IShapeItemPtr > shapeItems)
- QList< IBarcodeItemPtr > **barcodeItems** () const
- void **setBarcodeItems** (QList< IBarcodeItemPtr > barcodeItems)
- QTextStream & **print** (QTextStream &strm) const override

5.79.1 Member Function Documentation

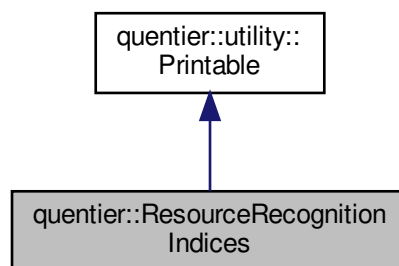
5.79.1.1 print()

```
QTextStream & quentier::ResourceRecognitionIndexItem::print (
    QTextStream & strm ) const [override], [virtual]
```

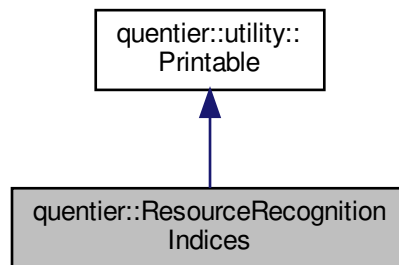
Implements [quentier::utility::Printable](#).

5.80 quentier::ResourceRecognitionIndices Class Reference

Inheritance diagram for quentier::ResourceRecognitionIndices:



Collaboration diagram for quantier::ResourceRecognitionIndices:



Public Member Functions

- **ResourceRecognitionIndices** (const QByteArray &rawRecognitionIndicesData)
- **ResourceRecognitionIndices** (const [ResourceRecognitionIndices](#) &other)
- **ResourceRecognitionIndices** ([ResourceRecognitionIndices](#) &&other) noexcept
- [ResourceRecognitionIndices](#) & **operator=** (const [ResourceRecognitionIndices](#) &other)
- [ResourceRecognitionIndices](#) & **operator=** ([ResourceRecognitionIndices](#) &&other) noexcept
- bool **isNull** () const
- bool **isValid** () const
- QString **objectId** () const
- QString **objectType** () const
- QString **recoType** () const
- QString **engineVersion** () const
- QString **docType** () const
- QString **lang** () const
- int **objectHeight** () const
- int **objectWidth** () const
- QVector< [ResourceRecognitionIndexItem](#) > **items** () const
- bool **setData** (const QByteArray &rawRecognitionIndicesData)
- QTextStream & [print](#) (QTextStream &strm) const override

5.80.1 Member Function Documentation

5.80.1.1 print()

```

QTextStream & quantier::ResourceRecognitionIndices::print (
    QTextStream & strm ) const [override], [virtual]
  
```

Implements [quantier::utility::Printable](#).

5.81 `quantier::Result< ValueType, ErrorType, typename >` Class Template Reference

Public Member Functions

- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr>`
Result (T1 t)
- **Result** (ErrorType error)
- **Result** (const [Result](#)< ValueType, ErrorType > &other)
- **Result** ([Result](#)< ValueType, ErrorType > &&other)
- [Result](#) & **operator=** (const [Result](#)< ValueType, ErrorType > &other)
- [Result](#) & **operator=** ([Result](#)< ValueType, ErrorType > &&other)
- `bool` [isValid](#) () const noexcept
- **operator bool** () const noexcept
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr>`
T1 & **get** ()
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr>`
const T1 & **get** () const
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr>`
T1 & **operator*** ()
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr>`
const T1 & **operator*** () const
- `const ErrorType` & **error** () const
- `ErrorType` & **error** ()

5.81.1 Member Function Documentation

5.81.1.1 `isValid()`

```
template<class ValueType , class ErrorType , typename = typename std::enable_if_t<!std::is_↵
void_v<std::decay_t<ErrorType>>>>
bool quantier::Result< ValueType, ErrorType, typename >::isValid ( ) const [inline], [noexcept]
```

Returns

boolean value indicating whether the result contains a value

5.82 `quantier::threading::detail::ResultTypeHelper< F, Arg, Enable >` Struct Template Reference

5.83 `quantier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > >` Struct Template Reference

Public Types

- using **ResultType** = `std::invoke_result_t< std::decay_t< F >, std::decay_t< Arg > >`

5.84 `quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference`

Public Types

- using **ResultType** = `std::invoke_result_t< std::decay_t< F >, QFuture< Arg > >`

5.85 `quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference`

Public Types

- using **ResultType** = `std::invoke_result_t< std::decay_t< F > >`

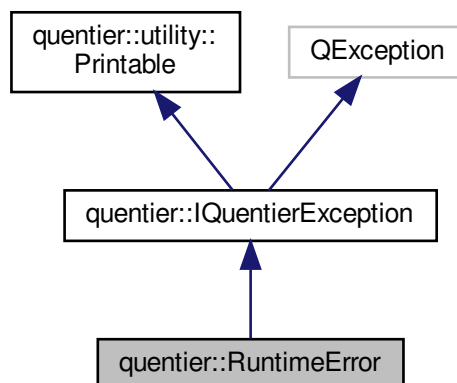
5.86 `quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference`

Public Types

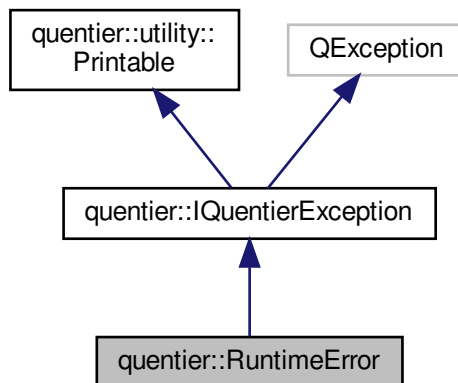
- using **ResultType** = `std::invoke_result_t< std::decay_t< F >, QFuture< void > >`

5.87 `quentier::RuntimeError` Class Reference

Inheritance diagram for `quentier::RuntimeError`:



Collaboration diagram for `quentier::RuntimeError`:



Public Member Functions

- **RuntimeError** ([ErrorString](#) message)
- [RuntimeError](#) * **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString [exceptionDisplayName](#) () const override

5.87.1 Member Function Documentation

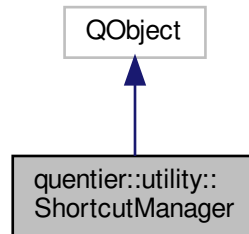
5.87.1.1 exceptionDisplayName()

```
QString quentier::RuntimeError::exceptionDisplayName ( ) const [override], [protected], [virtual]
```

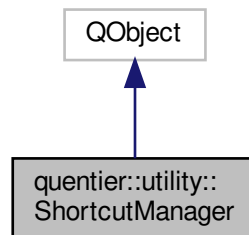
Implements [quentier::IQuentierException](#).

5.88 quantier::utility::ShortcutManager Class Reference

Inheritance diagram for quantier::utility::ShortcutManager:



Collaboration diagram for quantier::utility::ShortcutManager:



Public Types

- enum **QuantierShortcutKey** {
NewNote = 5000 , **NewTag** , **NewNotebook** , **NewSavedSearch** ,
AddAttachment , **SaveAttachment** , **OpenAttachment** , **CopyAttachment** ,
CutAttachment , **RemoveAttachment** , **RenameAttachment** , **AddAccount** ,
ExitAccount , **SwitchAccount** , **AccountInfo** , **NoteSearch** ,
NewNoteSearch , **ShowNotes** , **ShowNotebooks** , **ShowTags** ,
ShowSavedSearches , **ShowDeletedNotes** , **ShowStatusBar** , **ShowToolBar** ,
PasteUnformatted , **Font** , **UpperIndex** , **LowerIndex** ,
AlignLeft , **AlignCenter** , **AlignRight** , **AlignFull** ,
IncreaseIndentation , **DecreaseIndentation** , **IncreaseFontSize** , **DecreaseFontSize** ,
InsertNumberedList , **InsertBulletedList** , **Strikethrough** , **Highlight** ,
InsertTable , **InsertRow** , **InsertColumn** , **RemoveRow** ,
RemoveColumn , **InsertHorizontalLine** , **InsertToDoTag** , **EditHyperlink** ,
CopyHyperlink , **RemoveHyperlink** , **Encrypt** , **Decrypt** ,
DecryptPermanently , **BackupLocalStorage** , **RestoreLocalStorage** , **UpgradeLocalStorage** ,

LocalStorageStatus , **SpellCheck** , **SpellCheckIgnoreWord** , **SpellCheckAddWordToUserDictionary** , **SaveImage** , **AnnotateImage** , **ImageRotateClockwise** , **ImageRotateCounterClockwise** , **Synchronize** , **FullSync** , **ImportFolders** , **Preferences** , **ReleaseNotes** , **ViewLogs** , **About** , **UnknownKey** = 100000 }

Public Slots

- void **setUserShortcut** (int key, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})
- void **setNonStandardUserShortcut** (QString nonStandardKey, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})
- void **setDefaultShortcut** (int key, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})
- void **setNonStandardDefaultShortcut** (QString nonStandardKey, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})

Signals

- void **shortcutChanged** (int key, QKeySequence [shortcut](#), const [Account](#) &account, QString context)
- void **nonStandardShortcutChanged** (QString nonStandardKey, QKeySequence [shortcut](#), const [Account](#) &account, QString context)

Public Member Functions

- **ShortcutManager** (QObject *parent=nullptr)
- QKeySequence [shortcut](#) (int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [shortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const
- QKeySequence [defaultShortcut](#) (int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [defaultShortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const
- QKeySequence [userShortcut](#) (int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [userShortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const

5.88.1 Member Function Documentation

5.88.1.1 defaultShortcut() [1/2]

```
QKeySequence quentier::utility::ShortcutManager::defaultShortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {} ) const
```

Returns

Default shortcut for the non-standard key if present, otherwise empty key sequence

5.88.1.2 defaultShortcut() [2/2]

```
QKeySequence quotientier::utility::ShortcutManager::defaultShortcut (
    int key,
    const Account & account,
    const QString & context = {} ) const
```

Returns

Default shortcut for the standard key if present, otherwise empty key sequence

5.88.1.3 shortcut() [1/2]

```
QKeySequence quotientier::utility::ShortcutManager::shortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {} ) const
```

Returns

Active shortcut for the non-standard key - either the user defined shortcut (if present) or the default one (if present as well)

5.88.1.4 shortcut() [2/2]

```
QKeySequence quotientier::utility::ShortcutManager::shortcut (
    int key,
    const Account & account,
    const QString & context = {} ) const
```

Returns

Active shortcut for the standard key - either the user defined shortcut (if present) or the default one (if present as well)

5.88.1.5 userShortcut() [1/2]

```
QKeySequence quotientier::utility::ShortcutManager::userShortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {} ) const
```

Returns

User defined shortcut for the non-standard key if present, otherwise empty key sequence

5.88.1.6 userShortcut() [2/2]

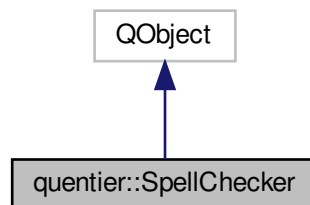
```
QKeySequence quentier::utility::ShortcutManager::userShortcut (
    int key,
    const Account & account,
    const QString & context = {} ) const
```

Returns

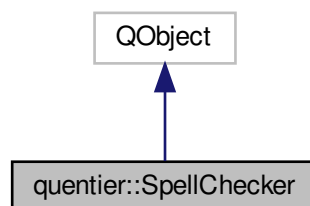
User defined shortcut for the standard key if present, otherwise empty key sequence

5.89 quentier::SpellChecker Class Reference

Inheritance diagram for quentier::SpellChecker:



Collaboration diagram for quentier::SpellChecker:



Signals

- void **ready** ()

Public Member Functions

- **SpellChecker** ([utility::FileIOProcessorAsync](#) *fileIOProcessorAsync, [Account](#) account, QObject *parent=nullptr, const QString &userDictionaryPath={})
- QList< std::pair< QString, bool > > **listAvailableDictionaries** () const
- void **setAccount** (const [Account](#) &account)
- void **enableDictionary** (const QString &language)
- void **disableDictionary** (const QString &language)
- bool **checkSpell** (const QString &word) const
- QStringList **spellCorrectionSuggestions** (const QString &misSpelledWord) const
- void **addToUserWordlist** (const QString &word)
- void **removeFromUserWordList** (const QString &word)
- void **ignoreWord** (const QString &word)
- void **removeWord** (const QString &word)
- bool **isReady** () const noexcept

5.90 quotient::utility::StringUtils Class Reference

Public Member Functions

- void **removePunctuation** (QString &str, const QList< QChar > &charactersToPreserve={}) const
- void **removeDiacritics** (QString &str) const
- void **removeNewlines** (QString &str) const

5.91 quotient::utility::SysInfo Class Reference

Public Member Functions

- qint64 **pageSize** ()
- qint64 **totalMemory** ()
- qint64 **freeMemory** ()
- QString **stackTrace** ()
- QString **platformName** ()

5.92 quotient::threading::TrackedTask< LockableObject, Function > Class Template Reference

```
#include <TrackedTask.h>
```

Public Member Functions

- template<typename SomeLockableObject, typename SomeFunction >
constexpr **TrackedTask** (SomeLockableObject &&someLockableObject, SomeFunction &&function)
- template<typename... Arguments, typename = std::enable_if_t< std::is_invocable_v<Function, Arguments...> || std::is_member_↵
function_pointer_v<Function>>>>
constexpr void **operator()** (Arguments &&... arguments)
- template<typename... Arguments, typename = std::enable_if_t< std::is_invocable_v<Function, Arguments...> || std::is_member_↵
function_pointer_v<Function>>>>
constexpr void **operator()** (Arguments &&... arguments) const

5.92.1 Detailed Description

```
template<typename LockableObject, typename Function>
class quantier::threading::TrackedTask< LockableObject, Function >
```

Wrapper class which automates checking for the state of a lockable object. With this class code like this

```
auto task = [selfWeak = weak_from_this()] { auto self = selfWeak.lock(); if (!self) { return; } // otherwise do something
};
```

can be written like this:

```
auto task = threading::TrackedTask{weak_from_this(), &MyClass::someMethod};
```

5.93 quantier::utility::UuidGenerator Class Reference

Static Public Member Functions

- static QString **generate** ()
- static QString **uidToString** (const QUuid &uid)

5.94 quantier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine Struct Reference

The [UseMine](#) conflict resolution means "override theirs version with mine version".

```
#include <ISyncConflictResolver.h>
```

5.94.1 Detailed Description

The [UseMine](#) conflict resolution means "override theirs version with mine version".

5.95 quantier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs Struct Reference

The [UseTheirs](#) conflict resolution means "override mine version with theirs version".

```
#include <ISyncConflictResolver.h>
```

5.95.1 Detailed Description

The [UseTheirs](#) conflict resolution means "override mine version with theirs version".

Chapter 6

File Documentation

6.1 ISkipRule.h

```
1 /*
2  * Copyright 2023-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/conversion_rules/MatchMode.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QtGlobal>
25
26 namespace quentier::enml::conversion_rules {
27
28 class ISkipRule : public utility::Printable
29 {
30 public:
31     ~ISkipRule() override;
32
33     enum class Target
34     {
35         Element,
36         AttributeName,
37         AttributeValue
38     };
39
40     friend QUINTIER_EXPORT QTextStream & operator<<(
41         QTextStream & strm, Target target);
42
43     friend QUINTIER_EXPORT QDebug & operator<<(QDebug & dbg, Target target);
44
45     [[nodiscard]] virtual Target target() const = 0;
46
47     [[nodiscard]] virtual QString value() const = 0;
48
49     [[nodiscard]] virtual MatchMode matchMode() const = 0;
50
51     [[nodiscard]] virtual bool includeContents() const = 0;
52
53     [[nodiscard]] virtual Qt::CaseSensitivity caseSensitivity() const = 0;
54
55 public: // utility::Printable
56     QTextStream & print(QTextStream & strm) const override;
57 };
58
59 } // namespace quentier::enml::conversion_rules
```

6.2 ISkipRuleBuilder.h

```

1  /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/conversion_rules/Fwd.h>
22 #include <quentier/enml/conversion_rules/ISkipRule.h>
23 #include <quentier/utility/Linkage.h>
24
25 namespace quentier::enml::conversion_rules {
26
27 class QUENTIER_EXPORT ISkipRuleBuilder
28 {
29 public:
30     virtual ~ISkipRuleBuilder();
31
32     virtual ISkipRuleBuilder & setTarget(ISkipRule::Target target) = 0;
33     virtual ISkipRuleBuilder & setValue(QString value) = 0;
34     virtual ISkipRuleBuilder & setMatchMode(MatchMode matchMode) = 0;
35     virtual ISkipRuleBuilder & setIncludeContents(bool includeContents) = 0;
36     virtual ISkipRuleBuilder & setCaseSensitivity(
37         Qt::CaseSensitivity caseSensitivity) = 0;
38
39     [[nodiscard]] virtual ISkipRulePtr build() = 0;
40 };
41
42 } // namespace quentier::enml::conversion_rules

```

6.3 MatchMode.h

```

1  /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 class QDebug;
24 class QTextStream;
25
26 namespace quentier::enml::conversion_rules {
27
28 enum class MatchMode
29 {
30     Equals,
31     StartsWith,
32     EndsWith,
33     Contains
34 };
35
36 QUENTIER_EXPORT QTextStream & operator<<

```

```

52     QTextStream & strm, MatchMode matchMode);
53
54 QUINTIER_EXPORT QDebug & operator«(QDebug & dbg, MatchMode matchMode);
55
56 } // namespace quantier::enml::conversion_rules

```

6.4 enml/conversion_rules/Factory.h

```

1  /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/enml/conversion_rules/Fwd.h>
22 #include <quantier/utility/Linkage.h>
23
24 namespace quantier::enml::conversion_rules {
25
26 [[nodiscard]] QUINTIER_EXPORT ISkipRuleBuilderPtr createSkipRuleBuilder();
27
28 } // namespace quantier::enml::conversion_rules

```

6.5 enml/Factory.h

```

1  /*
2  * Copyright 2023-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/enml/Fwd.h>
22 #include <quantier/utility/Fwd.h>
23 #include <quantier/utility/Linkage.h>
24
25 namespace quantier::enml {
26
27 [[nodiscard]] QUINTIER_EXPORT IDecryptedTextCachePtr
28     createDecryptedTextCache(utility::IEncryptorPtr encryptor = nullptr);
29
30 [[nodiscard]] QUINTIER_EXPORT IENMLTagsConverterPtr createEnmlTagsConverter();
31
32 [[nodiscard]] QUINTIER_EXPORT IConverterPtr
33     createConverter(IENMLTagsConverterPtr enmlTagsConverter = nullptr);
34
35 } // namespace quantier::enml

```

6.6 local_storage/Factory.h

```

1 /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/threading/Fwd.h>
23 #include <quentier/types/Fwd.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <QtGlobal>
27
28 class QDir;
29
30 namespace quentier::local_storage {
31
32 [[nodiscard]] QUENTIER_EXPORT ILocalStoragePtr createSqliteLocalStorage(
33     const Account & account, const QDir & localStorageDir,
34     threading::QThreadPtr thread = {});
35
36 } // namespace quentier::local_storage

```

6.7 synchronization/Factory.h

```

1 /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/synchronization/Fwd.h>
23 #include <quentier/threading/Fwd.h>
24 #include <quentier/utility/Fwd.h>
25 #include <quentier/utility/Linkage.h>
26
27 #include <qevercloud/Fwd.h>
28
29 #include <QString>
30 #include <QUrl>
31 #include <QtGlobal>
32
33 class QWidget;
34
35 namespace quentier::synchronization {
36
37 [[nodiscard]] QUENTIER_EXPORT IAuthenticatorPtr createQEverCloudAuthenticator(
38     QString consumerKey, QString consumerSecret, QUrl serverUrl,
39     threading::QThreadPtr uiThread, QWidget * parentWidget = nullptr);
40
41 [[nodiscard]] QUENTIER_EXPORT ISynchronizerPtr createSynchronizer(
42     const QUrl & userStoreUrl, IAuthenticatorPtr authenticator,

```

```

43     ISyncStateStoragePtr syncStateStorage = nullptr,
44     utility::IKeychainServicePtr keychainService = nullptr,
45     INoteStoreFactoryPtr noteStoreFactory = nullptr,
46     IUserStoreFactoryPtr userStoreFactory = nullptr,
47     qevercloud::IRequestContextPtr ctx = nullptr,
48     qevercloud::IRetryPolicyPtr retryPolicy = nullptr);
49
50 [[nodiscard]] QUENTIER_EXPORT ISyncConflictResolverPtr
51     createSimpleSyncConflictResolver(
52         local_storage::ILocalStoragePtr localStorage);
53
54 [[nodiscard]] QUENTIER_EXPORT ISyncStateStoragePtr
55     createSyncStateStorage(QObject * parent = nullptr);
56
57 } // namespace quentier::synchronization

```

6.8 threading/Factory.h

```

1 /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/threading/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 namespace quentier::threading {
25
26 [[nodiscard]] QUENTIER_EXPORT QThreadPoolPtr globalThreadPool();
27
28 } // namespace quentier::threading

```

6.9 utility/Factory.h

```

1 /*
2  * Copyright 2024-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QString>
25
26 namespace quentier::utility {
27
28 [[nodiscard]] QUENTIER_EXPORT IEncryptorPtr createOpenSslEncryptor();
29
30 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newQtKeychainService();

```

```

38
43 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr
44     newObfuscatingKeychainService();
45
46 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newCompositeKeychainService(
47     QString name, IKeychainServicePtr primaryKeychain,
48     IKeychainServicePtr secondaryKeychain);
49
50 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newMigratingKeychainService(
51     IKeychainServicePtr sourceKeychain, IKeychainServicePtr sinkKeychain);
52
53 } // namespace quentier::utility

```

6.10 HtmlUtils.h

```

1 /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorMessage.h>
22 #include <quentier/types/Result.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QFlags>
26
27 #include <memory>
28
29 namespace quentier::enml::utils {
30
31 [[nodiscard]] Result<QString, ErrorMessage> QUENTIER_EXPORT
32     convertHtmlToXml(const QString & html);
33
34 [[nodiscard]] Result<QString, ErrorMessage> QUENTIER_EXPORT
35     convertHtmlToXhtml(const QString & html);
36
37 [[nodiscard]] Result<QString, ErrorMessage> QUENTIER_EXPORT
38     cleanupHtml(const QString & html);
39
40 enum class EscapeStringOption
41 {
42     Simplify = 1 << 0,
43 };
44
45 Q_DECLARE_FLAGS(EscapeStringOptions, EscapeStringOption);
46
47 [[nodiscard]] QString QUENTIER_EXPORT htmlEscapeString(
48     QString str, EscapeStringOptions options = EscapeStringOptions{});
49
50 } // namespace quentier::enml::utils

```

6.11 IConverter.h

```

1 /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of

```

```

12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/Fwd.h>
22 #include <quentier/enml/conversion_rules/Fwd.h>
23 #include <quentier/types/ErrorMessage.h>
24 #include <quentier/types/Result.h>
25 #include <quentier/utility/Linkage.h>
26
27 #include <QList>
28 #include <QStringList>
29 #include <QTextDocument>
30
31 #include <qevercloud/types/Note.h>
32
33 namespace quentier::enml {
34
35     class QUENTIER_EXPORT IConverter
36     {
37     public:
38         virtual ~IConverter();
39
40         [[nodiscard]] virtual Result<QString, ErrorMessage> convertHtmlToEnml(
41             const QString & html, IDecryptedTextCache & decryptedTextCache,
42             const QList<conversion_rules::ISkipRulePtr> & skipRules = {}) const = 0;
43
44         [[nodiscard]] virtual Result<void, ErrorMessage> convertHtmlToDoc(
45             const QString & html, QTextDocument & doc,
46             const QList<conversion_rules::ISkipRulePtr> & skipRules = {}) const = 0;
47
48         [[nodiscard]] virtual Result<QString, ErrorMessage> convertHtmlToXml(
49             const QString & html) const = 0;
50
51         [[nodiscard]] virtual Result<QString, ErrorMessage> convertHtmlToXhtml(
52             const QString & html) const = 0;
53
54         [[nodiscard]] virtual Result<IHtmlDataPtr, ErrorMessage> convertEnmlToHtml(
55             const QString & enml,
56             IDecryptedTextCache & decryptedTextCache) const = 0;
57
58         [[nodiscard]] virtual Result<QString, ErrorMessage> convertEnmlToPlainText(
59             const QString & enml) const = 0;
60
61         [[nodiscard]] virtual Result<QStringList, ErrorMessage>
62             convertEnmlToWordsList(const QString & enml) const = 0;
63
64         [[nodiscard]] virtual QStringList convertPlainTextToWordsList(
65             const QString & plainText) const = 0;
66
67         [[nodiscard]] virtual Result<void, ErrorMessage> validateEnml(
68             const QString & enml) const = 0;
69
70         [[nodiscard]] virtual Result<QString, ErrorMessage> validateAndFixupEnml(
71             const QString & enml) const = 0;
72
73         enum class EnexExportTags
74         {
75             Yes = 0,
76             No
77         };
78
79         [[nodiscard]] virtual Result<QString, ErrorMessage> exportNotesToEnex(
80             const QList<qevercloud::Note> & notes,
81             const QHash<QString, QString> & tagNameByTagLocalIds,
82             EnexExportTags exportTagsOption,
83             const QString & version = {}) const = 0;
84
85         [[nodiscard]] virtual Result<QList<qevercloud::Note>, ErrorMessage>
86             importEnex(const QString & enex) const = 0;
87     };
88 } // namespace quentier::enml

```

6.12 IDecryptedTextCache.h

```

1 /*
2 * Copyright 2016-2025 Dmitry Ivanov

```

```

3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/IEncryptor.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QString>
25
26 #include <optional>
27 #include <utility>
28
29 class QDebug;
30 class QTextStream;
31
32 namespace quentier::enml {
33
34 class QUENTIER_EXPORT IDecryptedTextCache
35 {
36 public:
37     virtual ~IDecryptedTextCache();
38
39     enum class RememberForSession
40     {
41         Yes,
42         No
43     };
44
45     friend QUENTIER_EXPORT QDebug & operator«(
46         QDebug & dbg, RememberForSession rememberForSession);
47
48     friend QUENTIER_EXPORT QTextStream & operator«(
49         QTextStream & strm, RememberForSession rememberForSession);
50
51     virtual void addDecryptexTextInfo(
52         const QString & encryptedText, const QString & decryptedText,
53         const QString & passphrase, utility::IEncryptor::Cipher cipher,
54         RememberForSession rememberForSession) = 0;
55
56     [[nodiscard]] virtual std::optional<std::pair<QString, RememberForSession>>
57         findDecryptedTextInfo(const QString & encryptedText) const = 0;
58
59     [[nodiscard]] virtual std::optional<QString> updateDecryptedTextInfo(
60         const QString & originalEncryptedText,
61         const QString & newDecryptedText) = 0;
62
63     [[nodiscard]] virtual bool containsRememberedForSessionEntries() const = 0;
64
65     virtual void removeDecryptedTextInfo(const QString & encryptedText) = 0;
66     virtual void clearNonRememberedForSessionEntries() = 0;
67 };
68
69 } // namespace quentier::enml

```

6.13 IENMLTagsConverter.h

```

1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.

```

```

14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorMessage.h>
22 #include <quentier/types/Result.h>
23 #include <quentier/utility/IEncryptor.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <qevercloud/types/Fwd.h>
27
28 #include <QString>
29 #include <QtGlobal>
30
31 namespace quentier::enml {
32
33 class QUENTIER_EXPORT IENMLTagsConverter
34 {
35 public:
36     virtual ~IENMLTagsConverter();
37
38     [[nodiscard]] virtual QString convertEnToDo(
39         bool checked, quint32 index) const = 0;
40
41     [[nodiscard]] virtual QString convertEncryptedText(
42         const QString & encryptedText, const QString & hint,
43         utility::IEncryptor::Cipher cipher, quint32 index) const = 0;
44
45     [[nodiscard]] virtual QString convertDecryptedText(
46         const QString & decryptedText, const QString & encryptedText,
47         const QString & hint, utility::IEncryptor::Cipher cipher,
48         quint32 index) const = 0;
49
50     [[nodiscard]] virtual Result<QString, ErrorMessage> convertResource(
51         const qevercloud::Resource & resource) const = 0;
52 };
53
54 } // namespace quentier::enml

```

6.14 IHtmlData.h

```

1 /*
2 * Copyright 2023-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QString>
25 #include <QtGlobal>
26
27 namespace quentier::enml {
28
29 struct QUENTIER_EXPORT IHtmlData : public utility::Printable
30 {
31     [[nodiscard]] virtual QString html() const = 0;
32
33     [[nodiscard]] virtual quint32 numEnToDoNodes() const = 0;
34
35     [[nodiscard]] virtual quint32 numHyperlinkNodes() const = 0;
36
37     [[nodiscard]] virtual quint32 numEnCryptNodes() const = 0;
38
39     [[nodiscard]] virtual quint32 numEnDecryptedNodes() const = 0;
40

```

```

59
60 public: // utility::Printable
61     QTextStream & print(QTextStream & strm) const override;
62 };
63
64 } // namespace quantier::enml

```

6.15 InvalidArgument.h

```

1 /*
2  * Copyright 2021 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/exception/IQuantierException.h>
22
23 namespace quantier {
24
25 class QUANTIER_EXPORT InvalidArgument : public IQuantierException
26 {
27 public:
28     explicit InvalidArgument(ErrorString message);
29
30     [[nodiscard]] InvalidArgument * clone() const override;
31     void raise() const override;
32
33 protected:
34     [[nodiscard]] QString exceptionDisplayName() const override;
35 };
36
37 } // namespace quantier

```

6.16 IQuantierException.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/types/ErrorString.h>
22 #include <quantier/utility/Printable.h>
23
24 #include <QException>
25
26 namespace quantier {
27
28 class QUANTIER_EXPORT IQuantierException :
29     public utility::Printable,
30     public QException

```

```

39 {
40 public:
41     ~IQuentierException() noexcept override;
42
43     [[nodiscard]] QString errorMessage() const;
44     [[nodiscard]] QString localizedErrorMessage() const;
45     [[nodiscard]] QString nonLocalizedErrorMessage() const;
46
47     // std::exception
48     [[nodiscard]] const char * what() const noexcept override;
49
50     // utility::Printable
51     QTextStream & print(QTextStream & strm) const override;
52
53 protected:
54     explicit IQuentierException(QString message);
55     IQuentierException(const IQuentierException & other);
56     IQuentierException & operator=(const IQuentierException & other);
57
58     [[nodiscard]] virtual QString exceptionDisplayName() const = 0;
59
60 private:
61     QString m_message;
62     char * m_whatMessage = nullptr;
63 };
64
65 } // namespace quentier

```

6.17 OperationCanceled.h

```

1 /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier {
24
25 class QUENTIER_EXPORT OperationCanceled : public IQuentierException
26 {
27 public:
28     explicit OperationCanceled();
29
30     [[nodiscard]] OperationCanceled * clone() const override;
31     void raise() const override;
32
33 protected:
34     [[nodiscard]] QString exceptionDisplayName() const override;
35 };
36
37 } // namespace quentier

```

6.18 RuntimeError.h

```

1 /*
2  * Copyright 2021 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *

```

```

10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier {
24
25 class QUENTIER_EXPORT RuntimeError : public IQuentierException
26 {
27 public:
28     explicit RuntimeError(ErrorString message);
29     ~RuntimeError() noexcept override;
30
31     [[nodiscard]] RuntimeError * clone() const override;
32     void raise() const override;
33
34 protected:
35     [[nodiscard]] QString exceptionDisplayName() const override;
36 };
37
38 } // namespace quentier

```

6.19 ILocalStorage.h

```

1 /*
2 * Copyright 2020-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/local_storage/NoteSearchQuery.h>
23 #include <quentier/types/Fwd.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <qevercloud/types/LinkedNotebook.h>
27 #include <qevercloud/types/Note.h>
28 #include <qevercloud/types/Notebook.h>
29 #include <qevercloud/types/Resource.h>
30 #include <qevercloud/types/SavedSearch.h>
31 #include <qevercloud/types/SharedNotebook.h>
32 #include <qevercloud/types/Tag.h>
33 #include <qevercloud/types/User.h>
34
35 #include <QFlags>
36 #include <QFuture>
37 #include <QHash>
38 #include <QList>
39 #include <QStringList>
40 #include <QThreadPool>
41
42 #include <optional>
43 #include <utility>
44
45 class QDebug;
46 class QTextStream;
47 class QThreadPool;
48
49 namespace quentier::local_storage {
50
51 class QUENTIER_EXPORT ILocalStorage

```

```

52 {
53 public:
54     virtual ~ILocalStorage() = default;
55
56 public:
57     enum class StartupOption
58     {
59         ClearDatabase = 1 « 1,
60         OverrideLock = 1 « 2
61     };
62     Q_DECLARE_FLAGS(StartupOptions, StartupOption);
63
64     friend QUINTIER_EXPORT QTextStream & operator<<(
65         QTextStream & strm, StartupOption option);
66
67     friend QUINTIER_EXPORT QDebug & operator<<(
68         QDebug & dbg, StartupOption option);
69
70     friend QUINTIER_EXPORT QTextStream & operator<<(
71         QTextStream & strm, StartupOptions options);
72
73     friend QUINTIER_EXPORT QDebug & operator<<(
74         QDebug & dbg, StartupOptions options);
75
76
77
78     enum class ListObjectsFilter
79     {
80         Include,
81         Exclude
82     };
83
84     friend QUINTIER_EXPORT QTextStream & operator<<(
85         QTextStream & strm, ListObjectsFilter filter);
86
87     friend QUINTIER_EXPORT QDebug & operator<<(
88         QDebug & dbg, ListObjectsFilter filter);
89
90
91
92     struct QUINTIER_EXPORT ListObjectsFilters
93     {
94         std::optional<ListObjectsFilter> m_locallyModifiedFilter;
95         std::optional<ListObjectsFilter> m_withGuidFilter;
96         std::optional<ListObjectsFilter> m_localOnlyFilter;
97         std::optional<ListObjectsFilter> m_locallyFavoritedFilter;
98     };
99
100     friend QUINTIER_EXPORT QTextStream & operator<<(
101         QTextStream & strm, const ListObjectsFilters & filters);
102
103     friend QUINTIER_EXPORT QDebug & operator<<(
104         QDebug & dbg, const ListObjectsFilters & filters);
105
106
107
108     struct QUINTIER_EXPORT ListGuidsFilters
109     {
110         std::optional<ListObjectsFilter> m_locallyModifiedFilter;
111         std::optional<ListObjectsFilter> m_locallyFavoritedFilter;
112     };
113
114     friend QUINTIER_EXPORT QTextStream & operator<<(
115         QTextStream & strm, const ListGuidsFilters & filters);
116
117     friend QUINTIER_EXPORT QDebug & operator<<(
118         QDebug & dbg, const ListGuidsFilters & filters);
119
120
121
122     enum class OrderDirection
123     {
124         Ascending,
125         Descending
126     };
127
128     friend QUINTIER_EXPORT QTextStream & operator<<(
129         QTextStream & strm, OrderDirection orderDirection);
130
131     friend QUINTIER_EXPORT QDebug & operator<<(
132         QDebug & dbg, OrderDirection orderDirection);
133
134
135
136     enum class ListNotebooksOrder
137     {
138         NoOrder,
139         ByUpdateSequenceNumber,
140         ByNotebookName,
141         ByCreationTimestamp,
142         ByModificationTimestamp
143     };

```

```

144
145     friend QUINTIER_EXPORT QTextStream & operator«(
146         QTextStream & strm, ListNotebooksOrder order);
147
148     friend QUINTIER_EXPORT QDebug & operator«(
149         QDebug & dbg, ListNotebooksOrder order);
150
151
152
153     enum class ListLinkedNotebooksOrder
154     {
155         NoOrder,
156         ByUpdateSequenceNumber,
157         ByShareName,
158         ByUsername,
159     };
160
161     friend QUINTIER_EXPORT QTextStream & operator«(
162         QTextStream & strm, ListLinkedNotebooksOrder order);
163
164     friend QUINTIER_EXPORT QDebug & operator«(
165         QDebug & dbg, ListLinkedNotebooksOrder order);
166
167
168
169     enum class ListTagsOrder
170     {
171         NoOrder,
172         ByUpdateSequenceNumber,
173         ByName
174     };
175
176     friend QUINTIER_EXPORT QTextStream & operator«(
177         QTextStream & strm, ListTagsOrder order);
178
179     friend QUINTIER_EXPORT QDebug & operator«(
180         QDebug & dbg, ListTagsOrder order);
181
182
183
184     enum class ListNotesOrder
185     {
186         NoOrder,
187         ByUpdateSequenceNumber,
188         ByTitle,
189         ByCreationTimestamp,
190         ByModificationTimestamp,
191         ByDeletionTimestamp,
192         ByAuthor,
193         BySource,
194         BySourceApplication,
195         ByReminderTime,
196         ByPlaceName
197     };
198
199     friend QUINTIER_EXPORT QTextStream & operator«(
200         QTextStream & strm, ListNotesOrder order);
201
202     friend QUINTIER_EXPORT QDebug & operator«(
203         QDebug & dbg, ListNotesOrder order);
204
205
206
207     enum class ListSavedSearchesOrder
208     {
209         NoOrder,
210         ByUpdateSequenceNumber,
211         ByName,
212         ByFormat
213     };
214
215     friend QUINTIER_EXPORT QTextStream & operator«(
216         QTextStream & strm, ListSavedSearchesOrder order);
217
218     friend QUINTIER_EXPORT QDebug & operator«(
219         QDebug & dbg, ListSavedSearchesOrder order);
220
221
222
223     enum class Affiliation
224     {
225         Any,
226         User,
227         AnyLinkedNotebook,
228         ParticularLinkedNotebooks
229     };
230
231
232
233     friend QUINTIER_EXPORT QTextStream & operator«(
234         QTextStream & strm, Affiliation affiliation);
235
236     friend QUINTIER_EXPORT QDebug & operator«(
237         QDebug & dbg, Affiliation affiliation);

```

```

238
240
241 struct QUINTIER_EXPORT ListOptionsBase
242 {
243     ListOptionsBase() noexcept {}; // NOLINT
244
245     ListObjectsFilters m_filters = {};
246     quint64 m_limit = 0UL;
247     quint64 m_offset = 0UL;
248     OrderDirection m_direction = OrderDirection::Ascending;
249 };
250
251 struct QUINTIER_EXPORT ListNotebooksOptions : public ListOptionsBase
252 {
253     ListNotebooksOptions() noexcept {}; // NOLINT
254
255     ListNotebooksOrder m_order = ListNotebooksOrder::NoOrder;
256     Affiliation m_affiliation = Affiliation::Any;
257     QList<qevercloud::Guid> m_linkedNotebookGuids;
258 };
259
260 friend QUINTIER_EXPORT QTextStream & operator<<(
261     QTextStream & strm, const ListNotebooksOptions & options);
262
263 friend QUINTIER_EXPORT QDebug & operator<<(
264     QDebug & dbg, const ListNotebooksOptions & options);
265
266 struct QUINTIER_EXPORT ListLinkedNotebooksOptions : public ListOptionsBase
267 {
268     ListLinkedNotebooksOptions() noexcept {}; // NOLINT
269
270     ListLinkedNotebooksOrder m_order = ListLinkedNotebooksOrder::NoOrder;
271 };
272
273 friend QUINTIER_EXPORT QTextStream & operator<<(
274     QTextStream & strm, const ListLinkedNotebooksOptions & options);
275
276 friend QUINTIER_EXPORT QDebug & operator<<(
277     QDebug & dbg, const ListLinkedNotebooksOptions & options);
278
279 struct QUINTIER_EXPORT ListSavedSearchesOptions : public ListOptionsBase
280 {
281     ListSavedSearchesOptions() noexcept {}; // NOLINT
282
283     ListSavedSearchesOrder m_order = ListSavedSearchesOrder::NoOrder;
284 };
285
286 friend QUINTIER_EXPORT QTextStream & operator<<(
287     QTextStream & strm, const ListSavedSearchesOptions & options);
288
289 friend QUINTIER_EXPORT QDebug & operator<<(
290     QDebug & dbg, const ListSavedSearchesOptions & options);
291
292 struct QUINTIER_EXPORT ListNotesOptions : public ListOptionsBase
293 {
294     ListNotesOptions() noexcept {}; // NOLINT
295
296     ListNotesOrder m_order = ListNotesOrder::NoOrder;
297 };
298
299 friend QUINTIER_EXPORT QTextStream & operator<<(
300     QTextStream & strm, const ListNotesOptions & options);
301
302 friend QUINTIER_EXPORT QDebug & operator<<(
303     QDebug & dbg, const ListNotesOptions & options);
304
305 enum class TagNotesRelation
306 {
307     Any,
308     WithNotes,
309     WithoutNotes
310 };
311
312 struct QUINTIER_EXPORT ListTagsOptions : public ListOptionsBase
313 {
314     ListTagsOptions() noexcept {}; // NOLINT
315
316     ListTagsOrder m_order = ListTagsOrder::NoOrder;
317     Affiliation m_affiliation = Affiliation::Any;
318     QList<qevercloud::Guid> m_linkedNotebookGuids;
319     TagNotesRelation m_tagNotesRelation = TagNotesRelation::Any;
320 };
321
322 friend QUINTIER_EXPORT QTextStream & operator<<(
323     QTextStream & strm, const ListTagsOptions & options);
324
325 friend QUINTIER_EXPORT QDebug & operator<<(

```

```

331         QDebug & dbg, const ListTagsOptions & options);
332
333
334
335     enum class NoteCountOption
336     {
337         IncludeNonDeletedNotes = 1 « 1,
338         IncludeDeletedNotes = 1 « 2
339     };
340     Q_DECLARE_FLAGS(NoteCountOptions, NoteCountOption)
341
342     friend QUINTIER_EXPORT QTextStream & operator«(
343         QTextStream & strm, NoteCountOption option);
344
345     friend QUINTIER_EXPORT QDebug & operator«(
346         QDebug & dbg, NoteCountOption option);
347
348     friend QUINTIER_EXPORT QTextStream & operator«(
349         QTextStream & strm, NoteCountOptions options);
350
351     friend QUINTIER_EXPORT QDebug & operator«(
352         QDebug & dbg, NoteCountOptions options);
353
354
355     enum class UpdateNoteOption
356     {
357         UpdateResourceMetadata = 1 « 1,
358         UpdateResourceBinaryData = 1 « 2,
359         UpdateTags = 1 « 3
360     };
361     Q_DECLARE_FLAGS(UpdateNoteOptions, UpdateNoteOption)
362
363     friend QUINTIER_EXPORT QTextStream & operator«(
364         QTextStream & strm, UpdateNoteOption option);
365
366     friend QUINTIER_EXPORT QDebug & operator«(
367         QDebug & dbg, UpdateNoteOption option);
368
369     friend QUINTIER_EXPORT QTextStream & operator«(
370         QTextStream & strm, UpdateNoteOptions options);
371
372     friend QUINTIER_EXPORT QDebug & operator«(
373         QDebug & dbg, UpdateNoteOptions options);
374
375
376     enum class FetchNoteOption
377     {
378         WithResourceMetadata = 1 « 1,
379         WithResourceBinaryData = 1 « 2
380     };
381     Q_DECLARE_FLAGS(FetchNoteOptions, FetchNoteOption)
382
383     friend QUINTIER_EXPORT QTextStream & operator«(
384         QTextStream & strm, FetchNoteOption option);
385
386     friend QUINTIER_EXPORT QDebug & operator«(
387         QDebug & dbg, FetchNoteOption option);
388
389     friend QUINTIER_EXPORT QTextStream & operator«(
390         QTextStream & strm, FetchNoteOptions options);
391
392     friend QUINTIER_EXPORT QDebug & operator«(
393         QDebug & dbg, FetchNoteOptions options);
394
395     friend QUINTIER_EXPORT QDebug & operator«(
396         QDebug & dbg, FetchNoteOptions options);
397
398
399     enum class FetchResourceOption
400     {
401         WithBinaryData = 1 « 1
402     };
403     Q_DECLARE_FLAGS(FetchResourceOptions, FetchResourceOption)
404
405     friend QUINTIER_EXPORT QTextStream & operator«(
406         QTextStream & strm, FetchResourceOption option);
407
408     friend QUINTIER_EXPORT QDebug & operator«(
409         QDebug & dbg, FetchResourceOption option);
410
411     friend QUINTIER_EXPORT QTextStream & operator«(
412         QTextStream & strm, FetchResourceOptions options);
413
414     friend QUINTIER_EXPORT QDebug & operator«(
415         QDebug & dbg, FetchResourceOptions options);
416
417
418     enum class HighestUsnOption
419     {
420         WithinUserOwnContent,
421         WithinUserOwnContentAndLinkedNotebooks

```

```

423     };
424
425     friend QUINTIER_EXPORT QTextStream & operator«(
426         QTextStream & strm, HighestUsnOption option);
427
428     friend QUINTIER_EXPORT QDebug & operator«(
429         QDebug & dbg, HighestUsnOption option);
430
431 public:
432     // Versions/upgrade API
433     [[nodiscard]] virtual QFuture<bool> isVersionTooHigh() const = 0;
434     [[nodiscard]] virtual QFuture<bool> requiresUpgrade() const = 0;
435     [[nodiscard]] virtual QFuture<QList<IPatchPtr>> requiredPatches() const = 0;
436     [[nodiscard]] virtual QFuture<qint32> version() const = 0;
437     [[nodiscard]] virtual QFuture<qint32> highestSupportedVersion() const = 0;
438
439     // Users API
440     [[nodiscard]] virtual QFuture<quint32> userCount() const = 0;
441     [[nodiscard]] virtual QFuture<void> putUser(qevercloud::User user) = 0;
442
443     [[nodiscard]] virtual QFuture<std::optional<qevercloud::User>> findUserById(
444         qevercloud::UserID userId) const = 0;
445
446     [[nodiscard]] virtual QFuture<void> expungeUserById(
447         qevercloud::UserID userId) = 0;
448
449     // Notebooks API
450     [[nodiscard]] virtual QFuture<quint32> notebookCount() const = 0;
451
452     [[nodiscard]] virtual QFuture<void> putNotebook(
453         qevercloud::Notebook notebook) = 0;
454
455     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook>>
456         findNotebookByLocalId(QString notebookLocalId) const = 0;
457
458     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook>>
459         findNotebookByGuid(qevercloud::Guid guid) const = 0;
460
461     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook>>
462         findNotebookByName(
463             QString notebookName,
464             std::optional<qevercloud::Guid> linkedNotebookGuid =
465                 std::nullopt) const = 0;
466
467     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook>>
468         findDefaultNotebook() const = 0;
469
470     [[nodiscard]] virtual QFuture<void> expungeNotebookByLocalId(
471         QString notebookLocalId) = 0;
472
473     [[nodiscard]] virtual QFuture<void> expungeNotebookByGuid(
474         qevercloud::Guid notebookGuid) = 0;
475
476     [[nodiscard]] virtual QFuture<void> expungeNotebookByName(
477         QString name,
478         std::optional<qevercloud::Guid> linkedNotebookGuid = std::nullopt) = 0;
479
480     [[nodiscard]] virtual QFuture<QList<qevercloud::Notebook>> listNotebooks(
481         ListNotebooksOptions options = {}) const = 0;
482
483     [[nodiscard]] virtual QFuture<QList<qevercloud::SharedNotebook>>
484         listSharedNotebooks(qevercloud::Guid notebookGuid = {}) const = 0;
485
486     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid>> listNotebookGuids(
487         ListGuidsFilters filters,
488         std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
489
490     // Linked notebooks API
491     [[nodiscard]] virtual QFuture<quint32> linkedNotebookCount() const = 0;
492
493     [[nodiscard]] virtual QFuture<void> putLinkedNotebook(
494         qevercloud::LinkedNotebook linkedNotebook) = 0;
495
496     [[nodiscard]] virtual QFuture<std::optional<qevercloud::LinkedNotebook>>
497         findLinkedNotebookByGuid(qevercloud::Guid guid) const = 0;
498
499     [[nodiscard]] virtual QFuture<void> expungeLinkedNotebookByGuid(
500         qevercloud::Guid guid) = 0;
501
502     [[nodiscard]] virtual QFuture<QList<qevercloud::LinkedNotebook>>
503         listLinkedNotebooks(ListLinkedNotebooksOptions options = {}) const = 0;
504
505     // Notes API
506     [[nodiscard]] virtual QFuture<quint32> noteCount(
507         NoteCountOptions options = NoteCountOptions(
508             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
509

```

```

510 [[nodiscard]] virtual QFuture<quint32> noteCountPerNotebookLocalId(
511     QString notebookLocalId,
512     NoteCountOptions options = NoteCountOptions(
513         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
514
515 [[nodiscard]] virtual QFuture<quint32> noteCountPerTagLocalId(
516     QString tagLocalId,
517     NoteCountOptions options = NoteCountOptions(
518         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
519
520 [[nodiscard]] virtual QFuture<QHash<QString, quint32>> noteCountsPerTags(
521     ListTagsOptions listTagsOptions = {},
522     NoteCountOptions options = NoteCountOptions(
523         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
524
525 [[nodiscard]] virtual QFuture<quint32> noteCountPerNotebookAndTagLocalIds(
526     QStringList notebookLocalIds, QStringList tagLocalIds,
527     NoteCountOptions options = NoteCountOptions(
528         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
529
530 [[nodiscard]] virtual QFuture<void> putNote(qevercloud::Note note) = 0;
531
532 [[nodiscard]] virtual QFuture<void> updateNote(
533     qevercloud::Note note, UpdateNoteOptions options) = 0;
534
535 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Note>
536     findNoteByLocalId(
537         QString noteLocalId, FetchNoteOptions options) const = 0;
538
539 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Note>
540     findNoteByGuid(
541         qevercloud::Guid noteGuid, FetchNoteOptions options) const = 0;
542
543 [[nodiscard]] virtual QFuture<QList<qevercloud::Note> listNotes(
544     FetchNoteOptions fetchOptions,
545     ListNotesOptions listOptions = {}) const = 0;
546
547 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>
548     listNotesPerNotebookLocalId(
549         QString notebookLocalId, FetchNoteOptions fetchOptions,
550         ListNotesOptions listOptions = {}) const = 0;
551
552 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>
553     listNotesPerTagLocalId(
554         QString tagLocalId, FetchNoteOptions fetchOptions,
555         ListNotesOptions listOptions = {}) const = 0;
556
557 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>
558     listNotesPerNotebookAndTagLocalIds(
559         QStringList notebookLocalIds, QStringList tagLocalIds,
560         FetchNoteOptions fetchOptions,
561         ListNotesOptions listOptions = {}) const = 0;
562
563 [[nodiscard]] virtual QFuture<QList<qevercloud::Note> listNotesByLocalIds(
564     QStringList noteLocalIds, FetchNoteOptions fetchOptions,
565     ListNotesOptions listOptions = {}) const = 0;
566
567 [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid> listNoteGuids(
568     ListGuidsFilters filters,
569     std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
570
571 [[nodiscard]] virtual QFuture<QList<qevercloud::Note> queryNotes(
572     NoteSearchQuery query, FetchNoteOptions fetchOptions) const = 0;
573
574 [[nodiscard]] virtual QFuture<QStringList> queryNoteLocalIds(
575     NoteSearchQuery query) const = 0;
576
577 [[nodiscard]] virtual QFuture<void> expungeNoteByLocalId(
578     QString noteLocalId) = 0;
579
580 [[nodiscard]] virtual QFuture<void> expungeNoteByGuid(
581     qevercloud::Guid noteGuid) = 0;
582
583 // Tags API
584 [[nodiscard]] virtual QFuture<quint32> tagCount() const = 0;
585 [[nodiscard]] virtual QFuture<void> putTag(qevercloud::Tag tag) = 0;
586
587 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag>
588     findTagByLocalId(QString tagLocalId) const = 0;
589
590 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag> findTagByGuid(
591     qevercloud::Guid tagGuid) const = 0;
592
593 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag> findTagByName(
594     QString tagName,
595     std::optional<qevercloud::Guid> linkedNotebookGuid =
596     std::nullopt) const = 0;

```

```

597
598 [[nodiscard]] virtual QFuture<QList<qevercloud::Tag> listTags(
599     ListTagsOptions options = {}) const = 0;
600
601 [[nodiscard]] virtual QFuture<QList<qevercloud::Tag>
602     listTagsPerNoteLocalId(
603         QString noteLocalId, ListTagsOptions options = {}) const = 0;
604
605 [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid> listTagGuids(
606     ListGuidsFilters filters,
607     std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
608
609 [[nodiscard]] virtual QFuture<void> expungeTagByLocalId(
610     QString tagLocalId) = 0;
611
612 [[nodiscard]] virtual QFuture<void> expungeTagByGuid(
613     qevercloud::Guid tagGuid) = 0;
614
615 [[nodiscard]] virtual QFuture<void> expungeTagByName(
616     QString name,
617     std::optional<qevercloud::Guid> linkedNotebookGuid = std::nullopt) = 0;
618
619 // Resources API
620 [[nodiscard]] virtual QFuture<quint32> resourceCount(
621     NoteCountOptions options = NoteCountOptions(
622         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
623
624 [[nodiscard]] virtual QFuture<quint32> resourceCountPerNoteLocalId(
625     QString noteLocalId) const = 0;
626
627 [[nodiscard]] virtual QFuture<void> putResource(
628     qevercloud::Resource resource) = 0;
629
630 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Resource>
631     findResourceByLocalId(
632         QString resourceLocalId,
633         FetchResourceOptions options = {}) const = 0;
634
635 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Resource>
636     findResourceByGuid(
637         qevercloud::Guid resourceGuid,
638         FetchResourceOptions options = {}) const = 0;
639
640 [[nodiscard]] virtual QFuture<void> expungeResourceByLocalId(
641     QString resourceLocalId) = 0;
642
643 [[nodiscard]] virtual QFuture<void> expungeResourceByGuid(
644     qevercloud::Guid resourceGuid) = 0;
645
646 // Saved searches API
647 [[nodiscard]] virtual QFuture<quint32> savedSearchCount() const = 0;
648
649 [[nodiscard]] virtual QFuture<void> putSavedSearch(
650     qevercloud::SavedSearch search) = 0;
651
652 [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch>
653     findSavedSearchByLocalId(QString savedSearchLocalId) const = 0;
654
655 [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch>
656     findSavedSearchByGuid(qevercloud::Guid guid) const = 0;
657
658 [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch>
659     findSavedSearchByName(QString name) const = 0;
660
661 [[nodiscard]] virtual QFuture<QList<qevercloud::SavedSearch>
662     listSavedSearches(ListSavedSearchesOptions options = {}) const = 0;
663
664 [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid> listSavedSearchGuids(
665     ListGuidsFilters filters) const = 0;
666
667 [[nodiscard]] virtual QFuture<void> expungeSavedSearchByLocalId(
668     QString savedSearchLocalId) = 0;
669
670 [[nodiscard]] virtual QFuture<void> expungeSavedSearchByGuid(
671     qevercloud::Guid guid) = 0;
672
673 // Synchronization API
674 [[nodiscard]] virtual QFuture<qint32> highestUpdateSequenceNumber(
675     HighestUsnOption option) const = 0;
676
677 [[nodiscard]] virtual QFuture<qint32> highestUpdateSequenceNumber(
678     qevercloud::Guid linkedNotebookGuid) const = 0;
679
680 [[nodiscard]] virtual ILocalStorageNotifier * notifier() const = 0;
681 };
682
683 [[nodiscard]] QENTIER_EXPORT bool operator==(

```

```

690     const ILocalStorage::ListObjectsFilters & lhs,
691     const ILocalStorage::ListObjectsFilters & rhs) noexcept;
692
693 [[nodiscard]] QUENTIER_EXPORT bool operator==(
694     const ILocalStorage::ListOptionsBase & lhs,
695     const ILocalStorage::ListOptionsBase & rhs) noexcept;
696
697 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
698     const ILocalStorage::ListOptionsBase & lhs,
699     const ILocalStorage::ListOptionsBase & rhs) noexcept;
700
701 [[nodiscard]] QUENTIER_EXPORT bool operator==(
702     const ILocalStorage::ListNotebooksOptions & lhs,
703     const ILocalStorage::ListNotebooksOptions & rhs) noexcept;
704
705 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
706     const ILocalStorage::ListNotebooksOptions & lhs,
707     const ILocalStorage::ListNotebooksOptions & rhs) noexcept;
708
709 [[nodiscard]] QUENTIER_EXPORT bool operator==(
710     const ILocalStorage::ListLinkedNotebooksOptions & lhs,
711     const ILocalStorage::ListLinkedNotebooksOptions & rhs) noexcept;
712
713 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
714     const ILocalStorage::ListLinkedNotebooksOptions & lhs,
715     const ILocalStorage::ListLinkedNotebooksOptions & rhs) noexcept;
716
717 [[nodiscard]] QUENTIER_EXPORT bool operator==(
718     const ILocalStorage::ListSavedSearchesOptions & lhs,
719     const ILocalStorage::ListSavedSearchesOptions & rhs) noexcept;
720
721 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
722     const ILocalStorage::ListSavedSearchesOptions & lhs,
723     const ILocalStorage::ListSavedSearchesOptions & rhs) noexcept;
724
725 [[nodiscard]] QUENTIER_EXPORT bool operator==(
726     const ILocalStorage::ListNotesOptions & lhs,
727     const ILocalStorage::ListNotesOptions & rhs) noexcept;
728
729 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
730     const ILocalStorage::ListNotesOptions & lhs,
731     const ILocalStorage::ListNotesOptions & rhs) noexcept;
732
733 [[nodiscard]] QUENTIER_EXPORT bool operator==(
734     const ILocalStorage::ListTagsOptions & lhs,
735     const ILocalStorage::ListTagsOptions & rhs) noexcept;
736
737 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
738     const ILocalStorage::ListTagsOptions & lhs,
739     const ILocalStorage::ListTagsOptions & rhs) noexcept;
740
741 [[nodiscard]] QUENTIER_EXPORT bool operator==(
742     const ILocalStorage::ListGuidsFilters & lhs,
743     const ILocalStorage::ListGuidsFilters & rhs) noexcept;
744
745 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
746     const ILocalStorage::ListGuidsFilters & lhs,
747     const ILocalStorage::ListGuidsFilters & rhs) noexcept;
748
749 } // namespace quentier::local_storage

```

6.20 ILocalStorageNotifier.h

```

1  /*
2  * Copyright 2020-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20

```

```

21 #include <quentier/local_storage/ILocalStorage.h>
22
23 #include <QObject>
24
25 namespace quentier::local_storage {
26
27 class QUENTIER_EXPORT ILocalStorageNotifier : public QObject
28 {
29     Q_OBJECT
30 protected:
31     explicit ILocalStorageNotifier(QObject * parent = nullptr);
32
33 public:
34     ~ILocalStorageNotifier() override;
35
36 Q_SIGNALS:
37     // Notifications about user related events
38     void userPut(qevercloud::User user);
39     void userExpunged(qevercloud::UserID userId);
40
41     // Notifications about notebook related events
42     void notebookPut(qevercloud::Notebook notebook);
43     void notebookExpunged(QString notebookLocalId);
44
45     // Notifications about linked notebooks
46     void linkedNotebookPut(qevercloud::LinkedNotebook linkedNotebook);
47     void linkedNotebookExpunged(qevercloud::Guid linkedNotebookGuid);
48
49     // Notifications about note related events
50     void notePut(qevercloud::Note note);
51
52     void noteUpdated(
53         qevercloud::Note note, ILocalStorage::UpdateNoteOptions options);
54
55     void noteExpunged(QString noteLocalId);
56
57     // Notifications about tag related events
58     void tagPut(qevercloud::Tag tag);
59
60     void tagExpunged(QString tagLocalId, QStringList expungedChildTagLocalIds);
61
62     // Notifications about resource related events
63     void resourcePut(qevercloud::Resource resource);
64     void resourceMetadataPut(qevercloud::Resource resource);
65     void resourceExpunged(QString resourceLocalId);
66
67     // Notifications about saved search related events
68     void savedSearchPut(qevercloud::SavedSearch savedSearch);
69     void savedSearchExpunged(QString savedSearchLocalId);
70 };
71
72 } // namespace quentier::local_storage

```

6.21 IPatch.h

```

1 /*
2  * Copyright 2021-2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QFuture>
24
25 namespace quentier::local_storage {
26
27 class QUENTIER_EXPORT IPatch
28 {

```

```

35 public:
36     virtual ~IPatch() noexcept;
37
42     [[nodiscard]] virtual int fromVersion() const noexcept = 0;
43
48     [[nodiscard]] virtual int toVersion() const noexcept = 0;
49
53     [[nodiscard]] virtual QString patchShortDescription() const = 0;
54
58     [[nodiscard]] virtual QString patchLongDescription() const = 0;
59
69     [[nodiscard]] virtual QFuture<void> backupLocalStorage() = 0;
70
80     [[nodiscard]] virtual QFuture<void> restoreLocalStorageFromBackup() = 0;
81
92     [[nodiscard]] virtual QFuture<void> removeLocalStorageBackup() = 0;
93
101     [[nodiscard]] virtual QFuture<void> apply() = 0;
102 };
103
104 } // namespace quentier::local_storage

```

6.22 LocalStorageOpenException.h

```

1 /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier::local_storage {
24
25     class QUENTIER_EXPORT LocalStorageOpenException : public IQuentierException
26     {
27     public:
28         explicit LocalStorageOpenException(const ErrorString & message);
29
30         [[nodiscard]] LocalStorageOpenException * clone() const override;
31         void raise() const override;
32
33     protected:
34         [[nodiscard]] QString exceptionDisplayName() const override;
35     };
36
37 } // namespace quentier::local_storage

```

6.23 LocalStorageOperationException.h

```

1 /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License

```

```

16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier::local_storage {
24
25     class QUENTIER_EXPORT LocalStorageOperationException : public IQuentierException
26     {
27     public:
28         explicit LocalStorageOperationException(ErrorString message);
29
30         [[nodiscard]] LocalStorageOperationException * clone() const override;
31         void raise() const override;
32
33     protected:
34         [[nodiscard]] QString exceptionDisplayName() const override;
35     };
36
37 } // namespace quentier::local_storage

```

6.24 NoteSearchQuery.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22
23 #include <QList>
24 #include <QSharedDataPointer>
25
26 namespace quentier::local_storage {
27
28     class QUENTIER_EXPORT NoteSearchQuery : public utility::Printable
29     {
30     public:
31         explicit NoteSearchQuery();
32
33         NoteSearchQuery(const NoteSearchQuery & other);
34         NoteSearchQuery(NoteSearchQuery && other) noexcept;
35
36         NoteSearchQuery & operator=(const NoteSearchQuery & other);
37         NoteSearchQuery & operator=(NoteSearchQuery && other) noexcept;
38
39         ~NoteSearchQuery() override;
40
41         [[nodiscard]] bool isEmpty() const;
42
43         void clear();
44
45         [[nodiscard]] QString queryString() const;
46
47         [[nodiscard]] bool setQueryString(
48             const QString & queryString, ErrorString & error);
49
50         [[nodiscard]] QString notebookModifier() const;
51
52         [[nodiscard]] bool hasAnyModifier() const;
53
54         [[nodiscard]] const QStringList & tagNames() const;
55         [[nodiscard]] const QStringList & negatedTagNames() const;
56         [[nodiscard]] bool hasAnyTag() const;
57         [[nodiscard]] bool hasNegatedAnyTag() const;
58     };
59
60 }

```

```

67 [[nodiscard]] const QStringList & titleNames() const;
68 [[nodiscard]] const QStringList & negatedTitleNames() const;
69 [[nodiscard]] bool hasAnyTitleName() const;
70 [[nodiscard]] bool hasNegatedAnyTitleName() const;
71
72 [[nodiscard]] const QList<qint64> & creationTimestamps() const;
73 [[nodiscard]] const QList<qint64> & negatedCreationTimestamps() const;
74 [[nodiscard]] bool hasAnyCreationTimestamp() const;
75 [[nodiscard]] bool hasNegatedAnyCreationTimestamp() const;
76
77 [[nodiscard]] const QList<qint64> & modificationTimestamps() const;
78 [[nodiscard]] const QList<qint64> & negatedModificationTimestamps() const;
79 [[nodiscard]] bool hasAnyModificationTimestamp() const;
80 [[nodiscard]] bool hasNegatedAnyModificationTimestamp() const;
81
82 [[nodiscard]] const QStringList & resourceMimeTypes() const;
83 [[nodiscard]] const QStringList & negatedResourceMimeTypes() const;
84 [[nodiscard]] bool hasAnyResourceMimeType() const;
85 [[nodiscard]] bool hasNegatedAnyResourceMimeType() const;
86
87 [[nodiscard]] const QList<qint64> & subjectDateTimestamps() const;
88 [[nodiscard]] const QList<qint64> & negatedSubjectDateTimestamps() const;
89 [[nodiscard]] bool hasAnySubjectDateTimestamp() const;
90 [[nodiscard]] bool hasNegatedAnySubjectDateTimestamp() const;
91
92 [[nodiscard]] const QList<double> & latitudes() const;
93 [[nodiscard]] const QList<double> & negatedLatitudes() const;
94 [[nodiscard]] bool hasAnyLatitude() const;
95 [[nodiscard]] bool hasNegatedAnyLatitude() const;
96
97 [[nodiscard]] const QList<double> & longitudes() const;
98 [[nodiscard]] const QList<double> & negatedLongitudes() const;
99 [[nodiscard]] bool hasAnyLongitude() const;
100 [[nodiscard]] bool hasNegatedAnyLongitude() const;
101
102 [[nodiscard]] const QList<double> & altitudes() const;
103 [[nodiscard]] const QList<double> & negatedAltitudes() const;
104 [[nodiscard]] bool hasAnyAltitude() const;
105 [[nodiscard]] bool hasNegatedAnyAltitude() const;
106
107 [[nodiscard]] const QStringList & authors() const;
108 [[nodiscard]] const QStringList & negatedAuthors() const;
109 [[nodiscard]] bool hasAnyAuthor() const;
110 [[nodiscard]] bool hasNegatedAnyAuthor() const;
111
112 [[nodiscard]] const QStringList & sources() const;
113 [[nodiscard]] const QStringList & negatedSources() const;
114 [[nodiscard]] bool hasAnySource() const;
115 [[nodiscard]] bool hasNegatedAnySource() const;
116
117 [[nodiscard]] const QStringList & sourceApplications() const;
118 [[nodiscard]] const QStringList & negatedSourceApplications() const;
119 [[nodiscard]] bool hasAnySourceApplication() const;
120 [[nodiscard]] bool hasNegatedAnySourceApplication() const;
121
122 [[nodiscard]] const QStringList & contentClasses() const;
123 [[nodiscard]] const QStringList & negatedContentClasses() const;
124 [[nodiscard]] bool hasAnyContentClass() const;
125 [[nodiscard]] bool hasNegatedAnyContentClass() const;
126
127 [[nodiscard]] const QStringList & placeNames() const;
128 [[nodiscard]] const QStringList & negatedPlaceNames() const;
129 [[nodiscard]] bool hasAnyPlaceName() const;
130 [[nodiscard]] bool hasNegatedAnyPlaceName() const;
131
132 [[nodiscard]] const QStringList & applicationData() const;
133 [[nodiscard]] const QStringList & negatedApplicationData() const;
134 [[nodiscard]] bool hasAnyApplicationData() const;
135 [[nodiscard]] bool hasNegatedAnyApplicationData() const;
136
137 [[nodiscard]] const QList<qint64> & reminderOrders() const;
138 [[nodiscard]] const QList<qint64> & negatedReminderOrders() const;
139 [[nodiscard]] bool hasAnyReminderOrder() const;
140 [[nodiscard]] bool hasNegatedAnyReminderOrder() const;
141
142 [[nodiscard]] const QList<qint64> & reminderTimes() const;
143 [[nodiscard]] const QList<qint64> & negatedReminderTimes() const;
144 [[nodiscard]] bool hasAnyReminderTime() const;
145 [[nodiscard]] bool hasNegatedAnyReminderTime() const;
146
147 [[nodiscard]] const QList<qint64> & reminderDoneTimes() const;
148 [[nodiscard]] const QList<qint64> & negatedReminderDoneTimes() const;
149 [[nodiscard]] bool hasAnyReminderDoneTime() const;
150 [[nodiscard]] bool hasNegatedAnyReminderDoneTime() const;
151
152 [[nodiscard]] bool hasUnfinishedToDo() const;
153 [[nodiscard]] bool hasNegatedUnfinishedToDo() const;

```

```

154
155     [[nodiscard]] bool hasFinishedToDo() const;
156     [[nodiscard]] bool hasNegatedFinishedToDo() const;
157
158     [[nodiscard]] bool hasAnyToDo() const;
159     [[nodiscard]] bool hasNegatedAnyToDo() const;
160
161     [[nodiscard]] bool hasEncryption() const;
162     [[nodiscard]] bool hasNegatedEncryption() const;
163
164     [[nodiscard]] const QStringList & contentSearchTerms() const;
165     [[nodiscard]] const QStringList & negatedContentSearchTerms() const;
166     [[nodiscard]] bool hasAnyContentSearchTerms() const;
167
168     [[nodiscard]] bool isMatcheable() const;
169
170     // utility::Printable
171     QTextStream & print(QTextStream & strm) const override;
172
173 private:
174     class Data;
175     QSharedDataPointer<Data> d;
176 };
177
178 [[nodiscard]] QUINTIER_EXPORT bool operator==(
179     const NoteSearchQuery & lhs, const NoteSearchQuery & rhs) noexcept;
180
181 [[nodiscard]] QUINTIER_EXPORT bool operator!=(
182     const NoteSearchQuery & lhs, const NoteSearchQuery & rhs) noexcept;
183
184 } // namespace quantier::local_storage

```

6.25 MockILocalStorage.h

```

1 /*
2  * Copyright 2021-2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/local_storage/ILocalStorage.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quantier::local_storage::tests::mocks {
26
27     class MockILocalStorage : public ILocalStorage
28     {
29     public:
30         MOCK_METHOD(QFuture<bool>, isVersionTooHigh, (), (const, override));
31         MOCK_METHOD(QFuture<bool>, requiresUpgrade, (), (const, override));
32
33         MOCK_METHOD(
34             QFuture<QList<IPatchPtr>, requiredPatches, (), (const, override));
35
36         MOCK_METHOD(QFuture<qint32>, version, (), (const, override));
37
38         MOCK_METHOD(
39             QFuture<qint32>, highestSupportedVersion, (), (const, override));
40
41         MOCK_METHOD(QFuture<quint32>, userCount, (), (const, override));
42         MOCK_METHOD(QFuture<void>, putUser, (qevercloud::User user), (override));
43
44         MOCK_METHOD(
45             QFuture<std::optional<qevercloud::User>, findUserById,
46             (qevercloud::UserID userId), (const, override));
47
48         MOCK_METHOD(
49             QFuture<void>, expungeUserById, (qevercloud::UserID userId),

```

```

50         (override));
51
52     MOCK_METHOD(QFuture<quint32>, notebookCount, (), (const, override));
53
54     MOCK_METHOD(
55         QFuture<void>, putNotebook, (qevercloud::Notebook notebook),
56         (override));
57
58     MOCK_METHOD(
59         QFuture<std::optional<qevercloud::Notebook>>, findNotebookByLocalId,
60         (QString localId), (const, override));
61
62     MOCK_METHOD(
63         QFuture<std::optional<qevercloud::Notebook>>, findNotebookByGuid,
64         (qevercloud::Guid guid), (const, override));
65
66     MOCK_METHOD(
67         QFuture<std::optional<qevercloud::Notebook>>, findNotebookByName,
68         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
69         (const, override));
70
71     MOCK_METHOD(
72         QFuture<std::optional<qevercloud::Notebook>>, findDefaultNotebook, (),
73         (const, override));
74
75     MOCK_METHOD(
76         QFuture<void>, expungeNotebookByLocalId, (QString localId), (override));
77
78     MOCK_METHOD(
79         QFuture<void>, expungeNotebookByGuid, (qevercloud::Guid guid),
80         (override));
81
82     MOCK_METHOD(
83         QFuture<void>, expungeNotebookByName,
84         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
85         (override));
86
87     MOCK_METHOD(
88         QFuture<QList<qevercloud::Notebook>>, listNotebooks,
89         (ListNotebooksOptions options), (const, override));
90
91     MOCK_METHOD(
92         QFuture<QList<qevercloud::SharedNotebook>>, listSharedNotebooks,
93         (qevercloud::Guid notebookGuid), (const, override));
94
95     MOCK_METHOD(
96         QFuture<QSet<qevercloud::Guid>>, listNotebookGuids,
97         (ListGuidsFilters filters,
98         std::optional<qevercloud::Guid> linkedNotebookGuid),
99         (const, override));
100
101     MOCK_METHOD(QFuture<quint32>, linkedNotebookCount, (), (const, override));
102
103     MOCK_METHOD(
104         QFuture<void>, putLinkedNotebook,
105         (qevercloud::LinkedNotebook linkedNotebook), (override));
106
107     MOCK_METHOD(
108         QFuture<std::optional<qevercloud::LinkedNotebook>>,
109         findLinkedNotebookByGuid, (qevercloud::Guid guid), (const, override));
110
111     MOCK_METHOD(
112         QFuture<void>, expungeLinkedNotebookByGuid, (qevercloud::Guid guid),
113         (override));
114
115     MOCK_METHOD(
116         QFuture<QList<qevercloud::LinkedNotebook>>, listLinkedNotebooks,
117         (ListLinkedNotebooksOptions options), (const, override));
118
119     MOCK_METHOD(
120         QFuture<quint32>, noteCount, (NoteCountOptions options),
121         (const, override));
122
123     MOCK_METHOD(
124         QFuture<quint32>, noteCountPerNotebookLocalId,
125         (QString notebookLocalId, NoteCountOptions options), (const, override));
126
127     MOCK_METHOD(
128         QFuture<quint32>, noteCountPerTagLocalId,
129         (QString tagLocalId, NoteCountOptions options), (const, override));
130
131     MOCK_METHOD(
132         QFuture<QHash<QString, quint32>>, noteCountsPerTags,
133         (ListTagsOptions listTagsOptions, NoteCountOptions options),
134         (const, override));
135
136     MOCK_METHOD(

```

```

137         QFuture<quint32>, noteCountPerNotebookAndTagLocalIds,
138         (QStringList notebookLocalIds, QStringList tagLocalIds,
139          NoteCountOptions options),
140         (const, override));
141
142     MOCK_METHOD(QFuture<void>, putNote, (qevercloud::Note note), (override));
143
144     MOCK_METHOD(
145         QFuture<void>, updateNote,
146         (qevercloud::Note note, UpdateNoteOptions options), (override));
147
148     MOCK_METHOD(
149         QFuture<std::optional<qevercloud::Note>, findNoteByLocalId,
150         (QString localId, FetchNoteOptions options), (const, override));
151
152     MOCK_METHOD(
153         QFuture<std::optional<qevercloud::Note>, findNoteByGuid,
154         (qevercloud::Guid guid, FetchNoteOptions options), (const, override));
155
156     MOCK_METHOD(
157         QFuture<void>, expungeNoteByLocalId, (QString localId), (override));
158
159     MOCK_METHOD(
160         QFuture<void>, expungeNoteByGuid, (qevercloud::Guid guid), (override));
161
162     MOCK_METHOD(
163         QFuture<QList<qevercloud::Note>, listNotes,
164         (FetchNoteOptions fetchOptions, ListNotesOptions options),
165         (const, override));
166
167     MOCK_METHOD(
168         QFuture<QList<qevercloud::Note>, listNotesPerNotebookLocalId,
169         (QString notebookLocalId, FetchNoteOptions fetchOptions,
170          ListNotesOptions options),
171         (const, override));
172
173     MOCK_METHOD(
174         QFuture<QList<qevercloud::Note>, listNotesPerTagLocalId,
175         (QString tagLocalId, FetchNoteOptions fetchOptions,
176          ListNotesOptions options),
177         (const, override));
178
179     MOCK_METHOD(
180         QFuture<QList<qevercloud::Note>, listNotesPerNotebookAndTagLocalIds,
181         (QStringList notebookLocalIds, QStringList tagLocalIds,
182          FetchNoteOptions fetchOptions, ListNotesOptions options),
183         (const, override));
184
185     MOCK_METHOD(
186         QFuture<QList<qevercloud::Note>, listNotesByLocalIds,
187         (QStringList noteLocalIds, FetchNoteOptions fetchOptions,
188          ListNotesOptions options),
189         (const, override));
190
191     MOCK_METHOD(
192         QFuture<QSet<qevercloud::Guid>, listNoteGuids,
193         (ListGuidsFilters filters,
194          std::optional<qevercloud::Guid> linkedNotebookGuid),
195         (const, override));
196
197     MOCK_METHOD(
198         QFuture<QList<qevercloud::Note>, queryNotes,
199         (NoteSearchQuery query, FetchNoteOptions fetchOptions),
200         (const, override));
201
202     MOCK_METHOD(
203         QFuture<QStringList>, queryNoteLocalIds, (NoteSearchQuery query),
204         (const, override));
205
206     MOCK_METHOD(QFuture<quint32>, tagCount, (), (const, override));
207     MOCK_METHOD(QFuture<void>, putTag, (qevercloud::Tag tag), (override));
208
209     MOCK_METHOD(
210         QFuture<std::optional<qevercloud::Tag>, findTagByLocalId,
211         (QString tagLocalId), (const, override));
212
213     MOCK_METHOD(
214         QFuture<std::optional<qevercloud::Tag>, findTagByGuid,
215         (qevercloud::Guid tagGuid), (const, override));
216
217     MOCK_METHOD(
218         QFuture<std::optional<qevercloud::Tag>, findTagByName,
219         (QString tagName, std::optional<QString> linkedNotebookGuid),
220         (const, override));
221
222     MOCK_METHOD(
223         QFuture<QList<qevercloud::Tag>, listTags, (ListTagsOptions options),

```

```

224         (const, override));
225
226     MOCK_METHOD(
227         QFuture<QList<qevercloud::Tag>, listTagsPerNoteLocalId,
228         (QString noteLocalId, ListTagsOptions options), (const, override));
229
230     MOCK_METHOD(
231         QFuture<QSet<qevercloud::Guid>, listTagGuids,
232         (ListGuidsFilters filters,
233         std::optional<qevercloud::Guid> linkedNotebookGuid),
234         (const, override));
235
236     MOCK_METHOD(
237         QFuture<void>, expungeTagByLocalId, (QString tagLocalId), (override));
238
239     MOCK_METHOD(
240         QFuture<void>, expungeTagByGuid, (qevercloud::Guid tagGuid),
241         (override));
242
243     MOCK_METHOD(
244         QFuture<void>, expungeTagByName,
245         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
246         (override));
247
248     MOCK_METHOD(
249         QFuture<quint32>, resourceCount, (NoteCountOptions options),
250         (const, override));
251
252     MOCK_METHOD(
253         QFuture<quint32>, resourceCountPerNoteLocalId, (QString noteLocalId),
254         (const, override));
255
256     MOCK_METHOD(
257         QFuture<void>, putResource, (qevercloud::Resource resource),
258         (override));
259
260     MOCK_METHOD(
261         QFuture<std::optional<qevercloud::Resource>, findResourceByLocalId,
262         (QString resourceLocalId, FetchResourceOptions options),
263         (const, override));
264
265     MOCK_METHOD(
266         QFuture<std::optional<qevercloud::Resource>, findResourceByGuid,
267         (qevercloud::Guid resourceGuid, FetchResourceOptions options),
268         (const, override));
269
270     MOCK_METHOD(
271         QFuture<void>, expungeResourceByLocalId, (QString resourceLocalId),
272         (override));
273
274     MOCK_METHOD(
275         QFuture<void>, expungeResourceByGuid, (qevercloud::Guid resourceGuid),
276         (override));
277
278     MOCK_METHOD(QFuture<quint32>, savedSearchCount, (), (const, override));
279
280     MOCK_METHOD(
281         QFuture<void>, putSavedSearch, (qevercloud::SavedSearch search),
282         (override));
283
284     MOCK_METHOD(
285         QFuture<std::optional<qevercloud::SavedSearch>,
286         findSavedSearchByLocalId, (QString localId), (const, override));
287
288     MOCK_METHOD(
289         QFuture<std::optional<qevercloud::SavedSearch>, findSavedSearchByGuid,
290         (qevercloud::Guid guid), (const, override));
291
292     MOCK_METHOD(
293         QFuture<std::optional<qevercloud::SavedSearch>, findSavedSearchByName,
294         (QString name), (const, override));
295
296     MOCK_METHOD(
297         QFuture<QList<qevercloud::SavedSearch>, listSavedSearches,
298         (ListSavedSearchesOptions options), (const, override));
299
300     MOCK_METHOD(
301         QFuture<QSet<qevercloud::Guid>, listSavedSearchGuids,
302         (ListGuidsFilters filters), (const, override));
303
304     MOCK_METHOD(
305         QFuture<void>, expungeSavedSearchByLocalId, (QString localId),
306         (override));
307
308     MOCK_METHOD(
309         QFuture<void>, expungeSavedSearchByGuid, (qevercloud::Guid guid),
310         (override));

```

```

311
312     MOCK_METHOD(
313         QFuture<qint32>, highestUpdateSequenceNumber, (HighestUsnOption option),
314         (const, override));
315
316     MOCK_METHOD(
317         QFuture<qint32>, highestUpdateSequenceNumber,
318         (qevercloud::Guid linkedNotebookGuid), (const, override));
319
320     MOCK_METHOD(ILocalStorageNotifier *, notifier, (), (const, override));
321 };
322
323 } // namespace quentier::local_storage::tests::mocks

```

6.26 QuentierLogger.h

```

1  /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QDebug>
24 #include <QRegularExpression>
25 #include <QString>
26 #include <QTextStream>
27
28 namespace quentier {
29
30     enum class LogLevel
31     {
32         Trace,
33         Debug,
34         Info,
35         Warning,
36         Error
37     };
38
39     QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, LogLevel logLevel);
40
41     QUENTIER_EXPORT QTextStream & operator<<(QTextStream & strm, LogLevel logLevel);
42
43     void QUENTIER_EXPORT QuentierInitializeLogging();
44
45     void QUENTIER_EXPORT QuentierAddLogEntry(
46         const QString & sourceFileName, int sourceFileLineNumber,
47         const QString & component, const QString & message, LogLevel logLevel);
48
49     LogLevel QUENTIER_EXPORT QuentierMinLogLevel();
50
51     void QUENTIER_EXPORT QuentierSetMinLogLevel(LogLevel logLevel);
52
53     void QUENTIER_EXPORT QuentierAddStdOutLogDestination();
54
55     [[nodiscard]] bool QUENTIER_EXPORT QuentierIsLogLevelActive(LogLevel logLevel);
56
57     [[nodiscard]] QString QUENTIER_EXPORT QuentierLogFilesDirPath();
58
59     void QUENTIER_EXPORT QuentierRestartLogging();
60
61     [[nodiscard]] QRegularExpression QUENTIER_EXPORT QuentierLogComponentFilter();
62
63     void QUENTIER_EXPORT
64         QuentierSetLogComponentFilter(const QRegularExpression & filter);
65
66 } // namespace quentier

```

```

110 #define QNLOG_PRIVATE_BASE(component, message, level)
111 if (quentier::QuentierIsLogLevelActive(quentier::LogLevel::level)) {
112     QString msg;
113     QDebug dbg(&msg);
114     dbg.nospace();
115     dbg.noquote();
116     dbg << message;
117     quentier::QuentierAddLogEntry(
118         QStringLiteral(__FILE__), __LINE__, QString::fromUtf8(component),
119         msg, quentier::LogLevel::level);
120 }
121 // QNLOG_PRIVATE_BASE
122
123 #define QNTRACE(component, message)
124 QNLOG_PRIVATE_BASE(component, message, Trace)
125 // QNTRACE
126
127 #define QNDEBUG(component, message)
128 QNLOG_PRIVATE_BASE(component, message, Debug)
129 // QNDEBUG
130
131 #define QNINFO(component, message)
132 QNLOG_PRIVATE_BASE(component, message, Info)
133 // QNINFO
134
135 #define QNWARNING(component, message)
136 QNLOG_PRIVATE_BASE(component, message, Warning)
137 // QNWARNING
138
139 #define QNERROR(component, message)
140 QNLOG_PRIVATE_BASE(component, message, Error)
141 // QNERROR
142
143 #define QUENTIER_SET_MIN_LOG_LEVEL(level)
144 quentier::QuentierSetMinLogLevel(quentier::LogLevel::level)
145 // QUENTIER_SET_MIN_LOG_LEVEL
146
147 #define QUENTIER_INITIALIZE_LOGGING() quentier::QuentierInitializeLogging()
148 // QUENTIER_INITIALIZE_LOGGING
149
150 // clang-format off
151 #define QUENTIER_ADD_STDOUT_LOG_DESTINATION()
152 quentier::QuentierAddStdOutLogDestination()
153 // QUENTIER_ADD_STDOUT_LOG_DESTINATION
154 // clang-format on
155
156 #define QNLOG_FILE_LINENUMBER_DELIMITER ":"

```

6.27 INoteEditorBackend.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/Fwd.h>
22 #include <quentier/local_storage/Fwd.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QPalette>
26 #include <QPrinter>
27 #include <QStringList>
28 #include <QThread>
29 #include <QWidget>
30
31 class QTextStream;
32 class QUndoStack;
33

```

```

34 namespace quentier {
35
36 class Account;
37 class ErrorString;
38 class NoteEditor;
39 class SpellChecker;
40
41 class QUENTIER_EXPORT INoteEditorBackend
42 {
43 public:
44     virtual ~INoteEditorBackend() noexcept;
45
46     virtual void initialize(
47         local_storage::ILocalStoragePtr localStorage,
48         SpellChecker & spellChecker, const Account & account,
49         QThread * backgroundJobsThread,
50         enml::IDecryptedTextCachePtr decryptedTextCache) = 0;
51
52     [[nodiscard]] virtual QObject * object() = 0; // provide QObject interface
53     [[nodiscard]] virtual QWidget * widget() = 0; // provide QWidget interface
54
55     virtual void setAccount(const Account & account) = 0;
56     virtual void setUndoStack(QUndoStack * pUndoStack) = 0;
57
58     virtual void setInitialPageHtml(const QString & html) = 0;
59     virtual void setNoteNotFoundPageHtml(const QString & html) = 0;
60     virtual void setNoteDeletedPageHtml(const QString & html) = 0;
61     virtual void setNoteLoadingPageHtml(const QString & html) = 0;
62
63     [[nodiscard]] virtual bool isNoteLoaded() const = 0;
64     [[nodiscard]] virtual qint64 idleTime() const = 0;
65
66     virtual void convertToNote() = 0;
67     virtual void saveNoteToLocalStorage() = 0;
68     virtual void setNoteTitle(const QString & noteTitle) = 0;
69
70     virtual void setTagIds(
71         const QStringList & tagLocalUids, const QStringList & tagGuids) = 0;
72
73     virtual void undo() = 0;
74     virtual void redo() = 0;
75     virtual void cut() = 0;
76     virtual void copy() = 0;
77     virtual void paste() = 0;
78     virtual void pasteUnformatted() = 0;
79     virtual void selectAll() = 0;
80
81     virtual void formatSelectionAsSourceCode() = 0;
82
83     virtual void fontMenu() = 0;
84     virtual void textBold() = 0;
85     virtual void textItalic() = 0;
86     virtual void textUnderline() = 0;
87     virtual void textStrikethrough() = 0;
88     virtual void textHighlight() = 0;
89
90     virtual void alignLeft() = 0;
91     virtual void alignCenter() = 0;
92     virtual void alignRight() = 0;
93     virtual void alignFull() = 0;
94
95     [[nodiscard]] virtual QString selectedText() const = 0;
96     [[nodiscard]] virtual bool hasSelection() const = 0;
97
98     virtual void findNext(const QString & text, bool matchCase) const = 0;
99
100     virtual void findPrevious(const QString & text, bool matchCase) const = 0;
101
102     virtual void replace(
103         const QString & textToReplace, const QString & replacementText,
104         bool matchCase) = 0;
105
106     virtual void replaceAll(
107         const QString & textToReplace, const QString & replacementText,
108         bool matchCase) = 0;
109
110     virtual void insertToDoCheckbox() = 0;
111
112     virtual void insertInAppNoteLink(
113         const QString & userId, const QString & shardId,
114         const QString & noteGuid, const QString & linkText) = 0;
115
116     virtual void setSpellcheck(bool enabled) = 0;
117     [[nodiscard]] virtual bool spellCheckEnabled() const = 0;
118
119     virtual void setFont(const QFont & font) = 0;
120     virtual void setFontHeight(int height) = 0;

```

```

121     virtual void setFontColor(const QColor & color) = 0;
122     virtual void setBackgroundColor(const QColor & color) = 0;
123
124     [[nodiscard]] virtual QPalette defaultPalette() const = 0;
125     virtual void setDefaultPalette(const QPalette & pal) = 0;
126
127     [[nodiscard]] virtual const QFont * defaultFont() const = 0;
128     virtual void setDefaultFont(const QFont & font) = 0;
129
130     virtual void insertHorizontalLine() = 0;
131
132     virtual void increaseFontSize() = 0;
133     virtual void decreaseFontSize() = 0;
134
135     virtual void increaseIndentation() = 0;
136     virtual void decreaseIndentation() = 0;
137
138     virtual void insertBulletedList() = 0;
139     virtual void insertNumberedList() = 0;
140
141     virtual void insertTableDialog() = 0;
142
143     virtual void insertFixedWidthTable(
144         int rows, int columns, int widthInPixels) = 0;
145
146     virtual void insertRelativeWidthTable(
147         int rows, int columns, double relativeWidth) = 0;
148
149     virtual void insertTableRow() = 0;
150     virtual void insertTableColumn() = 0;
151     virtual void removeTableRow() = 0;
152     virtual void removeTableColumn() = 0;
153
154     virtual void addAttachmentDialog() = 0;
155     virtual void saveAttachmentDialog(const QByteArray & resourceHash) = 0;
156     virtual void saveAttachmentUnderCursor() = 0;
157     virtual void openAttachment(const QByteArray & resourceHash) = 0;
158     virtual void openAttachmentUnderCursor() = 0;
159     virtual void copyAttachment(const QByteArray & resourceHash) = 0;
160     virtual void copyAttachmentUnderCursor() = 0;
161     virtual void removeAttachment(const QByteArray & resourceHash) = 0;
162     virtual void removeAttachmentUnderCursor() = 0;
163     virtual void renameAttachment(const QByteArray & resourceHash) = 0;
164     virtual void renameAttachmentUnderCursor() = 0;
165
166     enum class Rotation
167     {
168         Clockwise,
169         Counterclockwise
170     };
171
172     friend QUENTIER_EXPORT QTextStream & operator«(
173         QTextStream & strm, Rotation rotation);
174
175     friend QUENTIER_EXPORT QDebug & operator«(QDebug & dbg, Rotation rotation);
176
177     virtual void rotateImageAttachment(
178         const QByteArray & resourceHash, Rotation rotationDirection) = 0;
179
180     virtual void rotateImageAttachmentUnderCursor(
181         Rotation rotationDirection) = 0;
182
183     virtual void encryptSelectedText() = 0;
184
185     virtual void decryptEncryptedTextUnderCursor() = 0;
186
187     virtual void decryptEncryptedText(
188         QString encryptedText, QString cipher, QString hint,
189         QString enCryptIndex) = 0;
190
191     virtual void hideDecryptedTextUnderCursor() = 0;
192
193     virtual void hideDecryptedText(
194         QString encryptedText, QString decryptedText, QString cipher,
195         QString hint, QString enDecryptedIndex) = 0;
196
197     virtual void editHyperlinkDialog() = 0;
198     virtual void copyHyperlink() = 0;
199     virtual void removeHyperlink() = 0;
200
201     virtual void onNoteLoadCancelled() = 0;
202
203     [[nodiscard]] virtual bool print(
204         QPrinter & printer, ErrorString & errorDescription) = 0;
205
206     [[nodiscard]] virtual bool exportToPdf(
207         const QString & absoluteFilePath, ErrorString & errorDescription) = 0;

```

```

208
209     [[nodiscard]] virtual bool exportToEnex(
210         const QStringList & tagNames, QString & enex,
211         ErrorString & errorDescription) = 0;
212
213     [[nodiscard]] virtual QString currentNoteLocalId() const = 0;
214     virtual void setCurrentNoteLocalId(const QString & noteLocalUid) = 0;
215
216     virtual void clear() = 0;
217
218     [[nodiscard]] virtual bool isModified() const = 0;
219     [[nodiscard]] virtual bool isEditorPageModified() const = 0;
220
221     virtual void setFocusToEditor() = 0;
222
223 protected:
224     INoteEditorBackend(NoteEditor * parent);
225     NoteEditor * m_pNoteEditor;
226 };
227
228 } // namespace quentier

```

6.28 NoteEditor.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/Fwd.h>
22 #include <quentier/local_storage/Fwd.h>
23 #include <quentier/types/ErrorString.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <qevercloud/types/Note.h>
27 #include <qevercloud/types/Notebook.h>
28
29 #include <QPrinter>
30 #include <QStringList>
31 #include <QThread>
32 #include <QWidget>
33
34 class QUndoStack;
35
36 namespace quentier {
37
38     class Account;
39     class INoteEditorBackend;
40     class SpellChecker;
41
42     class QUENTIER_EXPORT NoteEditor : public QWidget
43     {
44     public:
45         Q_OBJECT
46     public:
47         explicit NoteEditor(
48             QWidget * parent = nullptr,
49             Qt::WindowFlags flags = {});
50
51         #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
52             Qt::WindowFlags flags = {});
53         #else
54             Qt::WindowFlags flags = 0); // NOLINT
55         #endif
56
57         ~NoteEditor() noexcept override;
58
59         void initialize(
60             local_storage::ILocalStoragePtr localStorage,
61             SpellChecker & spellChecker, const Account & account,
62             QThread * backgroundJobsThread = nullptr,

```

```

84         enml::IDecryptedTextCachePtr decryptedTextCache = nullptr;
85
89     [[nodiscard]] INoteEditorBackend * backend() noexcept;
90
96     void setBackend(INoteEditorBackend * backend);
97
101    void setAccount(const Account & account);
102
106    [[nodiscard]] const QUndoStack * undoStack() const noexcept;
107
111    void setUndoStack(QUndoStack * pUndoStack);
112
116    void setInitialPageHtml(const QString & html);
117
122    void setNoteNotFoundPageHtml(const QString & html);
123
129    void setNoteDeletedPageHtml(const QString & html);
130
135    void setNoteLoadingPageHtml(const QString & html);
136
140    [[nodiscard]] QString currentNoteLocalId() const;
141
149    void setCurrentNoteLocalId(const QString & noteLocalId);
150
154    void clear();
155
160    [[nodiscard]] bool isModified() const noexcept;
161
166    [[nodiscard]] bool isEditorPageModified() const noexcept;
167
172    [[nodiscard]] bool isNoteLoaded() const noexcept;
173
179    [[nodiscard]] quint64 idleTime() const noexcept;
180
184    void setFocus();
185
186    [[nodiscard]] QString selectedText() const noexcept;
187    [[nodiscard]] bool hasSelection() const noexcept;
188
189    [[nodiscard]] bool spellCheckEnabled() const noexcept;
190
191    [[nodiscard]] bool print(
192        QPrinter & printer, ErrorString & errorDescription);
193
194    [[nodiscard]] bool exportToPdf(
195        const QString & absoluteFilePath, ErrorString & errorDescription);
196
197    [[nodiscard]] bool exportToEnex(
198        const QStringList & tagNames, QString & enex,
199        ErrorString & errorDescription);
200
208    [[nodiscard]] QPalette defaultPalette() const;
209
214    [[nodiscard]] const QFont * defaultFont() const;
215
216    Q_SIGNALS:
221        void contentChanged();
222
228        void noteAndNotebookFoundInLocalStorage(
229            qevercloud::Note note, qevercloud::Notebook notebook);
230
235        void noteNotFound(QString noteLocalId);
236
242        void noteDeleted(QString noteLocalId);
243
249        void noteModified();
250
255        void notifyError(ErrorString error);
256
261        void inAppNoteLinkClicked(
262            QString userId, QString shardId, QString noteGuid);
263
275        void inAppNoteLinkPasteRequested(
276            QString url, QString userId, QString shardId, QString noteGuid);
277
278        void convertedToNote(qevercloud::Note note);
279        void cantConvertToNote(ErrorString error);
280
281        void noteEditorHtmlUpdated(QString html);
282
283        void currentNoteChanged(qevercloud::Note note);
284
285        void spellCheckerNotReady();
286        void spellCheckerReady();
287
288        void noteLoaded();
289

```

```

296 void noteSavedToLocalStorage(QString noteLocalId);
297
302 void failedToSaveNoteToLocalStorage(
303     QString errorDescription, QString noteLocalId);
304
305 // Signals to notify anyone interested of the formatting at the current
306 // cursor position
307 void textBoldState(bool state);
308 void textItalicState(bool state);
309 void textUnderlineState(bool state);
310 void textStrikethroughState(bool state);
311 void textAlignLeftState(bool state);
312 void textAlignCenterState(bool state);
313 void textAlignRightState(bool state);
314 void textAlignFullState(bool state);
315 void textInsideOrderedListState(bool state);
316 void textInsideUnorderedListState(bool state);
317 void textInsideTableState(bool state);
318
319 void textFontFamilyChanged(QString fontFamily);
320 void textFontSizeChanged(int fontSize);
321
322 void insertTableDialogRequested();
323
324 public Q_SLOTS:
325 void convertToNote();
326
327 void saveNoteToLocalStorage();
328
329 void setNoteTitle(const QString & noteTitle);
330
331 void setTagIds(
332     const QStringList & tagLocalIds, const QStringList & tagGuids);
333
334 void undo();
335 void redo();
336 void cut();
337 void copy();
338 void paste();
339 void pasteUnformatted();
340 void selectAll();
341
342 void formatSelectionAsSourceCode();
343
344 void fontMenu();
345 void textBold();
346 void textItalic();
347 void textUnderline();
348 void textStrikethrough();
349 void textHighlight();
350
351 void alignLeft();
352 void alignCenter();
353 void alignRight();
354 void alignFull();
355
356 void findNext(const QString & text, bool matchCase) const;
357 void findPrevious(const QString & text, bool matchCase) const;
358
359 void replace(
360     const QString & textToReplace, const QString & replacementText,
361     bool matchCase);
362
363 void replaceAll(
364     const QString & textToReplace, const QString & replacementText,
365     bool matchCase);
366
367 void insertToDoCheckbox();
368
369 void insertInAppNoteLink(
370     const QString & userId, const QString & shardId,
371     const QString & noteGuid, const QString & linkText);
372
373 void setSpellcheck(bool enabled);
374
375 void setFont(const QFont & font);
376 void setFontHeight(int height);
377 void setFontColor(const QColor & color);
378 void setBackgroundColor(const QColor & color);
379
380 void setDefaultPalette(const QPalette & pal);
381
382 void setDefaultFont(const QFont & font);
383
384 void insertHorizontalLine();
385
386 void increaseFontSize();

```

```

440     void decreaseFontSize();
441
442     void increaseIndentation();
443     void decreaseIndentation();
444
445     void insertBulletedList();
446     void insertNumberedList();
447
448     void insertTableDialog();
449
450     void insertFixedWidthTable(int rows, int columns, int widthInPixels);
451
452     void insertRelativeWidthTable(int rows, int columns, double relativeWidth);
453
454     void insertTableRow();
455     void insertTableColumn();
456     void removeTableRow();
457     void removeTableColumn();
458
459     void addAttachmentDialog();
460     void saveAttachmentDialog(const QByteArray & resourceHash);
461     void saveAttachmentUnderCursor();
462     void openAttachment(const QByteArray & resourceHash);
463     void openAttachmentUnderCursor();
464     void copyAttachment(const QByteArray & resourceHash);
465     void copyAttachmentUnderCursor();
466
467     void encryptSelectedText();
468     void decryptEncryptedTextUnderCursor();
469
470     void editHyperlinkDialog();
471     void copyHyperlink();
472     void removeHyperlink();
473
474     void onNoteLoadCancelled();
475
476 protected:
477     void dragMoveEvent(QDragMoveEvent * pEvent) override;
478     void dropEvent(QDropEvent * pEvent) override;
479
480 private:
481     INoteEditorBackend * m_backend;
482 };
483
484 } // namespace quentier

```

6.29 SpellChecker.h

```

1  /*
2  * Copyright 2017-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QList>
25 #include <QObject>
26
27 #include <utility>
28
29 namespace quentier {
30
31 class Account;
32 class FileIOProcessorAsync;
33 class SpellCheckerPrivate;
34
35 class QUENTIER_EXPORT SpellChecker : public QObject

```

```

36 {
37     Q_OBJECT
38 public:
39     SpellChecker(
40         utility::FileIOProcessorAsync * fileIOProcessorAsync, Account account,
41         QObject * parent = nullptr, const QString & userDictionaryPath = {});
42
43     // The second bool in the pair indicates whether the dictionary
44     // is enabled or disabled
45     [[nodiscard]] QList<std::pair<QString, bool>> listAvailableDictionaries()
46         const;
47
48     void setAccount(const Account & account);
49
50     void enableDictionary(const QString & language);
51     void disableDictionary(const QString & language);
52
53     [[nodiscard]] bool checkSpell(const QString & word) const;
54
55     [[nodiscard]] QStringList spellCorrectionSuggestions(
56         const QString & misspelledWord) const;
57
58     void addToUserWordlist(const QString & word);
59     void removeFromUserWordlist(const QString & word);
60     void ignoreWord(const QString & word);
61     void removeWord(const QString & word);
62
63     [[nodiscard]] bool isReady() const noexcept;
64
65 Q_SIGNALS:
66     void ready();
67
68 private:
69     SpellCheckerPrivate * const d_ptr;
70     Q_DECLARE_PRIVATE(SpellChecker)
71 };
72
73 } // namespace quentier

```

6.30 IAuthenticator.h

```

1 /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/types/Account.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QFuture>
26
27 namespace quentier::synchronization {
28
29 class QUINTIER_EXPORT IAuthenticator
30 {
31 public:
32     virtual ~IAuthenticator() noexcept;
33
34     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr>
35         authenticateNewAccount() = 0;
36
37     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr> authenticateAccount(
38         Account account) = 0;
39 };
40
41 } // namespace quentier::synchronization

```

6.31 INoteStoreFactory.h

```

1 /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/Fwd.h>
24 #include <qevercloud/services/Fwd.h>
25 #include <qevercloud/types/TypeAliases.h>
26
27 #include <optional>
28
29 namespace quentier::synchronization {
30
31 class QUENTIER_EXPORT INoteStoreFactory
32 {
33 public:
34     virtual ~INoteStoreFactory();
35
36     [[nodiscard]] virtual qevercloud::INoteStorePtr createNoteStore(
37         QString noteStoreUrl = {},
38         std::optional<qevercloud::Guid> linkedNotebookGuid = {},
39         qevercloud::IRequestContextPtr ctx = {},
40         qevercloud::IRetryPolicyPtr retryPolicy = {}) = 0;
41 };
42
43 } // namespace quentier::synchronization

```

6.32 ISyncConflictResolver.h

```

1 /*
2  * Copyright 2021-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/types/Note.h>
24 #include <qevercloud/types/Notebook.h>
25 #include <qevercloud/types/SavedSearch.h>
26 #include <qevercloud/types/Tag.h>
27
28 #include <QFuture>
29
30 #include <variant>
31
32 class QDebug;
33 class QTextStream;
34
35 namespace quentier::synchronization {

```

```

36
41 class QUENTIER_EXPORT ISyncConflictResolver
42 {
43 public:
44     struct QUENTIER_EXPORT ConflictResolution
45     {
46         struct QUENTIER_EXPORT UseTheirs
47         {};
48
49         struct QUENTIER_EXPORT UseMine
50         {};
51
52         struct QUENTIER_EXPORT IgnoreMine
53         {};
54
55         template <class T>
56         struct MoveMine
57         {
58             using value_type = T;
59
60             T mine;
61         };
62     };
63
64     using NotebookConflictResolution = std::variant<
65         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
66         ConflictResolution::IgnoreMine,
67         ConflictResolution::MoveMine<qevercloud::Notebook>>;
68
69     using NoteConflictResolution = std::variant<
70         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
71         ConflictResolution::IgnoreMine,
72         ConflictResolution::MoveMine<qevercloud::Note>>;
73
74     using SavedSearchConflictResolution = std::variant<
75         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
76         ConflictResolution::IgnoreMine,
77         ConflictResolution::MoveMine<qevercloud::SavedSearch>>;
78
79     using TagConflictResolution = std::variant<
80         ConflictResolution::IgnoreMine, ConflictResolution::UseTheirs,
81         ConflictResolution::UseMine,
82         ConflictResolution::MoveMine<qevercloud::Tag>>;
83
84 public:
85     virtual ~ISyncConflictResolver() noexcept;
86
87     [[nodiscard]] virtual QFuture<NotebookConflictResolution>
88         resolveNotebookConflict(
89             qevercloud::Notebook theirs, qevercloud::Notebook mine) = 0;
90
91     [[nodiscard]] virtual QFuture<NoteConflictResolution> resolveNoteConflict(
92         qevercloud::Note theirs, qevercloud::Note mine) = 0;
93
94     [[nodiscard]] virtual QFuture<SavedSearchConflictResolution>
95         resolveSavedSearchConflict(
96             qevercloud::SavedSearch theirs, qevercloud::SavedSearch mine) = 0;
97
98     [[nodiscard]] virtual QFuture<TagConflictResolution> resolveTagConflict(
99         qevercloud::Tag theirs, qevercloud::Tag mine) = 0;
100 };
101
102 QUENTIER_EXPORT QTextStream & operator<<(
103     QTextStream & strm,
104     const ISyncConflictResolver::NotebookConflictResolution & resolution);
105
106 QUENTIER_EXPORT QDebug & operator<<(
107     QDebug & dbg,
108     const ISyncConflictResolver::NotebookConflictResolution & resolution);
109
110 QUENTIER_EXPORT QTextStream & operator<<(
111     QTextStream & strm,
112     const ISyncConflictResolver::NoteConflictResolution & resolution);
113
114 QUENTIER_EXPORT QDebug & operator<<(
115     QDebug & dbg,
116     const ISyncConflictResolver::NoteConflictResolution & resolution);
117
118 QUENTIER_EXPORT QTextStream & operator<<(
119     QTextStream & strm,
120     const ISyncConflictResolver::SavedSearchConflictResolution & resolution);
121
122 QUENTIER_EXPORT QDebug & operator<<(
123     QDebug & dbg,
124     const ISyncConflictResolver::SavedSearchConflictResolution & resolution);
125
126 QUENTIER_EXPORT QTextStream & operator<<(

```

```

155     QTextStream & strm,
156     const ISyncConflictResolver::TagConflictResolution & resolution);
157
158 QUENTIER_EXPORT QDebug & operator«(
159     QDebug & dbg,
160     const ISyncConflictResolver::TagConflictResolution & resolution);
161
162 } // namespace quentier::synchronization

```

6.33 ISyncEventsNotifier.h

```

1 /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/Fwd.h>
22 #include <quentier/synchronization/types/Fwd.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <qevercloud/types/LinkedNotebook.h>
26
27 #include <QList>
28 #include <QObject>
29
30 namespace quentier::synchronization {
31
32 class QUENTIER_EXPORT ISyncEventsNotifier : public QObject
33 {
34     Q_OBJECT
35 protected:
36     explicit ISyncEventsNotifier(QObject * parent = nullptr);
37
38 public:
39     ~ISyncEventsNotifier() override;
40
41 Q_SIGNALS:
42     void syncChunksDownloadProgress(
43         qint32 highestDownloadedUsn, qint32 highestServerUsn,
44         qint32 lastPreviousUsn);
45
46     void syncChunksDownloaded();
47
48     void syncChunksDataProcessingProgress(ISyncChunksDataCountersPtr counters);
49
50     void startLinkedNotebooksDataDownloading(
51         const QList<qevercloud::LinkedNotebook> & linkedNotebooks);
52
53     void linkedNotebookSyncChunksDownloadProgress(
54         qint32 highestDownloadedUsn, qint32 highestServerUsn,
55         qint32 lastPreviousUsn,
56         const qevercloud::LinkedNotebook & linkedNotebook);
57
58     void linkedNotebookSyncChunksDownloaded(
59         const qevercloud::LinkedNotebook & linkedNotebook);
60
61     void linkedNotebookSyncChunksDataProcessingProgress(
62         ISyncChunksDataCountersPtr counters,
63         const qevercloud::LinkedNotebook & linkedNotebook);
64
65     void notesDownloadProgress(
66         quint32 notesDownloaded, quint32 totalNotesToDownload);
67
68     void linkedNotebookNotesDownloadProgress(
69         quint32 notesDownloaded, quint32 totalNotesToDownload,
70         const qevercloud::LinkedNotebook & linkedNotebook);
71
72     void resourcesDownloadProgress(

```

```

166         quint32 resourcesDownloaded, quint32 totalResourcesToDownload);
167
179     void linkedNotebookResourcesDownloadProgress(
180         quint32 resourcesDownloaded, quint32 totalResourcesToDownload,
181         const qevercloud::LinkedNotebook & linkedNotebook);
182
191     void downloadFinished(bool dataDownloaded);
192
199     void userOwnSendStatusUpdate(ISendStatusPtr sendStatus);
200
209     void linkedNotebookSendStatusUpdate(
210         const qevercloud::Guid & linkedNotebookGuid, ISendStatusPtr sendStatus);
211 };
212
213 } // namespace quantier::synchronization

```

6.34 ISynchronizer.h

```

1 /*
2  * Copyright 2021-2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/local_storage/Fwd.h>
22 #include <quantier/synchronization/Fwd.h>
23 #include <quantier/synchronization/types/Fwd.h>
24 #include <quantier/types/Account.h>
25 #include <quantier/utility/Linkage.h>
26 #include <quantier/utility/cancelers/Fwd.h>
27
28 #include <qevercloud/types/TypeAliases.h>
29
30 #include <QFuture>
31
32 #include <memory>
33 #include <utility>
34
35 namespace quantier {
36
37     class Account;
38
39 } // namespace quantier
40
41 namespace quantier::synchronization {
42
43     class QUENTIER_EXPORT ISynchronizer
44     {
45     public:
46         virtual ~ISynchronizer() noexcept;
47
48         [[nodiscard]] virtual QFuture<std::pair<Account, IAuthenticationInfoPtr>>
49             authenticateNewAccount() = 0;
50
51         [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr> authenticateAccount(
52             Account account) = 0;
53
54         using SyncResult =
55             std::pair<QFuture<ISyncResultPtr>, ISyncEventsNotifier *>;
56
57         [[nodiscard]] virtual SyncResult synchronizeAccount(
58             Account account, local_storage::ILocalStoragePtr localStorage,
59             utility::cancelers::ICancelerPtr canceler,
60             ISyncOptionsPtr options = nullptr,
61             ISyncConflictResolverPtr syncConflictResolver = nullptr) = 0;
62
63         virtual void revokeAuthentication(qevercloud::UserID userId) = 0;
64     };
65
66 } // namespace quantier::synchronization

```

6.35 ISyncStateStorage.h

```

1  /*
2  * Copyright 2020-2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/types/Account.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QObject>
26
27 namespace quentier::synchronization {
28
29     class QUENTIER_EXPORT ISyncStateStorage : public QObject
30     {
31     public:
32         Q_OBJECT
33     protected:
34         explicit ISyncStateStorage(QObject * parent = nullptr);
35     public:
36         ~ISyncStateStorage() override;
37
38         [[nodiscard]] virtual ISyncStatePtr getSyncState(
39             const Account & account) = 0;
40
41         virtual void setSyncState(
42             const Account & account, ISyncStatePtr syncState) = 0;
43
44     Q_SIGNALS:
45         void notifySyncStateUpdated(Account account, ISyncStatePtr syncState);
46     };
47
48 } // namespace quentier::synchronization

```

6.36 IUserStoreFactory.h

```

1  /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/Fwd.h>
24 #include <qevercloud/services/Fwd.h>
25
26 #include <QString>
27
28 namespace quentier::synchronization {
29
30     class QUENTIER_EXPORT IUserStoreFactory

```

```

31 {
32 public:
33     virtual ~IUserStoreFactory();
34
35     [[nodiscard]] virtual qevercloud::IUserStorePtr createUserStore(
36         QString userStoreUrl = {}, qevercloud::IRequestContextPtr ctx = {},
37         qevercloud::IRetryPolicyPtr retryPolicy = {}) = 0;
38 };
39
40 } // namespace quantier::synchronization

```

6.37 MockIAuthenticator.h

```

1 /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/IAuthenticator.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quantier::synchronization::tests::mocks {
26
27 class MockIAuthenticator : public IAuthenticator
28 {
29 public:
30     MOCK_METHOD(
31         QFuture<IAuthenticationInfoPtr>, authenticateNewAccount, (),
32         (override));
33
34     MOCK_METHOD(
35         QFuture<IAuthenticationInfoPtr>, authenticateAccount, (Account account),
36         (override));
37 };
38
39 } // namespace quantier::synchronization::tests::mocks

```

6.38 MockINoteStoreFactory.h

```

1 /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/INoteStoreFactory.h>
22
23 #include <gmock/gmock.h>
24

```

```

25 namespace quantier::synchronization::tests::mocks {
26
27 class MockINoteStoreFactory : public INoteStoreFactory
28 {
29 public:
30     MOCK_METHOD(
31         ::qevercloud::INoteStorePtr, createNoteStore,
32         (QString noteStoreUrl,
33          std::optional<::qevercloud::Guid> linkedNotebookGuid,
34          ::qevercloud::IRequestContextPtr ctx,
35          ::qevercloud::IRetryPolicyPtr retryPolicy),
36         (override));
37 };
38
39 } // namespace quantier::synchronization::tests::mocks

```

6.39 MockISyncConflictResolver.h

```

1 /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/ISyncConflictResolver.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quantier::synchronization::tests::mocks {
26
27 class MockISyncConflictResolver : public ISyncConflictResolver
28 {
29 public:
30     MOCK_METHOD(
31         QFuture<NotebookConflictResolution>, resolveNotebookConflict,
32         (::qevercloud::Notebook theirs, ::qevercloud::Notebook mine),
33         (override));
34
35     MOCK_METHOD(
36         QFuture<NoteConflictResolution>, resolveNoteConflict,
37         (::qevercloud::Note theirs, ::qevercloud::Note mine), (override));
38
39     MOCK_METHOD(
40         QFuture<SavedSearchConflictResolution>, resolveSavedSearchConflict,
41         (::qevercloud::SavedSearch theirs, ::qevercloud::SavedSearch mine),
42         (override));
43
44     MOCK_METHOD(
45         QFuture<TagConflictResolution>, resolveTagConflict,
46         (::qevercloud::Tag theirs, ::qevercloud::Tag mine), (override));
47 };
48
49 } // namespace quantier::synchronization::tests::mocks

```

6.40 MockISyncStateStorage.h

```

1 /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *

```

```

10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/ISyncStateStorage.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quentier::synchronization::tests::mocks {
26
27 class MockISyncStateStorage : public ISyncStateStorage
28 {
29     Q_OBJECT
30 public:
31     MOCK_METHOD(
32         ISyncStatePtr, getSyncState, (const Account & account), (override));
33
34     MOCK_METHOD(
35         void, setSyncState, (const Account & account, ISyncStatePtr syncState),
36         (override));
37 };
38
39 } // namespace quentier::synchronization::tests::mocks

```

6.41 Errors.h

```

1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QtGlobal>
24
25 #include <optional>
26 #include <variant>
27
28 namespace quentier::synchronization {
29
30 struct QUENTIER_EXPORT RateLimitReachedError
31 {
32     std::optional<qint32> rateLimitDurationSec;
33 };
34
35 [[nodiscard]] QUENTIER_EXPORT bool operator==(
36     const RateLimitReachedError & lhs,
37     const RateLimitReachedError & rhs) noexcept;
38
39 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
40     const RateLimitReachedError & lhs,
41     const RateLimitReachedError & rhs) noexcept;
42
43 struct QUENTIER_EXPORT AuthenticationExpiredError
44 {};
45
46 [[nodiscard]] QUENTIER_EXPORT bool operator==(
47     const AuthenticationExpiredError & lhs,
48     const AuthenticationExpiredError & rhs) noexcept;
49
50 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
51     const AuthenticationExpiredError & lhs,
52     const AuthenticationExpiredError & rhs) noexcept;
53
54 }

```

```

68     const AuthenticationExpiredError & lhs,
69     const AuthenticationExpiredError & rhs) noexcept;
70
76 using StopSynchronizationError = std::variant<
77     RateLimitReachedError, AuthenticationExpiredError, std::monostate>;
78
79 } // namespace quantier::synchronization

```

6.42 IAuthenticationInfo.h

```

1  /*
2  * Copyright 2022-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/types/Fwd.h>
22 #include <quantier/utility/Linkage.h>
23 #include <quantier/utility/Printable.h>
24
25 #include <qevercloud/types/TypeAliases.h>
26
27 #include <QList>
28 #include <QNetworkCookie>
29 #include <QString>
30
31 namespace quantier::synchronization {
32
33 class QUENTIER_EXPORT IAuthenticationInfo : public utility::Printable
34 {
35 public:
36     [[nodiscard]] virtual qevercloud::UserID userId() const = 0;
37
38     [[nodiscard]] virtual QString authToken() const = 0;
39
40     [[nodiscard]] virtual qevercloud::Timestamp authTokenExpirationTime()
41         const = 0;
42
43     [[nodiscard]] virtual qevercloud::Timestamp authenticationTime() const = 0;
44
45     [[nodiscard]] virtual QString shardId() const = 0;
46
47     [[nodiscard]] virtual QString noteStoreUrl() const = 0;
48
49     [[nodiscard]] virtual QString webApiUrlPrefix() const = 0;
50
51     [[nodiscard]] virtual QList<QNetworkCookie> userStoreCookies() const = 0;
52 };
53
54 } // namespace quantier::synchronization

```

6.43 IAuthenticationInfoBuilder.h

```

1  /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```

13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/types/TypeAliases.h>
25
26 #include <QList>
27 #include <QNetworkCookie>
28 #include <QString>
29
30 namespace quentier::synchronization {
31
32 class QUENTIER_EXPORT IAuthenticationInfoBuilder
33 {
34 public:
35     virtual ~IAuthenticationInfoBuilder() noexcept;
36
37     virtual IAuthenticationInfoBuilder & setUserId(
38         qevercloud::UserID userId) = 0;
39
40     virtual IAuthenticationInfoBuilder & setAuthToken(QString token) = 0;
41
42     virtual IAuthenticationInfoBuilder & setAuthTokenExpirationTime(
43         qevercloud::Timestamp expirationTime) = 0;
44
45     virtual IAuthenticationInfoBuilder & setAuthenticationTime(
46         qevercloud::Timestamp authenticationTime) = 0;
47
48     virtual IAuthenticationInfoBuilder & setShardId(QString shardId) = 0;
49
50     virtual IAuthenticationInfoBuilder & setNoteStoreUrl(
51         QString noteStoreUrl) = 0;
52
53     virtual IAuthenticationInfoBuilder & setWebApiUrlPrefix(
54         QString webApiUrlPrefix) = 0;
55
56     virtual IAuthenticationInfoBuilder & setUserStoreCookies(
57         QList<QNetworkCookie> cookies) = 0;
58
59     [[nodiscard]] virtual IAuthenticationInfoPtr build() = 0;
60 };
61
62 [[nodiscard]] QUENTIER_EXPORT IAuthenticationInfoBuilderPtr
63 createAuthenticationInfoBuilder();
64
65 } // namespace quentier::synchronization

```

6.44 IDownloadNotesStatus.h

```

1 /*
2 * Copyright 2022-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Errors.h>
22 #include <quentier/synchronization/types/Fwd.h>
23 #include <quentier/utility/Linkage.h>
24 #include <quentier/utility/Printable.h>
25
26 #include <qevercloud/types/Note.h>
27 #include <qevercloud/types/TypeAliases.h>

```

```

28
29 #include <QException>
30 #include <QList>
31
32 #include <memory>
33 #include <utility>
34
35 namespace quantier::synchronization {
36
41 class QUENTIER_EXPORT IDownloadNotesStatus : public utility::Printable
42 {
43 public:
44     using QExceptionPtr = std::shared_ptr<QException>;
45     using NoteWithException = std::pair<qevercloud::Note, QExceptionPtr>;
46     using GuidWithException = std::pair<qevercloud::Guid, QExceptionPtr>;
47     using UpdateSequenceNumbersByGuid = QHash<qevercloud::Guid, qint32>;
48
49     [[nodiscard]] virtual quint64 totalNewNotes() const = 0;
50     [[nodiscard]] virtual quint64 totalUpdatedNotes() const = 0;
51     [[nodiscard]] virtual quint64 totalExpungedNotes() const = 0;
52
53     [[nodiscard]] virtual QList<NoteWithException> notesWhichFailedToDownload()
54         const = 0;
55
56     [[nodiscard]] virtual QList<NoteWithException> notesWhichFailedToProcess()
57         const = 0;
58
59     [[nodiscard]] virtual QList<GuidWithException>
60         noteGuidsWhichFailedToExpunge() const = 0;
61
62     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
63         processedNoteGuidsAndUsns() const = 0;
64
65     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
66         cancelledNoteGuidsAndUsns() const = 0;
67
68     [[nodiscard]] virtual QList<qevercloud::Guid> expungedNoteGuids() const = 0;
69
70     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
71         const = 0;
72 };
73
74 } // namespace quantier::synchronization

```

6.45 IDownloadResourcesStatus.h

```

1 /*
2  * Copyright 2022-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/types/Errors.h>
22 #include <quantier/utility/Linkage.h>
23 #include <quantier/utility/Printable.h>
24
25 #include <qevercloud/types/Resource.h>
26 #include <qevercloud/types/TypeAliases.h>
27
28 #include <QException>
29
30 #include <memory>
31 #include <utility>
32
33 namespace quantier::synchronization {
34
35 class QUENTIER_EXPORT IDownloadResourcesStatus : public utility::Printable
36 {
37 public:

```

```

38     ~IDownloadResourcesStatus() noexcept override;
39
40     using QExceptionPtr = std::shared_ptr<QException>;
41
42     using ResourceWithException =
43         std::pair<qevercloud::Resource, QExceptionPtr>;
44
45     using UpdateSequenceNumbersByGuid = QHash<qevercloud::Guid, qint32>;
46
47     [[nodiscard]] virtual quint64 totalNewResources() const = 0;
48     [[nodiscard]] virtual quint64 totalUpdatedResources() const = 0;
49
50     [[nodiscard]] virtual QList<ResourceWithException>
51         resourcesWhichFailedToDownload() const = 0;
52
53     [[nodiscard]] virtual QList<ResourceWithException>
54         resourcesWhichFailedToProcess() const = 0;
55
56     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
57         processedResourceGuidsAndUsns() const = 0;
58
59     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
60         cancelledResourceGuidsAndUsns() const = 0;
61
62     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
63         const = 0;
64 };
65
66 } // namespace quantier::synchronization

```

6.46 ISendStatus.h

```

1  /*
2  * Copyright 2022-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/types/Errors.h>
22 #include <quantier/utility/Linkage.h>
23 #include <quantier/utility/Printable.h>
24
25 #include <qevercloud/types/Note.h>
26 #include <qevercloud/types/Notebook.h>
27 #include <qevercloud/types/SavedSearch.h>
28 #include <qevercloud/types/Tag.h>
29 #include <qevercloud/types/TypeAliases.h>
30
31 #include <QException>
32 #include <QList>
33
34 #include <memory>
35 #include <utility>
36
37 namespace quantier::synchronization {
38
39     class QUENTIER_EXPORT ISendStatus : public utility::Printable
40     {
41     public:
42         using QExceptionPtr = std::shared_ptr<QException>;
43
44         using NoteWithException = std::pair<qevercloud::Note, QExceptionPtr>;
45
46         using NotebookWithException =
47             std::pair<qevercloud::Notebook, QExceptionPtr>;
48
49         using SavedSearchWithException =
50             std::pair<qevercloud::SavedSearch, QExceptionPtr>;
51
52     };
53
54 }

```

```

57     using TagWithException = std::pair<qevercloud::Tag, QExceptionPtr>;
58
59 public:
60     // Total
61
62     [[nodiscard]] virtual quint64 totalAttemptedToSendNotes() const = 0;
63
64     [[nodiscard]] virtual quint64 totalAttemptedToSendNotebooks() const = 0;
65
66     [[nodiscard]] virtual quint64 totalAttemptedToSendSavedSearches() const = 0;
67
68     [[nodiscard]] virtual quint64 totalAttemptedToSendTags() const = 0;
69
70     // Notes
71
72     [[nodiscard]] virtual quint64 totalSuccessfullySentNotes() const = 0;
73
74     [[nodiscard]] virtual QList<NoteWithException> failedToSendNotes()
75         const = 0;
76
77     // Notebooks
78
79     [[nodiscard]] virtual quint64 totalSuccessfullySentNotebooks() const = 0;
80
81     [[nodiscard]] virtual QList<NotebookWithException> failedToSendNotebooks()
82         const = 0;
83
84     // Saved searches
85
86     [[nodiscard]] virtual quint64 totalSuccessfullySentSavedSearches()
87         const = 0;
88
89     [[nodiscard]] virtual QList<SavedSearchWithException>
90         failedToSendSavedSearches() const = 0;
91
92     // Tags
93
94     [[nodiscard]] virtual quint64 totalSuccessfullySentTags() const = 0;
95
96     [[nodiscard]] virtual QList<TagWithException> failedToSendTags() const = 0;
97
98     // General
99
100    [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
101        const = 0;
102
103    [[nodiscard]] virtual bool needToRepeatIncrementalSync() const = 0;
104 };
105
106 } // namespace quentier::synchronization

```

6.47 ISyncChunksDataCounters.h

```

1  /*
2  * Copyright 2021-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QtGlobal>
25
26 namespace quentier::synchronization {
27
28 struct QUINTIER_EXPORT ISyncChunksDataCounters : public utility::Printable
29 {
30     // ===== Saved searches =====

```

```

36
40     [[nodiscard]] virtual quint64 totalSavedSearches() const noexcept = 0;
41
45     [[nodiscard]] virtual quint64 totalExpungedSavedSearches()
46         const noexcept = 0;
47
52     [[nodiscard]] virtual quint64 addedSavedSearches() const noexcept = 0;
53
58     [[nodiscard]] virtual quint64 updatedSavedSearches() const noexcept = 0;
59
64     [[nodiscard]] virtual quint64 expungedSavedSearches() const noexcept = 0;
65
66     // ===== Tags =====
67
71     [[nodiscard]] virtual quint64 totalTags() const noexcept = 0;
72
76     [[nodiscard]] virtual quint64 totalExpungedTags() const noexcept = 0;
77
81     [[nodiscard]] virtual quint64 addedTags() const noexcept = 0;
82
86     [[nodiscard]] virtual quint64 updatedTags() const noexcept = 0;
87
91     [[nodiscard]] virtual quint64 expungedTags() const noexcept = 0;
92
93     // ===== Linked notebooks =====
94
98     [[nodiscard]] virtual quint64 totalLinkedNotebooks() const noexcept = 0;
99
103    [[nodiscard]] virtual quint64 totalExpungedLinkedNotebooks()
104        const noexcept = 0;
105
110    [[nodiscard]] virtual quint64 addedLinkedNotebooks() const noexcept = 0;
111
116    [[nodiscard]] virtual quint64 updatedLinkedNotebooks() const noexcept = 0;
117
122    [[nodiscard]] virtual quint64 expungedLinkedNotebooks() const noexcept = 0;
123
124    // ===== Notebooks =====
125
129    [[nodiscard]] virtual quint64 totalNotebooks() const noexcept = 0;
130
134    [[nodiscard]] virtual quint64 totalExpungedNotebooks() const noexcept = 0;
135
139    [[nodiscard]] virtual quint64 addedNotebooks() const noexcept = 0;
140
144    [[nodiscard]] virtual quint64 updatedNotebooks() const noexcept = 0;
145
150    [[nodiscard]] virtual quint64 expungedNotebooks() const noexcept = 0;
151 };
152
153 } // namespace quantier::synchronization

```

6.48 ISyncOptions.h

```

1 /*
2  * Copyright 2022-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/utility/Linkage.h>
22 #include <quantier/utility/Printable.h>
23
24 #include <qevercloud/Fwd.h>
25
26 #include <QDir>
27 #include <QtGlobal>
28
29 #include <optional>

```

```

30
31 namespace quantier::synchronization {
32
36 class QUENTIER_EXPORT ISyncOptions : public utility::Printable
37 {
38 public:
39     ~ISyncOptions() noexcept override;
40
46     [[nodiscard]] virtual bool downloadNoteThumbnails() const = 0;
47
62     [[nodiscard]] virtual std::optional<QDir> inkNoteImagesStorageDir()
63         const = 0;
64
69     [[nodiscard]] virtual qevercloud::IRequestContextPtr requestContext()
70         const = 0;
71
76     [[nodiscard]] virtual qevercloud::IRetryPolicyPtr retryPolicy() const = 0;
77
84     [[nodiscard]] virtual std::optional<quint32> maxConcurrentNoteDownloads()
85         const = 0;
86
93     [[nodiscard]] virtual std::optional<quint32>
94         maxConcurrentResourceDownloads() const = 0;
95 };
96
97 } // namespace quantier::synchronization

```

6.49 ISyncOptionsBuilder.h

```

1 /*
2  * Copyright 2022-2023 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/types/Fwd.h>
22 #include <quantier/utility/Linkage.h>
23
24 #include <qevercloud/Fwd.h>
25
26 #include <QDir>
27
28 #include <optional>
29
30 namespace quantier::synchronization {
31
32 class QUENTIER_EXPORT ISyncOptionsBuilder
33 {
34 public:
35     virtual ~ISyncOptionsBuilder() noexcept;
36
37     virtual ISyncOptionsBuilder & setDownloadNoteThumbnails(bool value) = 0;
38
39     virtual ISyncOptionsBuilder & setInkNoteImagesStorageDir(
40         std::optional<QDir> dir) = 0;
41
42     virtual ISyncOptionsBuilder & setRequestContext(
43         qevercloud::IRequestContextPtr ctx) = 0;
44
45     virtual ISyncOptionsBuilder & setRetryPolicy(
46         qevercloud::IRetryPolicyPtr retryPolicy) = 0;
47
48     virtual ISyncOptionsBuilder & setMaxConcurrentNoteDownloads(
49         std::optional<quint32> maxConcurrentNoteDownloads) = 0;
50
51     virtual ISyncOptionsBuilder & setMaxConcurrentResourceDownloads(
52         std::optional<quint32> maxConcurrentResourceDownloads) = 0;
53
54     [[nodiscard]] virtual ISyncOptionsPtr build() = 0;

```

```

55 };
56
57 [[nodiscard]] QUENTIER_EXPORT ISyncOptionsBuilderPtr createSyncOptionsBuilder();
58
59 } // namespace quantier::synchronization

```

6.50 ISyncResult.h

```

1  /*
2  * Copyright 2022-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/Fwd.h>
22 #include <quantier/synchronization/types/Errors.h>
23 #include <quantier/synchronization/types/Fwd.h>
24 #include <quantier/utility/Linkage.h>
25 #include <quantier/utility/Printable.h>
26
27 #include <qevercloud/types/TypeAliases.h>
28
29 #include <QHash>
30 #include <QSet>
31
32 namespace quantier::synchronization {
33
34 class QUENTIER_EXPORT ISyncResult : public utility::Printable
35 {
36 public:
37     [[nodiscard]] virtual ISyncStatePtr syncState() const = 0;
38
39     [[nodiscard]] virtual ISyncChunksDataCountersPtr
40         userAccountSyncChunksDataCounters() const = 0;
41
42     [[nodiscard]] virtual QHash<qevercloud::Guid, ISyncChunksDataCountersPtr>
43         linkedNotebookSyncChunksDataCounters() const = 0;
44
45     [[nodiscard]] virtual bool userAccountSyncChunksDownloaded() const = 0;
46
47     [[nodiscard]] virtual QSet<qevercloud::Guid>
48         linkedNotebookGuidsWithSyncChunksDownloaded() const = 0;
49
50     [[nodiscard]] virtual IDownloadNotesStatusPtr
51         userAccountDownloadNotesStatus() const = 0;
52
53     [[nodiscard]] virtual QHash<qevercloud::Guid, IDownloadNotesStatusPtr>
54         linkedNotebookDownloadNotesStatuses() const = 0;
55
56     [[nodiscard]] virtual IDownloadResourcesStatusPtr
57         userAccountDownloadResourcesStatus() const = 0;
58
59     [[nodiscard]] virtual QHash<qevercloud::Guid, IDownloadResourcesStatusPtr>
60         linkedNotebookDownloadResourcesStatuses() const = 0;
61
62     [[nodiscard]] virtual ISendStatusPtr userAccountSendStatus() const = 0;
63
64     [[nodiscard]] virtual QHash<qevercloud::Guid, ISendStatusPtr>
65         linkedNotebookSendStatuses() const = 0;
66
67     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
68         const = 0;
69 };
70
71 } // namespace quantier::synchronization

```

6.51 ISyncState.h

```

1 /*
2  * Copyright 2022-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <qevercloud/types/TypeAliases.h>
25
26 #include <QHash>
27 #include <QString>
28
29 namespace quentier::synchronization {
30
31 class QUENTIER_EXPORT ISyncState : public utility::Printable
32 {
33 public:
34     [[nodiscard]] virtual qint32 userDataUpdateCount() const = 0;
35
36     [[nodiscard]] virtual qevercloud::Timestamp userDataLastSyncTime()
37         const = 0;
38
39     [[nodiscard]] virtual QHash<qevercloud::Guid, qint32>
40         linkedNotebookUpdateCounts() const = 0;
41
42     [[nodiscard]] virtual QHash<qevercloud::Guid, qevercloud::Timestamp>
43         linkedNotebookLastSyncTimes() const = 0;
44 };
45
46 } // namespace quentier::synchronization

```

6.52 ISyncStateBuilder.h

```

1 /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/types/TypeAliases.h>
25
26 #include <QHash>
27
28 namespace quentier::synchronization {
29
30 class QUENTIER_EXPORT ISyncStateBuilder
31 {
32 public:

```

```

33     virtual ~ISyncStateBuilder() noexcept;
34
35     virtual ISyncStateBuilder & setUserDataUpdateCount(qint32 updateCount) = 0;
36
37     virtual ISyncStateBuilder & setUserDataLastSyncTime(
38         qevercloud::Timestamp lastSyncTime) = 0;
39
40     virtual ISyncStateBuilder & setLinkedNotebookUpdateCounts(
41         QHash<qevercloud::Guid, qint32> updateCounts) = 0;
42
43     virtual ISyncStateBuilder & setLinkedNotebookLastSyncTimes(
44         QHash<qevercloud::Guid, qevercloud::Timestamp> lastSyncTimes) = 0;
45
46     [[nodiscard]] virtual ISyncStatePtr build() = 0;
47 };
48
49 [[nodiscard]] QUENTIER_EXPORT ISyncStateBuilderPtr createSyncStateBuilder();
50
51 } // namespace quantier::synchronization

```

6.53 AuthenticationInfo.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/types/Fwd.h>
22 #include <quantier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quantier::synchronization {
27
28 [[nodiscard]] QJsonObject QUENTIER_EXPORT
29     serializeAuthenticationInfoToJson(const IAuthenticationInfo & info);
30
31 [[nodiscard]] IAuthenticationInfoPtr QUENTIER_EXPORT
32     deserializeAuthenticationInfoFromJson(const QJsonObject & json);
33
34 } // namespace quantier::synchronization

```

6.54 DownloadNotesStatus.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20

```

```

21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
28 [[nodiscard]] QJsonObject QUENTIER_EXPORT
29     serializeDownloadNotesStatusToJson(const IDownloadNotesStatus & status);
30
31 [[nodiscard]] IDownloadNotesStatusPtr QUENTIER_EXPORT
32     deserializeDownloadNotesStatusFromJson(const QJsonObject & json);
33
34 } // namespace quentier::synchronization

```

6.55 DownloadResourcesStatus.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
28 [[nodiscard]] QJsonObject QUENTIER_EXPORT
29     serializeDownloadResourcesStatusToJson(
30         const IDownloadResourcesStatus & status);
31
32 [[nodiscard]] IDownloadResourcesStatusPtr QUENTIER_EXPORT
33     deserializeDownloadResourcesStatusFromJson(const QJsonObject & json);
34
35 } // namespace quentier::synchronization

```

6.56 SendStatus.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>

```

```

25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeSendStatusToJson(const ISendStatus & sendStatus);
33
39 [[nodiscard]] ISendStatusPtr QUENTIER_EXPORT
40     deserializeSendStatusFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization

```

6.57 SyncChunksDataCounters.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT serializeSyncChunksDataCountersToJson(
32     const ISyncChunksDataCounters & counters);
33
39 [[nodiscard]] ISyncChunksDataCountersPtr QUENTIER_EXPORT
40     deserializeSyncChunksDataCountersFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization

```

6.58 SyncResult.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeSyncResultToJson(const ISyncResult & syncResult);

```

```

33
39 [[nodiscard]] ISyncResultPtr QUENTIER_EXPORT
40     deserializeSyncResultFromJson(const QJsonObject & json);
41
42 } // namespace quantier::synchronization

```

6.59 SyncState.h

```

1 /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/synchronization/types/Fwd.h>
22 #include <quantier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quantier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeSyncStateToJson(const ISyncState & syncState);
33
39 [[nodiscard]] ISyncStatePtr QUENTIER_EXPORT
40     deserializeSyncStateFromJson(const QJsonObject & json);
41
42 } // namespace quantier::synchronization

```

6.60 Future.h

```

1 /*
2  * Copyright 2021-2022 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/utility/Linkage.h>
22
23 #include <QAbstractEventDispatcher>
24 #include <QFuture>
25 #include <QFutureWatcher>
26 #include <QMutex>
27 #include <QMutexLocker>
28 #include <QObject>
29 #include <QPointer>
30
31 #include <quantier/threading/QtFutureContinuations.h>
32
33 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
34 #include <QPromise>

```

```

35 #include <exception>
36 #else
37 #include <quentier/threading/Qt5Promise.h>
38 #endif
39
40 #include <algorithm>
41 #include <cmath>
42 #include <memory>
43 #include <type_traits>
44 #include <utility>
45
46 namespace quentier::threading {
47
48 template <class T>
49 [[nodiscard]] std::enable_if_t<
50     std::negation_v<std::is_same<std::decay_t<T>, void>>,
51     QFuture<std::decay_t<T>>
52     makeReadyFuture(T t)
53 {
54     QPromise<std::decay_t<T>> promise;
55     QFuture<std::decay_t<T>> future = promise.future();
56
57     promise.start();
58     promise.setResult(std::move(t));
59     promise.finish();
60
61     return future;
62 }
63
64 [[nodiscard]] QFuture<void> QUINTIER_EXPORT makeReadyFuture();
65
66 template <class T, class E>
67 [[nodiscard]] std::enable_if_t<std::is_base_of_v<QException, E>, QFuture<T>>
68     makeExceptionalFuture(const E & e)
69 {
70     QPromise<std::decay_t<T>> promise;
71     QFuture<std::decay_t<T>> future = promise.future();
72
73     promise.start();
74     promise.setException(e);
75     promise.finish();
76
77     return future;
78 }
79
80 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
81 template <class T>
82 [[nodiscard]] QFuture<T> makeExceptionalFuture(std::exception_ptr e)
83 {
84     QPromise<std::decay_t<T>> promise;
85     QFuture<std::decay_t<T>> future = promise.future();
86
87     promise.start();
88     promise.setException(std::move(e));
89     promise.finish();
90
91     return future;
92 }
93 #endif // QT_VERSION
94
95 template <class T, class U>
96 void bindCancellation(const QFuture<T> & from, QFuture<U> to)
97 {
98     auto watcher = std::make_unique<QFutureWatcher<T>>();
99     auto * rawWatcher = watcher.get();
100
101     QObject::connect(
102         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
103         [rawWatcher, to]() mutable {
104             to.cancel();
105             rawWatcher->deleteLater();
106         });
107
108     QObject::connect(
109         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
110         [rawWatcher] { rawWatcher->deleteLater(); });
111
112     watcher->setFuture(from);
113     Q_UNUSED(watcher.release());
114 }
115
116 [[nodiscard]] QFuture<void> QUINTIER_EXPORT
117     whenAll(QList<QFuture<void>> futures);
118
119 template <class T>
120 [[nodiscard]] std::enable_if_t<
121     !std::is_void_v<std::decay_t<T>, QFuture<QList<std::decay_t<T>>>

```

```

153     whenAll(QList<QFuture<std::decay_t<T>>> futures)
154 {
155     if (Q_UNLIKELY(futures.isEmpty())) {
156         return makeReadyFuture<QList<std::decay_t<T>>>({});
157     }
158
159     auto promise = std::make_shared<QPromise<QList<std::decay_t<T>>>>();
160     auto future = promise->future();
161
162     for (auto & f: futures) {
163         threading::bindCancellation(future, f);
164     }
165
166     const auto totalItemCount = futures.size();
167     promise->setProgressRange(0, static_cast<int>(totalItemCount));
168     promise->setProgressValue(0);
169
170     promise->start();
171
172     auto resultIndexedList =
173         std::make_shared<QList<std::pair<int, std::decay_t<T>>>>();
174
175     auto processedItemsCount = std::make_shared<int>(0);
176     auto exceptionFlag = std::make_shared<bool>(false);
177     auto mutex = std::make_shared<QMutex>();
178
179     for (int i = 0; i < futures.size(); ++i) {
180         auto & f = futures[i];
181         auto thenFuture = then(
182             std::move(f),
183             [promise, processedItemsCount, totalItemCount, exceptionFlag, mutex,
184              resultIndexedList, i](std::decay_t<T> result) {
185                 if (promise->isCanceled()) {
186                     return;
187                 }
188
189                 int count = 0;
190                 {
191                     const QMutexLocker locker{mutex.get()};
192
193                     if (*exceptionFlag) {
194                         return;
195                     }
196
197                     ++(*processedItemsCount);
198                     count = *processedItemsCount;
199                     promise->setProgressValue(count);
200
201                     resultIndexedList->append(
202                         std::make_pair(i, std::move(result)));
203                 }
204
205                 if (count == totalItemCount) {
206                     std::sort(
207                         resultIndexedList->begin(), resultIndexedList->end(),
208                         [](const auto & lhs, const auto & rhs) {
209                             return lhs.first < rhs.first;
210                         });
211
212                     auto resultList =
213                         std::make_shared<QList<std::decay_t<T>>>();
214                     resultList->reserve(resultIndexedList->size());
215                     for (auto & [i, v]: *resultIndexedList) {
216                         resultList->append(std::move(v));
217                     }
218
219                     promise->addResult(*resultList);
220                     promise->finish();
221                 }
222             });
223
224     onFailed(
225         std::move(thenFuture),
226         [promise, mutex, exceptionFlag](const QException & e) {
227             if (promise->isCanceled()) {
228                 return;
229             }
230
231             {
232                 const QMutexLocker locker{mutex.get()};
233
234                 if (*exceptionFlag) {
235                     return;
236                 }
237
238                 *exceptionFlag = true;
239             }

```

```

240
241         promise->setException(e);
242         promise->finish();
243     });
244 }
245
246     return future;
247 }
248
249 template <class T, class U>
250 void mapFutureProgress(
251     const QFuture<T> & future, const std::shared_ptr<QPromise<U>> & promise)
252 {
253     const auto futureProgressMinimum = future.progressMinimum();
254     const auto futureProgressRange =
255         future.progressMaximum() - futureProgressMinimum;
256
257     Q_ASSERT(futureProgressRange >= 0);
258
259     const auto promiseFuture = promise->future();
260     const auto promiseProgressMinimum = promiseFuture.progressMinimum();
261     const auto promiseProgressMaximum = promiseFuture.progressMaximum();
262
263     const auto promiseProgressRange =
264         promiseProgressMaximum - promiseProgressMinimum;
265
266     Q_ASSERT(promiseProgressRange >= 0);
267
268     auto futureWatcher = std::make_unique<QFutureWatcher<T>>();
269
270     QObject::connect(
271         futureWatcher.get(), &QFutureWatcher<T>::progressValueChanged,
272         futureWatcher.get(),
273         [promise, futureProgressMinimum, futureProgressRange,
274         promiseProgressRange, promiseProgressMinimum,
275         promiseProgressMaximum](int progressValue) {
276             if (Q_UNLIKELY(futureProgressRange == 0)) {
277                 promise->setProgressValue(0);
278                 return;
279             }
280
281             const auto progressPart =
282                 static_cast<double>(progressValue - futureProgressMinimum) /
283                 static_cast<double>(futureProgressRange);
284
285             const auto mappedProgressValue = static_cast<int>(
286                 std::round(progressPart * promiseProgressRange));
287
288             promise->setProgressValue(std::clamp(
289                 promiseProgressMinimum + mappedProgressValue,
290                 promiseProgressMinimum, promiseProgressMaximum));
291         });
292
293     QObject::connect(
294         futureWatcher.get(), &QFutureWatcher<T>::finished, futureWatcher.get(),
295         [futureWatcherWeak = QPointer<QFutureWatcher<T>>(futureWatcher.get())] {
296             if (!futureWatcherWeak.isNull()) {
297                 futureWatcherWeak->deleteLater();
298             }
299         });
300
301     QObject::connect(
302         futureWatcher.get(), &QFutureWatcher<T>::canceled, futureWatcher.get(),
303         [futureWatcherWeak = QPointer<QFutureWatcher<T>>(futureWatcher.get())] {
304             if (!futureWatcherWeak.isNull()) {
305                 futureWatcherWeak->deleteLater();
306             }
307         });
308
309     futureWatcher->setFuture(future);
310     Q_UNUSED(futureWatcher.release());
311 }
312
313 } // namespace quantier::threading

```

6.61 Post.h

```

1 /*
2  * Copyright 2021-2024 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify

```

```

7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QAbstractEventDispatcher>
22 #include <QMetaObject>
23 #include <QObject>
24
25 #include <QThread>
26
27 #include <memory>
28 #include <utility>
29
30 namespace quentier::threading {
31
32 template <typename Function>
33 void postToObject(QObject * object, Function && function)
34 {
35     Q_ASSERT(object);
36
37     QMetaObject::invokeMethod(
38         object, std::forward<Function>(function), Qt::QueuedConnection);
39 }
40
41 template <typename Function>
42 void postToThread(QThread * pThread, Function && function)
43 {
44     Q_ASSERT(pThread);
45     Q_ASSERT(!pThread->isFinished());
46
47     QObject * pObject = QAbstractEventDispatcher::instance(pThread);
48     if (!pObject) {
49         // Thread's event loop has not been started yet. Create a dummy QObject,
50         // move it to the target thread, set things up so that it would be
51         // destroyed after the job is done and use postToObject.
52         auto pDummyObj = std::make_unique<QObject>();
53         pDummyObj->moveToThread(pThread);
54         postToObject(
55             pDummyObj.get(),
56             [pObj = pDummyObj.get(),
57              function = std::forward<Function>(function)]() mutable {
58                 pObj->deleteLater();
59                 function();
60             });
61         Q_UNUSED(pDummyObj.release()) // NOLINT
62         return;
63     }
64
65     if (pThread == QThread::currentThread()) {
66         // Already on the target thread, executing the function right away
67         function();
68         return;
69     }
70
71     QMetaObject::invokeMethod(pObject, std::forward<Function>(function));
72 }
73
74 } // namespace quentier::threading

```

6.62 Qt5Promise.h

```

1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```

13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QFutureInterface>
22
23 #include <type_traits>
24
25 // Partial backport of QPromise from Qt6 to Qt5
26 template <typename T>
27 class QPromise
28 {
29     static_assert(
30         std::is_copy_constructible_v<T> || std::is_move_constructible_v<T> ||
31         std::is_same_v<T, void>,
32         "Type with copy or move constructors or type void is required");
33
34 public:
35     QPromise() = default;
36
37     Q_DISABLE_COPY(QPromise)
38
39     QPromise(QPromise<T> && other) noexcept : d(other.d)
40     {
41         other.d = QFutureInterface<T>();
42     }
43
44     QPromise(QFutureInterface<T> & other) : d(other) {}
45
46     QPromise & operator=(QPromise<T> && other) noexcept
47     {
48         QPromise<T> tmp(std::move(other));
49         tmp.swap(*this);
50         return *this;
51     }
52
53     ~QPromise()
54     {
55         // If QFutureInterface has no state, there is nothing to be done
56         if (d.queryState(QFutureInterfaceBase::State::NoState)) {
57             return;
58         }
59
60         // Otherwise, if computation is not finished at this point, cancel
61         // potential waits
62         if (!d.queryState(QFutureInterfaceBase::State::Finished)) {
63             d.cancel();
64             finish(); // required to finalize the state
65         }
66     }
67
68     // Core QPromise APIs
69     QFuture<T> future() const
70     {
71         return d.future();
72     }
73
74     template <
75         typename U,
76         typename = std::enable_if_t<
77             std::is_same_v<U, T> || std::is_convertible_v<U, T>>
78     > void addResult(U && result, int index = -1)
79     {
80         d.reportResult(std::forward<U>(result), index);
81     }
82
83     void setException(const QException & e)
84     {
85         d.reportException(e);
86     }
87
88     void start()
89     {
90         d.reportStarted();
91     }
92
93     void finish()
94     {
95         d.reportFinished();
96     }
97
98     void suspendIfRequested()
99     {
100         d.suspendIfRequested();
101     }

```

```

100
101     bool isCanceled() const
102 {
103     return d.isCanceled();
104 }
105
106 // Progress methods
107 void setProgressRange(int minimum, int maximum)
108 {
109     d.setProgressRange(minimum, maximum);
110 }
111 void setProgressValue(int progressValue)
112 {
113     d.setProgressValue(progressValue);
114 }
115 void setProgressValueAndText(
116     int progressValue, const QString & progressText)
117 {
118     d.setProgressValueAndText(progressValue, progressText);
119 }
120
121 void swap(QPromise<T> & other) noexcept
122 {
123     qSwap(this->d, other.d);
124 }
125
126 private:
127     mutable QFutureInterface<T> d = QFutureInterface<T>();
128 };
129
130 template <typename T>
131 inline void swap(QPromise<T> & a, QPromise<T> & b) noexcept
132 {
133     a.swap(b);
134 }

```

6.63 QtFutureContinuations.h

```

1 /*
2  * Copyright 2021-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QtGlobal>
22
23 #include <QFutureWatcher>
24 #include <QRunnable>
25 #include <QThreadPool>
26 #include <quentier/exception/RuntimeException.h>
27 #include <quentier/threading/Post.h>
28 #include <quentier/threading/QtFutureHelpers.h>
29
30 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
31 #include <quentier/threading/Qt5Promise.h>
32 #endif // QT_VERSION
33
34 #include <quentier/threading/Runnable.h>
35
36 #include <boost/core/demangle.hpp>
37
38 #include <memory>
39 #include <type_traits>
40 #include <typeinfo>
41
42 namespace quentier::threading {
43
44 // NOTE: "native" implementation of continuations for Qt6 is currently disabled
45 // due to bugs in their implementation, in particular (but not limited to)

```

```

46 // https://bugreports.qt.io/browse/QTBUG-119579 and
47 // https://bugreports.qt.io/browse/QTBUG-117918. It's a shame but it is what it
48 // is.
49 /*
50 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
51
52 template <class T, class Function>
53 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
54 QFuture<T> && future, Function && function)
55 {
56 return future.then(std::forward<decltype(function)>(function));
57 }
58
59 template <class T, class Function>
60 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
61 QFuture<T> && future, QtFuture::Launch policy, Function && function)
62 {
63 return future.then(policy, std::forward<decltype(function)>(function));
64 }
65
66 template <class T, class Function>
67 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
68 QFuture<T> && future, QThreadPool * pool, Function && function)
69 {
70 return future.then(pool, std::forward<decltype(function)>(function));
71 }
72
73 template <class T, class Function>
74 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
75 QFuture<T> && future, QObject * context, Function && function)
76 {
77 return future.then(context, std::forward<decltype(function)>(function));
78 }
79
80 template <class T, class Function>
81 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
82 onFailed(QFuture<T> && future, Function && handler)
83 {
84 return future.onFailed(std::forward<decltype(handler)>(handler));
85 }
86
87 template <class T, class Function>
88 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
89 onFailed(QFuture<T> && future, QObject * context, Function && handler)
90 {
91 return future.onFailed(context, std::forward<decltype(handler)>(handler));
92 }
93
94 #else // QT_VERSION
95
96 // implementation for Qt5
97 */
98
99 namespace detail {
100
101 template <class T, class Function>
102 void processParentFuture(
103     std::shared_ptr<
104         QPromise<typename ResultTypeHelper<Function, T>::ResultType>
105         promise,
106         QFuture<T> && future, Function && function)
107 {
108     Q_ASSERT(promise);
109
110     using ResultType = typename ResultTypeHelper<Function, T>::ResultType;
111
112     promise->start();
113
114     // If future contains exception, just forward it to the promise and
115     // don't call the function at all
116     try {
117         future.waitForFinished();
118     }
119     catch (const QException & e) {
120         promise->setException(e);
121         promise->finish();
122         return;
123     }
124     // NOTE: there cannot be other exception types in this context in Qt5
125     // because exception store can only contain QExceptions
126
127     // Try to run the handler, in case of success forward the result to promise
128     // (unless it is void), catch possible exceptions and if caught put them
129     // to the promise
130     try {
131         if constexpr (std::is_void_v<ResultType>) {
132             if constexpr (std::is_void_v<T>) {

```

```

133         function();
134     }
135     else {
136         if (future.resultCount() == 0) {
137             promise->setException(RuntimeError{ErrorString{
138                 QString::fromUtf8(
139                     "Invalid future continuation: detected future "
140                     "without result for type %1"
141                     .arg(QString::fromStdString(std::string{
142                         boost::core::demangle(typeid(T).name())})));
143             promise->finish();
144             return;
145         }
146     }
147     function(future.result());
148 }
149 }
150 else {
151     if constexpr (std::is_void_v<T>) {
152         promise->addResult(function());
153     }
154     else {
155         promise->addResult(function(future.result()));
156     }
157 }
158 }
159 catch (const QException & e) {
160     promise->setException(e);
161 }
162 catch (const std::exception & e) {
163     ErrorString error{QT_TRANSLATE_NOOP(
164         "utility", "Unknown std::exception in then future handler");
165     error.details() = QString::fromStdString(std::string{e.what()});
166     promise->setException(RuntimeError{std::move(error)});
167 }
168 catch (...) {
169     ErrorString error{QT_TRANSLATE_NOOP(
170         "utility", "Unknown exception in then future handler");
171     promise->setException(RuntimeError{std::move(error)});
172 }
173 }
174 promise->finish();
175 }
176 }
177 } // namespace detail
178
179 template <class T, class Function>
180 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
181     QFuture<T> && future, Function && function)
182 {
183     using ResultType =
184         typename detail::ResultTypeHelper<Function, T>::ResultType;
185
186     auto promise = std::make_shared<QPromise<ResultType>>();
187     auto result = promise->future();
188
189     if (future.isFinished()) {
190         detail::processParentFuture(
191             std::move(promise), std::move(future),
192             std::forward<decltype(function)>(function));
193         return result;
194     }
195
196     auto watcher = std::make_unique<QFutureWatcher<T>>();
197     auto * rawWatcher = watcher.get();
198     QObject::connect(
199         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
200         [rawWatcher, function = std::forward<decltype(function)>(function),
201         promise = std::move(promise)]() mutable {
202             detail::processParentFuture(
203                 std::move(promise), rawWatcher->future(),
204                 std::forward<decltype(function)>(function));
205             rawWatcher->deleteLater();
206         });
207
208     QObject::connect(
209         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
210         [rawWatcher] { rawWatcher->deleteLater(); });
211
212     watcher->setFuture(std::move(future));
213     Q_UNUSED(watcher.release())
214
215     return result;
216 }
217
218 template <class T, class Function>
219 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(

```

```

220     QFuture<T> && future, QtFuture::Launch policy, Function && function)
221 {
222     if (policy == QtFuture::Launch::Sync) {
223         return then(
224             std::move(future), std::forward<decltype(function)>(function));
225     }
226
227     return then(
228         std::move(future), QThreadPool::globalInstance(),
229         std::forward<decltype(function)>(function));
230 }
231
232 template <class T, class Function>
233 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
234     QFuture<T> && future, QThreadPool * pool, Function && function)
235 {
236     using ResultType =
237         typename detail::ResultTypeHelper<Function, T>::ResultType;
238
239     auto promise = std::make_shared<QPromise<ResultType>>();
240     auto result = promise->future();
241
242     if (future.isFinished()) {
243         auto * runnable = createFunctionRunnable(
244             [future = std::move(future), promise = std::move(promise),
245              function = std::forward<decltype(function)>(function)] () mutable {
246                 detail::processParentFuture(
247                     std::move(promise), std::move(future),
248                     std::forward<decltype(function)>(function));
249             });
250         runnable->setAutoDelete(true);
251         pool->start(runnable);
252         return result;
253     }
254
255     auto watcher = std::make_unique<QFutureWatcher<T>>();
256     auto * rawWatcher = watcher.get();
257     QObject::connect(
258         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
259         [rawWatcher, function = std::forward<decltype(function)>(function),
260          promise = std::move(promise), pool] () mutable {
261             auto * runnable = createFunctionRunnable(
262                 [function = std::forward<decltype(function)>(function),
263                  promise = std::move(promise),
264                  future = rawWatcher->future()] () mutable {
265                     detail::processParentFuture(
266                         std::move(promise), std::move(future),
267                         std::forward<decltype(function)>(function));
268                 });
269             runnable->setAutoDelete(true);
270             pool->start(runnable);
271             rawWatcher->deleteLater();
272         });
273
274     QObject::connect(
275         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
276         [rawWatcher] { rawWatcher->deleteLater(); });
277
278     watcher->setFuture(std::move(future));
279     Q_UNUSED(watcher.release())
280
281     return result;
282 }
283
284 template <class T, class Function>
285 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
286     QFuture<T> && future, QObject * context, Function && function)
287 {
288     using ResultType =
289         typename detail::ResultTypeHelper<Function, T>::ResultType;
290
291     auto promise = std::make_shared<QPromise<ResultType>>();
292     auto result = promise->future();
293
294     if (future.isFinished()) {
295         postToObject(
296             context,
297             [future = std::move(future), promise = std::move(promise),
298              function = std::forward<decltype(function)>(function)] () mutable {
299                 detail::processParentFuture(
300                     std::move(promise), std::move(future),
301                     std::forward<decltype(function)>(function));
302             });
303         return result;
304     }
305
306     auto watcher = std::make_unique<QFutureWatcher<T>>();

```

```

307     auto * rawWatcher = watcher.get();
308
309     QObject::connect(
310         rawWatcher, &QFutureWatcher<T>::finished, context,
311         [context, rawWatcher,
312         function = std::forward<decltype(function)>(function),
313         promise = std::move(promise)]() mutable {
314         postToObject(
315             context,
316             [function = std::forward<decltype(function)>(function),
317             promise = std::move(promise),
318             future = rawWatcher->future()]() mutable {
319                 detail::processParentFuture(
320                     std::move(promise), std::move(future),
321                     std::forward<decltype(function)>(function));
322             });
323         rawWatcher->deleteLater();
324     });
325
326     QObject::connect(
327         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
328         [rawWatcher] { rawWatcher->deleteLater(); });
329
330     watcher->setFuture(std::move(future));
331     Q_UNUSED(watcher.release())
332
333     return result;
334 }
335
336 namespace detail {
337
338 template <class T, class Function>
339 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, void>
340 processPossibleFutureException(
341     std::shared_ptr<QPromise<T>> promise, QFuture<T> && future,
342     Function && handler)
343 {
344     Q_ASSERT(promise);
345
346     using ArgType = typename QtPrivate::ArgResolver<Function>::First;
347     using ResultType =
348         typename ResultTypeHelper<Function, std::decay_t<ArgType>::ResultType;
349     static_assert(std::is_convertible_v<ResultType, T>);
350
351     promise->start();
352
353     try {
354         try {
355             future.waitForFinished();
356         }
357         catch (const ArgType & e) {
358             try {
359                 if constexpr (std::is_void_v<ResultType>) {
360                     handler(e);
361                 }
362                 else {
363                     promise->addResult(handler(e));
364                 }
365             }
366             catch (const QException & e) {
367                 promise->setException(e);
368             }
369             catch (const std::exception & e) {
370                 QString error{QT_TRANSLATE_NOOP(
371                     "utility",
372                     "Unknown std:exception in onFailed future handler")};
373                 error.details() = QString::fromStdString(std::string{e.what()});
374                 promise->setException(RuntimeError{std::move(error)});
375             }
376             catch (...) {
377                 QString error{QT_TRANSLATE_NOOP(
378                     "utility", "Unknown exception in onFailed future handler")};
379                 promise->setException(RuntimeError{std::move(error)});
380             }
381         }
382     }
383     // Exception doesn't match with handler's argument type, propagate
384     // the exception to be handled later.
385     catch (const QException & e) {
386         promise->setException(e);
387     }
388     catch (const std::exception & e) {
389         QString error{QT_TRANSLATE_NOOP(
390             "utility",
391             "Unknown std:exception which did not match with onFailed "
392             "future handler")};
393         error.details() = QString::fromStdString(std::string{e.what()});

```

```

394     promise->setException(RuntimeError{std::move(error)});
395 }
396 catch (...) {
397     ErrorString error{QT_TRANSLATE_NOOP(
398         "utility",
399         "Unknown which did not match with onFailed "
400         "future handler")};
401     promise->setException(RuntimeError{std::move(error)});
402 }
403
404 promise->finish();
405 }
406
407 } // namespace detail
408
409 // WARNING! "Chaining" of onFailed calls would only work properly with Qt5 if
410 // all involved exceptions subclass QException. It is due to the way exception
411 // storage is implemented in Qt5. In Qt6 it was made to store std::exception_ptr
412 // so there's no requirement to use QExceptions in Qt6.
413
414 template <class T, class Function>
415 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
416     onFailed(QFuture<T> && future, Function && handler)
417 {
418     auto promise = std::make_shared<QPromise<T>>();
419     auto result = promise->future();
420
421     if (future.isFinished()) {
422         detail::processPossibleFutureException(
423             std::move(promise), std::move(future),
424             std::forward<decltype(handler)>(handler));
425         return result;
426     }
427
428     auto watcher = std::make_unique<QFutureWatcher<T>>();
429     auto * rawWatcher = watcher.get();
430     QObject::connect(
431         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
432         [rawWatcher, promise = std::move(promise),
433          handler = std::forward<decltype(handler)>(handler)]() mutable {
434             auto future = rawWatcher->future();
435             rawWatcher->deleteLater();
436             detail::processPossibleFutureException(
437                 std::move(promise), std::move(future),
438                 std::forward<decltype(handler)>(handler));
439         });
440
441     QObject::connect(
442         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
443         [rawWatcher] { rawWatcher->deleteLater(); });
444
445     watcher->setFuture(std::move(future));
446     Q_UNUSED(watcher.release())
447
448     return result;
449 }
450
451 template <class T, class Function>
452 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
453     onFailed(QFuture<T> && future, QObject * context, Function && handler)
454 {
455     auto promise = std::make_shared<QPromise<T>>();
456     auto result = promise->future();
457
458     if (future.isFinished()) {
459         postToObject(
460             context,
461             [promise = std::move(promise), future = std::move(future),
462              handler = std::forward<decltype(handler)>(handler)]() mutable {
463                 detail::processPossibleFutureException(
464                     std::move(promise), std::move(future),
465                     std::forward<decltype(handler)>(handler));
466             });
467         return result;
468     }
469
470     auto watcher = std::make_unique<QFutureWatcher<T>>();
471     auto * rawWatcher = watcher.get();
472     QObject::connect(
473         rawWatcher, &QFutureWatcher<T>::finished, context,
474         [context, rawWatcher, promise = std::move(promise),
475          handler = std::forward<decltype(handler)>(handler)]() mutable {
476             postToObject(
477                 context,
478                 [promise = std::move(promise), future = rawWatcher->future(),
479                  handler = std::forward<decltype(handler)>(handler)]() mutable {
480                     detail::processPossibleFutureException(

```

```

481         std::move(promise), std::move(future),
482         std::forward<decltype(handler)>(handler));
483     });
484     rawWatcher->deleteLater();
485 });
486
487 QObject::connect(
488     rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
489     [rawWatcher] { rawWatcher->deleteLater(); });
490
491 watcher->setFuture(std::move(future));
492 Q_UNUSED(watcher.release())
493
494 return result;
495 }
496
497 // #endif // QT_VERSION
498
499 // Convenience functions for both Qt versions
500
501 template <class T, class U, class Function>
502 void thenOrFailed(
503     QFuture<T> && future, std::shared_ptr<QPromise<U>> promise,
504     Function && function)
505 {
506     auto thenFuture =
507         then(std::move(future), std::forward<decltype(function)>(function));
508
509     onFailed(std::move(thenFuture), [promise](const QException & e) {
510         promise->setException(e);
511         promise->finish();
512     });
513 }
514
515 template <class T, class U, class Function>
516 void thenOrFailed(
517     QFuture<T> && future, QThread * thread,
518     std::shared_ptr<QPromise<U>> promise, Function && function)
519 {
520     auto thenFuture =
521         then(std::move(future), thread, std::forward<Function>(function));
522
523     onFailed(std::move(thenFuture), thread, [promise](const QException & e) {
524         promise->setException(e);
525         promise->finish();
526     });
527 }
528
529 template <class T, class U>
530 void thenOrFailed(QFuture<T> && future, std::shared_ptr<QPromise<U>> promise)
531 {
532     thenOrFailed(std::move(future), promise, [promise] { promise->finish(); });
533 }
534
535 template <class T, class U>
536 void thenOrFailed(
537     QFuture<T> && future, QThread * thread,
538     std::shared_ptr<QPromise<U>> promise)
539 {
540     thenOrFailed(
541         std::move(future), thread, promise, [promise] { promise->finish(); });
542 }
543
544 } // namespace quentier::threading

```

6.64 QtFutureHelpers.h

```

1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.

```

```

17 */
18
19 #pragma once
20
21 #include <QFuture>
22
23 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
24 #include "Qt5Promise.h"
25 #endif
26
27 #include <type_traits>
28
29 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
30 // Backports of some helpers for QFuture continuations from Qt6 to Qt5
31 namespace QtFuture {
32
33 // Inherit option from Qt6 is not supported in Qt5
34 enum class Launch
35 {
36     Sync,
37     Async,
38 };
39
40 } // namespace QtFuture
41
42 namespace QtPrivate {
43
44 template <typename...>
45 struct ArgsType;
46
47 template <typename Arg, typename... Args>
48 struct ArgsType<Arg, Args...>
49 {
50     using First = Arg;
51     using PromiseType = void;
52     using IsPromise = std::false_type;
53     static const bool HasExtraArgs = (sizeof...(Args) > 0);
54
55     template <class Class, class Callable>
56     static const bool CanInvokeWithArgs =
57         std::is_invocable_v<Callable, Class, Arg, Args...>;
58 };
59
60 template <typename Arg, typename... Args>
61 struct ArgsType<QPromise<Arg> &, Args...>
62 {
63     using First = QPromise<Arg> &;
64     using PromiseType = Arg;
65     using IsPromise = std::true_type;
66     static const bool HasExtraArgs = (sizeof...(Args) > 0);
67
68     template <class Class, class Callable>
69     static const bool CanInvokeWithArgs =
70         std::is_invocable_v<Callable, Class, QPromise<Arg> &, Args...>;
71 };
72
73 template <>
74 struct ArgsType<>
75 {
76     using First = void;
77     using PromiseType = void;
78     using IsPromise = std::false_type;
79     static const bool HasExtraArgs = false;
80     using AllArgs = void;
81
82     template <class Class, class Callable>
83     static const bool CanInvokeWithArgs = std::is_invocable_v<Callable, Class>;
84 };
85
86 template <typename F>
87 struct ArgResolver : ArgResolver<decltype(&std::decay_t<F>::operator())>
88 {};
89
90 template <typename F>
91 struct ArgResolver<std::reference_wrapper<F> > :
92     ArgResolver<decltype(&std::decay_t<F>::operator())>
93 {};
94
95 template <typename R, typename... Args>
96 struct ArgResolver<R(Args...) > : public ArgsType<Args...>
97 {};
98
99 template <typename R, typename... Args>
100 struct ArgResolver<R (*) (Args...) > : public ArgsType<Args...>
101 {};
102
103 template <typename R, typename... Args>

```

```

104 struct ArgResolver<R (&)(Args...)> : public ArgsType<Args...>
105 {};
106
107 template <typename R, typename... Args>
108 struct ArgResolver<R (*const)(Args...)> : public ArgsType<Args...>
109 {};
110
111 template <typename R, typename... Args>
112 struct ArgResolver<R (&)(Args...)> : public ArgsType<Args...>
113 {};
114
115 template <typename Class, typename R, typename... Args>
116 struct ArgResolver<R (Class::*)(Args...)> : public ArgsType<Args...>
117 {};
118
119 template <typename Class, typename R, typename... Args>
120 struct ArgResolver<R (Class::*)(Args...) noexcept> : public ArgsType<Args...>
121 {};
122
123 template <typename Class, typename R, typename... Args>
124 struct ArgResolver<R (Class::*)(Args...) const> : public ArgsType<Args...>
125 {};
126
127 template <typename Class, typename R, typename... Args>
128 struct ArgResolver<R (Class::*)(Args...) const noexcept> :
129     public ArgsType<Args...>
130 {};
131
132 template <typename Class, typename R, typename... Args>
133 struct ArgResolver<R (Class::*const)(Args...) const> : public ArgsType<Args...>
134 {};
135
136 template <typename Class, typename R, typename... Args>
137 struct ArgResolver<R (Class::*const)(Args...) const noexcept> :
138     public ArgsType<Args...>
139 {};
140
141 } // namespace QtPrivate
142 #endif // QT_VERSION
143
144 namespace quentier::threading::detail {
145
146 template <typename F, typename Arg, typename Enable = void>
147 struct ResultTypeHelper
148 {};
149
150 // The callable takes an argument of type Arg
151 template <typename F, typename Arg>
152 struct ResultTypeHelper<
153     F, Arg,
154     typename std::enable_if_t<
155         !std::is_invocable_v<std::decay_t<F>, QFuture<Arg>>>
156 {
157     using ResultType = std::invoke_result_t<std::decay_t<F>, std::decay_t<Arg>;
158 };
159
160 // The callable takes an argument of type QFuture<Arg>
161 template <class F, class Arg>
162 struct ResultTypeHelper<
163     F, Arg,
164     typename std::enable_if_t<
165         std::is_invocable_v<std::decay_t<F>, QFuture<Arg>>>
166 {
167     using ResultType = std::invoke_result_t<std::decay_t<F>, QFuture<Arg>;
168 };
169
170 // The callable takes an argument of type QFuture<void>
171 template <class F>
172 struct ResultTypeHelper<
173     F, void,
174     typename std::enable_if_t<
175         std::is_invocable_v<std::decay_t<F>, QFuture<void>>>
176 {
177     using ResultType = std::invoke_result_t<std::decay_t<F>, QFuture<void>;
178 };
179
180 // The callable doesn't take argument
181 template <class F>
182 struct ResultTypeHelper<
183     F, void,
184     typename std::enable_if_t<
185         !std::is_invocable_v<std::decay_t<F>, QFuture<void>>>
186 {
187     using ResultType = std::invoke_result_t<std::decay_t<F>;
188 };
189
190 } // namespace quentier::threading::detail

```

6.65 Runnable.h

```

1  /*
2  * Copyright 2021 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <functional>
24
25 class QRunnable;
26
27 namespace quentier::threading {
28
29 [[nodiscard]] auto QUENTIER_EXPORT
30     createFunctionRunnable(std::function<void()> function) -> QRunnable *;
31
32 } // namespace quentier::threading

```

6.66 TrackedTask.h

```

1  /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <functional>
22 #include <type_traits>
23 #include <utility>
24
25 namespace quentier::threading {
26
27 namespace detail {
28
29 template <typename LockableObject, typename Function, typename... Arguments>
30 constexpr std::enable_if_t<std::is_invocable_v<Function, Arguments...> &&
31     LockableObject & lockableObject, Function & function,
32     Arguments &&... arguments>
33 {
34     const auto lockedObject = lockableObject.lock();
35     if (lockedObject) {
36         std::invoke(function, std::forward<Arguments>(arguments)...);
37     }
38 }
39
40 template <typename LockableObject, typename Function, typename... Arguments>
41 constexpr std::enable_if_t<
42     !std::is_invocable_v<Function, Arguments...> &&
43     std::is_member_function_pointer_v<Function>
44     && LockableObject & lockableObject, Function & function,
45     Arguments &&... arguments>

```

```

47 {
48     const auto lockedObject = lockableObject.lock();
49     if (lockedObject) {
50         std::invoke(
51             function, *lockedObject, std::forward<Arguments>(arguments)...);
52     }
53 }
54
55 } // namespace detail
56
57 template <typename LockableObject, typename Function>
58 class TrackedTask
59 {
60 public:
61     template <typename SomeLockableObject, typename SomeFunction>
62     constexpr TrackedTask(
63         SomeLockableObject && someLockableObject, SomeFunction && function) :
64         m_lockableObject{std::forward<SomeLockableObject>(someLockableObject)},
65         m_function{std::forward<SomeFunction>(function)}
66     {}
67
68     template <
69         typename... Arguments,
70         typename = std::enable_if_t<
71             std::is_invocable_v<Function, Arguments...> ||
72             std::is_member_function_pointer_v<Function>>
73     > constexpr void operator()(Arguments &&... arguments)
74     {
75         detail::invoke(
76             m_lockableObject, m_function,
77             std::forward<Arguments>(arguments)...);
78     }
79
80     template <
81         typename... Arguments,
82         typename = std::enable_if_t<
83             std::is_invocable_v<Function, Arguments...> ||
84             std::is_member_function_pointer_v<Function>>
85     > constexpr void operator()(Arguments &&... arguments) const
86     {
87         detail::invoke(
88             m_lockableObject, m_function,
89             std::forward<Arguments>(arguments)...);
90     }
91
92 private:
93     LockableObject m_lockableObject;
94     Function m_function;
95 };
96
97 template <typename LockableObject, typename Function>
98 TrackedTask(LockableObject, Function) -> TrackedTask<LockableObject, Function>;
99
100 } // namespace quentier::threading

```

6.67 Account.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Printable.h>
22
23 #include <Qevercloud/QEverCloud.h>
24
25 #include <QSharedDataPointer>
26 #include <QString>

```

```

27
28 namespace quentier {
29
30 class AccountData;
31
32 class QUINTIER_EXPORT Account : public utility::Printable
33 {
34 public:
35     enum class Type
36     {
37         Local,
38         Evernote
39     };
40
41     friend QUINTIER_EXPORT QTextStream & operator<<(
42         QTextStream & strm, Type type);
43
44     friend QUINTIER_EXPORT QDebug & operator<<(QDebug & dbg, Type type);
45
46     enum class EvernoteAccountType
47     {
48         Free,
49         Plus,
50         Premium,
51         Business
52     };
53
54     friend QUINTIER_EXPORT QTextStream & operator<<(
55         QTextStream & strm, EvernoteAccountType type);
56
57     friend QUINTIER_EXPORT QDebug & operator<<(
58         QDebug & dbg, EvernoteAccountType type);
59 public:
60     explicit Account();
61
62     explicit Account(
63         QString name, Type type, qevercloud::UserID userId = -1,
64         EvernoteAccountType evernoteAccountType = EvernoteAccountType::Free,
65         QString evernoteHost = {}, QString shardId = {});
66
67     Account(const Account & other);
68     Account(Account && other) noexcept;
69
70     Account & operator=(const Account & other);
71     Account & operator=(Account && other) noexcept;
72
73     ~Account() noexcept override;
74
75     [[nodiscard]] bool operator==(const Account & other) const noexcept;
76     [[nodiscard]] bool operator!=(const Account & other) const noexcept;
77
78     [[nodiscard]] bool isEmpty() const;
79
80     [[nodiscard]] QString name() const;
81
82     void setName(QString name);
83
84     [[nodiscard]] QString displayName() const;
85
86     void setDisplayName(QString displayName);
87
88     [[nodiscard]] Type type() const;
89
90     [[nodiscard]] qevercloud::UserID id() const;
91
92     [[nodiscard]] EvernoteAccountType evernoteAccountType() const;
93
94     [[nodiscard]] QString evernoteHost() const;
95
96     [[nodiscard]] QString shardId() const;
97
98     void setEvernoteAccountType(EvernoteAccountType evernoteAccountType);
99     void setEvernoteHost(QString evernoteHost);
100     void setShardId(QString shardId);
101
102     [[nodiscard]] quint32 mailLimitDaily() const;
103     [[nodiscard]] quint64 noteSizeMax() const;
104     [[nodiscard]] quint64 resourceSizeMax() const;
105     [[nodiscard]] quint32 linkedNotebookMax() const;
106     [[nodiscard]] quint32 noteCountMax() const;
107     [[nodiscard]] quint32 notebookCountMax() const;
108     [[nodiscard]] quint32 tagCountMax() const;
109     [[nodiscard]] quint32 noteTagCountMax() const;
110     [[nodiscard]] quint32 savedSearchCountMax() const;
111     [[nodiscard]] quint32 noteResourceCountMax() const;
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158

```

```

159     void setEvernoteAccountLimits(const qevercloud::AccountLimits & limits);
160
161     // utility::Printable
162     QTextStream & print(QTextStream & strm) const override;
163
164 private:
165     QSharedDataPointer<AccountData> d;
166 };
167
168 } // namespace quentier

```

6.68 ErrorString.h

```

1 /*
2  * Copyright 2017-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Printable.h>
22
23 #include <QSharedDataPointer>
24
25 namespace quentier {
26
27 class ErrorStringData;
28
29 class QUINTIER_EXPORT ErrorString : public utility::Printable
30 {
31 public:
32     explicit ErrorString(const char * error = nullptr);
33     explicit ErrorString(const QString & error);
34
35     ErrorString(const ErrorString & other);
36     ErrorString(ErrorString && other) noexcept;
37
38     ErrorString & operator=(const ErrorString & other);
39     ErrorString & operator=(ErrorString && other) noexcept;
40
41     ~ErrorString() override;
42
43     [[nodiscard]] const QString & base() const noexcept;
44     [[nodiscard]] QString & base();
45
46     [[nodiscard]] const QStringList & additionalBases() const noexcept;
47     [[nodiscard]] QStringList & additionalBases();
48
49     [[nodiscard]] const QString & details() const noexcept;
50     [[nodiscard]] QString & details();
51
52     void setBase(QString error);
53     void setBase(const char * error);
54
55     void appendBase(const QString & error);
56     void appendBase(const QStringList & errors);
57     void appendBase(const char * error);
58
59     void setDetails(const QString & error);
60     void setDetails(const char * error);
61
62     [[nodiscard]] bool isEmpty() const;
63     void clear();
64
65     [[nodiscard]] QString localizedString() const;
66     [[nodiscard]] QString nonLocalizedString() const;
67
68     QTextStream & print(QTextStream & strm) const override;
69
70 private:

```

```

84     QSharedDataPointer<ErrorStringData> d;
85 };
86
87 [[nodiscard]] QUENTIER_EXPORT bool operator==(
88     const ErrorString & lhs, const ErrorString & rhs) noexcept;
89
90 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
91     const ErrorString & lhs, const ErrorString & rhs) noexcept;
92
93 } // namespace quentier

```

6.69 enml/conversion_rules/Fwd.h

```

1  /*
2  * Copyright 2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::enml::conversion_rules {
24
25 class ISkipRule;
26 using ISkipRulePtr = std::shared_ptr<ISkipRule>;
27
28 class ISkipRuleBuilder;
29 using ISkipRuleBuilderPtr = std::shared_ptr<ISkipRuleBuilder>;
30
31 } // namespace quentier::enml::conversion_rules

```

6.70 enml/Fwd.h

```

1  /*
2  * Copyright 2016-2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::enml {
24
25 class IConverter;
26 using IConverterPtr = std::shared_ptr<IConverter>;
27
28 class IDecryptedTextCache;
29 using IDecryptedTextCachePtr = std::shared_ptr<IDecryptedTextCache>;
30
31 class IENMLTagsConverter;
32 using IENMLTagsConverterPtr = std::shared_ptr<IENMLTagsConverter>;

```

```

33
34 struct IHtmlData;
35 using IHtmlDataPtr = std::shared_ptr<IHtmlData>;
36
37 } // namespace quentier::enml

```

6.71 local_storage/Fwd.h

```

1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::local_storage {
24
25 class ILocalStorage;
26 using ILocalStoragePtr = std::shared_ptr<ILocalStorage>;
27
28 class ILocalStorageNotifier;
29
30 class IPatch;
31 using IPatchPtr = std::shared_ptr<IPatch>;
32
33 class NoteSearchQuery;
34
35 } // namespace quentier::local_storage

```

6.72 synchronization/Fwd.h

```

1 /*
2 * Copyright 2020-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::synchronization {
24
25 class IAuthenticator;
26 using IAuthenticatorPtr = std::shared_ptr<IAuthenticator>;
27
28 class INoteStoreFactory;
29 using INoteStoreFactoryPtr = std::shared_ptr<INoteStoreFactory>;
30
31 class ISyncConflictResolver;
32 using ISyncConflictResolverPtr = std::shared_ptr<ISyncConflictResolver>;
33

```

```

34 class ISynchronizer;
35 using ISynchronizerPtr = std::shared_ptr<ISynchronizer>;
36
37 class ISyncEventsNotifier;
38
39 class ISyncOptions;
40 using ISyncOptionsPtr = std::shared_ptr<ISyncOptions>;
41
42 class ISyncStateStorage;
43 using ISyncStateStoragePtr = std::shared_ptr<ISyncStateStorage>;
44
45 class IUserStoreFactory;
46 using IUserStoreFactoryPtr = std::shared_ptr<IUserStoreFactory>;
47
48 } // namespace quantier::synchronization

```

6.73 synchronization/types/Fwd.h

```

1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quantier::synchronization {
24
25 class IAuthenticationInfo;
26 using IAuthenticationInfoPtr = std::shared_ptr<IAuthenticationInfo>;
27
28 class IAuthenticationInfoBuilder;
29 using IAuthenticationInfoBuilderPtr =
30     std::shared_ptr<IAuthenticationInfoBuilder>;
31
32 class IDownloadNotesStatus;
33 using IDownloadNotesStatusPtr = std::shared_ptr<IDownloadNotesStatus>;
34
35 class IDownloadResourcesStatus;
36 using IDownloadResourcesStatusPtr = std::shared_ptr<IDownloadResourcesStatus>;
37
38 class ISendStatus;
39 using ISendStatusPtr = std::shared_ptr<ISendStatus>;
40
41 struct ISyncChunksDataCounters;
42 using ISyncChunksDataCountersPtr = std::shared_ptr<ISyncChunksDataCounters>;
43
44 class ISyncOptions;
45 using ISyncOptionsPtr = std::shared_ptr<ISyncOptions>;
46
47 class ISyncOptionsBuilder;
48 using ISyncOptionsBuilderPtr = std::shared_ptr<ISyncOptionsBuilder>;
49
50 class ISyncResult;
51 using ISyncResultPtr = std::shared_ptr<ISyncResult>;
52
53 class ISyncState;
54 using ISyncStatePtr = std::shared_ptr<ISyncState>;
55
56 class ISyncStateBuilder;
57 using ISyncStateBuilderPtr = std::shared_ptr<ISyncStateBuilder>;
58
59 } // namespace quantier::synchronization

```

6.74 threading/Fwd.h

```

1  /*

```

```
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 class QMutex;
24 class QThread;
25 class QThreadPool;
26
27 namespace quentier::threading {
28
29 using QMutexPtr = std::shared_ptr<QMutex>;
30 using QThreadPtr = std::shared_ptr<QThread>;
31 using QThreadPoolPtr = std::shared_ptr<QThreadPool>;
32
33 } // namespace quentier::threading
```

6.75 types/Fwd.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 namespace quentier {
22
23 class Account;
24 class ErrorString;
25 class ResourceRecognitionIndexItem;
26 class ResourceRecognitionIndices;
27
28 } // namespace quentier
```

6.76 utility/cancelers/Fwd.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
```

```

14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::utility::cancelers {
24
25     class AnyOfCanceler;
26     using AnyOfCancelerPtr = std::shared_ptr<AnyOfCanceler>;
27
28     class ICanceler;
29     using ICancelerPtr = std::shared_ptr<ICanceler>;
30
31     class ManualCanceler;
32     using ManualCancelerPtr = std::shared_ptr<ManualCanceler>;
33
34 } // namespace quentier::utility::cancelers

```

6.77 utility/Fwd.h

```

1 /*
2 * Copyright 2020-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::utility {
24
25     class ApplicationSettings;
26     class FileIOProcessorAsync;
27     class QuentierApplication;
28
29     struct IEncryptor;
30     using IEncryptorPtr = std::shared_ptr<IEncryptor>;
31
32     class IKeychainService;
33     using IKeychainServicePtr = std::shared_ptr<IKeychainService>;
34
35 } // namespace quentier::utility

```

6.78 NoteUtils.h

```

1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */

```

```

18
19 #pragma once
20
21 #include <quentier/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/types/Fwd.h>
25
26 #include <QStringList>
27
28 #include <utility>
29
30 namespace quentier {
31
32 [[nodiscard]] QUENTIER_EXPORT bool isInkNote(const qevercloud::Note & note);
33
34 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsCheckedToDo(
35     const QString & noteContent);
36
37 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsUncheckedToDo(
38     const QString & noteContent);
39
40 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsToDo(
41     const QString & noteContent);
42
43 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsEncryptedFragments(
44     const QString & noteContent);
45
46 [[nodiscard]] QUENTIER_EXPORT QString noteContentToPlainText(
47     const QString & noteContent, ErrorString * errorDescription = nullptr);
48
49 [[nodiscard]] QUENTIER_EXPORT QStringList noteContentToListOfWords(
50     const QString & noteContent, ErrorString * errorDescription = nullptr);
51
52 [[nodiscard]] QUENTIER_EXPORT std::pair<QString, QStringList>
53     noteContentToPlainTextAndListOfWords(
54     const QString & noteContent, ErrorString * errorDescription = nullptr);
55
56 } // namespace quentier

```

6.79 RegisterMetatypes.h

```

1 /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 namespace quentier {
24
25 QUENTIER_EXPORT void registerMetatypes();
26
27 } // namespace quentier

```

6.80 ResourceRecognitionIndexItem.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by

```

```

8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QByteArray>
25 #include <QList>
26 #include <QSharedDataPointer>
27
28 #include <memory>
29
30 namespace quentier {
31
32 class ResourceRecognitionIndexItemData;
33
34 class QUENTIER_EXPORT ResourceRecognitionIndexItem : public utility::Printable
35 {
36 public:
37     explicit ResourceRecognitionIndexItem();
38
39     ResourceRecognitionIndexItem(const ResourceRecognitionIndexItem & other);
40
41     ResourceRecognitionIndexItem(
42         ResourceRecognitionIndexItem && other) noexcept;
43
44     ResourceRecognitionIndexItem & operator=(
45         const ResourceRecognitionIndexItem & other);
46
47     ResourceRecognitionIndexItem & operator=(
48         ResourceRecognitionIndexItem && other) noexcept;
49
50     ~ResourceRecognitionIndexItem() override;
51
52     [[nodiscard]] bool isValid() const;
53
54     [[nodiscard]] int x() const;
55     void setX(int x);
56
57     [[nodiscard]] int y() const;
58     void setY(int y);
59
60     [[nodiscard]] int h() const;
61     void setH(int h);
62
63     [[nodiscard]] int w() const;
64     void setW(int w);
65
66     [[nodiscard]] int offset() const;
67     void setOffset(int offset);
68
69     [[nodiscard]] int duration() const;
70     void setDuration(int duration);
71
72     [[nodiscard]] QList<int> strokes() const;
73     void setStrokes(QList<int> strokes);
74
75     struct QUENTIER_EXPORT ITextItem
76     {
77         virtual ~ITextItem();
78
79         [[nodiscard]] virtual QString text() const = 0;
80         [[nodiscard]] virtual int weight() const = 0;
81     };
82
83     using ITextItemPtr = std::shared_ptr<ITextItem>;
84
85     [[nodiscard]] QList<ITextItemPtr> textItems() const;
86     void setTextItems(QList<ITextItemPtr> textItems);
87
88     struct QUENTIER_EXPORT IObjectItem
89     {
90         virtual ~IObjectItem();
91
92         [[nodiscard]] virtual QString objectType() const = 0;
93         [[nodiscard]] virtual int weight() const = 0;
94     };

```

```

95
96     using IObjectItemPtr = std::shared_ptr<IObjectItem>;
97
98     [[nodiscard]] QList<IObjectItemPtr> objectItems() const;
99     void setObjectItems(QList<IObjectItemPtr> objectItems);
100
101     struct QUENTIER_EXPORT IShapeItem
102     {
103         virtual ~IShapeItem();
104
105         [[nodiscard]] virtual QString shape() const = 0;
106         [[nodiscard]] virtual int weight() const = 0;
107     };
108
109     using IShapeItemPtr = std::shared_ptr<IShapeItem>;
110
111     [[nodiscard]] QList<IShapeItemPtr> shapeItems() const;
112     void setShapeItems(QList<IShapeItemPtr> shapeItems);
113
114     struct QUENTIER_EXPORT IBarcodeItem
115     {
116         virtual ~IBarcodeItem();
117
118         [[nodiscard]] virtual QString barcode() const = 0;
119         [[nodiscard]] virtual int weight() const = 0;
120     };
121
122     using IBarcodeItemPtr = std::shared_ptr<IBarcodeItem>;
123
124     [[nodiscard]] QList<IBarcodeItemPtr> barcodeItems() const;
125     void setBarcodeItems(QList<IBarcodeItemPtr> barcodeItems);
126
127     // utility::Printable
128     QTextStream & print(QTextStream & strm) const override;
129
130 private:
131     QSharedDataPointer<ResourceRecognitionIndexItemData> d;
132 };
133
134 } // namespace quentier

```

6.81 ResourceRecognitionIndices.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ResourceRecognitionIndexItem.h>
22
23 #include <QByteArray>
24 #include <QSharedDataPointer>
25 #include <QVector>
26
27 namespace quentier {
28
29 class ResourceRecognitionIndicesData;
30
31 class QUENTIER_EXPORT ResourceRecognitionIndices : public utility::Printable
32 {
33 public:
34     explicit ResourceRecognitionIndices();
35
36     explicit ResourceRecognitionIndices(
37         const QByteArray & rawRecognitionIndicesData);
38
39     ResourceRecognitionIndices(const ResourceRecognitionIndices & other);
40     ResourceRecognitionIndices(ResourceRecognitionIndices && other) noexcept;

```

```

41
42     ResourceRecognitionIndices & operator=(
43         const ResourceRecognitionIndices & other);
44
45     ResourceRecognitionIndices & operator=(
46         ResourceRecognitionIndices && other) noexcept;
47
48     ~ResourceRecognitionIndices() override;
49
50     [[nodiscard]] bool isNull() const;
51     [[nodiscard]] bool isValid() const;
52
53     [[nodiscard]] QString objectId() const;
54     [[nodiscard]] QString objectType() const;
55     [[nodiscard]] QString recoType() const;
56     [[nodiscard]] QString engineVersion() const;
57     [[nodiscard]] QString docType() const;
58     [[nodiscard]] QString lang() const;
59
60     [[nodiscard]] int objectHeight() const;
61     [[nodiscard]] int objectWidth() const;
62
63     [[nodiscard]] QVector<ResourceRecognitionIndexItem> items() const;
64
65     bool setData(const QByteArray & rawRecognitionIndicesData);
66
67     // utility::Printable
68     QTextStream & print(QTextStream & strm) const override;
69
70 private:
71     QSharedDataPointer<ResourceRecognitionIndicesData> d;
72 };
73
74 } // namespace quentier

```

6.82 ResourceUtils.h

```

1 /*
2  * Copyright 2020-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/types/Fwd.h>
24
25 #include <QString>
26
27 namespace quentier {
28
29 [[nodiscard]] QUENTIER_EXPORT QString
30     resourceDisplayName(const qevercloud::Resource & resource);
31
32 [[nodiscard]] QUENTIER_EXPORT QString
33     preferredFileSuffix(const qevercloud::Resource & resource);
34
35 } // namespace quentier

```

6.83 Result.h

```

1 /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier

```

```

5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/RuntimeError.h>
22 #include <quentier/types/ErrorMessage.h>
23
24 #include <type_traits>
25 #include <variant>
26
27 namespace quentier {
28
29 template <
30     class ValueType, class ErrorType,
31     typename =
32         typename std::enable_if_t<!std::is_void_v<std::decay_t<ErrorType>>>
33 class Result
34 {
35 private:
36     template <typename T>
37     struct ValueWrapper
38     {
39         T value;
40     };
41
42     using ValueWrapperInnerType = std::conditional_t<
43         std::is_void_v<std::decay_t<ValueType>, std::nullptr_t,
44         std::decay_t<ValueType>>;
45
46 public:
47     template <
48         typename T1 = ValueType,
49         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>>> * =
50             nullptr>
51     explicit Result(T1 t) :
52         m_valueOrError{ValueWrapper<std::decay_t<ValueType>>{std::move(t)}}
53     {}
54
55     template <
56         typename T1 = ValueType,
57         typename std::enable_if_t<std::is_void_v<std::decay_t<T1>>> * = nullptr>
58     explicit Result() : m_valueOrError{ValueWrapper<std::nullptr_t>{}}
59     {}
60
61     explicit Result(ErrorType error) : m_valueOrError{std::move(error)} {}
62
63     Result(const Result<ValueType, ErrorType> & other) :
64         m_valueOrError{other.m_valueOrError}
65     {}
66
67     Result(Result<ValueType, ErrorType> && other) :
68         m_valueOrError{std::move(other.m_valueOrError)}
69     {}
70
71     Result & operator=(const Result<ValueType, ErrorType> & other)
72     {
73         if (this != &other) {
74             m_valueOrError = other.m_valueOrError;
75         }
76
77         return *this;
78     }
79
80     Result & operator=(Result<ValueType, ErrorType> && other)
81     {
82         if (this != &other) {
83             m_valueOrError = std::move(other.m_valueOrError);
84         }
85
86         return *this;
87     }
88
89     [[nodiscard]] bool isValid() const noexcept
90     {
91         return std::holds_alternative<ValueWrapper<ValueWrapperInnerType>>(

```

```

95         m_valueOrError);
96     }
97
98     operator bool() const noexcept
99     {
100         return isValid();
101     }
102
103     template <
104         typename T1 = ValueType,
105         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
106             nullptr>
107     [[nodiscard]] T1 & get()
108     {
109         // NOTE: std::get also performs the check of what is stored inside the
110         // variant but it throws std::bad_variant_access which doesn't implement
111         // QException so this exception is not representable inside QFuture
112         // in Qt5. Due to this for Qt5 also performing another check and using
113         // another exception type
114 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
115         if (Q_UNLIKELY(!isValid())) {
116             throw RuntimeError{
117                 ErrorString{"Detected attempt to get value from empty Result"}};
118         }
119 #endif
120
121         return std::get<ValueWrapper<std::decay_t<ValueType>>>(m_valueOrError)
122             .value;
123     }
124
125     template <
126         typename T1 = ValueType,
127         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
128             nullptr>
129     [[nodiscard]] const T1 & get()const
130     {
131         // NOTE: std::get also performs the check of what is stored inside the
132         // variant but it throws std::bad_variant_access which doesn't implement
133         // QException so this exception is not representable inside QFuture
134         // in Qt5. Due to this for Qt5 also performing another check and using
135         // another exception type
136 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
137         if (Q_UNLIKELY(!isValid())) {
138             throw RuntimeError{
139                 ErrorString{"Detected attempt to get value from empty Result"}};
140         }
141 #endif
142
143         return std::get<ValueWrapper<std::decay_t<ValueType>>>(m_valueOrError)
144             .value;
145     }
146
147     template <
148         typename T1 = ValueType,
149         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
150             nullptr>
151     [[nodiscard]] T1 & operator*()
152     {
153         return get();
154     }
155
156     template <
157         typename T1 = ValueType,
158         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
159             nullptr>
160     [[nodiscard]] const T1 & operator*()const
161     {
162         return get();
163     }
164
165     [[nodiscard]] const ErrorType & error()const
166     {
167         // NOTE: std::get also performs the check of what is stored inside the
168         // variant but it throws std::bad_variant_access which doesn't implement
169         // QException so this exception is not representable inside QFuture
170         // in Qt5. Due to this for Qt5 also performing another check and using
171         // another exception type
172 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
173         if (Q_UNLIKELY(isValid())) {
174             throw RuntimeError{ErrorString{
175                 "Detected attempt to get error from non-empty Result"}};
176         }
177 #endif
178
179         return std::get<ErrorType>(m_valueOrError);
180     }
181

```

```

182     [[nodiscard]] ErrorType & error()
183     {
184         // NOTE: std::get also performs the check of what is stored inside the
185         // variant but it throws std::bad_variant_access which doesn't implement
186         // QException so this exception is not representable inside QFuture
187         // in Qt5. Due to this for Qt5 also performing another check and using
188         // another exception type
189 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
190         if (Q_UNLIKELY(isValid())) {
191             throw RuntimeError{ErrorString{
192                 "Detected attempt to get error from non-empty Result"}};
193         }
194 #endif
195         return std::get<ErrorType>(m_valueOrError);
196     }
197 }
198
199 private:
200     std::variant<ValueWrapper<ValueWrapperInnerType>, ErrorType> m_valueOrError;
201 };
202
203 } // namespace quentier

```

6.84 Validation.h

```

1 /*
2  * Copyright 2020-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 class QString;
25
26 namespace quentier {
27
28 [[nodiscard]] QUENTIER_EXPORT bool validateNoteTitle(
29     const QString & noteTitle,
30     ErrorString * errorDescription = nullptr) noexcept;
31
32 [[nodiscard]] QUENTIER_EXPORT bool validateNotebookName(
33     const QString & notebookName,
34     ErrorString * errorDescription = nullptr) noexcept;
35
36 [[nodiscard]] QUENTIER_EXPORT bool validateSavedSearchName(
37     const QString & savedSearchName,
38     ErrorString * errorDescription = nullptr) noexcept;
39
40 [[nodiscard]] QUENTIER_EXPORT bool validateTagName(
41     const QString & tagName, ErrorString * errorDescription = nullptr) noexcept;
42
43 } // namespace quentier

```

6.85 ApplicationSettings.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.

```

```

9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Account.h>
22
23 #include <QSettings>
24
25 #include <string_view>
26
27 namespace quentier::utility {
28
29 class QUINTIER_EXPORT ApplicationSettings :
30     public QSettings,
31     public utility::Printable
32 {
33     Q_OBJECT
34 public:
35     explicit ApplicationSettings(const QString & settingsName = {});
36
37     explicit ApplicationSettings(
38         const Account & account, const QString & settingsName = {});
39
40     ApplicationSettings(
41         const Account & account, const char * settingsName,
42         int settingsNameSize = -1);
43
44     ApplicationSettings(const Account & account, std::string_view settingsName);
45
46     ~ApplicationSettings() override;
47
48     struct ArrayCloser
49     {
50         ArrayCloser(ApplicationSettings & settings) : m_settings(settings) {}
51
52         ~ArrayCloser()
53         {
54             m_settings.endArray();
55             m_settings.sync();
56         }
57
58         ApplicationSettings & m_settings;
59     };
60
61     struct GroupCloser
62     {
63         GroupCloser(ApplicationSettings & settings) : m_settings(settings) {}
64
65         ~GroupCloser()
66         {
67             m_settings.endGroup();
68             m_settings.sync();
69         }
70
71         ApplicationSettings & m_settings;
72     };
73
74     void beginGroup(const QString & prefix);
75
76     void beginGroup(const char * prefix, int size = -1);
77
78     void beginGroup(std::string_view prefix);
79
80     [[nodiscard]] int beginReadArray(const QString & prefix);
81
82     [[nodiscard]] int beginReadArray(const char * prefix, int size = -1);
83
84     [[nodiscard]] int beginReadArray(std::string_view prefix);
85
86     void beginWriteArray(const QString & prefix, int arraySize = -1);
87
88     void beginWriteArray(
89         const char * prefix, int arraySize = -1, int prefixSize = -1);
90
91     void beginWriteArray(std::string_view prefix, int arraySize = -1);
92
93     [[nodiscard]] bool contains(const QString & key) const;
94
95     [[nodiscard]] bool contains(const char * key, int size = -1) const;

```

```

263
272     [[nodiscard]] bool contains(std::string_view key) const;
273
280     void remove(const QString & key);
281
292     void remove(const char * key, int size = -1);
293
301     void remove(std::string_view key);
302
310     void setValue(const QString & key, const QVariant & value);
311
323     void setValue(const char * key, const QVariant & value, int keySize = -1);
324
333     void setValue(std::string_view key, const QVariant & value);
334
345     [[nodiscard]] QVariant value(
346         const QString & key, const QVariant & defaultValue = {}) const;
347
362     [[nodiscard]] QVariant value(
363         const char * key, const QVariant & defaultValue = {},
364         int keySize = -1) const;
365
377     [[nodiscard]] QVariant value(
378         std::string_view key, const QVariant & defaultValue = {}) const;
379
380     // utility::Printable
381     QTextStream & print(QTextStream & strm) const override;
382
383 private:
384     Q_DISABLE_COPY(ApplicationSettings)
385 };
386
387 } // namespace quantier::utility

```

6.86 AnyOfCanceler.h

```

1 /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/utility/cancelers/Fwd.h>
22 #include <quantier/utility/cancelers/ICanceler.h>
23
24 #include <QList>
25
26 namespace quantier::utility::cancelers {
27
28     class QUANTIER_EXPORT AnyOfCanceler : public ICanceler
29     {
30     public:
31         explicit AnyOfCanceler(QList<ICancelerPtr> cancelers);
32         AnyOfCanceler(AnyOfCanceler && other) noexcept;
33         AnyOfCanceler & operator=(AnyOfCanceler && other) noexcept;
34         ~AnyOfCanceler() noexcept override;
35
36         [[nodiscard]] bool isCanceled() const noexcept override;
37
38     private:
39         class Impl;
40         std::unique_ptr<Impl> m_impl;
41     };
42
43 } // namespace quantier::utility::cancelers

```

6.87 FutureCanceler.h

```

1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/cancelers/ICanceler.h>
22
23 #include <QFuture>
24
25 namespace quentier::utility::cancelers {
26
27     template <class T>
28     class FutureCanceler : public ICanceler
29     {
30     public:
31         explicit FutureCanceler(QFuture<T> future) : m_future(std::move(future)) {}
32
33         [[nodiscard]] bool isCanceled() const noexcept override
34         {
35             return m_future.isCanceled();
36         }
37
38     private:
39         QFuture<T> m_future;
40     };
41
42 } // namespace quentier::utility::cancelers

```

6.88 ICanceler.h

```

1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 namespace quentier::utility::cancelers {
24
25     class QUENTIER_EXPORT ICanceler
26     {
27     public:
28         virtual ~ICanceler() = default;
29
30         [[nodiscard]] virtual bool isCanceled() const = 0;
31     };
32
33 } // namespace quentier::utility::cancelers

```

6.89 ManualCanceler.h

```

1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/cancelers/ICanceler.h>
22
23 #include <atomic>
24 #include <memory>
25
26 namespace quentier::utility::cancelers {
27
28     class QUENTIER_EXPORT ManualCanceler : public ICanceler
29     {
30     public:
31         ManualCanceler();
32         ManualCanceler(ManualCanceler && other) noexcept;
33         ManualCanceler & operator=(ManualCanceler && other) noexcept;
34         ~ManualCanceler() noexcept override;
35
36         void cancel() noexcept;
37
38         [[nodiscard]] bool isCanceled() const noexcept override;
39
40     private:
41         class Impl;
42         std::unique_ptr<Impl> m_impl;
43     };
44
45 } // namespace quentier::utility::cancelers

```

6.90 Compat.h

```

1  /*
2  * Copyright 2020-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QHash>
22 #include <QString>
23 #include <QtGlobal>
24
25 // Compatibility with boost parts which require to take a hash of QString
26
27 inline std::size_t hash_value(const QString & x) noexcept
28 {
29     return qHash(x);
30 }

```

6.91 DateTime.h

```

1  /*
2  * Copyright 2020-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QFlags>
24
25 namespace quentier::utility {
26
27 [[nodiscard]] constexpr int secondsToMilliseconds(int seconds) noexcept
28 {
29     return seconds * 1000;
30 }
31
32 enum class DateTimePrintOption
33 {
34     IncludeNumericTimestamp = 1 « 1,
35     IncludeMilliseconds = 1 « 2,
36     IncludeTimezone = 1 « 3
37 };
38
39 Q_DECLARE_FLAGS(DateTimePrintOptions, DateTimePrintOption)
40 Q_DECLARE_OPERATORS_FOR_FLAGS(DateTimePrintOptions)
41
42 [[nodiscard]] QString QUINTIER_EXPORT printableDateTimeFromTimestamp(
43     qint64 timestamp,
44     DateTimePrintOptions options = DateTimePrintOptions(
45         DateTimePrintOption::IncludeNumericTimestamp |
46         DateTimePrintOption::IncludeMilliseconds |
47         DateTimePrintOption::IncludeTimezone),
48     const char * customFormat = nullptr);
49
50 } // namespace quentier::utility

```

6.92 EventLoopWithExitStatus.h

```

1  /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QEventLoop>
25
26 class QDebug;
27 class QTextStream;

```

```

28
29 namespace quentier::utility {
30
31 class QUINTIER_EXPORT EventLoopWithExitStatus : public QEventLoop
32 {
33     Q_OBJECT
34 public:
35     explicit EventLoopWithExitStatus(QObject * parent = nullptr);
36
37     enum class ExitStatus
38     {
39         Success,
40         Failure,
41         Timeout
42     };
43
44     friend QDebug & operator<<(QDebug & dbg, ExitStatus status);
45     friend QTextStream & operator<<(QTextStream & strm, ExitStatus status);
46
47     [[nodiscard]] ExitStatus exitStatus() const;
48     [[nodiscard]] const QString & errorDescription() const;
49
50 public Q_SLOTS:
51     void exitAsSuccess();
52     void exitAsFailure();
53     void exitAsFailureWithError(QString errorDescription);
54     void exitAsFailureWithErrorString(QString errorDescription);
55     void exitAsTimeout();
56
57 private:
58     ExitStatus m_exitStatus;
59     QString m_errorDescription;
60 };
61
62 } // namespace quentier::utility

```

6.93 FileIOProcessorAsync.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorMessage.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QByteArray>
25 #include <QIODevice>
26 #include <QObject>
27 #include <QString>
28 #include <QUuid>
29
30 namespace quentier::utility {
31
32 class FileIOProcessorAsyncPrivate;
33
34 class QUINTIER_EXPORT FileIOProcessorAsync : public QObject
35 {
36     Q_OBJECT
37 public:
38     explicit FileIOProcessorAsync(QObject * parent = nullptr);
39
40     void setIdleTimePeriod(qint32 seconds);
41
42 public Q_SIGNALS:
43     void readyForIO();
44
45     void writeFileRequestProcessed(

```

```

75         bool success, ErrorString errorDescription, QUuid requestId);
76
88     void readFileRequestProcessed(
89         bool success, ErrorString errorDescription, QByteArray data,
90         QUuid requestId);
91
92 public Q_SLOTS:
93     void onWriteFileRequest(
94         QString absoluteFilePath, QByteArray data, QUuid requestId,
95         bool append);
96
97     void onReadFileRequest(QString absoluteFilePath, QUuid requestId);
98
99 private:
100     FileIOProcessorAsyncPrivate * const d_ptr;
101     Q_DECLARE_PRIVATE(FileIOProcessorAsync)
102 };
103
104 } // namespace quentier::utility

```

6.94 FileSystem.h

```

1 /*
2  * Copyright 2020-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QString>
25
26 namespace quentier::utility {
27
28 [[nodiscard]] QString QUENTIER_EXPORT relativePathFromAbsolutePath(
29     const QString & absolutePath, const QString & relativePathRootFolderPath);
30
31 [[nodiscard]] bool QUENTIER_EXPORT removeFile(const QString & filePath);
32
33 [[nodiscard]] bool QUENTIER_EXPORT removeDir(const QString & dirPath);
34
35 [[nodiscard]] QByteArray QUENTIER_EXPORT
36     readFileContents(const QString & filePath, ErrorString & errorDescription);
37
38 [[nodiscard]] bool QUENTIER_EXPORT renameFile(
39     const QString & from, const QString & to, ErrorString & errorDescription);
40
41 } // namespace quentier::utility

```

6.95 FileSystemWatcher.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.

```

```

14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QObject>
24 #include <QStringList>
25
26 namespace quentier::utility {
27
28 class QUENTIER_EXPORT FileSystemWatcher : public QObject
29 {
30     Q_OBJECT
31 public:
32     explicit FileSystemWatcher(
33         int removalTimeoutMsec = 500, QObject * parent = nullptr);
34
35     explicit FileSystemWatcher(
36         const QStringList & paths, int removalTimeoutMsec = 500,
37         QObject * parent = nullptr);
38
39     ~FileSystemWatcher() override;
40
41     void addPath(const QString & path);
42     void addPaths(const QStringList & paths);
43
44     [[nodiscard]] QStringList directories() const;
45     [[nodiscard]] QStringList files() const;
46
47     void removePath(const QString & path);
48     void removePaths(const QStringList & paths);
49
50 Q_SIGNALS:
51     void directoryChanged(const QString & path);
52     void directoryRemoved(const QString & path);
53
54     void fileChanged(const QString & path);
55     void fileRemoved(const QString & path);
56
57 private:
58     Q_DISABLE_COPY(FileSystemWatcher)
59
60 private:
61     class FileSystemWatcherPrivate;
62
63     FileSystemWatcherPrivate * d_ptr;
64     Q_DECLARE_PRIVATE(FileSystemWatcher)
65 };
66
67 } // namespace quentier::utility

```

6.96 IEncryptor.h

```

1 /*
2 * Copyright 2024-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorMessage.h>
22 #include <quentier/types/Result.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QString>
26

```

```

27 class QDebug;
28 class QTextStream;
29
30 namespace quantier::utility {
31
32 struct QUINTIER_EXPORT IEncryptor
33 {
34     virtual ~IEncryptor() noexcept;
35
36     enum class Cipher
37     {
38         RC2,
39         AES
40     };
41
42     friend QUINTIER_EXPORT QDebug & operator«(QDebug & dbg, Cipher cipher);
43
44     friend QUINTIER_EXPORT QTextStream & operator«(
45         QTextStream & strm, Cipher cipher);
46
47     [[nodiscard]] virtual Result<QString, ErrorString> encrypt(
48         const QString & text, const QString & passphrase) = 0;
49
50     [[nodiscard]] virtual Result<QString, ErrorString> decrypt(
51         const QString & encryptedText, const QString & passphrase,
52         Cipher cipher) = 0;
53 };
54
55 } // namespace quantier::utility

```

6.97 IKeychainService.h

```

1 /*
2  * Copyright 2018–2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/exception/IQuantierException.h>
22 #include <quantier/types/ErrorString.h>
23
24 #include <quantier/utility/Fwd.h>
25 #include <quantier/utility/Linkage.h>
26
27 #include <QFuture>
28
29 class QDebug;
30
31 namespace quantier::utility {
32
33 class QUINTIER_EXPORT IKeychainService
34 {
35 public:
36     virtual ~IKeychainService() noexcept;
37
38     enum class ErrorCode
39     {
40         NoError,
41         EntryNotFound,
42         CouldNotDeleteEntry,
43         AccessDeniedByUser,
44         AccessDenied,
45         NoBackendAvailable,
46         NotImplemented,
47         OtherError
48     };
49
50     friend QUINTIER_EXPORT QTextStream & operator«(
51         QTextStream & strm, ErrorCode errorCode);

```

```

83
84     friend QUINTIER_EXPORT QDebug & operator<<(
85         QDebug & dbg, ErrorCode errorCode);
86
87     class QUINTIER_EXPORT Exception : public IQuentierException
88     {
89     public:
90         explicit Exception(ErrorCode errorCode) noexcept;
91
92         explicit Exception(
93             ErrorCode errorCode, ErrorString errorDescription) noexcept;
94
95         [[nodiscard]] ErrorCode errorCode() const noexcept;
96         [[nodiscard]] QString exceptionDisplayName() const override;
97
98         void raise() const override;
99         [[nodiscard]] Exception * clone() const override;
100
101     private:
102         const ErrorCode m_errorCode;
103     };
104
105 public:
106     [[nodiscard]] virtual QFuture<void> writePassword(
107         QString service, QString key, QString password) = 0;
108
109     [[nodiscard]] virtual QFuture<QString> readPassword(
110         QString service, QString key) const = 0;
111
112     [[nodiscard]] virtual QFuture<void> deletePassword(
113         QString service, QString key) = 0;
114 };
115
116 } // namespace quentier::utility

```

6.98 Initialize.h

```

1 /*
2  * Copyright 2020-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 namespace quentier::utility {
24
25 void QUINTIER_EXPORT initializeLibquentier();
26
27 } // namespace quentier::utility

```

6.99 LRUCache.hpp

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```

13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QHash>
22
23 #include <cstddef>
24 #include <list>
25
26 namespace quentier::utility {
27
28 template <
29     class Key, class Value,
30     class Allocator = std::allocator<std::pair<Key, Value>>
31 class LRUCache
32 {
33 public:
34     explicit LRUCache(const std::size_t maxSize = 100) : m_maxSize{maxSize} {}
35
36     using key_type = Key;
37     using mapped_type = Value;
38     using allocator_type = Allocator;
39     using value_type = std::pair<key_type, mapped_type>;
40     using container_type = std::list<value_type, allocator_type>;
41     using size_type = typename container_type::size_type;
42     using difference_type = typename container_type::difference_type;
43     using iterator = typename container_type::iterator;
44     using const_iterator = typename container_type::const_iterator;
45     using reverse_iterator = std::reverse_iterator<iterator>;
46     using const_reverse_iterator = std::reverse_iterator<const_iterator>;
47
48     using reference = value_type &;
49     using const_reference = const value_type &;
50     using pointer = typename std::allocator_traits<allocator_type>::pointer;
51
52     using const_pointer =
53         typename std::allocator_traits<allocator_type>::const_pointer;
54
55     [[nodiscard]] iterator begin() noexcept
56     {
57         return m_container.begin();
58     }
59
60     [[nodiscard]] const_iterator begin() const noexcept
61     {
62         return m_container.begin();
63     }
64
65     [[nodiscard]] reverse_iterator rbegin() noexcept
66     {
67         return m_container.rbegin();
68     }
69
70     [[nodiscard]] const_reverse_iterator rbegin() const noexcept
71     {
72         return m_container.rbegin();
73     }
74
75     [[nodiscard]] iterator end() noexcept
76     {
77         return m_container.end();
78     }
79
80     [[nodiscard]] const_iterator end() const noexcept
81     {
82         return m_container.end();
83     }
84
85     [[nodiscard]] reverse_iterator rend() noexcept
86     {
87         return m_container.rend();
88     }
89
90     [[nodiscard]] const_reverse_iterator rend() const noexcept
91     {
92         return m_container.rend();
93     }
94
95     [[nodiscard]] bool empty() const noexcept
96     {
97         return m_container.empty();
98     }
99

```

```

100     [[nodiscard]] std::size_t size() const noexcept
101     {
102         return m_currentSize;
103     }
104
105     [[nodiscard]] std::size_t max_size() const noexcept
106     {
107         return m_maxSize;
108     }
109
110     void clear()
111     {
112         m_container.clear();
113         m_mapper.clear();
114         m_currentSize = 0;
115     }
116
117     void put(const key_type & key, const mapped_type & value)
118     {
119         Q_UNUSED(remove(key))
120
121         m_container.push_front(value_type(key, value));
122         m_mapper[key] = m_container.begin();
123         ++m_currentSize;
124
125         fixupSize();
126     }
127
128     [[nodiscard]] const mapped_type * get(const key_type & key) const noexcept
129     {
130         auto mapperIt = m_mapper.find(key);
131         if (mapperIt == m_mapper.end()) {
132             return nullptr;
133         }
134
135         auto it = mapperIt.value();
136         if (it == m_container.end()) {
137             return nullptr;
138         }
139
140         m_container.splice(m_container.begin(), m_container, it);
141         mapperIt.value() = m_container.begin();
142         return &(mapperIt.value()->second);
143     }
144
145     [[nodiscard]] bool exists(const key_type & key) const noexcept
146     {
147         const auto mapperIt = m_mapper.find(key);
148         if (mapperIt == m_mapper.end()) {
149             return false;
150         }
151
152         const auto it = mapperIt.value();
153         return (it != m_container.end());
154     }
155
156     bool remove(const key_type & key) noexcept
157     {
158         const auto mapperIt = m_mapper.find(key);
159         if (mapperIt == m_mapper.end()) {
160             return false;
161         }
162
163         const auto it = mapperIt.value();
164         Q_UNUSED(m_container.erase(it))
165         Q_UNUSED(m_mapper.erase(mapperIt))
166
167         if (m_currentSize != 0) {
168             --m_currentSize;
169         }
170
171         return true;
172     }
173
174     void setMaxSize(const std::size_t maxSize)
175     {
176         if (maxSize >= m_maxSize) {
177             m_maxSize = maxSize;
178             return;
179         }
180
181         std::size_t diff = m_maxSize - maxSize;
182         for (std::size_t i = 0; (i < diff) && !m_container.empty(); ++i) {
183             auto lastIt = m_container.end();
184             --lastIt;
185
186             const key_type & lastElementKey = lastIt->first;

```

```

187         Q_UNUSED(m_mapper.remove(lastElementKey))
188         Q_UNUSED(m_container.erase(lastIt))
189
190         if (m_currentSize != 0) {
191             --m_currentSize;
192         }
193     }
194 }
195
196 private:
197     void fixupSize()
198     {
199         if (m_currentSize <= m_maxSize) {
200             return;
201         }
202
203         if (Q_UNLIKELY(m_container.empty())) {
204             return;
205         }
206
207         auto lastIt = m_container.end();
208         --lastIt;
209
210         const key_type & lastElementKey = lastIt->first;
211
212         Q_UNUSED(m_mapper.remove(lastElementKey))
213         Q_UNUSED(m_container.erase(lastIt))
214
215         if (m_currentSize != 0) {
216             --m_currentSize;
217         }
218     }
219
220 private:
221     mutable container_type m_container;
222     std::size_t m_currentSize = 0;
223     std::size_t m_maxSize;
224
225     mutable QHash<Key, iterator> m_mapper;
226 };
227
228 } // namespace quantier::utility

```

6.100 MessageBox.h

```

1  /*
2  * Copyright 2017-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/utility/Linkage.h>
22
23 #include <QMessageBox>
24
25 namespace quantier::utility {
26
27 int QUENTIER_EXPORT genericMessageBox(
28     QWidget * parent, const QString & title, const QString & briefText,
29     const QString & detailedText = {},
30     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
31
32 int QUENTIER_EXPORT informationMessageBox(
33     QWidget * parent, const QString & title, const QString & briefText,
34     const QString & detailedText = {},
35     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
36
37 int QUENTIER_EXPORT warningMessageBox(
38     QWidget * parent, const QString & title, const QString & briefText,

```

```

39     const QString & detailedText = {},
40     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
41
42 int QUINTIER_EXPORT criticalMessageBox(
43     QWidget * parent, const QString & title, const QString & briefText,
44     const QString & detailedText = {},
45     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
46
47 [[nodiscard]] int QUINTIER_EXPORT questionMessageBox(
48     QWidget * parent, const QString & title, const QString & briefText,
49     const QString & detailedText = {},
50     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok |
51     QMessageBox::Cancel);
52
53 void QUINTIER_EXPORT
54     internalErrorMessageBox(QWidget * parent, QString detailedText = {});
55
56 } // namespace quentier::utility

```

6.101 PlatformUtils.h

```

1  /*
2  * Copyright 2020-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24 #include <QUrl>
25
26 namespace quentier::utility {
27
28 [[nodiscard]] QString QUINTIER_EXPORT getCurrentUserName();
29
30 [[nodiscard]] QString QUINTIER_EXPORT getCurrentUserFullName();
31
32 void QUINTIER_EXPORT openUrl(const QUrl & url);
33
34 } // namespace quentier::utility

```

6.102 Printable.h

```

1  /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>

```

```

22
23 #include <QDebug>
24 #include <QHash>
25 #include <QIODevice>
26 #include <QSet>
27 #include <QString>
28 #include <QTextStream>
29
30 namespace quentier::utility {
31
32     class QUINTIER_EXPORT Printable
33     {
34     public:
35         virtual ~Printable() noexcept;
36
37         virtual QTextStream & print(QTextStream & strm) const = 0;
38
39         [[nodiscard]] QString toString() const;
40
41         friend QUINTIER_EXPORT QTextStream & operator<<
42             (QTextStream & strm, const Printable & printable);
43
44         friend QUINTIER_EXPORT QDebug & operator<<
45             (QDebug & debug, const Printable & printable);
46     };
47
48 // namespace quentier::utility
49
50 // printing operators for existing classes not inheriting from Printable
51
52 template <class T>
53 [[nodiscard]] QString ToString(const T & object)
54 {
55     QString str;
56     QTextStream strm(&str, QIODevice::WriteOnly);
57     strm << object;
58     return str;
59 }
60
61 template <class TKey, class TValue>
62 [[nodiscard]] QString ToString(const QHash<TKey, TValue> & object)
63 {
64     QString str;
65     QTextStream strm(&str, QIODevice::WriteOnly);
66     strm << QStringLiteral("QHash:  \n");
67
68     using CIter = typename QHash<TKey, TValue>::const_iterator;
69     CIter hashEnd = object.end();
70     for (CIter it = object.begin(); it != hashEnd; ++it) {
71         strm << QStringLiteral("(") << it.key() << QStringLiteral(" = ")
72             << it.value() << QStringLiteral("; \n");
73     }
74     return str;
75 }
76
77 template <class T>
78 [[nodiscard]] QString ToString(const QSet<T> & object)
79 {
80     QString str;
81     QTextStream strm(&str, QIODevice::WriteOnly);
82     strm << QStringLiteral("QSet:  \n");
83
84     using CIter = typename QSet<T>::const_iterator;
85     CIter setEnd = object.end();
86     for (CIter it = object.begin(); it != setEnd; ++it) {
87         strm << QStringLiteral("{") << *it << QStringLiteral("; \n");
88     }
89     return str;
90 }
91
92 #define QUINTIER_DECLARE_PRINTABLE(type, ...)
93 \
94 QUINTIER_EXPORT QTextStream & operator<<
95 (QTextStream & strm, const type & obj);
96 inline QDebug & operator<<(QDebug & debug, const type & obj)
97 {
98     debug << ToString<type, #__VA_ARGS__>(obj);
99     return debug;
100 }
101 // QUINTIER_DECLARE_PRINTABLE

```

6.103 QuentierApplication.h

1 /*

```

2 * Copyright 2016-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QApplication>
24
25 namespace quentier::utility {
26
27 class QUENTIER_EXPORT QuentierApplication : public QApplication
28 {
29     Q_OBJECT
30 public:
31     QuentierApplication(int & argc, char * argv[]); // NOLINT
32     ~QuentierApplication() noexcept override;
33
34     [[nodiscard]] bool notify(QObject * object, QEvent * event) override;
35     [[nodiscard]] bool event(QEvent * event) override;
36 };
37
38 } // namespace quentier::utility

```

6.104 QuentierUndoCommand.h

```

1 /*
2 * Copyright 2016-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorMessage.h>
22
23 #include <QObject>
24 #include <QUndoCommand>
25
26 namespace quentier::utility {
27
28 class QUENTIER_EXPORT QuentierUndoCommand : public QObject, public QUndoCommand
29 {
30     Q_OBJECT
31 public:
32     QuentierUndoCommand(QUndoCommand * parent = nullptr);
33     QuentierUndoCommand(const QString & text, QUndoCommand * parent = nullptr);
34     ~QuentierUndoCommand() noexcept override;
35
36     void undo() final;
37     void redo() final;
38
39     [[nodiscard]] bool onceUndoExecuted() const noexcept
40     {
41         return m_onceUndoExecuted;
42     }
43 };
44
45 }

```

```

69 Q_SIGNALS:
70     void notifyError(ErrorString error);
71
72 protected:
73     virtual void undoImpl() = 0;
74     virtual void redoImpl() = 0;
75
76 private:
77     bool m_onceUndoExecuted = false;
78 };
79
80 } // namespace quantier::utility

```

6.105 ShortcutManager.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/types/Account.h>
22 #include <quantier/utility/Linkage.h>
23
24 #include <QKeySequence>
25 #include <QObject>
26
27 namespace quantier::utility {
28
29 class QUANTIER_EXPORT ShortcutManager : public QObject
30 {
31     Q_OBJECT
32 public:
33     explicit ShortcutManager(QObject * parent = nullptr);
34
35     enum QuantierShortcutKey
36     {
37         NewNote = 5000,
38         NewTag,
39         NewNotebook,
40         NewSavedSearch,
41         AddAttachment,
42         SaveAttachment,
43         OpenAttachment,
44         CopyAttachment,
45         CutAttachment,
46         RemoveAttachment,
47         RenameAttachment,
48         AddAccount,
49         ExitAccount,
50         SwitchAccount,
51         AccountInfo,
52         NoteSearch,
53         NewNoteSearch,
54         ShowNotes,
55         ShowNotebooks,
56         ShowTags,
57         ShowSavedSearches,
58         ShowDeletedNotes,
59         ShowStatusBar,
60         ShowToolBar,
61         PasteUnformatted,
62         Font,
63         UpperIndex,
64         LowerIndex,
65         AlignLeft,
66         AlignCenter,
67         AlignRight,
68         AlignFull,

```

```

69         IncreaseIndentation,
70         DecreaseIndentation,
71         IncreaseFontSize,
72         DecreaseFontSize,
73         InsertNumberedList,
74         InsertBulletedList,
75         Strikethrough,
76         Highlight,
77         InsertTable,
78         InsertRow,
79         InsertColumn,
80         RemoveRow,
81         RemoveColumn,
82         InsertHorizontalLine,
83         InsertToDoTag,
84         EditHyperlink,
85         CopyHyperlink,
86         RemoveHyperlink,
87         Encrypt,
88         Decrypt,
89         DecryptPermanently,
90         BackupLocalStorage,
91         RestoreLocalStorage,
92         UpgradeLocalStorage,
93         LocalStorageStatus,
94         SpellCheck,
95         SpellCheckIgnoreWord,
96         SpellCheckAddWordToUserDictionary,
97         SaveImage,
98         AnnotateImage,
99         ImageRotateClockwise,
100        ImageRotateCounterClockwise,
101        Synchronize,
102        FullSync,
103        ImportFolders,
104        Preferences,
105        ReleaseNotes,
106        ViewLogs,
107        About,
108        UnknownKey = 100000
109    };
110
111    [[nodiscard]] QKeySequence shortcut(
112        int key, const Account & account, const QString & context = {}) const;
113
114    [[nodiscard]] QKeySequence shortcut(
115        const QString & nonStandardKey, const Account & account,
116        const QString & context = {}) const;
117
118    [[nodiscard]] QKeySequence defaultShortcut(
119        int key, const Account & account, const QString & context = {}) const;
120
121    [[nodiscard]] QKeySequence defaultShortcut(
122        const QString & nonStandardKey, const Account & account,
123        const QString & context = {}) const;
124
125    [[nodiscard]] QKeySequence userShortcut(
126        int key, const Account & account, const QString & context = {}) const;
127
128    [[nodiscard]] QKeySequence userShortcut(
129        const QString & nonStandardKey, const Account & account,
130        const QString & context = {}) const;
131
132    Q_SIGNALS:
133    void shortcutChanged(
134        int key, QKeySequence shortcut, const Account & account,
135        QString context);
136
137    void nonStandardShortcutChanged(
138        QString nonStandardKey, QKeySequence shortcut, const Account & account,
139        QString context);
140
141    public Q_SLOTS:
142    void setUserShortcut(
143        int key, const QKeySequence & shortcut, const Account & account,
144        QString context = {});
145
146    void setNonStandardUserShortcut(
147        QString nonStandardKey, const QKeySequence & shortcut,
148        const Account & account, QString context = {});
149
150    void setDefaultShortcut(
151        int key, const QKeySequence & shortcut, const Account & account,
152        QString context = {});
153
154    void setNonStandardDefaultShortcut(
155        QString nonStandardKey, const QKeySequence & shortcut,

```

```

182         const Account & account, QString context = {});
183
184 private:
185     class ShortcutManagerPrivate;
186
187     ShortcutManagerPrivate * const d_ptr;
188     Q_DECLARE_PRIVATE(ShortcutManager)
189 };
190
191 } // namespace quantier::utility

```

6.106 Size.h

```

1 /*
2  * Copyright 2020-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/utility/Linkage.h>
22
23 #include <QString>
24
25 namespace quantier::utility {
26
27 [[nodiscard]] QString QUENTIER_EXPORT humanReadableSize(quint64 bytes);
28
29 } // namespace quantier::utility

```

6.107 StandardPaths.h

```

1 /*
2  * Copyright 2017-2025 Dmitry Ivanov
3  *
4  * This file is part of libquantier
5  *
6  * libquantier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquantier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quantier/types/Account.h>
22 #include <quantier/utility/Linkage.h>
23
24 #define LIBQUENTIER_PERSISTENCE_STORAGE_PATH \
25 "LIBQUENTIER_PERSISTENCE_STORAGE_PATH"
26
27 namespace quantier::utility {
28
29 constexpr const char * gLibquantierPersistenceStoragePath =
30     "LIBQUENTIER_PERSISTENCE_STORAGE_PATH";
31
32 [[nodiscard]] QString QUENTIER_EXPORT
33     applicationPersistentStoragePath(bool * nonStandardLocation = nullptr);
34
35 }

```

```

56 [[nodiscard]] QString QUENTIER_EXPORT
57     accountPersistentStoragePath(const Account & account);
58
63 [[nodiscard]] QString QUENTIER_EXPORT applicationTemporaryStoragePath();
64
70 [[nodiscard]] QString QUENTIER_EXPORT homePath();
71
75 [[nodiscard]] QString QUENTIER_EXPORT documentsPath();
76
77 } // namespace quentier::utility

```

6.108 StringUtils.h

```

1 /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QList>
24 #include <QSet>
25 #include <QString>
26
27 namespace quentier::utility {
28
29 class QUENTIER_EXPORT StringUtils
30 {
31 public:
32     StringUtils();
33     ~StringUtils() noexcept;
34
35     void removePunctuation(
36         QString & str, const QList<QChar> & charactersToPreserve = {} ) const;
37
38     void removeDiacritics(QString & str) const;
39     void removeNewlines(QString & str) const;
40
41 private:
42     class StringUtilsPrivate;
43     StringUtilsPrivate * const d_ptr;
44     Q_DECLARE_PRIVATE(StringUtils);
45 };
46
47 } // namespace quentier::utility

```

6.109 SuppressWarnings.h

```

1 /*
2  * Copyright 2020-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.

```

```

17 */
18
19 #pragma once
20
21 // Common macros
22
23 #define STRINGIFY(a) #a
24
25 // Define empty macros doing nothing for supported compilers, they would be used
26 // as fallback when any of these compilers are not actually used
27
28 #define SAVE_WARNINGS
29
30 #define CLANG_SUPPRESS_WARNING(warning)
31 #define GCC_SUPPRESS_WARNING(warning)
32 #define MSVC_SUPPRESS_WARNING(warning)
33
34 #define RESTORE_WARNINGS
35
36 // Clang implementation
37
38 #if defined(__clang__)
39
40 #undef CLANG_SUPPRESS_WARNING
41
42 #define CLANG_SUPPRESS_WARNING(warning) \
43     _Pragma(Stringify(clang diagnostic ignored #warning))
44
45 #undef SAVE_WARNINGS
46
47 #define SAVE_WARNINGS _Pragma("clang diagnostic push")
48
49 #undef RESTORE_WARNINGS
50
51 #define RESTORE_WARNINGS _Pragma("clang diagnostic pop")
52
53 #endif // clang
54
55 // GCC implementation
56
57 // Clang can mimic gcc so need to ensure it's indeed gcc
58 #if defined(__GNUC__) && !defined(__clang__)
59
60 #undef GCC_SUPPRESS_WARNING
61
62 #define GCC_SUPPRESS_WARNING(warning) \
63     _Pragma(Stringify(GCC diagnostic ignored #warning))
64
65 #undef SAVE_WARNINGS
66
67 #define SAVE_WARNINGS _Pragma("GCC diagnostic push")
68
69 #undef RESTORE_WARNINGS
70
71 #define RESTORE_WARNINGS _Pragma("GCC diagnostic pop")
72
73 #endif // GCC
74
75 // MSVC implementation
76
77 #if defined(_MSC_VER)
78
79 #undef MSVC_SUPPRESS_WARNING
80
81 #define MSVC_SUPPRESS_WARNING(number) __pragma(warning(disable : number))
82
83 #undef SAVE_WARNINGS
84
85 #define SAVE_WARNINGS __pragma(warning(push))
86
87 #undef RESTORE_WARNINGS
88
89 #define RESTORE_WARNINGS __pragma(warning(pop))
90
91 #endif // MSVC

```

6.110 SysInfo.h

```

1 /*
2 * Copyright 2016-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *

```

```

6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24
25 namespace quentier::utility {
26
27 class QUENTIER_EXPORT SysInfo
28 {
29 public:
30     SysInfo();
31     ~SysInfo() noexcept;
32
33     [[nodiscard]] quint64 pageSize();
34     [[nodiscard]] quint64 totalMemory();
35     [[nodiscard]] quint64 freeMemory();
36
37     [[nodiscard]] QString stackTrace();
38
39     [[nodiscard]] QString platformName();
40
41 private:
42     Q_DISABLE_COPY(SysInfo)
43
44 private:
45     class SysInfoPrivate;
46     SysInfoPrivate * const d_ptr;
47     Q_DECLARE_PRIVATE(SysInfo)
48 };
49
50 } // namespace quentier::utility

```

6.111 TagSortByParentChildRelations.h

```

1 /*
2 * Copyright 2017-2025 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/types/Fwd.h>
25
26 #include <QList>
27
28 namespace quentier::utility {
29
30 bool QUENTIER_EXPORT sortTagsByParentChildRelations(
31     QList<qevercloud::Tag> & tagList, QString & errorDescription);
32
33 } // namespace quentier::utility

```

6.112 MockIKeychainService.h

```

1  /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #include <quentier/utility/IKeychainService.h>
20
21 #include <gmock/gmock.h>
22
23 namespace quentier::utility::tests::mocks {
24
25 class MockIKeychainService : public IKeychainService
26 {
27 public:
28     MOCK_METHOD(
29         QFuture<void>, writePassword,
30         (QString service, QString key, QString password), (override));
31
32     MOCK_METHOD(
33         QFuture<QString>, readPassword, (QString service, QString key),
34         (const, override));
35
36     MOCK_METHOD(
37         QFuture<void>, deletePassword, (QString service, QString key),
38         (override));
39 };
40
41 } // namespace quentier::utility::tests::mocks

```

6.113 UidGenerator.h

```

1  /*
2  * Copyright 2016-2025 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24 #include <QUuid>
25
26 namespace quentier::utility {
27
28 class QUENTIER_EXPORT UidGenerator
29 {
30 public:
31     [[nodiscard]] static QString generate();
32     [[nodiscard]] static QString uidToString(const QUuid & uid);
33 };
34
35 } // namespace quentier::utility

```

6.114 Unreachable.h

```
1 /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QtGlobal>
22
23 #ifdef _MSC_VER
24 #define UNREACHABLE
25 do {
26     Q_ASSERT(false);
27     __assume(0);
28 } while (false)
29 #else
30 #define UNREACHABLE
31 do {
32     Q_ASSERT(false);
33     __builtin_unreachable();
34 } while (false)
35 #endif
```

Index

- ~ApplicationSettings
 - quentier::utility::ApplicationSettings, [23](#)
- AccessDenied
 - quentier::utility::IKeychainService, [64](#)
- AccessDeniedByUser
 - quentier::utility::IKeychainService, [64](#)
- Account.h, [234](#)
- addedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCountersCipher
 - [91](#)
- addedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters,
 - [91](#)
- addedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters,
 - [91](#)
- addedTags
 - quentier::synchronization::ISyncChunksDataCounters,
 - [91](#)
- AES
 - quentier::utility::IEncryptor, [57](#)
- Affiliation
 - quentier::local_storage::ILocalStorage, [70](#)
- Any
 - quentier::local_storage::ILocalStorage, [71](#)
- AnyOfCanceler.h, [250](#)
- ApplicationSettings
 - quentier::utility::ApplicationSettings, [21](#), [22](#)
- ApplicationSettings.h, [248](#)
- apply
 - quentier::local_storage::IPatch, [78](#)
- AttributeName
 - quentier::enml::conversion_rules::ISkipRule, [88](#)
- AttributeValue
 - quentier::enml::conversion_rules::ISkipRule, [88](#)
- AuthenticationInfo.h, [215](#)
- authenticationTime
 - quentier::synchronization::IAuthenticationInfo, [45](#)
- authToken
 - quentier::synchronization::IAuthenticationInfo, [45](#)
- authTokenExpirationTime
 - quentier::synchronization::IAuthenticationInfo, [45](#)
- backend
 - quentier::NoteEditor, [133](#)
- backupLocalStorage
 - quentier::local_storage::IPatch, [78](#)
- beginGroup
 - quentier::utility::ApplicationSettings, [23](#), [24](#)
- beginReadArray
 - quentier::utility::ApplicationSettings, [24](#), [25](#)
- beginWriteArray
 - quentier::utility::ApplicationSettings, [25](#), [26](#)
- cancel
 - quentier::utility::cancelers::ManualCanceler, [121](#)
- caseSensitivity
 - quentier::enml::conversion_rules::ISkipRule, [88](#)
- contains
 - quentier::utility::IEncryptor, [57](#)
- clear
 - quentier::NoteEditor, [133](#)
- Compat.h, [252](#)
- convertDecryptedText
 - quentier::enml::IENMLTagsConverter, [59](#)
- convertEncryptedText
 - quentier::enml::IENMLTagsConverter, [59](#)
- convertEnmlToHtml
 - quentier::enml::IConverter, [49](#)
- convertEnmlToPlainText
 - quentier::enml::IConverter, [49](#)
- convertEnmlToWordsList
 - quentier::enml::IConverter, [50](#)
- convertEnToDo
 - quentier::enml::IENMLTagsConverter, [60](#)
- convertHtmlToDoc
 - quentier::enml::IConverter, [50](#)
- convertHtmlToEnml
 - quentier::enml::IConverter, [51](#)
- convertHtmlToXhtml
 - quentier::enml::IConverter, [51](#)
- convertHtmlToXml
 - quentier::enml::IConverter, [51](#)
- convertPlainTextToWordsList
 - quentier::enml::IConverter, [52](#)
- convertResource
 - quentier::enml::IENMLTagsConverter, [60](#)
- convertToNote
 - quentier::NoteEditor, [134](#)
- CouldNotDeleteEntry
 - quentier::utility::IKeychainService, [64](#)
- currentNoteLocalId
 - quentier::NoteEditor, [134](#)
- Date.h, [253](#)
- decrypt
 - quentier::utility::IEncryptor, [58](#)

- defaultFont
 - quentier::NoteEditor, 134
- defaultPalette
 - quentier::NoteEditor, 134
- defaultShortcut
 - quentier::utility::ShortcutManager, 156
- deletePassword
 - quentier::utility::IKeychainService, 65
- displayName
 - quentier::Account, 17
- downloadFinished
 - quentier::synchronization::ISyncEventsNotifier, 96
- DownloadNotesStatus.h, 215
- downloadNoteThumbnails
 - quentier::synchronization::ISyncOptions, 102
- DownloadResourcesStatus.h, 216
- Element
 - quentier::enml::conversion_rules::ISkipRule, 88
- encrypt
 - quentier::utility::IEncryptor, 58
- EntryNotFound
 - quentier::utility::IKeychainService, 64
- ErrorCode
 - quentier::utility::IKeychainService, 64
- Errors.h, 205
- ErrorString.h, 236
- EventLoopWithExitStatus.h, 253
- evernoteAccountType
 - quentier::Account, 17
- evernoteHost
 - quentier::Account, 17
- exceptionDisplayName
 - quentier::InvalidArgument, 77
 - quentier::local_storage::LocalStorageOpenException, 117
 - quentier::local_storage::LocalStorageOperationException, 118
 - quentier::OperationCanceled, 143
 - quentier::RuntimeError, 154
 - quentier::utility::IKeychainService::Exception, 37
- exportNotesToEnex
 - quentier::enml::IConverter, 52
- expungedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, 92
- expungedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, 92
- expungedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters, 92
- expungedTags
 - quentier::synchronization::ISyncChunksDataCounters, 92
- Factory.h, 163–165
- failedToSendNotes
 - quentier::synchronization::ISendStatus, 83
- failedToSendSavedSearches
 - quentier::synchronization::ISendStatus, 83
- failedToSendTags
 - quentier::synchronization::ISendStatus, 84
- FileIOProcessorAsync.h, 254
- FileSystem.h, 255
- FileSystemWatcher.h, 255
- fromVersion
 - quentier::local_storage::IPatch, 78
- Future.h, 218
- FutureCanceler.h, 251
- Fwd.h, 237–241
- html
 - quentier::enml::IHtmlData, 62
- HtmlUtils.h, 166
- IAuthenticationInfo.h, 206
- IAuthenticationInfoBuilder.h, 206
- IAuthenticator.h, 197
- ICanceler.h, 251
- IConverter.h, 166
- id
 - quentier::Account, 17
- IDecryptedTextCache.h, 167
- idleTime
 - quentier::NoteEditor, 134
- IDownloadNotesStatus.h, 207
- IDownloadResourcesStatus.h, 208
- IEncryptor.h, 256
- IENMLTagsConverter.h, 168
- IHtmlData.h, 169
- IKeychainService.h, 257
- ILocalStorage.h, 172
- ILocalStorageNotifier.h, 180
- importEnex
 - quentier::enml::IConverter, 52
- inAppNoteLinkPasteRequested
 - quentier::NoteEditor, 134
- includeContents
 - quentier::enml::conversion_rules::ISkipRule, 88
- initialize
 - quentier::NoteEditor, 135
- Initialize.h, 258
- inkNoteImagesStorageDir
 - quentier::synchronization::ISyncOptions, 102
- INoteEditorBackend.h, 190
- INoteStoreFactory.h, 198
- InvalidArgument.h, 170
- IPatch.h, 181
- IQuentierException.h, 170
- isCanceled
 - quentier::utility::cancelers::AnyOfCanceler, 20
 - quentier::utility::cancelers::FutureCanceler< T >, 43
 - quentier::utility::cancelers::ManualCanceler, 121
- isEditorPageModified

- quentier::NoteEditor, 135
- isEmpty
 - quentier::Account, 17
- ISendStatus.h, 209
- ISkipRule.h, 161
- ISkipRuleBuilder.h, 162
- isModified
 - quentier::NoteEditor, 135
- isNoteLoaded
 - quentier::NoteEditor, 136
- isValid
 - quentier::Result< ValueType, ErrorType, typename >, 152
- ISyncChunksDataCounters.h, 210
- ISyncConflictResolver.h, 198
- ISyncEventsNotifier.h, 200
- ISynchronizer.h, 201
- ISyncOptions.h, 211
- ISyncOptionsBuilder.h, 212
- ISyncResult.h, 213
- ISyncState.h, 214
- ISyncStateBuilder.h, 214
- ISyncStateStorage.h, 202
- IUserStoreFactory.h, 202
- linkedNotebookNotesDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 97
- linkedNotebookResourcesDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 97
- linkedNotebookSendStatusUpdate
 - quentier::synchronization::ISyncEventsNotifier, 97
- linkedNotebookSyncChunksDataProcessingProgress
 - quentier::synchronization::ISyncEventsNotifier, 98
- linkedNotebookSyncChunksDownloaded
 - quentier::synchronization::ISyncEventsNotifier, 98
- linkedNotebookSyncChunksDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 98
- LocalStorageOpenException.h, 182
- LocalStorageOperationException.h, 182
- LRUCache.hpp, 258
- ManualCanceler.h, 252
- matchMode
 - quentier::enml::conversion_rules::ISkipRule, 88
- MatchMode.h, 162
- maxConcurrentNoteDownloads
 - quentier::synchronization::ISyncOptions, 103
- maxConcurrentResourceDownloads
 - quentier::synchronization::ISyncOptions, 103
- MessageBox.h, 261
- mine
 - quentier::synchronization::ISyncConflictResolver::ConflictResolutionMode, 129
- MockIAuthenticator.h, 203
- MockIKeychainService.h, 271
- MockILocalStorage.h, 185
- MockINoteStoreFactory.h, 203
- MockISyncConflictResolver.h, 204
- MockISyncStateStorage.h, 204
- name
 - quentier::Account, 18
- needToRepeatIncrementalSync
 - quentier::synchronization::ISendStatus, 84
- NoBackendAvailable
 - quentier::utility::IKeychainService, 64
- NoError
 - quentier::utility::IKeychainService, 64
- notebookModifier
 - quentier::local_storage::NoteSearchQuery, 142
- NoteEditor.h, 193
- notesDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 99
- NoteSearchQuery.h, 183
- noteStoreUrl
 - quentier::synchronization::IAuthenticationInfo, 45
- NoteUtils.h, 241
- notifier
 - quentier::local_storage::ILocalStorage, 71
- notifySyncStateUpdated
 - quentier::synchronization::ISyncStateStorage, 108
- NotImplemented
 - quentier::utility::IKeychainService, 64
- numEncryptNodes
 - quentier::enml::IHtmlData, 62
- numDecryptNodes
 - quentier::enml::IHtmlData, 62
- numToDoNodes
 - quentier::enml::IHtmlData, 63
- numHyperlinkNodes
 - quentier::enml::IHtmlData, 63
- onReadFileRequest
 - quentier::utility::FileIOProcessorAsync, 39
- onWriteFileRequest
 - quentier::utility::FileIOProcessorAsync, 39
- OperationCanceled.h, 171
- OtherError
 - quentier::utility::IKeychainService, 64
- patchLongDescription
 - quentier::local_storage::IPatch, 79
- patchShortDescription
 - quentier::local_storage::IPatch, 79
- PlatformUtils.h, 262
- Post.h, 221
- print
 - quentier::Account, 18
 - quentier::enml::conversion_rules::ISkipRule, 89
 - quentier::enml::IHtmlData, 63
 - quentier::ErrorString, 34
 - quentier::local_storage::NoteSearchQuery, 142
 - quentier::ResourceRecognitionIndexItem, 150
 - quentier::ResourceRecognitionIndices, 151
 - quentier::utility::ApplicationSettings, 27
- Printable.h, 262
- QPromise< T >, 145

- Qt5Promise.h, 222
- QtFutureContinuations.h, 224
- QtFutureHelpers.h, 230
- quentier::Account, 15
 - displayName, 17
 - evernoteAccountType, 17
 - evernoteHost, 17
 - id, 17
 - isEmpty, 17
 - name, 18
 - print, 18
 - setDisplayName, 18
 - shardId, 18
 - type, 18
- quentier::enml::conversion_rules::ISkipRule, 87
 - AttributeName, 88
 - AttributeValue, 88
 - caseSensitivity, 88
 - Element, 88
 - includeContents, 88
 - matchMode, 88
 - print, 89
 - Target, 88
 - target, 89
 - value, 89
- quentier::enml::conversion_rules::ISkipRuleBuilder, 89
- quentier::enml::IConverter, 48
 - convertEnmlToHtml, 49
 - convertEnmlToPlainText, 49
 - convertEnmlToWordsList, 50
 - convertHtmlToDoc, 50
 - convertHtmlToEnml, 51
 - convertHtmlToXhtml, 51
 - convertHtmlToXml, 51
 - convertPlainTextToWordsList, 52
 - exportNotesToEnex, 52
 - importEnex, 52
 - validateAndFixupEnml, 53
 - validateEnml, 53
- quentier::enml::IDecryptedTextCache, 54
- quentier::enml::IENMLTagsConverter, 58
 - convertDecryptedText, 59
 - convertEncryptedText, 59
 - convertEnToDo, 60
 - convertResource, 60
- quentier::enml::IHtmlData, 61
 - html, 62
 - numEncryptNodes, 62
 - numDecryptNodes, 62
 - numEnToDoNodes, 63
 - numHyperlinkNodes, 63
 - print, 63
- quentier::ErrorString, 33
 - print, 34
- quentier::INoteEditorBackend, 73
- quentier::InvalidArgument, 76
 - exceptionDisplayName, 77
- quentier::IQuentierException, 80
 - print, 81
- quentier::local_storage::ILocalStorage, 66
 - Affiliation, 70
 - Any, 71
 - notifier, 71
 - TagNotesRelation, 70
 - WithNotes, 71
 - WithoutNotes, 71
- quentier::local_storage::ILocalStorage::ListGuidsFilters, 109
- quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions, 109
- quentier::local_storage::ILocalStorage::ListNotebooksOptions, 110
- quentier::local_storage::ILocalStorage::ListNotesOptions, 111
- quentier::local_storage::ILocalStorage::ListObjectsFilters, 112
- quentier::local_storage::ILocalStorage::ListOptionsBase, 113
- quentier::local_storage::ILocalStorage::ListSavedSearchesOptions, 114
- quentier::local_storage::ILocalStorage::ListTagsOptions, 115
- quentier::local_storage::ILocalStorageNotifier, 71
- quentier::local_storage::IPatch, 78
 - apply, 78
 - backupLocalStorage, 78
 - fromVersion, 78
 - patchLongDescription, 79
 - patchShortDescription, 79
 - removeLocalStorageBackup, 79
 - restoreLocalStorageFromBackup, 79
 - toVersion, 80
- quentier::local_storage::LocalStorageOpenException, 116
 - exceptionDisplayName, 117
- quentier::local_storage::LocalStorageOperationException, 117
 - exceptionDisplayName, 118
- quentier::local_storage::NoteSearchQuery, 139
 - notebookModifier, 142
 - print, 142
 - queryString, 142
- quentier::local_storage::tests::mocks::MockILocalStorage, 123
- quentier::NoteEditor, 130
 - backend, 133
 - clear, 133
 - convertToNote, 134
 - currentNoteLocalId, 134
 - defaultFont, 134
 - defaultPalette, 134
 - idleTime, 134
 - inAppNoteLinkPasteRequested, 134
 - initialize, 135
 - isEditorPageModified, 135
 - isModified, 135

- isNoteLoaded, 136
- saveNoteToLocalStorage, 136
- setAccount, 136
- setBackend, 136
- setCurrentNoteLocalId, 136
- setDefaultFont, 137
- setDefaultPalette, 137
- setFocus, 137
- setInitialPageHtml, 137
- setNoteDeletedPageHtml, 138
- setNoteLoadingPageHtml, 138
- setNoteNotFoundPageHtml, 138
- setNoteTitle, 138
- setTagIds, 138
- setUndoStack, 139
- undoStack, 139
- quentier::OperationCanceled, 142
 - exceptionDisplayName, 143
- quentier::ResourceRecognitionIndexItem, 148
 - print, 150
- quentier::ResourceRecognitionIndexItem::IBarcodeItem, 47
- quentier::ResourceRecognitionIndexItem::IObjectItem, 77
- quentier::ResourceRecognitionIndexItem::IShapeItem, 86
- quentier::ResourceRecognitionIndexItem::ITextItem, 108
- quentier::ResourceRecognitionIndices, 150
 - print, 151
- quentier::Result< ValueType, ErrorType, typename >, 152
 - isValid, 152
- quentier::RuntimeError, 153
 - exceptionDisplayName, 154
- quentier::SpellChecker, 158
- quentier::synchronization::AuthenticationExpiredError, 32
- quentier::synchronization::IAuthenticationInfo, 44
 - authenticationTime, 45
 - authToken, 45
 - authTokenExpirationTime, 45
 - noteStoreUrl, 45
 - shardId, 45
 - userId, 46
 - userStoreCookies, 46
 - webApiUrlPrefix, 46
- quentier::synchronization::IAuthenticationInfoBuilder, 46
- quentier::synchronization::IAuthenticator, 47
- quentier::synchronization::IDownloadNotesStatus, 54
- quentier::synchronization::IDownloadResourcesStatus, 56
- quentier::synchronization::INoteStoreFactory, 76
- quentier::synchronization::ISendStatus, 82
 - failedToSendNotebooks, 83
 - failedToSendNotes, 83
 - failedToSendSavedSearches, 83
 - failedToSendTags, 84
- needToRepeatIncrementalSync, 84
- stopSynchronizationError, 84
- totalAttemptedToSendNotebooks, 84
- totalAttemptedToSendNotes, 84
- totalAttemptedToSendSavedSearches, 85
- totalAttemptedToSendTags, 85
- totalSuccessfullySentNotebooks, 85
- totalSuccessfullySentNotes, 85
- totalSuccessfullySentSavedSearches, 86
- totalSuccessfullySentTags, 86
- quentier::synchronization::ISyncChunksDataCounters, 90
 - addedLinkedNotebooks, 91
 - addedNotebooks, 91
 - addedSavedSearches, 91
 - addedTags, 91
 - expungedLinkedNotebooks, 92
 - expungedNotebooks, 92
 - expungedSavedSearches, 92
 - expungedTags, 92
 - totalExpungedLinkedNotebooks, 92
 - totalExpungedNotebooks, 92
 - totalExpungedSavedSearches, 92
 - totalExpungedTags, 93
 - totalLinkedNotebooks, 93
 - totalNotebooks, 93
 - totalSavedSearches, 93
 - totalTags, 93
 - updatedLinkedNotebooks, 93
 - updatedNotebooks, 93
 - updatedSavedSearches, 94
 - updatedTags, 94
- quentier::synchronization::ISyncConflictResolver, 94
- quentier::synchronization::ISyncConflictResolver::ConflictResolution, 32
- quentier::synchronization::ISyncConflictResolver::ConflictResolution::Ignore, 61
- quentier::synchronization::ISyncConflictResolver::ConflictResolution::Move, 129
 - T >, 129
 - mine, 129
- quentier::synchronization::ISyncConflictResolver::ConflictResolution::Use, 160
- quentier::synchronization::ISyncConflictResolver::ConflictResolution::Use, 160
- quentier::synchronization::ISyncEventsNotifier, 95
 - downloadFinished, 96
 - linkedNotebookNotesDownloadProgress, 97
 - linkedNotebookResourcesDownloadProgress, 97
 - linkedNotebookSendStatusUpdate, 97
 - linkedNotebookSyncChunksDataProcessingProgress, 98
 - linkedNotebookSyncChunksDownloaded, 98
 - linkedNotebookSyncChunksDownloadProgress, 98
 - notesDownloadProgress, 99
 - resourcesDownloadProgress, 99
 - startLinkedNotebooksDataDownloading, 99
 - syncChunksDataProcessingProgress, 100
 - syncChunksDownloaded, 100

- syncChunksDownloadProgress, 100
- userOwnSendStatusUpdate, 100
- quentier::synchronization::ISynchronizer, 101
- quentier::synchronization::ISyncOptions, 101
 - downloadNoteThumbnails, 102
 - inkNoteImagesStorageDir, 102
 - maxConcurrentNoteDownloads, 103
 - maxConcurrentResourceDownloads, 103
 - requestContext, 103
 - retryPolicy, 103
- quentier::synchronization::ISyncOptionsBuilder, 104
- quentier::synchronization::ISyncResult, 104
- quentier::synchronization::ISyncState, 105
- quentier::synchronization::ISyncStateBuilder, 106
- quentier::synchronization::ISyncStateStorage, 107
 - notifySyncStateUpdated, 108
- quentier::synchronization::IUserStoreFactory, 108
- quentier::synchronization::RateLimitReachedError, 148
 - rateLimitDurationSec, 148
- quentier::synchronization::tests::mocks::MockIAuthenticator, 121
- quentier::synchronization::tests::mocks::MockINoteStoreFactory, 126
- quentier::synchronization::tests::mocks::MockISyncConflictResolver, 127
- quentier::synchronization::tests::mocks::MockISyncStateStorage, 128
- quentier::threading::detail::ResultTypeHelper< F, Arg, Enable >, 152
- quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > >, 152
- quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > >, 153
- quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > >, 153
- quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > >, 153
- quentier::threading::TrackedTask< LockableObject, Function >, 159
- quentier::utility::ApplicationSettings, 20
 - ~ApplicationSettings, 23
 - ApplicationSettings, 21, 22
 - beginGroup, 23, 24
 - beginReadArray, 24, 25
 - beginWriteArray, 25, 26
 - contains, 26, 27
 - print, 27
 - remove, 27, 28
 - setValue, 28, 29
 - value, 29, 30
- quentier::utility::ApplicationSettings::ArrayCloser, 31
- quentier::utility::ApplicationSettings::GroupCloser, 43
- quentier::utility::cancelers::AnyOfCanceler, 19
 - isCanceled, 20
- quentier::utility::cancelers::FutureCanceler< T >, 42
 - isCanceled, 43
- quentier::utility::cancelers::ICanceler, 47
- quentier::utility::cancelers::ManualCanceler, 120
 - cancel, 121
 - isCanceled, 121
- quentier::utility::EventLoopWithExitStatus, 35
- quentier::utility::FileIOProcessorAsync, 37
 - onReadFileRequest, 39
 - onWriteFileRequest, 39
 - readFileRequestProcessed, 39
 - setIdleTimePeriod, 40
 - writeFileRequestProcessed, 40
- quentier::utility::FileSystemWatcher, 41
- quentier::utility::IEncryptor, 57
 - AES, 57
 - Cipher, 57
 - decrypt, 58
 - encrypt, 58
- quentier::utility::IKeychainService, 63
 - AccessDenied, 64
 - AccessDeniedByUser, 64
 - CouldNotDeleteEntry, 64
 - deletePassword, 65
 - EntryNotFound, 64
 - ErrorCode, 64
 - NoBackendAvailable, 64
 - NoError, 64
 - NotImplemented, 64
 - OtherError, 64
 - readPassword, 65
 - writePassword, 65
- quentier::utility::IKeychainService::Exception, 36
 - exceptionDisplayName, 37
- quentier::utility::LRUCache< Key, Value, Allocator >, 119
- quentier::utility::Printable, 144
- quentier::utility::QuentierApplication, 145
- quentier::utility::QuentierUndoCommand, 146
- quentier::utility::ShortcutManager, 155
 - defaultShortcut, 156
 - shortcut, 157
 - userShortcut, 157
- quentier::utility::StringUtils, 159
- quentier::utility::SysInfo, 159
- quentier::utility::tests::mocks::MockIKeychainService, 122
- quentier::utility::UidGenerator, 160
- QuentierApplication.h, 263
- QuentierLogger.h, 189
- QuentierUndoCommand.h, 264
- queryString
 - quentier::local_storage::NoteSearchQuery, 142

- rateLimitDurationSec
 - quentier::synchronization::RateLimitReachedError, 148
- RC2
 - quentier::utility::IEncryptor, 57
- readFileRequestProcessed
 - quentier::utility::FileIOProcessorAsync, 39
- readPassword
 - quentier::utility::IKeychainService, 65
- RegisterMetatypes.h, 242
- remove
 - quentier::utility::ApplicationSettings, 27, 28
- removeLocalStorageBackup
 - quentier::local_storage::IPatch, 79
- requestContext
 - quentier::synchronization::ISyncOptions, 103
- ResourceRecognitionIndexItem.h, 242
- ResourceRecognitionIndices.h, 244
- resourcesDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 99
- ResourceUtils.h, 245
- restoreLocalStorageFromBackup
 - quentier::local_storage::IPatch, 79
- Result.h, 245
- retryPolicy
 - quentier::synchronization::ISyncOptions, 103
- Runnable.h, 233
- RuntimeError.h, 171
- saveNoteToLocalStorage
 - quentier::NoteEditor, 136
- SendStatus.h, 216
- setAccount
 - quentier::NoteEditor, 136
- setBackend
 - quentier::NoteEditor, 136
- setCurrentNoteLocalId
 - quentier::NoteEditor, 136
- setDefaultFont
 - quentier::NoteEditor, 137
- setDefaultPalette
 - quentier::NoteEditor, 137
- setDisplayName
 - quentier::Account, 18
- setFocus
 - quentier::NoteEditor, 137
- setIdleTimePeriod
 - quentier::utility::FileIOProcessorAsync, 40
- setInitialPageHtml
 - quentier::NoteEditor, 137
- setNoteDeletedPageHtml
 - quentier::NoteEditor, 138
- setNoteLoadingPageHtml
 - quentier::NoteEditor, 138
- setNoteNotFoundPageHtml
 - quentier::NoteEditor, 138
- setNoteTitle
 - quentier::NoteEditor, 138
- setTagIds
 - quentier::NoteEditor, 138
- setUndoStack
 - quentier::NoteEditor, 139
- setValue
 - quentier::utility::ApplicationSettings, 28, 29
- shardId
 - quentier::Account, 18
 - quentier::synchronization::IAuthenticationInfo, 45
- shortcut
 - quentier::utility::ShortcutManager, 157
- ShortcutManager.h, 265
- Size.h, 267
- SpellChecker.h, 196
- StandardPaths.h, 267
- startLinkedNotebooksDataDownloading
 - quentier::synchronization::ISyncEventsNotifier, 99
- stopSynchronizationError
 - quentier::synchronization::ISendStatus, 84
- StringUtils.h, 268
- SuppressWarnings.h, 268
- SyncChunksDataCounters.h, 217
- syncChunksDataProcessingProgress
 - quentier::synchronization::ISyncEventsNotifier, 100
- syncChunksDownloaded
 - quentier::synchronization::ISyncEventsNotifier, 100
- syncChunksDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 100
- SyncResult.h, 217
- SyncState.h, 218
- SysInfo.h, 269
- TagNotesRelation
 - quentier::local_storage::ILocalStorage, 70
- TagSortByParentChildRelations.h, 270
- Target
 - quentier::enml::conversion_rules::ISkipRule, 88
- target
 - quentier::enml::conversion_rules::ISkipRule, 89
- totalAttemptedToSendNotebooks
 - quentier::synchronization::ISendStatus, 84
- totalAttemptedToSendNotes
 - quentier::synchronization::ISendStatus, 84
- totalAttemptedToSendSavedSearches
 - quentier::synchronization::ISendStatus, 85
- totalAttemptedToSendTags
 - quentier::synchronization::ISendStatus, 85
- totalExpungedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, 92
- totalExpungedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, 92
- totalExpungedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters, 92
- totalExpungedTags

- quentier::synchronization::ISyncChunksDataCounters::value
 - 93
- totalLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters,
 - 93
- totalNotebooks
 - quentier::synchronization::ISyncChunksDataCounters,
 - 93
- totalSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters,
 - 93
- totalSuccessfullySentNotebooks
 - quentier::synchronization::ISendStatus, 85
- totalSuccessfullySentNotes
 - quentier::synchronization::ISendStatus, 85
- totalSuccessfullySentSavedSearches
 - quentier::synchronization::ISendStatus, 86
- totalSuccessfullySentTags
 - quentier::synchronization::ISendStatus, 86
- totalTags
 - quentier::synchronization::ISyncChunksDataCounters,
 - 93
- toVersion
 - quentier::local_storage::IPatch, 80
- TrackedTask.h, 233
- type
 - quentier::Account, 18
- UidGenerator.h, 271
- undoStack
 - quentier::NoteEditor, 139
- Unreachable.h, 272
- updatedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters,
 - 93
- updatedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters,
 - 93
- updatedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters,
 - 94
- updatedTags
 - quentier::synchronization::ISyncChunksDataCounters,
 - 94
- userId
 - quentier::synchronization::IAuthenticationInfo, 46
- userOwnSendStatusUpdate
 - quentier::synchronization::ISyncEventsNotifier,
 - 100
- userShortcut
 - quentier::utility::ShortcutManager, 157
- userStoreCookies
 - quentier::synchronization::IAuthenticationInfo, 46
- validateAndFixupEnml
 - quentier::enml::IConverter, 53
- validateEnml
 - quentier::enml::IConverter, 53
- Validation.h, 248
- quentier::enml::conversion_rules::ISkipRule, 89
 - quentier::utility::ApplicationSettings, 29, 30
 - webApiUrlPrefix
 - quentier::synchronization::IAuthenticationInfo, 46
 - WithNotes
 - quentier::local_storage::ILocalStorage, 71
 - WithoutNotes
 - quentier::local_storage::ILocalStorage, 71
 - writeFileRequestProcessed
 - quentier::utility::FileIOProcessorAsync, 40
 - writePassword
 - quentier::utility::IKeychainService, 65