
GroupDocs Python SDK Documentation

Release 1.4.0

GroupDocs

October 23, 2015

1	Requirements	1
2	Installation	3
3	Usage Example	5
4	Contents:	7
4.1	Auto Generated documentation	7
5	Sign, Manage, Annotate, Assemble, Compare and Convert Documents with GroupDocs	49
	Python Module Index	51

Requirements

- SDK requires Python 2.6 (or later). If you need Python 3 version of SDK please go to [groupdocs-python3](#)

Installation

You can use the [Pip](#) to download and install SDK. GroupDocs SDK is now in [PyPi](#).

Usage Example

```
apiClient = ApiClient(GroupDocsRequestSigner(privateKey))
api = AntApi(apiClient)
response = api.ListAnnotations(userId, fileId)
```


4.1 Auto Generated documentation

4.1.1 AntApi Module

class groupdocs.AntApi.**AntApi** (*ApiClient*)

Bases: object

AddAnnotationCollaborator (*userId, fileId, **kwargs*)

Add an annotation collaborator

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required) *body*, ReviewerInfo: Reviewer Info (optional)

Returns: AddCollaboratorResponse

CreateAnnotation (*userId, fileId, body, **kwargs*)

Create annotation

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required) *body*, AnnotationInfo: annotation (required)

Returns: CreateAnnotationResponse

CreateAnnotationReply (*userId, annotationId, body, **kwargs*)

Create annotation reply

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *body*, AnnotationReplyInfo: Reply (required)

Returns: AddReplyResponse

DeleteAnnotation (*userId, annotationId, **kwargs*)

Delete annotation

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required)

Returns: DeleteAnnotationResponse

DeleteAnnotationReply (*userId, replyGuid, **kwargs*)

Delete annotation reply

Args: *userId*, str: User GUID (required) *replyGuid*, str: Reply GUID (required)

Returns: DeleteReplyResponse

DeleteDocumentAnnotations (*userId, fileId, **kwargs*)

Delete document annotations

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required)

Returns: DeleteDocumentAnnotationsResponse

DeleteDocumentReviewer (*userId, fileId, reviewerId, **kwargs*)

Delete document reviewer

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required) *reviewerId*, str: Reviewer ID (required)

Returns: AddCollaboratorResponse

EditAnnotationReply (*userId, replyGuid, body, **kwargs*)

Edit annotation reply

Args: *userId*, str: User GUID (required) *replyGuid*, str: Reply GUID (required) *body*, str: Message (required)

Returns: EditReplyResponse

GetAnnotationCollaborators (*userId, fileId, **kwargs*)

Get list of annotation collaborators

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required)

Returns: GetCollaboratorsResponse

GetReviewerContacts (*userId, **kwargs*)

Get list of reviewer contacts

Args: *userId*, str: User GUID (required)

Returns: GetReviewerContactsResponse

GetSharedLinkAccessRights (*userId, fileId, **kwargs*)

Get Shared Link Access Rights

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required)

Returns: GetSharedLinkAccessRightsResponse

ListAnnotationReplies (*userId, annotationId, after, **kwargs*)

Get list of annotation replies

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *after*, long: After (required)

Returns: ListRepliesResponse

ListAnnotations (*userId, fileId, **kwargs*)

Get list of annotations

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required)

Returns: ListAnnotationsResponse

MoveAnnotation (*userId, annotationId, body, **kwargs*)

Move annotation

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *body*, Point: position (required)

Returns: MoveAnnotationResponse

MoveAnnotationMarker (*userId, annotationId, body, **kwargs*)

Move Annotation Marker

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *body*, MarkerPosition: Annotation marker position (required)

Returns: MoveAnnotationResponse

ResizeAnnotation (*userId, annotationId, body, **kwargs*)

Resize annotation

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *body*, AnnotationSizeInfo: position (required)

Returns: ResizeAnnotationResponse

SaveTextField (*userId, annotationId, body, **kwargs*)

Save Text Of Text Field

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *body*, TextFieldInfo: Text (required)

Returns: SaveAnnotationTextResponse

SetAnnotationAccess (*userId, annotationId, body, **kwargs*)

Set Annotation Access

Args: *userId*, str: User GUID (required) *annotationId*, str: Annotation ID (required) *body*, int: Annotation Access (required)

Returns: SetAnnotationAccessResponse

SetAnnotationCollaborators (*userId, fileId, version, **kwargs*)

Set annotation collaborators

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required) *version*, str: Annotation version (required) *body*, List[str]: Collaborators (optional)

Returns: SetCollaboratorsResponse

SetReviewerContacts (*userId, **kwargs*)

Get list of reviewer contacts

Args: *userId*, str: User GUID (required) *body*, List[ReviewerContactInfo]: Reviewer Contacts Array (optional)

Returns: GetReviewerContactsResponse

SetReviewerRights (*userId, fileId, body, **kwargs*)

Set Reviewer Rights

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required) *body*, List[ReviewerInfo]: Collaborators (required)

Returns: SetReviewerRightsResponse

SetSessionCallbackUrl (*userId, fileId, body, **kwargs*)

Set Session Web Hook Callback Url

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *body*, str: Callback Url (required)

Returns: SetSessionCallbackUrlResponse

SetSharedLinkAccessRights (*userId, fileId, body, **kwargs*)

Set Shared Link Access Rights

Args: `userId`, str: User GUID (required) `fileId`, str: File ID (required) `body`, str: Access Rights for the collaborate link (required)

Returns: `SetSharedLinkAccessRightsResponse`

SetTextfieldColor (*userId, annotationId, body, **kwargs*)
Save Text Of Text Field

Args: `userId`, str: User GUID (required) `annotationId`, str: Annotation ID (required) `body`, int: Font Color (required)

Returns: `SaveAnnotationTextResponse`

basePath

4.1.2 ApiClient Module

class `groupdocs.ApiClient.ApiClient` (*requestSigner=None*)
Bases: `object`

Generic API client for Swagger client library builds

addHeaders (***headers*)

callAPI (*apiServer, resourcePath, method, queryParams, postData, headerParams=None, returnType=<type 'str'>*)

deserialize (*obj, objClass*)

Derialize a JSON string into an object.

Args: `obj` – string or object to be deserialized `objClass` – class literal for deserialzied object, or string of class name

Returns: object – deserialized object

static encodeURI (*url*)

static encodeURIComponent (*url*)

static readAsDataURL (*filePath*)

sanitizeForSerialization (*obj*)

Dump an object into JSON for POSTing.

setDebug (*flag, logFilepath=None*)

toPathValue (*obj*)

Serialize a list to a CSV string, if necessary.

Args: `obj` – data object to be serialized

Returns: string – json serialization of object

exception `groupdocs.ApiClient.ApiException` (*code, *args*)
Bases: `exceptions.Exception`

class `groupdocs.ApiClient.DefaultRequestSigner`
Bases: `groupdocs.ApiClient.RequestSigner`

signContent (*requestBody, headers*)

signUrl (*url*)

class `groupdocs.ApiClient.MethodRequest` (**args, **kwargs*)
Bases: `urllib2.Request`

`get_method()`

class `groupdocs.ApiClient.RequestSigner`

Bases: `object`

signContent (*requestBody, headers*)

signUrl (*url*)

4.1.3 AsyncApi Module

class `groupdocs.AsyncApi.AsyncApi` (*apiClient*)

Bases: `object`

AddJobDocument (*userId, jobId, fileId, checkOwnership, **kwargs*)

Add job document

Args: `userId`, str: User GUID (required) `jobId`, str: Job id or guid (required) `fileId`, str: File GUID (required) `checkOwnership`, bool: Check Document Ownership (required) `formats`, str: Formats (optional)

Returns: `AddJobDocumentResponse`

AddJobDocumentUrl (*userId, jobId, absoluteUrl, **kwargs*)

Add job document url

Args: `userId`, str: User GUID (required) `jobId`, str: Job id (required) `absoluteUrl`, str: Absolute Url (required) `formats`, str: Formats (optional)

Returns: `AddJobDocumentResponse`

Convert (*userId, fileId, **kwargs*)

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `emailResults`, str: Email results (optional) `description`, str: Description (optional) `printScript`, bool: Print (optional) `callbackUrl`, str: Callback url (optional) `new_type`, str: Target type (optional)

Returns: `ConvertResponse`

CreateJob (*userId, body, **kwargs*)

Create job

Args: `userId`, str: User GUID (required) `body`, `JobInfo`: Job (required)

Returns: `CreateJobResponse`

DeleteJob (*userId, jobGuid, **kwargs*)

Delete draft job

Args: `userId`, str: User GUID (required) `jobGuid`, str: Job Guid (required)

Returns: `DeleteResult`

DeleteJobDocument (*userId, jobGuid, documentId, **kwargs*)

Delete document from job

Args: `userId`, str: User GUID (required) `jobGuid`, str: Job Guid (required) `documentId`, str: Document GUID (required)

Returns: `DeleteResponse`

GetJobDocuments (*userId, jobId, **kwargs*)

Get job documents

Args: `userId`, str: User GUID (required) `jobId`, str: Job id or guid (required) `format`, str: Format (optional)

Returns: GetJobDocumentsResponse

GetJobJson (*userId, jobId, **kwargs*)

Get job json

Args: *userId*, str: User GUID (required) *jobId*, str: Job id (required)

Returns: GetJobResponse

GetJobResources (*userId, statusIds, **kwargs*)

Get job resources

Args: *userId*, str: User GUID (required) *statusIds*, str: Comma separated job status identifiers (required) *actions*, str: Actions (optional) *excludedActions*, str: Excluded actions (optional)

Returns: GetJobResourcesResponse

GetJobs (*userId, **kwargs*)

Get jobs

Args: *userId*, str: User GUID (required) *pageIndex*, str: Page Index (optional) *pageSize*, str: Page Size (optional) *datetime*, str: Date (optional) *statusIds*, str: Comma separated status identifiers (optional) *actions*, str: Actions (optional) *excludedActions*, str: Excluded actions (optional) *jobName*, str: Filtered job name (optional) *orderBy*, str: Sorted column name (optional) *orderAsc*, bool: Order ASC (optional)

Returns: GetJobsResponse

GetJobsDocuments (*userId, **kwargs*)

Get jobs documents

Args: *userId*, str: User GUID (required) *pageIndex*, str: Page Index (optional) *pageSize*, str: Page Size (optional) *actions*, str: Actions (optional) *excludedActions*, str: Excluded actions (optional) *orderBy*, str: Order by (optional) *orderAsc*, bool: Order asc (optional)

Returns: GetJobsDocumentsResponse

GetPossibleConversions (*userId, fileExt, **kwargs*)

Get Possible Conversions

Args: *userId*, str: User GUID (required) *fileExt*, str: File extension to check (required)

Returns: GetPossibleConversions

UpdateJob (*userId, jobId, body, **kwargs*)

Update job

Args: *userId*, str: User GUID (required) *jobId*, str: Job id or Guid (required) *body*, JobInfo: Job (required)

Returns: UpdateJobResponse

basePath

4.1.4 ComparisonApi Module

class groupdocs.ComparisonApi.**ComparisonApi** (*apiClient*)

Bases: object

Compare (*userId, sourceFileId, targetFileId, callbackUrl, **kwargs*)

Args: *userId*, str: User GUID (required) *sourceFileId*, str: Source File GUID (required) *targetFileId*, str: Target File GUID (required) *callbackUrl*, str: Callback Url (required)

Returns: CompareResponse

DownloadResult (*resultFileId*, ***kwargs*)

Download comparison result file

Args: resultFileId, str: Comparison result file GUID (required) format, str: Comparison result file format (optional)

Returns: stream

GetChanges (*resultFileId*, ***kwargs*)

Get changes

Args: resultFileId, str: Comparison result file GUID (required)

Returns: ChangesResponse

GetDocumentDetails (*userId*, *guid*, ***kwargs*)

Get document details

Args: userId, str: User GUID (required) guid, str: Document GUID (required)

Returns: DocumentDetailsResponse

UpdateChanges (*resultFileId*, *body*, ***kwargs*)

Update changes

Args: resultFileId, str: Comparison result file GUID (required) body, List[ChangeInfo]: Comparison changes to update (accept or reject) (required)

Returns: ChangesResponse

basePath

4.1.5 DocApi Module

class groupdocs.DocApi.DocApi (*apiClient*)

Bases: object

GetDocumentAccessInfo (*userId*, *fileId*, ***kwargs*)

Get document access info

Args: userId, str: User GUID (required) fileId, str: File GUID (required)

Returns: DocumentAccessInfoResponse

GetDocumentContent (*userId*, *fileId*, *contentType*, ***kwargs*)

Returns document content

Args: userId, str: GroupDocs user global unique identifier. (required) fileId, str: Document global unique identifier. (required) contentType, str: Content type. (required)

Returns: GetDocumentContentResponse

GetDocumentFormats (*userId*, *fileId*, ***kwargs*)

Get document formats

Args: userId, str: User GUID (required) fileId, str: File GUID (required)

Returns: GetDocumentForeignTypesResponse

GetDocumentHyperlinks (*userId*, *fileId*, ***kwargs*)

Returns document hyperlinks.

Args: userId, str: GroupDocs user global unique identifier. (required) fileId, str: Document global unique identifier. (required)

Returns: GetHyperlinksResponse

GetDocumentMetadata (*userId, fileId, **kwargs*)

Get document metadata

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required)

Returns: GetDocumentInfoResponse

GetDocumentMetadataByPath (*userId, path, **kwargs*)

Returns document metadata

Args: *userId*, str: User GUID (required) *path*, str: File path to return metadata for (required)

Returns: GetDocumentInfoResponse

GetDocumentPageImage (*userId, fileId, pageNumber, dimension, **kwargs*)

Returns a stream of bytes representing a particular document page image.

Args: *userId*, str: GroupDocs user global unique identifier. (required) *fileId*, str: Document global unique identifier. (required) *pageNumber*, int: Document page number to get image for. (required) *dimension*, str: Image dimension in format ‘<width>x<height>’ (one or both values can be omitted). (required) *quality*, int: Image quality in range 1-100. (optional) *usePdf*, bool: A flag indicating whether a document should be converted to PDF format before generating the image. (optional) *expiresOn*, bool: The date and time in milliseconds since epoch the URL expires. (optional)

Returns: stream

GetDocumentPagesImageUrls (*userId, fileId, dimension, **kwargs*)

Returns a list of URLs pointing to document page images.

Args: *userId*, str: GroupDocs user global unique identifier. (required) *fileId*, str: Document global unique identifier. (required) *firstPage*, int: Document page number to start from. (optional) *pageCount*, int: Page count to return URLs for. (optional) *dimension*, str: Image dimension in format ‘<width>x<height>’ (one or both values can be omitted). (required) *quality*, int: Image quality in range 1-100. (optional) *usePdf*, bool: A flag indicating whether a document should be converted to PDF format before generating the image. (optional) *token*, str: A document viewer session token returned by the View Document request. (optional)

Returns: GetImageUrlsResponse

GetDocumentTags (*userId, fileId, **kwargs*)

Returns tags assigned to the document.

Args: *userId*, str: GroupDocs user global unique identifier. (required) *fileId*, str: Document global unique identifier. (required)

Returns: GetTagsResponse

GetDocumentViews (*userId, **kwargs*)

Get Document Views

Args: *userId*, str: User GUID (required) *startIndex*, str: A historical view entry to start from. Historical entries are sorted from the recent to old ones (optional) *pageSize*, str: The total number of requested entries. If *pageSize* is equal to -1, all available entries will be returned (optional)

Returns: DocumentViewsResponse

GetEditLock (*userId, fileId, **kwargs*)

Locks a document for editing and returns editing metadata.

Args: *userId*, str: GroupDocs user global unique identifier. (required) *fileId*, str: Document global unique identifier. (required)

Returns: GetEditLockResponse

GetFolderSharers (*userId, folderId, **kwargs*)

Get folder sharers

Args: *userId*, str: User GUID (required) *folderId*, str: Folder Id (required)

Returns: SharedUsersResponse

GetSharedDocuments (*userId, sharesTypes, **kwargs*)

Get shared documents

Args: *userId*, str: User GUID (required) *sharesTypes*, str: Shares types (optional) *pageIndex*, str: Page index (optional) *pageSize*, str: Page size (optional) *orderBy*, str: Order by (optional) *orderAsc*, bool: Order asc (optional)

Returns: SharedDocumentsResponse

GetTemplateEditorFields (*userId, fileId, **kwargs*)

Get template editor fields

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required)

Returns: TemplateEditorFieldsResponse

GetTemplateFields (*userId, fileId, **kwargs*)

Get template fields

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *includeGeometry*, bool: Include geometry (optional)

Returns: TemplateFieldsResponse

RemoveDocumentTags (*userId, fileId, **kwargs*)

Removes tags assigned to the document.

Args: *userId*, str: GroupDocs user global unique identifier. (required) *fileId*, str: Document global unique identifier. (required)

Returns: RemoveTagsResponse

RemoveEditLock (*userId, fileId, lockId, **kwargs*)

Removes edit lock for a document and replaces the document with its edited copy.

Args: *userId*, str: GroupDocs user global unique identifier. (required) *fileId*, str: Document global unique identifier. (required) *lockId*, str: Lock Id. (required)

Returns: RemoveEditLockResponse

SetDocumentAccessMode (*userId, fileId, **kwargs*)

Set document access mode

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *mode*, str: Mode (optional)

Returns: DocumentAccessInfoResponse

SetDocumentPassword (*userId, fileId, body, **kwargs*)

Set document password

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *body*, str: Password (required)

Returns: SetDocumentPasswordResponse

SetDocumentTags (*userId, fileId, **kwargs*)

Assign tags to the document.

Args: `userId`, str: GroupDocs user global unique identifier. (required) `fileId`, str: Document global unique identifier. (required)

Returns: `SetTagsResponse`

SetDocumentUserStatus (*userId, fileId, status, **kwargs*)

Set document user status

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `status`, str: Status (required)

Returns: `DocumentUserStatusResponse`

ShareDocument (*userId, fileId, body, **kwargs*)

Share document

Args: `userId`, str: User GUID (required) `fileId`, str: File ID - decimal type (required) `body`, List[str]: Sharers (required)

Returns: `SharedUsersResponse`

ShareFolder (*userId, folderId, body, **kwargs*)

Share folder

Args: `userId`, str: User GUID (required) `folderId`, str: Folder Id (required) `body`, List[str]: Sharers (required)

Returns: `SharedUsersResponse`

UnshareDocument (*userId, fileId, **kwargs*)

Unshare document

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required)

Returns: `SharedUsersResponse`

UnshareFolder (*userId, folderId, **kwargs*)

Unshare folder

Args: `userId`, str: User GUID (required) `folderId`, str: Folder Id (required)

Returns: `SharedUsersResponse`

ViewDocument (*userId, fileId, **kwargs*)

View Document

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `pageNumber`, str: Page Number (optional) `pageCount`, str: Page Count (optional) `width`, str: Width (optional) `quality`, str: Quality (optional) `usePdf`, str: Use Pdf (optional) `passwordSalt`, str: A password hash for password protected documents (optional)

Returns: `ViewDocumentResponse`

basePath

4.1.6 FileStream Module

class `groupdocs.FileStream.FileStream` (*filePath=None, response=None, stream=None*)

Bases: `object`

This class encapsulates data needed either for file upload or download. All properties are initialized lazily on first access.

To use this class for file upload initialize it with absolute path to file on your filesystem.

To use this class for file download call `fromHttp(response)` method.

contentType**fileName****classmethod fromFile** (*filePath*)

filePath is an absolute path to file on your filesystem.

classmethod fromHttp (*response*)

response is a file-like object with two additional methods: geturl() and info()

classmethod fromStream (*stream, size, contentType='application/octet-stream'*)

stream is a file-like object, i.e. stream = fopen(filename)

inputStream

returns file object

size

4.1.7 GroupDocsRequestSigner Module

class groupdocs.GroupDocsRequestSigner.**GroupDocsRequestSigner** (*privateKey*)Bases: *groupdocs.ApiClient.RequestSigner***signContent** (*requestBody, headers*)**signUrl** (*url*)

4.1.8 MergeApi Module

class groupdocs.MergeApi.**MergeApi** (*apiClient*)

Bases: object

AddDataSource (*userId, body, **kwargs*)

Add datasource

Args: userId, str: User GUID (required) body, Datasource: Datasource (required)

Returns: AddDatasourceResponse

AddDocumentQuestionnaire (*userId, fileId, questionnaireId, **kwargs*)

Add document questionnaire

Args: userId, str: User GUID (required) fileId, str: File GUID (required) questionnaireId, str: Questionnaire id (required)

Returns: AddDocumentQuestionnaireResponse

AddJobDocumentDataSource (*userId, jobId, fileId, datasourceId, **kwargs*)

Add job document datasource

Args: userId, str: User GUID (required) jobId, int: Job identifier (required) fileId, int: File identifier (required) datasourceId, int: Datasource identifier (required)

Returns: AddDocumentDataSourceResponse

AddJobDocumentDataSourceFields (*userId, jobId, fileId, body, **kwargs*)

Add job document datasource fields

Args: userId, str: User GUID (required) jobId, str: Job id (required) fileId, str: File GUID (required) body, List[DataSourceField]: Fields (required)

Returns: AddDocumentDataSourceResponse

AddQuestionnaireCollector (*userId, questionnaireId, body, **kwargs*)

Add questionnaire collector

Args: *userId*, str: User global unique identifier (required) *questionnaireId*, str: Questionnaire identifier (required) *body*, QuestionnaireCollectorInfo: Questionnaire collector to be added (required)

Returns: AddQuestionnaireCollectorResponse

AddQuestionnaireExecution (*userId, collectorId, body, **kwargs*)

Add questionnaire execution

Args: *userId*, str: User GUID (required) *collectorId*, str: Questionnaire collector id (required) *body*, QuestionnaireExecutionInfo: Execution (required)

Returns: AddQuestionnaireExecutionResponse

AddQuestionnaireTemplateFields (*userId, fileId, body, **kwargs*)

Add template editor fields to the specific document

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *body*, List[TemplateEditorField]: Array of template fields which should be added (required)

Returns: GetQuestionnaireTemplateResponse

CopyFileToTemplates (*userId, path, **kwargs*)

Copy file to template

Args: *userId*, str: User GUID (required) *path*, str: Path (required) *mode*, str: Mode (optional) Groupdocs_Copy, str: File ID (copy) (optional) Groupdocs_Move, str: File ID (move) (optional)

Returns: GetTemplatesResponse

CreateDocumentQuestionnaire (*userId, fileId, body, **kwargs*)

Create document questionnaire

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *body*, QuestionnaireInfo: Questionnaire (required)

Returns: AddDocumentQuestionnaireResponse

CreateQuestionnaire (*userId, body, **kwargs*)

Create questionnaire

Args: *userId*, str: User GUID (required) *body*, QuestionnaireInfo: Questionnaire (required)

Returns: CreateQuestionnaireResponse

CreateQuestionnaireTemplate (*userId, fileId, body, **kwargs*)

Create questionnaire template from file

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *body*, List[TemplateEditorField]: Array of template fields (required)

Returns: CreateQuestionnaireTemplateResponse

DecorateQuestionnaireCollector (*userId, collectorId, body, **kwargs*)

Decorate questionnaire collector

Args: *userId*, str: User global unique identifier (required) *collectorId*, str: Questionnaire collector global unique identifier (required) *body*, QuestionnaireCollectorStyle: Questionnaire collector style (required)

Returns: AddQuestionnaireCollectorStyleResponse

DeleteDataSource (*userId, datasourceId, **kwargs*)

Delete datasource fields

Args: `userId`, str: User GUID (required) `datasourceId`, str: Datasource id (required)

Returns: `DeleteDatasourceResponse`

DeleteDataSourceList (*userId*, ***kwargs*)

Delete list of datasource fields

Args: `userId`, str: User GUID (required) `body`, List[str]: List of Id (optional)

Returns: `DeleteDatasourceListResponse`

DeleteDocumentQuestionnaire (*userId*, *fileId*, *questionnaireId*, ***kwargs*)

Delete document questionnaire

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `questionnaireId`, str: Questionnaire id (required)

Returns: `DeleteDocumentQuestionnaireResponse`

DeleteQuestionnaire (*userId*, *questionnaireId*, ***kwargs*)

Delete questionnaire

Args: `userId`, str: User GUID (required) `questionnaireId`, str: Questionnaire id (required)

Returns: `DeleteQuestionnaireResponse`

DeleteQuestionnaireCollector (*userId*, *collectorId*, ***kwargs*)

Removes questionnaire collector

Args: `userId`, str: User global unique identifier (required) `collectorId`, str: Questionnaire collector global unique identifier (required)

Returns: `DeleteQuestionnaireCollectorResponse`

DeleteQuestionnaireCollectorList (*userId*, ***kwargs*)

Removes questionnaire collector

Args: `userId`, str: User global unique identifier (required) `body`, List[str]: Collector's GUID list (optional)

Returns: `DeleteQuestionnaireCollectorListResponse`

DeleteQuestionnaireExecution (*userId*, *executionId*, ***kwargs*)

Removes questionnaire execution

Args: `userId`, str: User GUID (required) `executionId`, str: Execution id (required)

Returns: `DeleteQuestionnaireExecutionResponse`

DeleteQuestionnaireExecutionList (*userId*, ***kwargs*)

Removes questionnaire execution

Args: `userId`, str: User GUID (required) `body`, List[str]: List of GUID (optional)

Returns: `DeleteQuestionnaireExecutionListResponse`

DeleteQuestionnaireTemplateFields (*userId*, *fileId*, *body*, ***kwargs*)

Delete template's fields

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `body`, List[TemplateEditorField]: Array of template fields which should be deleted (required)

Returns: `GetQuestionnaireTemplateResponse`

DeleteQuestionnairesList (*userId*, ***kwargs*)

Delete list of questionnaires by GUIDs

Args: `userId`, str: User GUID (required) `body`, List[str]: List of GUID (optional)

Returns: DeleteQuestionnaireListResponse

FillExecution (*userId, executionId, datasourceId, **kwargs*)

Scheduled questionnaire execution fulfillment job

Args: *userId*, str: User global unique identifier (required) *executionId*, str: Execution global unique identifier to fill (required) *datasourceId*, str: Datasource identifier (required) *targetType*, str: Filled document type (optional) *emailResults*, str: Email results (optional) *callbackUrl*, str: Callback url (optional)

Returns: MergeTemplateResponse

FillQuestionnaire (*userId, collectorId, datasourceId, **kwargs*)

Merge datasource

Args: *userId*, str: User global unique identifier (required) *collectorId*, str: Collector global unique identifier to fill (required) *datasourceId*, str: Datasource identifier (required) *targetType*, str: Filled document type (optional) *emailResults*, str: Email results (optional) *callbackUrl*, str: Callback url (optional)

Returns: MergeTemplateResponse

GetDataSource (*userId, datasourceId, **kwargs*)

Get datasource

Args: *userId*, str: User GUID (required) *datasourceId*, str: Datasource id (required) *fields*, str: Fields (optional)

Returns: GetDataSourceResponse

GetDocumentByQuestionnaire (*userId, questionnaireId, **kwargs*)

Get associated document by questionnaire

Args: *userId*, str: User GUID (required) *questionnaireId*, str: Questionnaire GUID (required)

Returns: GetQuestionnaireDocumentResponse

GetDocumentQuestionnaires (*userId, fileId, **kwargs*)

Get document questionnaires

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required)

Returns: GetDocumentQuestionnairesResponse

GetQuestionnaire (*userId, questionnaireId, **kwargs*)

Get questionnaire

Args: *userId*, str: User GUID (required) *questionnaireId*, str: Questionnaire id (required)

Returns: GetQuestionnaireResponse

GetQuestionnaireByCollector (*userId, collectorId, **kwargs*)

Get questionnaire by collector guid

Args: *userId*, str: User GUID (required) *collectorId*, str: Collector GUID (required)

Returns: GetQuestionnaireResponse

GetQuestionnaireCollector (*userId, collectorId, **kwargs*)

Returns questionnaire collector

Args: *userId*, str: User global unique identifier (required) *collectorId*, str: Questionnaire collector global unique identifier (required)

Returns: GetQuestionnaireCollectorResponse

GetQuestionnaireCollectorExecutions (*userId, collectorId, **kwargs*)

Get questionnaire collector executions

Args: *userId*, str: User GUID (required) *collectorId*, str: Questionnaire collector global unique identifier (required)

Returns: GetQuestionnaireExecutionsResponse

GetQuestionnaireCollectorStyle (*userId, collectorId, **kwargs*)

Returns questionnaire collector style

Args: *userId*, str: User global unique identifier (required) *collectorId*, str: Questionnaire collector global unique identifier (required)

Returns: GetQuestionnaireCollectorStyleResponse

GetQuestionnaireCollectors (*userId, questionnaireId, orderBy, isAsc, **kwargs*)

Get questionnaire collectors

Args: *userId*, str: User global unique identifier (required) *questionnaireId*, str: Questionnaire identifier (required) *orderBy*, str: Order by column (required) *isAsc*, bool: Order by ascending or descending (required)

Returns: GetQuestionnaireCollectorsResponse

GetQuestionnaireDataSources (*userId, questionnaireId, **kwargs*)

Get questionnaire datasources

Args: *userId*, str: User GUID (required) *questionnaireId*, str: QuestionnaireId id (required) *includeFields*, bool: Include fields (optional)

Returns: GetDatasourcesResponse

GetQuestionnaireExecution (*userId, executionId, **kwargs*)

Get questionnaire execution

Args: *userId*, str: User GUID (required) *executionId*, str: Questionnaire execution global unique identifier (required)

Returns: GetQuestionnaireExecutionResponse

GetQuestionnaireExecutions (*userId, questionnaireId, **kwargs*)

Get questionnaire executions

Args: *userId*, str: User GUID (required) *questionnaireId*, str: Questionnaire global unique identifier (required)

Returns: GetQuestionnaireExecutionsResponse

GetQuestionnaireFields (*userId, questionnaireId, **kwargs*)

Get questionnaire fields

Args: *userId*, str: User global unique identifier (required) *questionnaireId*, str: Questionnaire global unique identifier (required) *includeGeometry*, bool: A flag indicating whether fields coordinates and size should be included into the response. (optional)

Returns: TemplateFieldsResponse

GetQuestionnaireMetadata (*userId, questionnaireId, **kwargs*)

Get questionnaire metadata

Args: *userId*, str: User global unique identifier (required) *questionnaireId*, str: Questionnaire global unique identifier (required)

Returns: GetQuestionnaireMetadataResponse

GetQuestionnaires (*userId, **kwargs*)

Get questionnaires

Args: *userId*, str: User GUID (required) *status*, str: Questionnaire status to filter by (optional) *pageNumber*, int: Page number to return questionnaires on (optional) *pageSize*, int: Number of questionnaires to return (optional) *orderBy*, str: Order by column (optional) *isAscending*, bool: Order by ascending or descending (optional)

Returns: GetQuestionnairesResponse

GetQuestionnairesByName (*userId, **kwargs*)

Get questionnaires

Args: *userId*, str: User GUID (required) *name*, str: Questionnaire name (optional) *status*, str: Questionnaire status to filter by (optional) *pageNumber*, int: Page number to return questionnaires on (optional) *pageSize*, int: Number of questionnaires to return (optional) *orderBy*, str: Order by column (optional) *isAscending*, bool: Order by ascending or descending (optional)

Returns: GetQuestionnairesResponse

GetTemplates (*userId, **kwargs*)

Get template documents information

Args: *userId*, str: User global unique identifier (required) *orderBy*, str: Order by column (optional) *isAscending*, bool: Order by ascending or descending (optional)

Returns: GetTemplatesResponse

MergeDatasource (*userId, fileId, datasourceId, **kwargs*)

Merge datasource

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *datasourceId*, str: Datasource id (required) *targetType*, str: Target type (optional) *emailResults*, str: Email results (optional)

Returns: MergeTemplateResponse

MergeDatasourceFields (*userId, fileId, body, **kwargs*)

Merge datasource fields

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required) *targetType*, str: Target type (optional) *emailResults*, str: Email results (optional) *assemblyName*, str: Assembly name (optional) *body*, List[DatasourceField]: Fields (required)

Returns: MergeTemplateResponse

UpdateDataSource (*userId, datasourceId, body, **kwargs*)

Update datasource

Args: *userId*, str: User GUID (required) *datasourceId*, str: Datasource id (required) *body*, Datasource: Datasource (required)

Returns: AddDatasourceResponse

UpdateDataSourceFields (*userId, datasourceId, body, **kwargs*)

Update datasource fields

Args: *userId*, str: User GUID (required) *datasourceId*, str: Datasource id (required) *body*, Datasource: Datasource (required)

Returns: AddDatasourceResponse

UpdateQuestionnaire (*userId, questionnaireId, body, **kwargs*)

Update questionnaire

Args: `userId`, str: User GUID (required) `questionnaireId`, str: Questionnaire id (required) `body`, `QuestionnaireInfo`: Questionnaire (required)

Returns: `UpdateQuestionnaireResponse`

UpdateQuestionnaireCollector (*userId, collectorId, body, **kwargs*)

Update questionnaire collector

Args: `userId`, str: User global unique identifier (required) `collectorId`, str: Questionnaire collector global unique identifier (required) `body`, `QuestionnaireCollectorInfo`: Questionnaire collector data (required)

Returns: `UpdateQuestionnaireCollectorResponse`

UpdateQuestionnaireExecution (*userId, executionId, body, **kwargs*)

Update questionnaire execution

Args: `userId`, str: User GUID (required) `executionId`, str: Execution id (required) `body`, `QuestionnaireExecutionInfo`: Execution (required)

Returns: `UpdateQuestionnaireExecutionResponse`

UpdateQuestionnaireExecutionStatus (*userId, executionId, body, **kwargs*)

Update questionnaire execution status

Args: `userId`, str: User GUID (required) `executionId`, str: Execution id (required) `body`, str: Status (required)

Returns: `UpdateQuestionnaireExecutionResponse`

UpdateQuestionnaireMetadata (*userId, questionnaireId, body, **kwargs*)

Update questionnaire metadata

Args: `userId`, str: User global unique identifier (required) `questionnaireId`, str: Questionnaire global unique identifier (required) `body`, `QuestionnaireMetadata`: Questionnaire metadata to update (required)

Returns: `UpdateQuestionnaireResponse`

UpdateQuestionnaireTemplateFields (*userId, fileId, body, **kwargs*)

Update template's fields

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `body`, `List[TemplateEditorField]`: Array of template fields which should be updated (required)

Returns: `GetQuestionnaireTemplateResponse`

basePath

4.1.9 MgmtApi Module

class `groupdocs.MgmtApi.MgmtApi` (*ApiClient*)

Bases: `object`

AddStorageProvider (*userId, provider, body, **kwargs*)

Adds a new storage provider configuration.

Args: `userId`, str: User GUID (required) `provider`, str: Storage provider name (required) `body`, `StorageProviderInfo`: Storage provider configuration details (required)

Returns: `AddStorageProviderResponse`

ChangeAlienUserPassword (*callerId, userId, body, **kwargs*)

Change alien user password

Args: callerId, str: Caller GUID (required) userId, str: User GUID (required) body, UserPasswordInfo: Password (required)

Returns: ChangePasswordResponse

ChangeUserPassword (*userId, body, **kwargs*)

Change user password

Args: userId, str: User GUID (required) body, UserPasswordInfo: Password (required)

Returns: ChangePasswordResponse

CreateUser (*callerId, body, **kwargs*)

Create user

Args: callerId, str: Caller GUID (required) body, UserInfo: Payload (required)

Returns: CreateUserResponse

CreateUserLogin (*callerId, userId, password, **kwargs*)

Create user login

Args: callerId, str: Caller GUID (required) userId, str: User name (required) password, str: Password (required)

Returns: UserInfoResponse

DeleteAccount (*userId, **kwargs*)

Closes user's account.

Args: userId, str: User global unique identifier (required)

Returns: DeleteAccountResponse

DeleteAccountUser (*adminId, userName, **kwargs*)

Delete account user.

Args: adminId, str: Administrator GUID (required) userName, str: User name (required)

Returns: DeleteAccountUserResponse

GenerateKeyForUser (*userId, area, **kwargs*)

Generates new active user embed key.

Args: userId, str: User GUID (required) area, str: Application area where the key is (required)

Returns: GetUserEmbedKeyResponse

GetAccount (*userId, **kwargs*)

Returns an account information.

Args: userId, str: User GUID (required)

Returns: GetAccountResponse

GetAccountUsers (*adminId, **kwargs*)

Returns account user list.

Args: adminId, str: Administrator GUID (required)

Returns: GetAccountUsersResponse

GetAlienUserProfile (*callerId, userId, **kwargs*)

Get alien user profile

Args: callerId, str: Caller GUID (required) userId, str: User GUID (required)

Returns: UserInfoResponse

GetRoles (*userId, **kwargs*)

Returns a list of user roles.

Args: *userId*, str: User GUID (required)

Returns: GetRolesResponse

GetStorageProviders (*userId, **kwargs*)

Returns user's storage providers.

Args: *userId*, str: User GUID (required)

Returns: GetStorageProvidersResponse

GetUserEmbedKey (*userId, area, **kwargs*)

Returns active user embed key.

Args: *userId*, str: User GUID (required) *area*, str: Application area where the key is (required)

Returns: GetUserEmbedKeyResponse

GetUserEmbedKeyFromGuid (*callerId, guid, **kwargs*)

Returns embed key by GUID.

Args: *callerId*, str: UserId invoked the service (required) *guid*, str: Key GUID (required)

Returns: GetUserEmbedKeyResponse

GetUserProfile (*userId, **kwargs*)

Get user profile

Args: *userId*, str: User GUID (required)

Returns: UserInfoResponse

GetUserProfileByClaimedToken (*callerId, token, **kwargs*)

Get user profile by claimed token

Args: *callerId*, str: Caller GUID (required) *token*, str: Token (required)

Returns: UserInfoResponse

GetUserProfileByResetToken (*callerId, token, **kwargs*)

Get user profile by reset token

Args: *callerId*, str: Caller GUID (required) *token*, str: Token (required)

Returns: UserInfoResponse

GetUserProfileByVerifToken (*callerId, token, **kwargs*)

Get user profile by verif token

Args: *callerId*, str: Caller GUID (required) *token*, str: Token (required)

Returns: UserInfoResponse

GetUserRoles (*callerId, userId, **kwargs*)

Returns a list of user roles.

Args: *callerId*, str: Caller GUID (required) *userId*, str: User GUID (required)

Returns: GetRolesResponse

RemoveAccount (*userId, email, nonce, **kwargs*)

Remove account by request.

Args: *userId*, str: User id (required) *email*, str: User email (required) *nonce*, str: Removal nonce (required)

Returns: AccountRemovalResponse

ResetUserPassword (*callerId, userId, **kwargs*)

Reset user password

Args: callerId, str: Caller GUID (required) userId, str: User GUID (required)

Returns: ResetPasswordResponse

Revoke (*userId, **kwargs*)

Revoke private key

Args: userId, str: User GUID (required)

Returns: RevokeResponse

SetUserRoles (*callerId, userId, body, **kwargs*)

Set user roles.

Args: callerId, str: Caller GUID (required) userId, str: User GUID (required) body, List[RoleInfo]: A list of user roles (required)

Returns: SetUserRolesResponse

UpdateAccountUser (*adminId, userName, body, **kwargs*)

Create or update account user.

Args: adminId, str: Administrator GUID (required) userName, str: User name (required) body, UserInfo: User details (required)

Returns: UpdateAccountUserResponse

UpdateAlienUserProfile (*callerId, userId, body, **kwargs*)

Update alien user profile

Args: callerId, str: Caller GUID (required) userId, str: User GUID (required) body, UserInfo: Info (required)

Returns: UpdateUserResponse

UpdateStorageProvider (*userId, provider, body, **kwargs*)

Updates user's storage provider configuration.

Args: userId, str: User GUID (required) provider, str: Storage provider name (required) body, StorageProviderInfo: Storage provider configuration details (required)

Returns: UpdateStorageProviderResponse

UpdateUserProfile (*userId, body, **kwargs*)

Update user profile

Args: userId, str: User GUID (required) body, UserInfo: Info (required)

Returns: UpdateUserResponse

basePath

4.1.10 PostApi Module

class groupdocs.PostApi.**PostApi** (*apiClient*)

Bases: object

CompressByPost (*userId, fileId, archiveType, **kwargs*)

Compress by post

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `archiveType`, str: Archive Type (required)

Returns: `CompressResponse`

DeleteByPost (*userId, fileId, **kwargs*)

Delete by post

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required)

Returns: `DeleteResponse`

DeleteFromFolderByPost (*userId, path, **kwargs*)

Delete from folder by post

Args: `userId`, str: User GUID (required) `path`, str: Path (required)

Returns: `DeleteResponse`

RenameByPost (*userId, fileId, newName, **kwargs*)

Rename by post

Args: `userId`, str: User GUID (required) `fileId`, str: File GUID (required) `newName`, str: New name (required)

Returns: `RenameResponse`

basePath

4.1.11 SharedApi Module

class `groupdocs.SharedApi.SharedApi` (*ApiClient*)

Bases: `object`

Download (*guid, fileName, **kwargs*)

Args: `guid`, str: GUID (required) `fileName`, str: File name (required) `render`, bool: Render (optional)

Returns: `stream`

GetHtml (*guid, **kwargs*)

Get html

Args: `guid`, str: GUID (required)

Returns: `stream`

GetPackage (*path, **kwargs*)

Get package

Args: `path`, str: Path (required)

Returns: `stream`

GetXml (*guid, **kwargs*)

Get xml

Args: `guid`, str: GUID (required)

Returns: `stream`

LoginUser (*userName, body, **kwargs*)

Logins user using user name and password

Args: `userName`, str: User name (required) `body`, str: Password (required)

Returns: UserInfoResponse

basePath

4.1.12 SignatureApi Module

class groupdocs.SignatureApi.**SignatureApi** (*ApiClient*)

Bases: object

AddContact (*userId, body, **kwargs*)

Add contact

Args: *userId*, str: User GUID (required) *body*, SignatureContactSettingsInfo: Contact data (required)

Returns: SignatureContactResponse

AddContactIntegration (*userId, **kwargs*)

Add Contact Integration Authorization

Args: *userId*, str: User GUID (required) *body*, SignatureContactIntegrationSettings: Authorization settings (optional)

Returns: SignatureStatusResponse

AddPredefinedList (*userId, body, **kwargs*)

Add predefined list

Args: *userId*, str: User GUID (required) *body*, SignaturePredefinedListSettingsInfo: List data (required)

Returns: SignaturePredefinedListResponse

AddSignatureEnvelopeDocument (*userId, envelopeGuid, documentGuid, **kwargs*)

Add document in envelope

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *documentGuid*, str: Document GUID (required) *order*, int: Document order (optional) *parseFields*, bool: Try to parse fields in document (optional)

Returns: SignatureEnvelopeDocumentResponse

AddSignatureEnvelopeField (*userId, envelopeGuid, documentGuid, recipientGuid, fieldGuid, **kwargs*)

Add signature field for document in envelope

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *documentGuid*, str: Document GUID (required) *recipientGuid*, str: Recipient GUID (required) *fieldGuid*, str: Field GUID (required) *body*, SignatureEnvelopeFieldSettingsInfo: Settings of the field (optional)

Returns: SignatureEnvelopeFieldsResponse

AddSignatureEnvelopeRecipient (*userId, envelopeGuid, recipientEmail, recipientFirstName, recipientLastName, roleGuid, **kwargs*)

Add signature envelope recipient

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *recipientEmail*, str: Recipient email (required) *recipientFirstName*, str: Recipient first name (required) *recipientLastName*, str: Recipient last name (required) *roleGuid*, str: Recipient role GUID (required) *order*, int: Recipient order (optional)

Returns: SignatureEnvelopeRecipientResponse

AddSignatureFormDocument (*userId, formGuid, documentGuid, **kwargs*)

Add document in form

Args: `userId`, str: User GUID (required) `formGuid`, str: Form GUID (required) `documentGuid`, str: Document GUID (required) `order`, int: Document order (optional) `parseFields`, bool: Try to parse fields in document (optional)

Returns: `SignatureFormDocumentResponse`

AddSignatureFormField (*userId, formGuid, documentGuid, fieldGuid, **kwargs*)

Add signature field for document in form

Args: `userId`, str: User GUID (required) `formGuid`, str: Form GUID (required) `documentGuid`, str: Document GUID (required) `fieldGuid`, str: Field GUID (required) `body`, `SignatureFormFieldSettingsInfo`: Settings of the field (optional)

Returns: `SignatureFormFieldResponse`

AddSignatureTemplateDocument (*userId, templateGuid, documentGuid, **kwargs*)

Add document to template

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required) `documentGuid`, str: Document GUID (required) `order`, int: Display order of the document (optional) `parseFields`, bool: Try to parse fields in document (optional)

Returns: `SignatureTemplateDocumentResponse`

AddSignatureTemplateField (*userId, templateGuid, documentGuid, recipientGuid, fieldGuid, **kwargs*)

Add signature template field

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required) `documentGuid`, str: Document GUID (required) `recipientGuid`, str: Recipient GUID (required) `fieldGuid`, str: Field GUID (required) `body`, `SignatureTemplateFieldSettingsInfo`: Settings of the field (optional)

Returns: `SignatureTemplateFieldResponse`

AddSignatureTemplateRecipient (*userId, templateGuid, nickname, roleGuid, **kwargs*)

Add recipient to the template

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required) `nickname`, str: Nickname of the recipient (required) `roleGuid`, str: Role GUID (required) `order`, int: Display order of the recipient (optional)

Returns: `SignatureTemplateRecipientResponse`

ArchiveSignatureEnvelope (*userId, envelopeGuid, **kwargs*)

Archive envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required)

Returns: `SignatureStatusResponse`

ArchiveSignatureForm (*userId, formGuid, **kwargs*)

Archive signature form

Args: `userId`, str: User GUID (required) `formGuid`, str: Form GUID (required)

Returns: `SignatureStatusResponse`

AssignSignatureEnvelopeField (*userId, envelopeGuid, documentGuid, fieldGuid, **kwargs*)

Assign signature envelope field

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `documentGuid`, str: Document GUID (required) `fieldGuid`, str: Field GUID (required) `body`, `SignatureEnvelopeAssignFieldSettingsInfo`: Settings of the field (optional)

Returns: `SignatureEnvelopeFieldResponse`

AssignSignatureTemplateField (*userId, templateGuid, documentGuid, fieldGuid, **kwargs*)

Assign signature template field

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *documentGuid*, str: Document GUID (required) *fieldGuid*, str: Field GUID (required) *body*, SignatureTemplateAssignFieldSettingsInfo: Settings of the field (optional)

Returns: SignatureTemplateFieldResponse

CancelSignatureEnvelope (*userId, envelopeGuid, **kwargs*)

Cancel envelope

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required)

Returns: SignatureStatusResponse

CompleteSignatureForm (*userId, formGuid, **kwargs*)

Complete signature form

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required)

Returns: SignatureStatusResponse

CreateSignature (*userId, name, **kwargs*)

Create user signature

Args: *userId*, str: User GUID (required) *name*, str: Signature name (required) *body*, SignatureSignatureSettingsInfo: Settings of the signature (optional)

Returns: SignatureSignatureResponse

CreateSignatureEnvelope (*userId, **kwargs*)

Create signature envelope

Args: *userId*, str: User GUID (required) *name*, str: Envelope name (optional) *templateGuid*, str: A templateGuid of the template which will be used to created the new envelope (optional) *envelopeGuid*, str: A envelopeGuid of the envelope which will be used to created the new envelope (optional) *documentGuid*, str: A documentGuid of the document which will be added to the new created envelope (optional) *parseFields*, bool: Try to parse fields in document (optional) *body*, SignatureEnvelopeSettingsInfo: Settings of the new envelope (optional)

Returns: SignatureEnvelopeResponse

CreateSignatureField (*userId, **kwargs*)

Create signature field

Args: *userId*, str: User GUID (required) *body*, SignatureFieldSettingsInfo: Settings of the new field (optional)

Returns: SignatureFieldResponse

CreateSignatureForm (*userId, **kwargs*)

Create signature form

Args: *userId*, str: User GUID (required) *name*, str: Form name (optional) *templateGuid*, str: A templateGuid of the template which will be used to created the new form (optional) *assemblyGuid*, str: A guid of the assembly which will be used to created the new form (optional) *formGuid*, str: A formGuid of the form which will be used to created the new form (optional) *body*, SignatureFormSettingsInfo: Settings of the new form (optional)

Returns: SignatureFormResponse

CreateSignatureTemplate (*userId, name, **kwargs*)

Create template

Args: `userId`, str: User GUID (required) `name`, str: Template name (required) `templateGuid`, str: Template GUID of the template that will be used to create the new template (optional) `envelopeGuid`, str: Envelope GUID of the envelope that will be used to create the new template (optional) `body`, `SignatureTemplateSettingsInfo`: Settings of the template (optional)

Returns: `SignatureTemplateResponse`

DeclineEnvelope (*userId, envelopeGuid, recipientGuid, **kwargs*)

Decline envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `recipientGuid`, str: Recipient GUID (required)

Returns: `SignatureStatusResponse`

DelegateEnvelopeRecipient (*userId, envelopeGuid, recipientGuid, recipientEmail, recipientFirstName, recipientLastName, **kwargs*)

Delegate envelope recipient

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `recipientGuid`, str: Recipient GUID (required) `recipientEmail`, str: Delegated recipient email (required) `recipientFirstName`, str: Delegated recipient first name (required) `recipientLastName`, str: Delegated recipient last name (required)

Returns: `SignatureStatusResponse`

DeleteContact (*userId, contactGuid, **kwargs*)

Delete contact

Args: `userId`, str: User GUID (required) `contactGuid`, str: Contact GUID (required)

Returns: `SignatureContactResponse`

DeletePredefinedList (*userId, listGuid, **kwargs*)

Delete predefined list

Args: `userId`, str: User GUID (required) `listGuid`, str: List GUID (required)

Returns: `SignaturePredefinedListsResponse`

DeleteSignature (*userId, signatureGuid, **kwargs*)

Delete user signature

Args: `userId`, str: User GUID (required) `signatureGuid`, str: Signature GUID (required)

Returns: `SignatureStatusResponse`

DeleteSignatureEnvelope (*userId, envelopeGuid, **kwargs*)

Delete signature envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required)

Returns: `SignatureStatusResponse`

DeleteSignatureEnvelopeDocument (*userId, envelopeGuid, documentGuid, **kwargs*)

Delete document from envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `documentGuid`, str: Document GUID (required)

Returns: `SignatureStatusResponse`

DeleteSignatureEnvelopeField (*userId, envelopeGuid, fieldGuid, **kwargs*)

Delete signature envelope field

Args: userId, str: User GUID (required) envelopeGuid, str: Envelope GUID (required) fieldGuid, str: Field GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureEnvelopeFieldLocation (*userId, envelopeGuid, fieldGuid, locationGuid, **kwargs*)

Remove signature envelope field location

Args: userId, str: User GUID (required) envelopeGuid, str: Envelope GUID (required) fieldGuid, str: Field GUID (required) locationGuid, str: Field location GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureEnvelopeRecipient (*userId, envelopeGuid, recipientGuid, **kwargs*)

Delete signature envelope recipient

Args: userId, str: User GUID (required) envelopeGuid, str: Envelope GUID (required) recipientGuid, str: Recipient GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureField (*userId, fieldGuid, **kwargs*)

Delete signature field

Args: userId, str: User GUID (required) fieldGuid, str: Field GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureForm (*userId, formGuid, **kwargs*)

Delete signature form

Args: userId, str: User GUID (required) formGuid, str: Form GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureFormDocument (*userId, formGuid, documentGuid, **kwargs*)

Delete document from form

Args: userId, str: User GUID (required) formGuid, str: Form GUID (required) documentGuid, str: Document GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureFormField (*userId, formGuid, fieldGuid, **kwargs*)

Delete signature form field

Args: userId, str: User GUID (required) formGuid, str: Form GUID (required) fieldGuid, str: Field GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureFormFieldLocation (*userId, formGuid, fieldGuid, locationGuid, **kwargs*)

Remove signature form field location

Args: userId, str: User GUID (required) formGuid, str: Form GUID (required) fieldGuid, str: Field GUID (required) locationGuid, str: Field location GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureTemplate (*userId, templateGuid, **kwargs*)

Delete template

Args: userId, str: User GUID (required) templateGuid, str: Template GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureTemplateDocument (*userId, templateGuid, documentGuid, **kwargs*)

Remove document from template

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *documentGuid*, str: Document GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureTemplateField (*userId, templateGuid, fieldGuid, **kwargs*)

Delete signature template field

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *fieldGuid*, str: Field GUID (required)

Returns: SignatureTemplateResponse

DeleteSignatureTemplateFieldLocation (*userId, templateGuid, fieldGuid, locationGuid, **kwargs*)

Delete signature template field location

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *fieldGuid*, str: Field GUID (required) *locationGuid*, str: Field location GUID (required)

Returns: SignatureStatusResponse

DeleteSignatureTemplateRecipient (*userId, templateGuid, recipientGuid, **kwargs*)

Remove recipient from template

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *recipientGuid*, str: Recipient GUID (required)

Returns: SignatureStatusResponse

FillEnvelopeField (*userId, envelopeGuid, documentGuid, recipientGuid, fieldGuid, **kwargs*)

Fill envelope field

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *documentGuid*, str: Document GUID (required) *recipientGuid*, str: Recipient GUID (required) *fieldGuid*, str: Field GUID (required) *signatureGuid*, str: SignatureId GUID (optional) *body*, stream: Data to be placed in field (optional)

Returns: SignatureEnvelopeFieldResponse

GetContacts (*userId, **kwargs*)

Get contacts

Args: *userId*, str: User GUID (required) *page*, int: Page number (optional) *records*, int: Records count to be returned (optional) *firstName*, str: Filter by firstName (optional) *lastName*, str: Filter by lastName (optional) *email*, str: Filter by email (optional) *useAnd*, bool: Filter uses AND instead of default OR between parameters (optional)

Returns: SignatureContactsResponse

GetEnvelopeAuditLogs (*userId, envelopeGuid, **kwargs*)

Get envelope audit logs

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required)

Returns: SignatureEnvelopeAuditLogsResponse

GetFieldsList (*userId, **kwargs*)

Get signature fields

Args: *userId*, str: User GUID (required) *fieldGuid*, str: Filter fields by id (optional)

Returns: SignatureFieldsResponse

GetFormAuditLogs (*userId, formGuid, **kwargs*)

Get form audit logs

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required)

Returns: SignatureFormAuditLogsResponse

GetRecipientSignatureEnvelopes (*userId, **kwargs*)

Get signature envelopes where the user is recipient

Args: *userId*, str: User GUID (required) *statusId*, str: Filter envelopes by statusId (optional) *page*, int: Show records for page number (optional) *records*, int: Show records count (optional)

Returns: SignatureEnvelopesResponse

GetRolesList (*userId, **kwargs*)

Get signature roles

Args: *userId*, str: User GUID (required) *roleGuid*, str: Filter roles by GUID (optional)

Returns: SignatureRolesResponse

GetSignDocumentStatus (*userId, jobGuid, **kwargs*)

Get sign documents status

Args: *userId*, str: User GUID (required) *jobGuid*, str: Job GUID (required)

Returns: SignatureSignDocumentStatusResponse

GetSignatureEnvelope (*userId, envelopeGuid, **kwargs*)

Get signature envelope

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required)

Returns: SignatureEnvelopeResponse

GetSignatureEnvelopeDocuments (*userId, envelopeGuid, **kwargs*)

Get documents in envelope

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required)

Returns: SignatureEnvelopeDocumentsResponse

GetSignatureEnvelopeFieldData (*envelopeGuid, recipientGuid, fieldGuid, **kwargs*)

Get signed envelope field data

Args: *envelopeGuid*, str: Envelope GUID (required) *recipientGuid*, str: Recipient GUID (required) *fieldGuid*, str: Field GUID (required)

Returns: stream

GetSignatureEnvelopeFields (*userId, envelopeGuid, **kwargs*)

Get signature field for document in envelope per recipient

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *documentGuid*, str: Document GUID (optional) *recipientGuid*, str: Recipient GUID (optional) *fieldGuid*, str: Field GUID (optional)

Returns: SignatureEnvelopeFieldsResponse

GetSignatureEnvelopeRecipients (*userId, envelopeGuid, **kwargs*)

Get signature envelope recipients

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required)

Returns: SignatureEnvelopeRecipientsResponse

GetSignatureEnvelopeResources (*userId, **kwargs*)

Get envelope resources

Args: *userId*, str: User GUID (required) *statusIds*, str: Envelope status identifier - comma separated list (optional)

Returns: SignatureEnvelopeResourcesResponse

GetSignatureEnvelopes (*userId, **kwargs*)

Get signature envelopes

Args: *userId*, str: User GUID (required) *statusId*, int: Filter envelopes by statusId (optional) *page*, int: Show records for page number (optional) *records*, int: Show records count (optional) *originalDocumentMD5*, str: Filter envelopes by original document md5 checksum (optional) *recipientEmail*, str: Filter envelopes by recipient email (optional) *datetime*, str: Filter envelopes by date (optional) *name*, str: Filter envelopes by name (optional) *tag*, str: Filter envelopes by tag (optional)

Returns: SignatureEnvelopesResponse

GetSignatureForm (*userId, formGuid, **kwargs*)

Get signature form

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required)

Returns: SignatureFormResponse

GetSignatureFormDocumentFields (*userId, formGuid, documentGuid, **kwargs*)

Get form fields for document in form per participant

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required) *documentGuid*, str: Document GUID (required) *fieldGuid*, str: Field GUID (optional)

Returns: SignatureFormFieldsResponse

GetSignatureFormDocuments (*userId, formGuid, **kwargs*)

Get documents in form

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required)

Returns: SignatureFormDocumentsResponse

GetSignatureFormFields (*userId, formGuid, **kwargs*)

Get form fields for document in form per participant

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required) *fieldGuid*, str: Field GUID (optional)

Returns: SignatureFormFieldsResponse

GetSignatureFormParticipant (*formGuid, participantGuid, **kwargs*)

Get signature form participant

Args: *formGuid*, str: Form GUID (required) *participantGuid*, str: Participant GUID (required)

Returns: SignatureFormParticipantResponse

GetSignatureFormParticipants (*userId, formGuid, **kwargs*)

Get form participants

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required)

Returns: SignatureFormParticipantsResponse

GetSignatureFormResources (*userId, statusIds, **kwargs*)

Get form resources

Args: `userId`, str: User GUID (required) `statusIds`, str: Form status identifier - comma separated list (required)

Returns: `SignatureFormResourcesResponse`

GetSignatureForms (*userId*, ***kwargs*)

Get signature forms

Args: `userId`, str: User GUID (required) `statusId`, int: Filter forms by statusId (optional) `page`, int: Show records for page number (optional) `records`, int: Show records count (optional) `originalDocumentMD5`, str: Filter forms by original document MD5 (optional) `datetime`, str: Filter forms by date (optional) `name`, str: Filter forms by name (optional) `tag`, str: Filter forms by tag (optional)

Returns: `SignatureFormsResponse`

GetSignaturePredefinedLists (*userId*, ***kwargs*)

Get user predefined lists

Args: `userId`, str: User GUID (required)

Returns: `SignaturePredefinedListsResponse`

GetSignatureTemplate (*userId*, *templateGuid*, ***kwargs*)

Get template

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required)

Returns: `SignatureTemplateResponse`

GetSignatureTemplateDocuments (*userId*, *templateGuid*, ***kwargs*)

Get documents in template

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required)

Returns: `SignatureTemplateDocumentsResponse`

GetSignatureTemplateFields (*userId*, *templateGuid*, ***kwargs*)

Get template fields

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required) `documentGuid`, str: Document GUID (optional) `recipientGuid`, str: Recipient GUID (optional) `fieldGuid`, str: Field GUID (optional)

Returns: `SignatureTemplateFieldsResponse`

GetSignatureTemplateRecipients (*userId*, *templateGuid*, ***kwargs*)

Get template recipients

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required)

Returns: `SignatureTemplateRecipientsResponse`

GetSignatureTemplateResources (*userId*, ***kwargs*)

Get template resources

Args: `userId`, str: User GUID (required)

Returns: `SignatureTemplateResourcesResponse`

GetSignatureTemplates (*userId*, ***kwargs*)

Get templates

Args: `userId`, str: User GUID (required) `page`, int: Page number (optional) `records`, int: Records count (optional) `documentGuid`, str: Filter templates by document originalMD5 (optional) `recipientName`, str: Filter templates by recipient nickname (optional) `name`, str: Filter templates by signatureTemplate name (optional) `tag`, str: Filter templates by tag (optional)

Returns: SignatureTemplatesResponse

GetSignatures (*userId*, ***kwargs*)

Get user signatures

Args: *userId*, str: User GUID (required) *page*, int: Show records for page number (optional) *records*, int: Show records count (optional) *name*, str: Filter by signature name (optional)

Returns: SignatureSignaturesResponse

GetSignedEnvelopeDocument (*userId*, *envelopeGuid*, *documentGuid*, ***kwargs*)

Get signed envelope document

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *documentGuid*, str: Document GUID (required)

Returns: stream

GetSignedEnvelopeDocuments (*userId*, *envelopeGuid*, ***kwargs*)

Get signed envelope documents

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required)

Returns: stream

GetSignedFormDocuments (*userId*, *formGuid*, ***kwargs*)

Get signed form documents

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required)

Returns: stream

GetUserEmailTemplate (*userId*, *templateType*, ***kwargs*)

Get user email template

Args: *userId*, str: User GUID (required) *templateType*, int: Template type (required)

Returns: stream

ImportContacts (*userId*, ***kwargs*)

Import contacts

Args: *userId*, str: User GUID (required) *body*, List[SignatureContactSettingsInfo]: Array of SignatureContactSettingsInfo (optional)

Returns: SignatureContactsImportResponse

ModifyContact (*userId*, *contactGuid*, ***kwargs*)

Update contact

Args: *userId*, str: User GUID (required) *contactGuid*, str: Contact GUID (required) *body*, SignatureContactSettingsInfo: Contact data (optional)

Returns: SignatureContactResponse

ModifySignatureEnvelope (*userId*, *envelopeGuid*, ***kwargs*)

Modify signature envelope

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *body*, SignatureEnvelopeSettingsInfo: Settings of the envelope (optional)

Returns: SignatureEnvelopeResponse

ModifySignatureEnvelopeField (*userId*, *envelopeGuid*, *documentGuid*, *recipientGuid*, *fieldGuid*, ***kwargs*)

Modify signature envelope field

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `documentGuid`, str: Document GUID (required) `recipientGuid`, str: Recipient GUID (required) `fieldGuid`, str: Field GUID (required) `body`, `SignatureEnvelopeFieldSettingsInfo`: Settings of the field (optional)

Returns: `SignatureEnvelopeFieldResponse`

ModifySignatureEnvelopeFieldLocation (*userId, envelopeGuid, documentGuid, recipientGuid, fieldGuid, locationGuid, **kwargs*)

Modify signature envelope field location

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `documentGuid`, str: Document GUID (required) `recipientGuid`, str: Recipient GUID (required) `fieldGuid`, str: Field GUID (required) `locationGuid`, str: Field location GUID (required) `body`, `SignatureEnvelopeFieldLocationSettingsInfo`: Settings of the field location (optional)

Returns: `SignatureEnvelopeFieldResponse`

ModifySignatureEnvelopeFieldLocationOrder (*userId, envelopeGuid, documentGuid, recipientGuid, fieldGuid, locationGuid, **kwargs*)

Modify signature envelope field location

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `documentGuid`, str: Document GUID (required) `recipientGuid`, str: Recipient GUID (required) `fieldGuid`, str: Field GUID (required) `locationGuid`, str: Field location GUID (required) `order`, int: Location order (optional)

Returns: `SignatureEnvelopeFieldResponse`

ModifySignatureEnvelopeRecipient (*userId, envelopeGuid, recipientGuid, recipientEmail, recipientFirstName, recipientLastName, roleGuid, **kwargs*)

Modify signature envelope recipient

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `recipientGuid`, str: Recipient GUID (required) `recipientEmail`, str: Recipient email (required) `recipientFirstName`, str: Recipient first name (required) `recipientLastName`, str: Recipient last name (required) `roleGuid`, str: Recipient role GUID (required) `order`, int: Recipient order (optional)

Returns: `SignatureEnvelopeRecipientResponse`

ModifySignatureField (*userId, fieldGuid, **kwargs*)

Modify signature field

Args: `userId`, str: User GUID (required) `fieldGuid`, str: Field GUID (required) `body`, `SignatureFieldSettingsInfo`: Settings of the field (optional)

Returns: `SignatureFieldResponse`

ModifySignatureForm (*userId, formGuid, **kwargs*)

Modify signature form

Args: `userId`, str: User GUID (required) `formGuid`, str: Form GUID (required) `body`, `SignatureFormSettingsInfo`: Settings of the form (optional)

Returns: `SignatureFormResponse`

ModifySignatureFormDocument (*userId, formGuid, documentGuid, **kwargs*)

Modify signature form document

Args: `userId`, str: User GUID (required) `formGuid`, str: Form GUID (required) `documentGuid`, str: Form Document GUID (required) `body`, `SignatureFormDocumentSettingsInfo`: Settings of the document (optional)

Returns: SignatureFormDocumentResponse

ModifySignatureFormField (*userId, formGuid, documentGuid, fieldGuid, **kwargs*)

Modify signature form field

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required) *documentGuid*, str: Document GUID (required) *fieldGuid*, str: Field GUID (required) *body*, SignatureFormFieldSettingsInfo: Settings of the field (optional)

Returns: SignatureFormFieldResponse

ModifySignatureFormFieldLocation (*userId, formGuid, documentGuid, fieldGuid, locationGuid, **kwargs*)

Modify signature form field location

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required) *documentGuid*, str: Document GUID (required) *fieldGuid*, str: Field GUID (required) *locationGuid*, str: Field location GUID (required) *body*, SignatureFormFieldLocationSettingsInfo: Settings of the field location (optional)

Returns: SignatureFormFieldResponse

ModifySignatureTemplate (*userId, templateGuid, **kwargs*)

Modify template

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *body*, SignatureTemplateSettingsInfo: Settings of the template (optional)

Returns: SignatureTemplateResponse

ModifySignatureTemplateField (*userId, templateGuid, documentGuid, recipientGuid, fieldGuid, **kwargs*)

Modify signature template field

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *documentGuid*, str: Document GUID (required) *recipientGuid*, str: Recipient GUID (required) *fieldGuid*, str: Field GUID (required) *body*, SignatureTemplateFieldSettingsInfo: Settings of the field (optional)

Returns: SignatureTemplateFieldResponse

ModifySignatureTemplateFieldLocation (*userId, templateGuid, documentGuid, recipientGuid, fieldGuid, locationGuid, **kwargs*)

Modify signature template field location

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *documentGuid*, str: Document GUID (required) *recipientGuid*, str: Recipient GUID (required) *fieldGuid*, str: Field GUID (required) *locationGuid*, str: Field location GUID (required) *body*, SignatureTemplateFieldLocationSettingsInfo: Settings of the field location (optional)

Returns: SignatureTemplateFieldResponse

ModifySignatureTemplateRecipient (*userId, templateGuid, recipientGuid, nickname, roleGuid, **kwargs*)

Update template recipient

Args: *userId*, str: User GUID (required) *templateGuid*, str: Template GUID (required) *recipientGuid*, str: Recipient GUID (required) *nickname*, str: Nickname of the recipient (required) *roleGuid*, str: Role GUID (required) *order*, str: Display order of the recipient (optional)

Returns: SignatureTemplateRecipientResponse

PublicFillEnvelopeField (*envelopeGuid, documentGuid, recipientGuid, fieldGuid, **kwargs*)

Fill envelope field

Args: envelopeGuid, str: Envelope GUID (required) documentGuid, str: Document GUID (required)
recipientGuid, str: Recipient GUID (required) fieldGuid, str: Field GUID (required) body, stream:
Data to be placed in field (optional)

Returns: SignatureEnvelopeFieldResponse

PublicFillFormField (*formGuid, documentGuid, fieldGuid, authSignature, participantIdId,*
***kwargs*)

Fill form field

Args: formGuid, str: Form GUID (required) documentGuid, str: Document GUID (required) fieldGuid,
str: Field GUID (required) authSignature, str: Authentication signature (required) body, stream: Data
to be placed in field (optional) participantIdId, str: Participant GUID (required)

Returns: SignatureFormFieldResponse

PublicFillSignatureForm (*formGuid, **kwargs*)

Fill signature form

Args: formGuid, str: Form GUID (required)

Returns: SignatureFormParticipantResponse

PublicFormParticipantResentValidationCode (*formGuid, participantGuid, **kwargs*)

Request validation code email

Args: formGuid, str: Form GUID (required) participantGuid, str: Participant GUID (required)

Returns: SignatureStatusResponse

PublicGetDefaultEmailTemplate (*templateType, **kwargs*)

Get default email template

Args: templateType, int: Template type (required)

Returns: stream

PublicGetDocumentFields (*documentGuid, **kwargs*)

Sign document

Args: documentGuid, str: Document GUID (required)

Returns: SignatureDocumentFieldsResponse

PublicGetEnvelopeDocuments (*envelopeGuid, recipientGuid, **kwargs*)

Get documents in envelope

Args: envelopeGuid, str: Envelope GUID (required) recipientGuid, str: Recipient GUID (required)

Returns: SignatureEnvelopeDocumentsResponse

PublicGetEnvelopeRecipients (*envelopeGuid, **kwargs*)

Get signature envelope recipients

Args: envelopeGuid, str: Envelope GUID (required)

Returns: SignatureEnvelopeRecipientsResponse

PublicGetSignDocumentStatus (*jobGuid, **kwargs*)

Get signed document status

Args: jobGuid, str: Job GUID (required)

Returns: SignatureSignDocumentStatusResponse

PublicGetSignatureEnvelope (*envelopeGuid, recipientGuid, **kwargs*)

Get signature envelope

Args: envelopeGuid, str: Envelope GUID (required) recipientGuid, str: Recipient GUID (required)

Returns: SignatureEnvelopeResponse

PublicGetSignatureEnvelopeFields (*envelopeGuid*, *documentGuid*, *recipientGuid*, ***kwargs*)

Get signature field for document in envelope per recipient

Args: envelopeGuid, str: Envelope GUID (required) documentGuid, str: Document GUID (required) recipientGuid, str: Recipient GUID (required)

Returns: SignatureEnvelopeFieldsResponse

PublicGetSignatureForm (*formGuid*, ***kwargs*)

Get signature form

Args: formGuid, str: Form GUID (required)

Returns: SignatureFormResponse

PublicGetSignatureFormDocuments (*formGuid*, ***kwargs*)

Get documents in form

Args: formGuid, str: Form GUID (required)

Returns: SignatureFormDocumentsResponse

PublicGetSignatureFormFields (*formGuid*, *documentGuid*, *participantGuid*, ***kwargs*)

Get form fields for document in form per participant

Args: formGuid, str: Form GUID (required) documentGuid, str: Document GUID (required) participantGuid, str: Participant GUID (required)

Returns: SignatureFormFieldsResponse

PublicGetSignedEnvelopeDocuments (*envelopeGuid*, *recipientGuid*, ***kwargs*)

Get signed envelope documents

Args: envelopeGuid, str: Envelope GUID (required) recipientGuid, str: Recipient GUID (required)

Returns: stream

PublicGetSignedFormDocuments (*formGuid*, *participantGuid*, ***kwargs*)

Get signed form documents

Args: formGuid, str: Form GUID (required) participantGuid, str: Participant GUID (required)

Returns: stream

PublicModifyFormParticipant (*formGuid*, *participantGuid*, *participantEmail*, ***kwargs*)

Update Form Participant

Args: formGuid, str: Form GUID (required) participantGuid, str: Participant GUID (required) participantEmail, str: Participant Email to be set (required)

Returns: SignatureFormParticipantResponse

PublicSignDocument (*documentGuid*, ***kwargs*)

Sign document

Args: documentGuid, str: Document GUID (required) body, PublicSignatureSignDocumentSignerSettingsInfo: Settings of the signing document (optional)

Returns: SignatureSignDocumentResponse

PublicSignEnvelope (*envelopeGuid*, *recipientGuid*, ***kwargs*)

Sign envelope

Args: envelopeGuid, str: Envelope GUID (required) recipientGuid, str: Recipient GUID (required) body, SignatureSignEnvelopeSettingsInfo: Sign settings (optional)

Returns: SignatureStatusResponse

PublicSignForm (*formGuid, participantGuid, participantName, authSignature, **kwargs*)
Sign Form

Args: formGuid, str: Form GUID (required) participantGuid, str: Participant GUID (required) participantName, str: Participant Name (required) authSignature, str: Authentication signature (required) authData, stream: Base64 encoded Signer Image (optional) body, SignatureSignFormSettingsInfo: Sign settings (optional)

Returns: SignatureStatusResponse

PublicValidateFormParticipant (*formGuid, participantGuid, validationCode, **kwargs*)
Validate Participant Identity

Args: formGuid, str: Form GUID (required) participantGuid, str: Participant GUID (required) validationCode, str: Code to be validated (required)

Returns: SignatureStatusResponse

PublicVerifyDocument (*body, **kwargs*)
Verify

Args: body, stream: Document to verify (required)

Returns: SignatureVerifyDocumentResponse

PublishSignatureForm (*userId, formGuid, **kwargs*)
Publish signature form

Args: userId, str: User GUID (required) formGuid, str: Form GUID (required) body, WebhookInfo: Webhook Callback Url (optional)

Returns: SignatureStatusResponse

RenameSignatureEnvelope (*userId, envelopeGuid, name, **kwargs*)
Rename signature envelope

Args: userId, str: User GUID (required) envelopeGuid, str: Envelope GUID (required) name, str: New envelope name (required)

Returns: SignatureEnvelopeResponse

RenameSignatureEnvelopeDocument (*userId, envelopeGuid, documentGuid, **kwargs*)
Rename signature envelope document

Args: userId, str: User GUID (required) envelopeGuid, str: Envelope GUID (required) documentGuid, str: Document GUID (required) newName, str: New name of the document (optional)

Returns: SignatureEnvelopeDocumentResponse

RenameSignatureForm (*userId, formGuid, name, **kwargs*)
Rename signature form

Args: userId, str: User GUID (required) formGuid, str: Form GUID (required) name, str: New form name (required)

Returns: SignatureFormResponse

RenameSignatureFormDocument (*userId, formGuid, documentGuid, **kwargs*)
Rename signature form document

Args: `userId`, str: User GUID (required) `formGuid`, str: Form GUID (required) `documentGuid`, str: Form Document GUID (required) `newName`, str: New name of the document (optional)

Returns: `SignatureFormDocumentResponse`

RenameSignatureTemplate (*userId, templateGuid, name, **kwargs*)

Rename template

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required) `name`, str: New template name (required)

Returns: `SignatureTemplateResponse`

RenameSignatureTemplateDocument (*userId, templateGuid, documentGuid, **kwargs*)

Rename signature template document

Args: `userId`, str: User GUID (required) `templateGuid`, str: Template GUID (required) `documentGuid`, str: Template Document GUID (required) `newName`, str: New name of the document (optional)

Returns: `SignatureTemplateDocumentResponse`

ResendSignatureEnvelopeEmailNotification (*userId, envelopeGuid, **kwargs*)

Resend envelope email notification

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required)

Returns: `SignatureStatusResponse`

RestartExpiredSignatureEnvelope (*userId, envelopeGuid, **kwargs*)

Restart expired envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required)

Returns: `SignatureStatusResponse`

RetrySignEnvelope (*userId, envelopeGuid, **kwargs*)

Retry sign envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required)

Returns: `SignatureStatusResponse`

SignDocument (*userId, **kwargs*)

Sign document

Args: `userId`, str: User GUID (required) `body`, `SignatureSignDocumentSettingsInfo`: Settings of the signing document (optional)

Returns: `SignatureSignDocumentResponse`

SignEnvelope (*userId, envelopeGuid, recipientGuid, **kwargs*)

Sign envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `recipientGuid`, str: Recipient GUID (required) `body`, `SignatureSignEnvelopeSettingsInfo`: Sign settings (optional)

Returns: `SignatureStatusResponse`

SignatureEnvelopeSend (*userId, envelopeGuid, **kwargs*)

Send envelope

Args: `userId`, str: User GUID (required) `envelopeGuid`, str: Envelope GUID (required) `body`, `WebhookInfo`: Webhook Callback Url (optional)

Returns: `SignatureEnvelopeSendResponse`

UpdateEnvelopeFromTemplate (*userId, envelopeGuid, templateGuid, **kwargs*)

Update envelope from template

Args: *userId*, str: User GUID (required) *envelopeGuid*, str: Envelope GUID (required) *templateGuid*, str: Template GUID (required)

Returns: SignatureEnvelopeResponse

UpdateSignatureFormFromTemplate (*userId, formGuid, templateGuid, **kwargs*)

Add signature form fields from template

Args: *userId*, str: User GUID (required) *formGuid*, str: Form GUID (required) *templateGuid*, str: Template GUID (required)

Returns: SignatureFormResponse

basePath

4.1.13 StorageApi Module

class groupdocs.StorageApi.StorageApi (*apiClient*)

Bases: object

CancelFileUpload (*userId, fileId, filePath, **kwargs*)

Cancel file upload

Args: *userId*, str: User GUID (optional) *fileId*, str: file Id (optional) *filePath*, str: File path (optional)

Returns: CancelFileUploadResponse

Compress (*userId, fileId, archiveType, **kwargs*)

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required) *archiveType*, str: Archive type (optional)

Returns: CompressResponse

Create (*userId, path, **kwargs*)

Args: *userId*, str: User GUID (required) *path*, str: Path (required)

Returns: CreateFolderResponse

CreatePackage (*userId, packageName, **kwargs*)

Create Package

Args: *userId*, str: User GUID (required) *packageName*, str: Package Name (required) *storeRelativePath*, bool: Store files using relative paths (optional) *body*, List[str]: Paths (optional)

Returns: CreatePackageResponse

Decompress (*userId, path, body, **kwargs*)

UploadAndUnzip

Args: *userId*, str: User GUID (required) *path*, str: Path (required) *description*, str: Description (optional) *archiveType*, str: Archive type (optional) *body*, stream: Stream (required)

Returns: UploadResponse

Delete (*userId, fileId, **kwargs*)

Args: *userId*, str: User GUID (required) *fileId*, str: File ID (required)

Returns: DeleteResponse

DeleteFromFolder (*userId, path, **kwargs*)

Delete from folder

Args: *userId*, str: User GUID (required) *path*, str: Path (required)

Returns: DeleteResponse

GetFile (*userId, fileId, **kwargs*)

Get file

Args: *userId*, str: User GUID (required) *fileId*, str: File GUID (required)

Returns: stream

GetSharedFile (*userEmail, filePath, **kwargs*)

Get shared file

Args: *userEmail*, str: User Email (required) *filePath*, str: File path (required)

Returns: stream

GetStorageInfo (*userId, **kwargs*)

Get storage info

Args: *userId*, str: User GUID (required)

Returns: StorageInfoResponse

ListEntities (*userId, path, **kwargs*)

List entities

Args: *userId*, str: User GUID (required) *path*, str: Path (optional) *pageIndex*, int: Page Index (optional) *pageSize*, int: Page Size (optional) *orderBy*, str: Order By (optional) *orderAsc*, bool: Order Asc (optional) *filter*, str: Filter (optional) *fileTypes*, str: File Types (optional) *extended*, bool: Indicates whether an extended information should be returned (optional)

Returns: ListEntitiesResponse

MoveFile (*userId, path, **kwargs*)

Move file

Args: *userId*, str: User GUID (required) *path*, str: Path (required) *mode*, str: Mode (optional) Groupdocs_Move, str: File ID (move) (optional) Groupdocs_Copy, str: File ID (copy) (optional)

Returns: FileMoveResponse

MoveFolder (*userId, path, **kwargs*)

Move folder

Args: *userId*, str: User GUID (required) *path*, str: Destination Path (required) *mode*, str: Mode (optional) Groupdocs_Move, str: Source path (move) (optional) Groupdocs_Copy, str: Source path (copy) (optional)

Returns: FolderMoveResponse

MoveToTrash (*userId, path, **kwargs*)

Move to trash

Args: *userId*, str: User GUID (required) *path*, str: Path (required)

Returns: DeleteResponse

RestoreFromTrash (*userId, path, **kwargs*)

Restore from trash

Args: *userId*, str: User GUID (required) *path*, str: Path (required)

Returns: DeleteResponse

Upload (*userId, path, body, **kwargs*)

Args: *userId*, str: User GUID (required) *path*, str: Path (required) *description*, str: Description (optional) *callbackUrl*, str: Callback url (optional) *overrideMode*, int: Override mode (optional) *body*, stream: Stream (required)

Returns: UploadResponse

UploadGoogle (*userId, url, description, accessToken, **kwargs*)

Upload Google

Args: *userId*, str: User GUID (required) *url*, str: Url (required) *description*, str: Description (required) *accessToken*, str: AccessToken (required)

Returns: UploadResponse

UploadWeb (*userId, url, **kwargs*)

Upload Web

Args: *userId*, str: User GUID (required) *url*, str: Url (required)

Returns: UploadResponse

basePath

4.1.14 SystemApi Module

class groupdocs.SystemApi.**SystemApi** (*apiClient*)

Bases: object

GetCountries (*callerId, **kwargs*)

Get countries

Args: *callerId*, str: User GUID (required)

Returns: GetCountriesResponse

GetCreditCard (*userId, **kwargs*)

Get user credit card

Args: *userId*, str: User GUID (required)

Returns: GetCreditCardResponse

GetInvoices (*callerId, **kwargs*)

Get invoices

Args: *callerId*, str: User GUID (required) *pageNumber*, str: Show records for page number (optional) *pageSize*, int: Show records count (optional)

Returns: GetInvoicesResponse

GetPurchaseWizardInfo (*userId, **kwargs*)

Returns purchase wizard info from billing provider

Args: *userId*, str: User global unique identifier (required)

Returns: GetPurchaseWizardResponse

GetStates (*callerId, countryName, **kwargs*)

Get states

Args: *callerId*, str: User GUID (required) *countryName*, str: Country Name (required)

Returns: GetStatesResponse

GetSubscriptionPlanUsage (*userId*, ***kwargs*)

Get subscription plans

Args: *userId*, str: User GUID (required)

Returns: GetSubscriptionPlanUsageResponse

GetSubscriptionPlans (*callerId*, *family*, ***kwargs*)

Get subscription plans

Args: *callerId*, str: User GUID (required) *family*, str: Product Family Name (required)

Returns: GetSubscriptionPlansResponse

GetTermSuggestions (*userId*, *term*, ***kwargs*)

Returns suggestions for a specified term

Args: *userId*, str: User global unique identifier (required) *term*, str: A term to return suggestions for (required)

Returns: GetTermSuggestionsResponse

GetUserPlan (*callerId*, ***kwargs*)

Get user plan

Args: *callerId*, str: User GUID (required)

Returns: GetPlanResponse

GetUserSubscriptionPlan (*callerId*, ***kwargs*)

Get user plan

Args: *callerId*, str: User GUID (required)

Returns: GetUserSubscriptionPlanResponse

SetBillingAddress (*userId*, *body*, ***kwargs*)

Set user billing address

Args: *userId*, str: User GUID (required) *body*, BillingAddressInfo: Billing Address (required)

Returns: GetBillingAddressResponse

SetCreditCard (*userId*, *body*, ***kwargs*)

Set user credit card

Args: *userId*, str: User GUID (required) *body*, CreditCardInfo: Credit Card (required)

Returns: GetCreditCardResponse

SetSubscriptionPlan (*userId*, *productId*, *body*, ***kwargs*)

Set subscription plan user plan

Args: *userId*, str: User GUID (required) *productId*, str: Product ID (required) *body*, SubscriptionPlanInfo: Subscription Plan (required)

Returns: SetUserSubscriptionPlanResponse

SimulateAssessForPaymentPlan (*userId*, *discountCode*, *paymentPlanId*, ***kwargs*)

Simulate Assess For Payment Plan

Args: *userId*, str: User GUID (required) *discountCode*, str: Discount Code (optional) *paymentPlanId*, str: Payment PlanId on the payment service (optional)

Returns: GetInvoicesResponse

SimulateAssessForPricingPlan (*userId, discountCode, planId, **kwargs*)

Simulate Assess For Pricing Plan

Args: *userId*, str: User GUID (required) *discountCode*, str: Discount Code (optional) *planId*, str: Subscription Plan Id (optional)

Returns: GetInvoicesResponse

UpdateSubscriptionPlan (*userId, productId, body, **kwargs*)

Update subscription plan user plan

Args: *userId*, str: User GUID (required) *productId*, str: Product ID (required) *body*, UpdateSubscriptionPlanInfo: Subscription Plan Update parameters (required)

Returns: SetUserSubscriptionPlanResponse

basePath

Sign, Manage, Annotate, Assemble, Compare and Convert Documents with GroupDocs

1. Sign documents online with GroupDocs Signature
2. PDF, Word and Image Annotation with GroupDocs Annotation
3. Online DOC, DOCX, PPT Document Comparison with GroupDocs Comparison
4. Online Document Management with GroupDocs Dashboard
5. Doc to PDF, Doc to Docx, PPT to PDF, and other Document Conversions with GroupDocs Viewer
6. Online Document Automation with GroupDocs Assembly

g

`groupdocs.AntApi`, 7
`groupdocs.ApiClient`, 10
`groupdocs.AsyncApi`, 11
`groupdocs.ComparisonApi`, 12
`groupdocs.DocApi`, 13
`groupdocs.FileStream`, 16
`groupdocs.GroupDocsRequestSigner`, 17
`groupdocs.MergeApi`, 17
`groupdocs.MgmtApi`, 23
`groupdocs.PostApi`, 26
`groupdocs.SharedApi`, 27
`groupdocs.SignatureApi`, 28
`groupdocs.StorageApi`, 44
`groupdocs.SystemApi`, 46

A

- AddAnnotationCollaborator() (groupdocs.AntApi.AntApi method), 7
- AddContact() (groupdocs.SignatureApi.SignatureApi method), 28
- AddContactIntegration() (groupdocs.SignatureApi.SignatureApi method), 28
- AddDataSource() (groupdocs.MergeApi.MergeApi method), 17
- AddDocumentQuestionnaire() (groupdocs.MergeApi.MergeApi method), 17
- addHeaders() (groupdocs.ApiClient.ApiClient method), 10
- AddJobDocument() (groupdocs.AsyncApi.AsyncApi method), 11
- AddJobDocumentDataSource() (groupdocs.MergeApi.MergeApi method), 17
- AddJobDocumentDataSourceFields() (groupdocs.MergeApi.MergeApi method), 17
- AddJobDocumentUrl() (groupdocs.AsyncApi.AsyncApi method), 11
- AddPredefinedList() (groupdocs.SignatureApi.SignatureApi method), 28
- AddQuestionnaireCollector() (groupdocs.MergeApi.MergeApi method), 17
- AddQuestionnaireExecution() (groupdocs.MergeApi.MergeApi method), 18
- AddQuestionnaireTemplateFields() (groupdocs.MergeApi.MergeApi method), 18
- AddSignatureEnvelopeDocument() (groupdocs.SignatureApi.SignatureApi method), 28
- AddSignatureEnvelopeField() (groupdocs.SignatureApi.SignatureApi method), 28
- AddSignatureEnvelopeRecipient() (groupdocs.SignatureApi.SignatureApi method), 28
- AddSignatureFormDocument() (groupdocs.SignatureApi.SignatureApi method), 28
- AddSignatureFormField() (groupdocs.SignatureApi.SignatureApi method), 29
- AddSignatureTemplateDocument() (groupdocs.SignatureApi.SignatureApi method), 29
- AddSignatureTemplateField() (groupdocs.SignatureApi.SignatureApi method), 29
- AddSignatureTemplateRecipient() (groupdocs.SignatureApi.SignatureApi method), 29
- AddStorageProvider() (groupdocs.MgmtApi.MgmtApi method), 23
- AntApi (class in groupdocs.AntApi), 7
- ApiClient (class in groupdocs.ApiClient), 10
- ApiException, 10
- ArchiveSignatureEnvelope() (groupdocs.SignatureApi.SignatureApi method), 29
- ArchiveSignatureForm() (groupdocs.SignatureApi.SignatureApi method), 29
- AssignSignatureEnvelopeField() (groupdocs.SignatureApi.SignatureApi method), 29
- AssignSignatureTemplateField() (groupdocs.SignatureApi.SignatureApi method), 29
- AsyncApi (class in groupdocs.AsyncApi), 11

B

- basePath (groupdocs.AntApi.AntApi attribute), 10
- basePath (groupdocs.AsyncApi.AsyncApi attribute), 12
- basePath (groupdocs.ComparisonApi.ComparisonApi attribute), 13
- basePath (groupdocs.DocApi.DocApi attribute), 16
- basePath (groupdocs.MergeApi.MergeApi attribute), 23

- basePath (groupdocs.MgmtApi.MgmtApi attribute), 26
- basePath (groupdocs.PostApi.PostApi attribute), 27
- basePath (groupdocs.SharedApi.SharedApi attribute), 28
- basePath (groupdocs.SignatureApi.SignatureApi attribute), 44
- basePath (groupdocs.StorageApi.StorageApi attribute), 46
- basePath (groupdocs.SystemApi.SystemApi attribute), 48
- C**
- callAPI() (groupdocs.ApiClient.ApiClient method), 10
- CancelFileUpload() (groupdocs.StorageApi.StorageApi method), 44
- CancelSignatureEnvelope() (groupdocs.SignatureApi.SignatureApi method), 30
- ChangeAlienUserPassword() (groupdocs.MgmtApi.MgmtApi method), 23
- ChangeUserPassword() (groupdocs.MgmtApi.MgmtApi method), 24
- Compare() (groupdocs.ComparisonApi.ComparisonApi method), 12
- ComparisonApi (class in groupdocs.ComparisonApi), 12
- CompleteSignatureForm() (groupdocs.SignatureApi.SignatureApi method), 30
- Compress() (groupdocs.StorageApi.StorageApi method), 44
- CompressByPost() (groupdocs.PostApi.PostApi method), 26
- contentType (groupdocs.FileStream.FileStream attribute), 16
- Convert() (groupdocs.AsyncApi.AsyncApi method), 11
- CopyFileToTemplates() (groupdocs.MergeApi.MergeApi method), 18
- Create() (groupdocs.StorageApi.StorageApi method), 44
- CreateAnnotation() (groupdocs.AntApi.AntApi method), 7
- CreateAnnotationReply() (groupdocs.AntApi.AntApi method), 7
- CreateDocumentQuestionnaire() (groupdocs.MergeApi.MergeApi method), 18
- CreateJob() (groupdocs.AsyncApi.AsyncApi method), 11
- CreatePackage() (groupdocs.StorageApi.StorageApi method), 44
- CreateQuestionnaire() (groupdocs.MergeApi.MergeApi method), 18
- CreateQuestionnaireTemplate() (groupdocs.MergeApi.MergeApi method), 18
- CreateSignature() (groupdocs.SignatureApi.SignatureApi method), 30
- CreateSignatureEnvelope() (groupdocs.SignatureApi.SignatureApi method), 30
- CreateSignatureField() (groupdocs.SignatureApi.SignatureApi method), 30
- CreateSignatureForm() (groupdocs.SignatureApi.SignatureApi method), 30
- CreateSignatureTemplate() (groupdocs.SignatureApi.SignatureApi method), 30
- CreateUser() (groupdocs.MgmtApi.MgmtApi method), 24
- CreateUserLogin() (groupdocs.MgmtApi.MgmtApi method), 24
- D**
- DeclineEnvelope() (groupdocs.SignatureApi.SignatureApi method), 31
- Decompress() (groupdocs.StorageApi.StorageApi method), 44
- DecorateQuestionnaireCollector() (groupdocs.MergeApi.MergeApi method), 18
- DefaultRequestSigner (class in groupdocs.ApiClient), 10
- DelegateEnvelopeRecipient() (groupdocs.SignatureApi.SignatureApi method), 31
- Delete() (groupdocs.StorageApi.StorageApi method), 44
- DeleteAccount() (groupdocs.MgmtApi.MgmtApi method), 24
- DeleteAccountUser() (groupdocs.MgmtApi.MgmtApi method), 24
- DeleteAnnotation() (groupdocs.AntApi.AntApi method), 7
- DeleteAnnotationReply() (groupdocs.AntApi.AntApi method), 7
- DeleteByPost() (groupdocs.PostApi.PostApi method), 27
- DeleteContact() (groupdocs.SignatureApi.SignatureApi method), 31
- DeleteDataSource() (groupdocs.MergeApi.MergeApi method), 18
- DeleteDataSourceList() (groupdocs.MergeApi.MergeApi method), 19
- DeleteDocumentAnnotations() (groupdocs.AntApi.AntApi method), 7
- DeleteDocumentQuestionnaire() (groupdocs.MergeApi.MergeApi method), 19
- DeleteDocumentReviewer() (groupdocs.AntApi.AntApi method), 8
- DeleteFromFolder() (groupdocs.StorageApi.StorageApi method), 44
- DeleteFromFolderByPost() (groupdocs.PostApi.PostApi method), 27
- DeleteJob() (groupdocs.AsyncApi.AsyncApi method), 11

- DeleteJobDocument() (groupdocs.AsyncApi.AsyncApi method), 11
- DeletePredefinedList() (groupdocs.SignatureApi.SignatureApi method), 31
- DeleteQuestionnaire() (groupdocs.MergeApi.MergeApi method), 19
- DeleteQuestionnaireCollector() (groupdocs.MergeApi.MergeApi method), 19
- DeleteQuestionnaireCollectorList() (groupdocs.MergeApi.MergeApi method), 19
- DeleteQuestionnaireExecution() (groupdocs.MergeApi.MergeApi method), 19
- DeleteQuestionnaireExecutionList() (groupdocs.MergeApi.MergeApi method), 19
- DeleteQuestionnairesList() (groupdocs.MergeApi.MergeApi method), 19
- DeleteQuestionnaireTemplateFields() (groupdocs.MergeApi.MergeApi method), 19
- DeleteSignature() (groupdocs.SignatureApi.SignatureApi method), 31
- DeleteSignatureEnvelope() (groupdocs.SignatureApi.SignatureApi method), 31
- DeleteSignatureEnvelopeDocument() (groupdocs.SignatureApi.SignatureApi method), 31
- DeleteSignatureEnvelopeField() (groupdocs.SignatureApi.SignatureApi method), 31
- DeleteSignatureEnvelopeFieldLocation() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureEnvelopeRecipient() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureField() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureForm() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureFormDocument() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureFormField() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureFormFieldLocation() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureTemplate() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureTemplateDocument() (groupdocs.SignatureApi.SignatureApi method), 32
- DeleteSignatureTemplateField() (groupdocs.SignatureApi.SignatureApi method), 33
- DeleteSignatureTemplateFieldLocation() (groupdocs.SignatureApi.SignatureApi method), 33
- DeleteSignatureTemplateRecipient() (groupdocs.SignatureApi.SignatureApi method), 33
- deserialize() (groupdocs.ApiClient.ApiClient method), 10
- DocApi (class in groupdocs.DocApi), 13
- Download() (groupdocs.SharedApi.SharedApi method), 27
- DownloadResult() (groupdocs.ComparisonApi.ComparisonApi method), 12
- ## E
- EditAnnotationReply() (groupdocs.AntApi.AntApi method), 8
- encodeURI() (groupdocs.ApiClient.ApiClient static method), 10
- encodeURIComponent() (groupdocs.ApiClient.ApiClient static method), 10
- ## F
- fileName (groupdocs.FileStream.FileStream attribute), 17
- FileStream (class in groupdocs.FileStream), 16
- FillEnvelopeField() (groupdocs.SignatureApi.SignatureApi method), 33
- FillExecution() (groupdocs.MergeApi.MergeApi method), 20
- FillQuestionnaire() (groupdocs.MergeApi.MergeApi method), 20
- fromFile() (groupdocs.FileStream.FileStream class method), 17
- fromHttp() (groupdocs.FileStream.FileStream class method), 17
- fromStream() (groupdocs.FileStream.FileStream class method), 17
- ## G
- GenerateKeyForUser() (groupdocs.MgmtApi.MgmtApi method), 24
- get_method() (groupdocs.ApiClient.MethodRequest method), 10
- GetAccount() (groupdocs.MgmtApi.MgmtApi method), 24

GetAccountUsers() (groupdocs.MgmtApi.MgmtApi method), 24
 GetAlienUserProfile() (groupdocs.MgmtApi.MgmtApi method), 24
 GetAnnotationCollaborators() (groupdocs.AntApi.AntApi method), 8
 GetChanges() (groupdocs.ComparisonApi.ComparisonApi method), 13
 GetContacts() (groupdocs.SignatureApi.SignatureApi method), 33
 GetCountries() (groupdocs.SystemApi.SystemApi method), 46
 GetCreditCard() (groupdocs.SystemApi.SystemApi method), 46
 GetDataSource() (groupdocs.MergeApi.MergeApi method), 20
 GetDocumentAccessInfo() (groupdocs.DocApi.DocApi method), 13
 GetDocumentByQuestionnaire() (groupdocs.MergeApi.MergeApi method), 20
 GetDocumentContent() (groupdocs.DocApi.DocApi method), 13
 GetDocumentDetails() (groupdocs.ComparisonApi.ComparisonApi method), 13
 GetDocumentFormats() (groupdocs.DocApi.DocApi method), 13
 GetDocumentHyperlinks() (groupdocs.DocApi.DocApi method), 13
 GetDocumentMetadata() (groupdocs.DocApi.DocApi method), 14
 GetDocumentMetadataByPath() (groupdocs.DocApi.DocApi method), 14
 GetDocumentPageImage() (groupdocs.DocApi.DocApi method), 14
 GetDocumentPagesImageUrls() (groupdocs.DocApi.DocApi method), 14
 GetDocumentQuestionnaires() (groupdocs.MergeApi.MergeApi method), 20
 GetDocumentTags() (groupdocs.DocApi.DocApi method), 14
 GetDocumentViews() (groupdocs.DocApi.DocApi method), 14
 GetEditLock() (groupdocs.DocApi.DocApi method), 14
 GetEnvelopeAuditLogs() (groupdocs.SignatureApi.SignatureApi method), 33
 GetFieldsList() (groupdocs.SignatureApi.SignatureApi method), 33
 GetFile() (groupdocs.StorageApi.StorageApi method), 45
 GetFolderSharers() (groupdocs.DocApi.DocApi method), 15
 GetFormAuditLogs() (groupdocs.SignatureApi.SignatureApi method), 34
 GetHtml() (groupdocs.SharedApi.SharedApi method), 27
 GetInvoices() (groupdocs.SystemApi.SystemApi method), 46
 GetJobDocuments() (groupdocs.AsyncApi.AsyncApi method), 11
 GetJobJson() (groupdocs.AsyncApi.AsyncApi method), 12
 GetJobResources() (groupdocs.AsyncApi.AsyncApi method), 12
 GetJobs() (groupdocs.AsyncApi.AsyncApi method), 12
 GetJobsDocuments() (groupdocs.AsyncApi.AsyncApi method), 12
 GetPackage() (groupdocs.SharedApi.SharedApi method), 27
 GetPossibleConversions() (groupdocs.AsyncApi.AsyncApi method), 12
 GetPurchaseWizardInfo() (groupdocs.SystemApi.SystemApi method), 46
 GetQuestionnaire() (groupdocs.MergeApi.MergeApi method), 20
 GetQuestionnaireByCollector() (groupdocs.MergeApi.MergeApi method), 20
 GetQuestionnaireCollector() (groupdocs.MergeApi.MergeApi method), 20
 GetQuestionnaireCollectorExecutions() (groupdocs.MergeApi.MergeApi method), 20
 GetQuestionnaireCollectors() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaireCollectorStyle() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaireDataSources() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaireExecution() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaireExecutions() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaireFields() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaireMetadata() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnaires() (groupdocs.MergeApi.MergeApi method), 21
 GetQuestionnairesByName() (groupdocs.MergeApi.MergeApi method), 22
 GetRecipientSignatureEnvelopes() (groupdocs.SignatureApi.SignatureApi method), 34
 GetReviewerContacts() (groupdocs.AntApi.AntApi method), 8
 GetRoles() (groupdocs.MgmtApi.MgmtApi method), 24
 GetRolesList() (groupdocs.SignatureApi.SignatureApi method), 34

GetSharedDocuments()	(groupdocs.DocApi.DocApi method), 15	GetSignatures()	(groupdocs.SignatureApi.SignatureApi method), 37
GetSharedFile()	(groupdocs.StorageApi.StorageApi method), 45	GetSignatureTemplate()	(groupdocs.SignatureApi.SignatureApi method), 36
GetSharedLinkAccessRights()	(groupdocs.AntApi.AntApi method), 8	GetSignatureTemplateDocuments()	(groupdocs.SignatureApi.SignatureApi method), 36
GetSignatureEnvelope()	(groupdocs.SignatureApi.SignatureApi method), 34	GetSignatureTemplateFields()	(groupdocs.SignatureApi.SignatureApi method), 36
GetSignatureEnvelopeDocuments()	(groupdocs.SignatureApi.SignatureApi method), 34	GetSignatureTemplateRecipients()	(groupdocs.SignatureApi.SignatureApi method), 36
GetSignatureEnvelopeFieldData()	(groupdocs.SignatureApi.SignatureApi method), 34	GetSignatureTemplateResources()	(groupdocs.SignatureApi.SignatureApi method), 36
GetSignatureEnvelopeFields()	(groupdocs.SignatureApi.SignatureApi method), 34	GetSignatureTemplates()	(groupdocs.SignatureApi.SignatureApi method), 36
GetSignatureEnvelopeRecipients()	(groupdocs.SignatureApi.SignatureApi method), 34	GetSignDocumentStatus()	(groupdocs.SignatureApi.SignatureApi method), 34
GetSignatureEnvelopeResources()	(groupdocs.SignatureApi.SignatureApi method), 34	GetSignedEnvelopeDocument()	(groupdocs.SignatureApi.SignatureApi method), 37
GetSignatureEnvelopes()	(groupdocs.SignatureApi.SignatureApi method), 35	GetSignedEnvelopeDocuments()	(groupdocs.SignatureApi.SignatureApi method), 37
GetSignatureForm()	(groupdocs.SignatureApi.SignatureApi method), 35	GetSignedFormDocuments()	(groupdocs.SignatureApi.SignatureApi method), 37
GetSignatureFormDocumentFields()	(groupdocs.SignatureApi.SignatureApi method), 35	GetStates()	(groupdocs.SystemApi.SystemApi method), 46
GetSignatureFormDocuments()	(groupdocs.SignatureApi.SignatureApi method), 35	GetStorageInfo()	(groupdocs.StorageApi.StorageApi method), 45
GetSignatureFormFields()	(groupdocs.SignatureApi.SignatureApi method), 35	GetStorageProviders()	(groupdocs.MgmtApi.MgmtApi method), 25
GetSignatureFormParticipant()	(groupdocs.SignatureApi.SignatureApi method), 35	GetSubscriptionPlans()	(groupdocs.SystemApi.SystemApi method), 47
GetSignatureFormParticipants()	(groupdocs.SignatureApi.SignatureApi method), 35	GetSubscriptionPlanUsage()	(groupdocs.SystemApi.SystemApi method), 47
GetSignatureFormResources()	(groupdocs.SignatureApi.SignatureApi method), 35	GetTemplateEditorFields()	(groupdocs.DocApi.DocApi method), 15
GetSignatureForms()	(groupdocs.SignatureApi.SignatureApi method), 36	GetTemplateFields()	(groupdocs.DocApi.DocApi method), 15
GetSignaturePredefinedLists()	(groupdocs.SignatureApi.SignatureApi method), 36	GetTemplates()	(groupdocs.MergeApi.MergeApi method), 22
		GetTermSuggestions()	(groupdocs.SystemApi.SystemApi method), 47
		GetUserEmailTemplate()	(groupdocs.SignatureApi.SignatureApi method), 37

GetUserEmbedKey() (groupdocs.MgmtApi.MgmtApi method), 25	MergeDatasource() (groupdocs.MergeApi.MergeApi method), 22
GetUserEmbedKeyFromGuid() (groupdocs.MgmtApi.MgmtApi method), 25	MergeDatasourceFields() (groupdocs.MergeApi.MergeApi method), 22
GetUserPlan() (groupdocs.SystemApi.SystemApi method), 47	MethodRequest (class in groupdocs.ApiClient), 10
GetUserProfile() (groupdocs.MgmtApi.MgmtApi method), 25	MgmtApi (class in groupdocs.MgmtApi), 23
GetUserProfileByClaimedToken() (groupdocs.MgmtApi.MgmtApi method), 25	ModifyContact() (groupdocs.SignatureApi.SignatureApi method), 37
GetUserProfileByResetToken() (groupdocs.MgmtApi.MgmtApi method), 25	ModifySignatureEnvelope() (groupdocs.SignatureApi.SignatureApi method), 37
GetUserProfileByVerifToken() (groupdocs.MgmtApi.MgmtApi method), 25	ModifySignatureEnvelopeField() (groupdocs.SignatureApi.SignatureApi method), 37
GetUserRoles() (groupdocs.MgmtApi.MgmtApi method), 25	ModifySignatureEnvelopeFieldLocation() (groupdocs.SignatureApi.SignatureApi method), 38
GetUserSubscriptionPlan() (groupdocs.SystemApi.SystemApi method), 47	ModifySignatureEnvelopeFieldLocationOrder() (groupdocs.SignatureApi.SignatureApi method), 38
GetXml() (groupdocs.SharedApi.SharedApi method), 27	ModifySignatureEnvelopeRecipient() (groupdocs.SignatureApi.SignatureApi method), 38
groupdocs.AntApi (module), 7	ModifySignatureField() (groupdocs.SignatureApi.SignatureApi method), 38
groupdocs.ApiClient (module), 10	ModifySignatureForm() (groupdocs.SignatureApi.SignatureApi method), 38
groupdocs.AsyncApi (module), 11	ModifySignatureFormDocument() (groupdocs.SignatureApi.SignatureApi method), 38
groupdocs.ComparisonApi (module), 12	ModifySignatureFormField() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.DocApi (module), 13	ModifySignatureFormFieldLocation() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.FileStream (module), 16	ModifySignatureTemplate() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.GroupDocsRequestSigner (module), 17	ModifySignatureTemplateField() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.MergeApi (module), 17	ModifySignatureTemplateFieldLocation() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.MgmtApi (module), 23	ModifySignatureTemplateRecipient() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.PostApi (module), 26	ModifySignatureTemplateRecipient() (groupdocs.SignatureApi.SignatureApi method), 39
groupdocs.SharedApi (module), 27	MoveAnnotation() (groupdocs.AntApi.AntApi method), 8
groupdocs.SignatureApi (module), 28	MoveAnnotationMarker() (groupdocs.AntApi.AntApi method), 8
groupdocs.StorageApi (module), 44	
groupdocs.SystemApi (module), 46	
GroupDocsRequestSigner (class in groupdocs.GroupDocsRequestSigner), 17	
I	
ImportContacts() (groupdocs.SignatureApi.SignatureApi method), 37	
inputStream (groupdocs.FileStream.FileStream attribute), 17	
L	
ListAnnotationReplies() (groupdocs.AntApi.AntApi method), 8	
ListAnnotations() (groupdocs.AntApi.AntApi method), 8	
ListEntities() (groupdocs.StorageApi.StorageApi method), 45	
LoginUser() (groupdocs.SharedApi.SharedApi method), 27	
M	
MergeApi (class in groupdocs.MergeApi), 17	

MoveFile() (groupdocs.StorageApi.StorageApi method), 45	PublicGetSignedFormDocuments() docs.SignatureApi.SignatureApi (group-method), 41
MoveFolder() (groupdocs.StorageApi.StorageApi method), 45	PublicModifyFormParticipant() docs.SignatureApi.SignatureApi (group-method), 41
MoveToTrash() (groupdocs.StorageApi.StorageApi method), 45	PublicSignDocument() docs.SignatureApi.SignatureApi (group-method), 41
P	PublicSignEnvelope() docs.SignatureApi.SignatureApi (group-method), 41
PostApi (class in groupdocs.PostApi), 26	PublicSignForm() docs.SignatureApi.SignatureApi (group-method), 42
PublicFillEnvelopeField() docs.SignatureApi.SignatureApi (group-method), 39	PublicValidateFormParticipant() docs.SignatureApi.SignatureApi (group-method), 42
PublicFillFormField() docs.SignatureApi.SignatureApi (group-method), 40	PublicVerifyDocument() docs.SignatureApi.SignatureApi (group-method), 42
PublicFillSignatureForm() docs.SignatureApi.SignatureApi (group-method), 40	PublishSignatureForm() docs.SignatureApi.SignatureApi (group-method), 42
PublicFormParticipantResentValidationCode() docs.SignatureApi.SignatureApi (group-method), 40	R
PublicGetDefaultEmailTemplate() docs.SignatureApi.SignatureApi (group-method), 40	readAsDataURL() (groupdocs.ApiClient.ApiClient static method), 10
PublicGetDocumentFields() docs.SignatureApi.SignatureApi (group-method), 40	RemoveAccount() (groupdocs.MgmtApi.MgmtApi method), 25
PublicGetEnvelopeDocuments() docs.SignatureApi.SignatureApi (group-method), 40	RemoveDocumentTags() (groupdocs.DocApi.DocApi method), 15
PublicGetEnvelopeRecipients() docs.SignatureApi.SignatureApi (group-method), 40	RemoveEditLock() (groupdocs.DocApi.DocApi method), 15
PublicGetSignatureEnvelope() docs.SignatureApi.SignatureApi (group-method), 40	RenameByPost() (groupdocs.PostApi.PostApi method), 27
PublicGetSignatureEnvelopeFields() docs.SignatureApi.SignatureApi (group-method), 41	RenameSignatureEnvelope() docs.SignatureApi.SignatureApi (group-method), 42
PublicGetSignatureForm() docs.SignatureApi.SignatureApi (group-method), 41	RenameSignatureEnvelopeDocument() docs.SignatureApi.SignatureApi (group-method), 42
PublicGetSignatureFormDocuments() docs.SignatureApi.SignatureApi (group-method), 41	RenameSignatureForm() docs.SignatureApi.SignatureApi (group-method), 42
PublicGetSignatureFormFields() docs.SignatureApi.SignatureApi (group-method), 41	RenameSignatureFormDocument() docs.SignatureApi.SignatureApi (group-method), 42
PublicGetSignDocumentStatus() docs.SignatureApi.SignatureApi (group-method), 40	RenameSignatureTemplate() docs.SignatureApi.SignatureApi (group-method), 43
PublicGetSignedEnvelopeDocuments() docs.SignatureApi.SignatureApi (group-method), 41	RenameSignatureTemplateDocument() docs.SignatureApi.SignatureApi (group-method), 43

RequestSigner (class in groupdocs.ApiClient), 11
 ResendSignatureEnvelopeEmailNotification() (groupdocs.SignatureApi.SignatureApi method), 43
 ResetUserPassword() (groupdocs.MgmtApi.MgmtApi method), 26
 ResizeAnnotation() (groupdocs.AntApi.AntApi method), 9
 RestartExpiredSignatureEnvelope() (groupdocs.SignatureApi.SignatureApi method), 43
 RestoreFromTrash() (groupdocs.StorageApi.StorageApi method), 45
 RetrySignEnvelope() (groupdocs.SignatureApi.SignatureApi method), 43
 Revoke() (groupdocs.MgmtApi.MgmtApi method), 26

S

sanitizeForSerialization() (groupdocs.ApiClient.ApiClient method), 10
 SaveTextField() (groupdocs.AntApi.AntApi method), 9
 SetAnnotationAccess() (groupdocs.AntApi.AntApi method), 9
 SetAnnotationCollaborators() (groupdocs.AntApi.AntApi method), 9
 SetBillingAddress() (groupdocs.SystemApi.SystemApi method), 47
 SetCreditCard() (groupdocs.SystemApi.SystemApi method), 47
 setDebug() (groupdocs.ApiClient.ApiClient method), 10
 SetDocumentAccessMode() (groupdocs.DocApi.DocApi method), 15
 SetDocumentPassword() (groupdocs.DocApi.DocApi method), 15
 SetDocumentTags() (groupdocs.DocApi.DocApi method), 15
 SetDocumentUserStatus() (groupdocs.DocApi.DocApi method), 16
 SetReviewerContacts() (groupdocs.AntApi.AntApi method), 9
 SetReviewerRights() (groupdocs.AntApi.AntApi method), 9
 SetSessionCallbackUrl() (groupdocs.AntApi.AntApi method), 9
 SetSharedLinkAccessRights() (groupdocs.AntApi.AntApi method), 9
 SetSubscriptionPlan() (groupdocs.SystemApi.SystemApi method), 47
 SetTextFieldColor() (groupdocs.AntApi.AntApi method), 10
 SetUserRoles() (groupdocs.MgmtApi.MgmtApi method), 26
 SharedApi (class in groupdocs.SharedApi), 27

ShareDocument() (groupdocs.DocApi.DocApi method), 16
 ShareFolder() (groupdocs.DocApi.DocApi method), 16
 SignatureApi (class in groupdocs.SignatureApi), 28
 SignatureEnvelopeSend() (groupdocs.SignatureApi.SignatureApi method), 43
 signContent() (groupdocs.ApiClient.DefaultRequestSigner method), 10
 signContent() (groupdocs.ApiClient.RequestSigner method), 11
 signContent() (groupdocs.GroupDocsRequestSigner.GroupDocsRequestSigner method), 17
 SignDocument() (groupdocs.SignatureApi.SignatureApi method), 43
 SignEnvelope() (groupdocs.SignatureApi.SignatureApi method), 43
 signUrl() (groupdocs.ApiClient.DefaultRequestSigner method), 10
 signUrl() (groupdocs.ApiClient.RequestSigner method), 11
 signUrl() (groupdocs.GroupDocsRequestSigner.GroupDocsRequestSigner method), 17
 SimulateAssessForPaymentPlan() (groupdocs.SystemApi.SystemApi method), 47
 SimulateAssessForPricingPlan() (groupdocs.SystemApi.SystemApi method), 47
 size (groupdocs.FileStream.FileStream attribute), 17
 StorageApi (class in groupdocs.StorageApi), 44
 SystemApi (class in groupdocs.SystemApi), 46

T

toPathValue() (groupdocs.ApiClient.ApiClient method), 10

U

UnshareDocument() (groupdocs.DocApi.DocApi method), 16
 UnshareFolder() (groupdocs.DocApi.DocApi method), 16
 UpdateAccountUser() (groupdocs.MgmtApi.MgmtApi method), 26
 UpdateAlienUserProfile() (groupdocs.MgmtApi.MgmtApi method), 26
 UpdateChanges() (groupdocs.ComparisonApi.ComparisonApi method), 13
 UpdateDataSource() (groupdocs.MergeApi.MergeApi method), 22
 UpdateDataSourceFields() (groupdocs.MergeApi.MergeApi method), 22
 UpdateEnvelopeFromTemplate() (groupdocs.SignatureApi.SignatureApi method), 43

UpdateJob() (groupdocs.AsyncApi.AsyncApi method),
12

UpdateQuestionnaire() (groupdocs.MergeApi.MergeApi
method), 22

UpdateQuestionnaireCollector() (group-
docs.MergeApi.MergeApi method), 23

UpdateQuestionnaireExecution() (group-
docs.MergeApi.MergeApi method), 23

UpdateQuestionnaireExecutionStatus() (group-
docs.MergeApi.MergeApi method), 23

UpdateQuestionnaireMetadata() (group-
docs.MergeApi.MergeApi method), 23

UpdateQuestionnaireTemplateFields() (group-
docs.MergeApi.MergeApi method), 23

UpdateSignatureFormFromTemplate() (group-
docs.SignatureApi.SignatureApi method),
44

UpdateStorageProvider() (group-
docs.MgmtApi.MgmtApi method), 26

UpdateSubscriptionPlan() (group-
docs.SystemApi.SystemApi method), 48

UpdateUserProfile() (groupdocs.MgmtApi.MgmtApi
method), 26

Upload() (groupdocs.StorageApi.StorageApi method), 46

UploadGoogle() (groupdocs.StorageApi.StorageApi
method), 46

UploadWeb() (groupdocs.StorageApi.StorageApi
method), 46

V

ViewDocument() (groupdocs.DocApi.DocApi method),
16