

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

API Reference

The ImageRight API is organized around REST (http://en.wikipedia.org/wiki/Representational_state_transfer). Our REST API is designed to have predictable, resource-oriented URLs and to use HTTP response codes to indicate API errors. We use built-in HTTP features, like HTTP authentication and HTTP verbs, which can be understood by off-the-shelf HTTP clients.

Access to resources is available by sending an HTTP request to the ImageRight Application Server. The server replies with a response that either contains the data you requested, or a status indicator, or both. JSON or XML responses are returned based on the requested content type.

Authentication

ImageRight security infrastructure ensures that only the intended people are able to access the files, folders, documents, pages, tasks dairies, etc. All methods are available only for authenticated users if another was not specified.

An initial post request should be made to `api/authenticate` with username and password to get an access token. The access token should be submitted as part of the authorization header.

```
Authorization: AccessToken QWxhZGRpbjpvcGVuIHN1c2FtZQ==
```

ImageRight also supports ADFS-based authentication for SAML11 and SAML2 tokens. A serialized security token should be submitted as part of the authorization header in base64 format.

```
Authorization: SecurityToken QWxhZGRpbjpvcGVuIHN1c2FtZQ==
```

ImageRight supports JSON Web Tokens (defined by RFC 7519 (<https://tools.ietf.org/html/rfc7519>) standard) signed with Certificate. Token is used to authenticate and authorize user for every request.

Token payload includes next properties:

- "sub" - identifies ImageRight user name which will be authorized within request
- "nbf" - identifies the time before which the token MUST NOT be accepted
- "exp" - identifies the expiration time after which the token MUST NOT be accepted for processing
- "iat" - identifies the time at which the JWT was issued
- "iss" - identifies the party that this token was issued by
- "aud" - identifies the party that this token was issued for

Properties "sub", "nbf", "exp", "iss" and "aud" are taken into account during validation.

Token must be signed with private key of Certificate using RS256 algorithm. JSON Web Signature defined by RFC 7515 (<https://tools.ietf.org/html/rfc7515>) standard. The public key of this Certificate is used on ImageRight side to ensure that token received from the party that has authorization to use ImageRight API.

The JSON web token represents base64 encoded string that should be submitted as part of the authorization header.

```
Authorization: JWT eyJxOjc0NMzdBNUEiLCJ0eXAiOiJQifQ.eyJzdWIiOiJJbGcmNvbiIsImF1IjIkmInaHqifQ.RzpJSKzzz92vp1wpG4NN
```

Status and error codes

ImageRight uses subset of conventional HTTP status codes to indicate success or failure of an API request.

- 200 - OK
- 201 - Created
- 400 - Bad Request
- 401 - Unauthorized

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

- 403 - Forbidden
- 404 - Not Found

- 500 - Internal Server Error

In case of 400 response status code additional info about error can be retrieved from the body of the response:

```
{
  "Message": "Some error occurred",
  "ErrorCode": 123
}
```

List of possible error codes included into each method description.

Push Notifications

ImageRight allows a connection to be established via SignalR to subscribe to interested topics and receive push notifications whenever an event occurs. Message delivery is not guaranteed and when all SignalR connections are removed, the corresponding subscriptions will be removed. In the event of an interruption in service or an interruption in the consuming client, the notifications are not queued.

An initial post request should be made to the standard REST API route `api/authenticate` with the username and password of the subscriber to get an access token. The access token should be submitted as part of the query string when connecting to SignalR. The documentation for adding a query string parameter can be found at the help page for the respective SignalR client from Microsoft. The JavaScript documentation, for example, is located here: <http://www.asp.net/signalr/overview/guide-to-the-api/hubs-api-guide-javascript-client#querystring> (`http://www.asp.net/signalr/overview/guide-to-the-api/hubs-api-guide-javascript-client#querystring`) The query string parameter name should be "AccessToken". The parameter value should be the access token received, with URL encoding such that any characters such as plus sign characters are encoded.

The SignalR route is "{appServerRoute}:8093/SignalR". For example, if the REST API Help Page is located at "https://{machineName}:8093/api/help", the SignalR route will be located at "https://{machineName}:8093/signalr".

Errors in invoking hub methods will throw an exception upon an access token validation error, a cache lock taking more than 10 seconds, or server errors (such as network connectivity issues in a clustered application server environment).

Hubs

- Hub Name: "NotificationHub" | Methods Available to Invoke
 - `Subscribe(SubscriptionModel subscriptionModel)`

This method is used to add a subscription for the current user. A valid subscription will always be added, but if the user does not have read permissions on notifications that match the subscription, they will not receive notifications about those objects.

Parameter: A SubscriptionModel is an object with the properties

 - Enum: ActionType (Create = 0, Update = 1, Delete = 2, Copy = 3, Move = 4)
 - Nullable long: DocumentType
 - Nullable long: DrawerType
 - Nullable long: FileType
 - Enumerable of long: FolderTypes
 - bool: MatchAllFolderTypes

This object represents the topics you are interested in subscribing to. For example, if you wanted to be notified whenever any object is created that has a parent with a filetype of 5, then you would pass in an ActionType property of 0, with a FileType property of 5. Multiple types are additional constraints, so passing in a filetype of 5 and a drawertype of 8 means that you will only be notified of objects that have both a filetype of 5 and a drawertype of 8 for its parents. For folder types, the boolean MatchAllFolderTypes determines whether all of the folder types have to be present in the parents of the affected object when set to true (e.g. foldertypes of 10 and 11 mean that you will be notified about objects with a folder with type 10 and a folder of type 11 in its parent path), or any one is sufficient when set to false (e.g. if either 10 or 11 are present but not necessarily both, you are interested in being notified about it).

Returns: A GUID (System.Guid object) representing the subscription that was added.
 - `Unsubscribe(Guid subscriptionGuid)`

This method is used to remove a subscription for the current user.

Returns: True if it was removed successfully or the subscription did not exist, false otherwise.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

- UnsubscribeAll()
This method is used to remove all subscriptions for the current user.
Returns: True if all subscriptions were removed successfully, false otherwise.
- UpdateAccessToken()
This method is used to update the current access token. This method needs to be called periodically using a new AccessToken, before the access token initially used to connect to SignalR expires. Otherwise, when the access token expires, all subscriptions will be invalidated and removed.
Returns: True if successful, false otherwise.
- GetAllSubscriptions()
This method is used to get a list of Subscription objects that the user is currently subscribed to.
Returns: An list of objects with the properties
 - Enum: ActionType (Create = 0, Update = 1, Delete = 2, Copy = 3, Move = 4)
 - Nullable long: DocumentType
 - Nullable long: DrawerType
 - Nullable long: FileType
 - Enumerable of long: FolderTypes
 - bool: MatchAllFolderTypes
 - Guid: SubscriptionGuid

The client contract that needs to be implemented is the following:

- void ReceiveNotification(NotificationModel notification)
The method that implements your action upon receiving a notification. The NotificationModel is an object that contains the following
 - Nullable long: BatchId
 - long: Id
 - Enum: NotificationType (Create = 0, Update = 1, Delete = 2, Copy = 3, Move = 4)
 - Enum: ObjectClass (File = 0, Folder = 1, Document = 2, Page = 3)
 - Nullable long: OldId
 - Nullable long: OldParentId
 - long: ParentId
- void Disconnect()
A method that closes the SignalR connection to the server. This method be implemented to prevent any possible issues with leaking connections, the connection cannot be closed from the server side.

Accounts

Accounts object allows you to get current user account, members of account

AddMemberToGroup: Adds a member(group or user) to an ImageRight group.

Method:	POST
URL:	api/accounts/groups/{groupid}/members/add
Parameters:	
Name	groupid
Description	The id of the group
Name	accountid
Description	The id of the account to add as a member of the specified group
Request:	api/accounts/groups/1/members/add
	2
Response:	200 OK
Error codes:	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

11	The account was not found.
30	The provided data are invalid for this operation.

AssignGroupToGroup: Allows the assigned group to inherit the permissions of the parent group while also maintaining permissions which are granted to the child group. These permissions are cumulative which means the group has more than one set of permissions. If a permission is explicitly denied in one location, that user is denied at all levels.

Method:	POST
URL:	api/accounts/groups/{groupIdParent}/assigngroup/{groupIdChild}
Parameters:	
Name	groupIdParent
Description	The group Id of the parent group.
Name	groupIdChild
Description	The group Id of the child group.

Request:	api/accounts/groups/1/assigngroup/2
Response:	200 OK

Error codes:	
1403	The group is not found.
1400	The group already exists.

AssignGroupToRole: A role is a collection of permissions that are standard across users within a specified job function. Groups can be added to roles and will be granted the permissions of the assigned role. These permissions are cumulative which means the user has more than one set of permissions. If a permission is explicitly denied in one location, that user is denied at all levels.

Method:	POST
URL:	api/accounts/groups/{groupId}/assignrole/{roleId}
Parameters:	
Name	groupId
Description	The group Id.
Name	roleId
Description	The role Id.

Request:	api/accounts/groups/1/assignrole/2
Response:	200 OK

Error codes:	
1403	The group is not found.
1402	The role is not found.
1401	The role already exists.

CreateGroup: Allows the creation of a group within ImageRight. Groups can be assigned permissions and are a collection of users within ImageRight. When users are assigned to groups, they inherit the permissions of that group.

Method:	POST
URL:	api/accounts/groups

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Parameters:

Name	groupModel
Description	The data for creating a new group.

Request: api/accounts/groups

```
{
  "Name": "NameStr1",
  "Description": "DescriptionStr2"
}
```

Response: 201 Created

```
1
```

Error codes:

1400	The group already exists.
-------------	---------------------------

CreateRole: Allows the creation of a role within ImageRight. A role is a collection of permissions that are standard across users within a specified job function.

Method: POST
URL: api/accounts/roles

Parameters:

Name	roleModel
Description	The data for creating a new role.

Request: api/accounts/roles

```
{
  "Name": "NameStr1",
  "Description": "DescriptionStr2"
}
```

Response: 201 Created

```
1
```

Error codes:

1401	The role already exists.
-------------	--------------------------

GetAccount: Returns account data for users, groups or roles identified in the ImageRight system.

Method: GET
URL: api/accounts/{accountId}?type={type}

Parameters:

Name	accountId
Description	The account Id.
Name	type
Description	Type of members retrieval. Optional , default value is null (empty) .
Values	List, DirectMembers, AllMembers

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Request: api/accounts/1?type=list

Response: 200 OK

```
{
  "Id": 1,
  "ExternalId": "ExternalIdStr2",
  "Name": "NameStr3",
  "FriendlyName": "FriendlyNameStr4",
  "Type": "user",
  "Enabled": true,
  "Description": "DescriptionStr6"
}
```

Error codes:

11 The account was not found.

GetAccountGroups: Returns a list of groups in which a specified ImageRight user or group is member of.

Method: GET

URL: api/accounts/{accountId}/groups

Parameters:

Name	accountId
Description	The account Id.

Request: api/accounts/accountIdStr1/groups

Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

Error codes:

11 The account was not found.
221 User id parameter is invalid.

GetAccounts: Returns all members(including disabled) of users, groups or roles accounts identified in the ImageRight system

Method: GET

URL: api/accounts/members?accountsId[0]={accountsId[0]}&accountsId[1]={accountsId[1]}

Parameters:

Name	accountsId
Description	The array accounts Ids.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Request: `api/accounts/members?accountsId=system.int32[]`

Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

Error codes:

11 The account was not found.

GetAllAccounts: Returns the list of accounts within the ImageRight system including users, groups and roles.

Method: GET

URL: `api/accounts`

Request: `api/accounts`

Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

Error codes:

15 The data you requested exists but is not ready now. Try again later.

GetCurrentUserAccount: Returns ImageRight account information of the current ImageRight user.

Method: GET

URL: `api/accounts/current`

Request: `api/accounts/current`

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Id": 1,
  "ExternalId": "ExternalIdStr2",
  "Name": "NameStr3",
  "FriendlyName": "FriendlyNameStr4",
  "Type": "user",
  "Enabled": true,
  "Description": "DescriptionStr6"
}
```

GetCurrentUserGroups: Returns a list of ImageRight groups that the current user is a member of.

Method: GET
URL: api/accounts/current/groups
Request: api/accounts/current/groups
Response: 200 OK

```
{
  "Id": 1,
  "ExternalId": "ExternalIdStr2",
  "Name": "NameStr3",
  "FriendlyName": "FriendlyNameStr4",
  "Type": "user",
  "Enabled": true,
  "Description": "DescriptionStr6"
}
```

GetFullAccountsTree: Get all accounts as tree. Users that are not in any group will be added to "Individuals"

Method: GET
URL: api/accounts/tree
Request: api/accounts/tree
Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Members": [
      {
        "Members": [
          {
            "Members": [],
            "Id": 25,
            "ExternalId": "ExternalIdStr26",
            "Name": "NameStr27",
            "FriendlyName": "FriendlyNameStr28",
            "Type": "user",
            "Enabled": true,
            "Description": "DescriptionStr30"
          },
          { ... },
          { ... }
        ],
        "Id": 31,
        "ExternalId": "ExternalIdStr32",
        "Name": "NameStr33",
        "FriendlyName": "FriendlyNameStr34",
        "Type": "user",
        "Enabled": true,
        "Description": "DescriptionStr36"
      },
      { ... },
      { ... }
    ],
    "Id": 37,
    "ExternalId": "ExternalIdStr38",
    "Name": "NameStr39",
    "FriendlyName": "FriendlyNameStr40",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr42"
  },
  { ... },
  { ... }
]
```

Error codes:

11	The account was not found.
----	----------------------------

GetFunctionalityRights: Gets a list of functionality rights and whether or not they are valid for the current user

Method: GET
URL: api/accounts/functionalityrights
Request: api/accounts/functionalityrights
Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Name": "NameStr1",
    "DisplayName": "DisplayNameStr2",
    "IsAllowed": true,
    "IsDenied": true
  },
  {...},
  {...}
]
```

GetGroups: Returns a list of all ImageRight groups.

Method: GET
URL: api/accounts/groups
Request: api/accounts/groups
Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

Error codes:

11	The account was not found.
221	User id parameter is invalid.

GetRootGroups: Returns a list of groups that are not members of any other groups in the ImageRight system. If a group is a member of another group, the parent will be returned and all child groups will be excluded from this list.

Method: POST
URL: api/accounts/groups/roots
Parameters:

Name	includedAccounts
Description	Not Available

Request: api/accounts/groups/roots

```
[
  "Str1",
  {...},
  {...}
]
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

GetUsersThatAreNotInAnyGroup: Returns a list of all ImageRight users that do not belong to any groups or roles.

Method: GET

URL: api/accounts/notInGroup

Request: api/accounts/notInGroup

Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

Error codes:

15

The data you requested exists but is not ready now. Try again later.

MapAccount: Maps an external account to the internal security system and returns the internal id.

Method: POST

URL: api/accounts/{externalid}/map

Parameters:

Name externalid

Description The externalid id of the account to be mapped

Request: api/accounts/externalidStr1/map

Response: 200 OK

```
1
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Error codes:

11

The account was not found.

30

The provided data are invalid for this operation.

RemoveMemberFromGroup:

Removes a member(group or user) from an ImageRight group.

Method:

POST

URL:

api/accounts/groups/{groupid}/members/remove

Parameters:

Name

groupid

Description

The id of the group

Name

accountid

Description

The id of the account to add as a member of the specified group

Request:

api/accounts/groups/1/members/remove

2

Response:

200 OK

Error codes:

11

The account was not found.

30

The provided data are invalid for this operation.

UnassignGroupFromGroup:

Removes an ImageRight group from another group that it is currently a member of. This will also remove the inherited permissions that were previously granted through the parent group.

Method:

POST

URL:

api/accounts/groups/{groupIdParent}/unassigngroup/{groupIdChild}

Parameters:

Name

groupIdParent

Description

The group Id of the parent group.

Name

groupIdChild

Description

The group Id of the child group.

Request:

api/accounts/groups/1/unassigngroup/2

Response:

200 OK

Error codes:

1403

The group is not found.

UnassignGroupFromRole:

Removes an ImageRight group from an ImageRight role. This will also remove the inherited permissions that were previously granted through the assigned role.

Method:

POST

URL:

api/accounts/groups/{groupId}/unassignrole/{roleId}

Parameters:

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Name	groupId
Description	The group Id.

Request: api/accounts/groups/1/unassignrole/2
Response: 200 OK

Error codes:	
1403	The group is not found.
1402	The role is not found.

ValidatePermission: Returns whether or not a functionality right for the current user is validated. Rights may be set in the ImageRight Enterprise Management Console via the functionality rights feature under user management.

Method:	GET
URL:	api/accounts/functionalityrights/validate?funcRightName={funcRightName}
Parameters:	
Name	funcRightName
Description	The name of the functionality right to validate. Example: WebAccess-OCRFormsManager

Request: api/accounts/functionalityrights/validate?funcRightName=funcRightNameStr1
Response: 200 OK

Error codes:	
1500	The functionality right is not found.

Attributes

Attributes controller that allows you to retrieve attributes.

GetAttributeByAttId: Gets a single attribute from a specific object by the attribute ID.

Method:	GET
URL:	api/containers/{objId}/attributes/{attId}
Parameters:	
Name	objId
Description	Object ID to get the attribute data from.
Name	attId
Description	The Attribute ID.

Request: api/containers/1/attributes/2
Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "DisplayName": "DisplayNameStr1",
  "Name": "NameStr2",
  "Value": {...},
  "AttributeType": "atInt"
}
```

Error codes:

602	The attribute is not allowed.
600	The attribute was not found.
601	The attribute is required.
603	The attribute value is invalid.

GetAttributeByAttName: Gets a single attribute from a specific object by the attribute name.

Method: GET

URL: `api/containers/{objId}/attributes/{attName}`

Parameters:

Name	objId
Description	Object ID to get the attribute data from.
Name	attName
Description	The attribute name.

Request: `api/containers/1/attributes/attNameStr2`

Response: 200 OK

```
{
  "DisplayName": "DisplayNameStr1",
  "Name": "NameStr2",
  "Value": {...},
  "AttributeType": "atInt"
}
```

Error codes:

602	The attribute is not allowed.
600	The attribute was not found.
601	The attribute is required.
603	The attribute value is invalid.

GetAttributesByObject: Gets all the attributes on a specific object.

Method: GET

URL: `api/attributes/{objId}`

Parameters:

Name	objId
Description	Object ID to get attributes from.

Request: `api/attributes/1`

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "DisplayName": "DisplayNameStr1",
    "Name": "NameStr2",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
]
```

Error codes:		
602		The attribute is not allowed.
600		The attribute was not found.
601		The attribute is required.
603		The attribute value is invalid.

GetAttributesDefinitions: Get definitions for all attributes

Method:	GET
URL:	api/attributes/definitions
Request:	api/attributes/definitions
Response:	200 OK

```
[
  {
    "DisplayName": "DisplayNameStr1",
    "Name": "NameStr2",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
]
```

Error codes:		
602		The attribute is not allowed.
600		The attribute was not found.
601		The attribute is required.
603		The attribute value is invalid.

Authentication

Authentication object allows you authenticate with REST API server to get Security Token. Required before making any other calls.

Authenticate: Returns an authentication token by user name and password. The resulting token must be included within the headers of every other API call.

This method is available anonymously.

Method:	POST
URL:	api/authenticate
Parameters:	
Name	userModel

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description

The user model. Username and Password are both required.

Request:

api/authenticate

```
{  "UserName": "UserNameStr1",  "Password": "PasswordStr2"}
```

Response:

200 OK

"QWxhZGRpbjpvcGVuIHN1c2FtZQ=="

Error codes:

1

The username and/or the password is incorrect.

3

Account is locked. Try again later.

4

Password expired.

Authenticate:

Returns an authentication token by special token. The resulting token must be included within the headers of every other API call.

This method is available anonymously.

Method:

POST

URL:

api/authenticate/token

Parameters:

Name

token

Description

The special short lived token for authentication.

Request:

api/authenticate/token

"Str1"

Response:

200 OK

"QWxhZGRpbjpvcGVuIHN1c2FtZQ=="

Error codes:

1

The username and/or the password is incorrect.

3

Account is locked. Try again later.

4

Password expired.

ValidTo:

No Help Available

This method is available anonymously.

Method:

POST

URL:

api/validto

Request:

api/validto

Response:

200 OK

"2019-04-30T07:10:47.7597219-04:00"

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Error codes:

24

The specified type is not valid for this operation.

Batches

Batches object allows you to create, update, delete and retrieve list of Batches.

CreateBatch: Creates a batch.

Method: POST

URL: api/batches

Parameters:

Name

batchModel

Description

The model for creating a batch. Application is required - DateTime is optional.

Request: api/batches

```
{
  "Application": "NoSelection",
  "CapturedDate": "2019-04-30T07:10:47.7247219-04:00"
}
```

Response: 201 Created

```
1
```

ClientActions

Controller for actions taken by the client that need to be recorded in Event Store

CreateClientActionEvents: Recieves a list of actions taken by the client and records those actions in Event Store

Method: POST

URL: api/clientactions

Parameters:

Name

clientActions

Description

Not Available

Request: api/clientactions

```
[
  {
    "Id": 1,
    "Description": "ContentPrinted",
    "Instance": "Page"
  },
  {...},
  {...}
]
```

Response: 200 OK

CreateFileClosedEvents: Recieves an array of file ids and registers a fileClosed event in the eventStore

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Method: POST
URL: api/clientactions/fileclosed
Parameters:

Name	fields
Description	Not Available

Request: api/clientactions/fileclosed

```
[
  1,
  {...},
  {...}
]
```

Response: 200 OK

Containers

Containers object allows you to retrieve a list of containers.

GetContainers: API call to retrieve the ID's and descriptions of all containers (folders and documents) under a specified file id or folder id.

Method: GET
URL: api/containers/{containerId}?recursive={recursive}&filter={filter}

Parameters:

Name	containerId
Description	The File or Folder ID to retrieve containers from.
Name	recursive
Description	If recursive is true, it will retrieve all folders and documents under that specified ID. If recursive is false, it will only retrieve the immediate children of that ID.
Name	filter
Description	If filter is true, it will apply any applicable document filters before returning the result. Optional , default value is False .

Request: api/containers/1?recursive=true&filter=true

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "ContainerData": {
      "Id": 1,
      "Description": "DescriptionStr2",
      "EffectivePermissions": 3
    },
    "ChildContainers": [
      {
        "ContainerData": {
          "Id": 4,
          "Description": "DescriptionStr5",
          "EffectivePermissions": 6
        },
        "ChildContainers": [
          {
            "ContainerData": {
              "Id": 7,
              "Description": "DescriptionStr8",
              "EffectivePermissions": 9
            },
            "ChildContainers": []
          },
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

Error codes: 400 The document structure is invalid.

Dashboard

Dashboard Controller

GetTaskCount: Returns count of tasks according to the specified filter

Method:	POST
URL:	api/dashboard/taskcount
Parameters:	
Name	filter
Description	Task filter
Request:	api/dashboard/taskcount

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "TaskFilter": {
    "DrawerFileTypesTree": [
      {
        "Id": 1,
        "Children": [
          2,
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ],
    "FlowStepsTree": [
      {
        "Id": 3,
        "Children": [
          4,
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ],
    "StepDurationSteps": [
      5,
      {...},
      {...}
    ],
    "Tasks": [
      6,
      {...},
      {...}
    ],
    "AvailableDateStart": "2019-04-30T07:10:47.8497219-04:00",
    "AvailableDateEnd": "2019-04-30T07:10:47.8497219-04:00",
    "Flows": [
      9,
      {...},
      {...}
    ],
    "ExcludeFlows": true,
    "Steps": [
      11,
      {...},
      {...}
    ],
    "ExcludeSteps": true,
    "AssignedTo": [
      13,
      {...},
      {...}
    ],
    "AgeModelSet": [
      {
        "Id": 14,
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
      "From": 15,
      "To": 16
    },
    {...},
    {...}
  ],
  "AgeCalculationAlgorithm": "DateInitiated",
  "TaskStatus": [
    17,
    {...},
    {...}
  ],
  "ExcludeStatus": true,
  "Lockable": true,
  "OrderBy": "OrderByStr20",
  "LockedBy": [
    21,
    {...},
    {...}
  ],
  "Debug": true,
  "FileTypes": [
    23,
    {...},
    {...}
  ],
  "ExcludeFileTypes": true,
  "Drawers": [
    25,
    {...},
    {...}
  ],
  "ExcludeDrawers": true
},
"GroupBy": "None",
"Having": [
  27,
  {...},
  {...}
]
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "Id": 9,
    "AgeModel": {
      "Id": 10,
      "From": 11,
      "To": 12
    },
    "AgeCalculationAlgorithm": "DateInitiated",
    "Count": 13,
    "Name": "NameStr14"
  },
  {...},
  {...}
]
```

DashboardViews

A dashboard view object allows you to create, update, delete and retrieve dashboard views, manage default dashboard views.

CreateView: Creates a dashboard view owned by specified user.

Method:	POST
URL:	api/users/{userId}/views
Parameters:	
Name	userId
Description	The id of a view owner.
Name	model
Description	The dashboard view model to be created.
Request:	api/users/1/views

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Name": "NameStr2",
  "Filter": {
    "AssignedTo": [
      3,
      {...},
      {...}
    ],
    "DrawersAndFileTypes": [
      {
        "Key": 4,
        "Value": [
          5
        ]
      },
      {...},
      {...}
    ],
    "WorkflowsAndSteps": [
      {
        "Key": 6,
        "Value": [
          7
        ]
      },
      {...},
      {...}
    ]
  },
  "SlaIds": [
    8,
    {...},
    {...}
  ]
}
```

Response: 201 Created

1

DeleteView: Deletes a dashboard view.

Method:	DELETE
URL:	api/views/{viewId}
Parameters:	
Name	viewId
Description	The id of the dashboard view.
Request:	api/views/1
Response:	200 OK

GetAllViews: Retrieves a list of all user's dashboard views.

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Method: GET
URL: api/users/{userId}/views
Parameters:

Name	
Description	The id of the user.

Request: api/users/1/views
Response: 200 OK

```
[
  {
    "Id": 2,
    "Name": "NameStr3",
    "Filter": {
      "AssignedTo": [
        4,
        {...},
        {...}
      ],
      "DrawersAndFileTypes": [
        {
          "Key": 5,
          "Value": [
            6
          ]
        },
        {...},
        {...}
      ],
      "WorkflowsAndSteps": [
        {
          "Key": 7,
          "Value": [
            8
          ]
        },
        {...},
        {...}
      ]
    },
    "SlaIds": [
      1,
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

GetDefaultView: Retrieves default dashboard view.

Method: GET
URL: api/users/{userId}/views/default
Parameters:

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

Authentication

Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

Attributes

Batches

Containers

DashboardViews

DocumentFilter

DocumentsV2

Encryption

Folders

Images

Name	userId
Description	The id of a view owner.

Request: api/users/1/views/default

Response: 200 OK

```
1
```

GetView: Retrieves a dashboard view.

Method: GET
URL: `api/views/{viewId}`

Parameters:

Name	viewId
Description	The id of the dashboard view.

Request: `api/views/1`

Response: 200 OK

```
{
  "Name": "NameStr1",
  "Filter": {
    "AssignedTo": [
      2,
      {...},
      {...}
    ],
    "DrawersAndFileTypes": [
      {
        "Key": 3,
        "Value": [
          4
        ]
      },
      {...},
      {...}
    ],
    "WorkflowsAndSteps": [
      {
        "Key": 5,
        "Value": [
          6
        ]
      },
      {...},
      {...}
    ]
  },
  "SlaIds": [
    7,
    {...},
    {...}
  ]
}
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

SetDefaultView: Set specified dashboard view as default view.

Method:

PUT

URL:

api/users/{userId}/views/default

Parameters:

Name	userId
Description	The id of a view owner.
Name	viewId
Description	The id of the dashboard view.

Request:

api/users/1/views/default

2

Response:

200 OK

UpdateView: Updates the dashboard view.

Method:

PUT

URL:

api/views/{viewId}

Parameters:

Name	viewId
Description	The id of the dashboard view to be updated.
Name	model
Description	The dashboard view model to be updated.

Request:

api/views/1

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Name": "NameStr2",
  "Filter": {
    "AssignedTo": [
      3,
      {...},
      {...}
    ],
    "DrawersAndFileTypes": [
      {
        "Key": 4,
        "Value": [
          5
        ]
      },
      {...},
      {...}
    ],
    "WorkflowsAndSteps": [
      {
        "Key": 6,
        "Value": [
          7
        ]
      },
      {...},
      {...}
    ]
  },
  "SlaIds": [
    8,
    {...},
    {...}
  ]
}
```

Response: 200 OK

DataValidation

Performs CRUD operations pertaining to data validation

GetPageCountainsValidatedField: Gets a boolean value indicating whether a given page has at least one validated field

Method:	GET
URL:	api/validation/page/{pageId}
Parameters:	
Name	pageId
Description	Id of the page for which data validation is being performed
Request:	api/validation/page/1
Response:	200 OK

GetValidationFields: Gets a ValidationSetModel which represents the data for fields to be validated

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Method: GET
URL: `api/validation/file/{fileId}/page/{pageId}/isEditMode/{isEditMode}`

Parameters:

Name	fileId
Description	Id of the file for which data validation is being performed
Name	pageId
Description	Id of the page for which data validation is being performed
Name	isEditMode
Description	Boolean indicating whether we are in edit mode

Request: `api/validation/file/1/page/2/isEditMode/3`

Response: 200 OK

UpdateOcrDataAfterValidation: Updates the OcrData row after validation has been completed IsValidated attribute becomes true Value attribute is updated with any changes that may have been applied

Method: PUT
URL: `api/validation/ocrdata`

Parameters:

Name	ocrDataModel
Description	Contains attributes for updating an OcrData row after validation

Request: `api/validation/ocrdata`

```
{
  "Id": 1,
  "Value": "ValueStr2",
  "IsEdited": 3
}
```

Response: 200 OK

UpdateRemainingOcrDataValidationFields: Updates remaining unvalidated OcrData rows after validation has been completed IsValidated attribute becomes true

Method: PUT
URL: `api/validation/ocrdata/remaining`

Parameters:

Name	ocrDataIds
Description	Contains Ids for updating OcrData rows after validation

Request: `api/validation/ocrdata/remaining`

```
[
  {...},
  {...},
  {...}
]
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

DocumentFilter

DocumentFilter object allows you to create and retrieve Document Filters.

AreMappingsDefined: Determines if Document Filters have been defined.

Method: GET
URL: api/documentfilters/AreMappingsDefined
Request: api/documentfilters/AreMappingsDefined
Response: 200 OK

true

Create: Creates a new document filter mapping.

Method: POST
URL: api/documentfilters
Parameters:

Name	documentFilterData
Description	Structure of Document Filter to save

Request: api/documentfilters

```
{
  "DocumentFilterId": 1,
  "Description": "DescriptionStr2",
  "LocationId": 3,
  "FilterBy": "CreatedDate",
  "ShowOnly": 4,
  "ShowOnlyQualifier": "Months",
  "CreatedById": 5,
  "LastModified": "2019-04-30T07:10:47.8427219-04:00",
  "LastModifiedById": 7,
  "FileTypes": [
    {
      "DocumentFilterId": 8,
      "FilterFileTypeId": 9,
      "Description": "DescriptionStr10",
      "DrawerName": "DrawerNameStr11"
    },
    {...},
    {...}
  ],
  "Exclusions": [
    {
      "DocumentFilterId": 12,
      "TypeId": 13,
      "ClassId": 14,
      "Description": "DescriptionStr15"
    },
    {...},
    {...}
  ]
}
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 201 Created

```
[
  {
    "DocumentFilterId": 1,
    "Description": "DescriptionStr2",
    "LocationId": 3,
    "FilterBy": "CreatedDate",
    "ShowOnly": 4,
    "ShowOnlyQualifier": "Months",
    "CreatedById": 5,
    "LastModified": "2019-04-30T07:10:47.8457219-04:00",
    "LastModifiedById": 7,
    "FileTypes": [
      {
        "DocumentFilterId": 8,
        "FilterFileTypeId": 9,
        "Description": "DescriptionStr10",
        "DrawerName": "DrawerNameStr11"
      },
      {...},
      {...}
    ],
    "Exclusions": [
      {
        "DocumentFilterId": 12,
        "TypeId": 13,
        "ClassId": 14,
        "Description": "DescriptionStr15"
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

Error codes:	2200	A document filter with this description already exists.
	2201	A document filter already exists for this drawer and file type combination.

DeleteExclusion: Deletes a document filter exclusion

Method:

DELETE

URL:

api/documentfilters/{documentFilterId}/exclusions?typeId={typeId}&classId={classId}

Parameters:

Name

documentFilterId

Description

ID of the Document Filter tp delete Exclusion from

Name

typeId

Description

Type ID of type of exclusion to delete

Name

classId

Description

Class ID of exclusion to delete

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Request: api/documentfilters/1/exclusions?typeId=2&classId=3

Response: 200 OK

true

DeleteFileType: Deletes a document filter file type

Method: DELETE

URL: api/documentfilters/{documentFilterId}/filetypes?fileTypeId={fileTypeId}

Parameters:

Name	documentFilterId
Description	ID of the Document Filter to delete File Type from

Name	fileTypeId
Description	ID of File Type to delete

Request: api/documentfilters/1/filetypes?fileTypeId=2

Response: 200 OK

true

DeleteMapping: Deletes a document filter mapping

Method: DELETE

URL: api/documentfilters/{documentFilterId}

Parameters:

Name	documentFilterId
Description	ID of the Document Filter Mapping to delete

Request: api/documentfilters/1

Response: 200 OK

true

GetMappings: Gets a single document filter. ID of the Document Filter Mapping you would like to get

Method: GET

URL: api/documentfilters/{documentFilterId}

Parameters:

Name	documentFilterId
Description	Not Available

Request: api/documentfilters/1

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "DocumentFilterId": 1,
    "Description": "DescriptionStr2",
    "LocationId": 3,
    "FilterBy": "CreatedDate",
    "ShowOnly": 4,
    "ShowOnlyQualifier": "Months",
    "CreatedById": 5,
    "LastModified": "2019-04-30T07:10:47.8457219-04:00",
    "LastModifiedById": 7,
    "FileTypes": [
      {
        "DocumentFilterId": 8,
        "FilterFileTypeId": 9,
        "Description": "DescriptionStr10",
        "DrawerName": "DrawerNameStr11"
      },
      {...},
      {...}
    ],
    "Exclusions": [
      {
        "DocumentFilterId": 12,
        "TypeId": 13,
        "ClassId": 14,
        "Description": "DescriptionStr15"
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

GetMappings: Gets a list of all document filter that have been defined.

Method: GET
URL: api/documentfilters
Request: api/documentfilters
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "DocumentFilterId": 1,
    "Description": "DescriptionStr2",
    "LocationId": 3,
    "FilterBy": "CreatedDate",
    "ShowOnly": 4,
    "ShowOnlyQualifier": "Months",
    "CreatedById": 5,
    "LastModified": "2019-04-30T07:10:47.8457219-04:00",
    "LastModifiedById": 7,
    "FileTypes": [
      {
        "DocumentFilterId": 8,
        "FilterFileTypeId": 9,
        "Description": "DescriptionStr10",
        "DrawerName": "DrawerNameStr11"
      },
      {...},
      {...}
    ],
    "Exclusions": [
      {
        "DocumentFilterId": 12,
        "TypeId": 13,
        "ClassId": 14,
        "Description": "DescriptionStr15"
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

IsMappingDefined: Determines if a document filter has been defined for a specific file.

Method: GET
URL: api/documentfilters/IsMappingDefined/{fileId}
Parameters:

Name	fileId
Description	Id of the file being checked to see if any document filters apply to it.

Request: api/documentfilters/IsMappingDefined/1
Response: 200 OK

true

Update: Updates a document filter mapping.

Method: PUT
URL: api/documentfilters/{documentFilterId}

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Parameters:

Name	documentFilterData
Description	Data that should be updated
Name	documentFilterId
Description	ID of the record that should be updated

Request: api/documentfilters/16

```
{
  "DocumentFilterId": 1,
  "Description": "DescriptionStr2",
  "LocationId": 3,
  "FilterBy": "CreatedDate",
  "ShowOnly": 4,
  "ShowOnlyQualifier": "Months",
  "CreatedById": 5,
  "LastModified": "2019-04-30T07:10:47.8457219-04:00",
  "LastModifiedById": 7,
  "FileTypes": [
    {
      "DocumentFilterId": 8,
      "FilterFileTypeId": 9,
      "Description": "DescriptionStr10",
      "DrawerName": "DrawerNameStr11"
    },
    {...},
    {...}
  ],
  "Exclusions": [
    {
      "DocumentFilterId": 12,
      "TypeId": 13,
      "ClassId": 14,
      "Description": "DescriptionStr15"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "DocumentFilterId": 1,
    "Description": "DescriptionStr2",
    "LocationId": 3,
    "FilterBy": "CreatedDate",
    "ShowOnly": 4,
    "ShowOnlyQualifier": "Months",
    "CreatedById": 5,
    "LastModified": "2019-04-30T07:10:47.8467219-04:00",
    "LastModifiedById": 7,
    "FileTypes": [
      {
        "DocumentFilterId": 8,
        "FilterFileTypeId": 9,
        "Description": "DescriptionStr10",
        "DrawerName": "DrawerNameStr11"
      },
      {...},
      {...}
    ],
    "Exclusions": [
      {
        "DocumentFilterId": 12,
        "TypeId": 13,
        "ClassId": 14,
        "Description": "DescriptionStr15"
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

Error codes:		
2200		A document filter with this description already exists.
2201		A document filter already exists for this drawer and file type combination.

Documents

Documents object allows you to create, update, delete and retrieve list of Documents.

CreateDocument: Creates a document.

Method:	POST	
	URL: api/documents	
Parameters:	Name	documentDataInfo
	Description	Creation data for the document. All properties are required except for ReceivedDate and Attributes.
Request:	api/documents	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "ParentId": 1,
  "DocumentTypeId": 2,
  "Description": "DescriptionStr3",
  "DocumentDate": "2019-04-30T07:10:47.7657219-04:00",
  "ReceivedDate": "2019-04-30T07:10:47.7657219-04:00",
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr6",
      "Name": "NameStr7",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ]
}
```

Response: 201 Created

1

Error codes:

400	The document structure is invalid.
602	The attribute is not allowed.
600	The attribute was not found.
601	The attribute is required.
603	The attribute value is invalid.

FindDocuments: Searches for documents.

Method:	POST
URL:	api/documents/find
Parameters:	
Name	documentSearchObject
Description	Search parameters. The only required property is Filed.
Request:	api/documents/find

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "ParentId": 1,
  "FileId": 2,
  "Description": "DescriptionStr3",
  "CreatedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7657219-04:00",
    "EndDate": "2019-04-30T07:10:47.7657219-04:00"
  },
  "LastModifiedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7657219-04:00",
    "EndDate": "2019-04-30T07:10:47.7657219-04:00"
  },
  "DocumentDateRange": {
    "StartDate": "2019-04-30T07:10:47.7657219-04:00",
    "EndDate": "2019-04-30T07:10:47.7657219-04:00"
  },
  "Deleted": true,
  "DocumentTypeIds": [
    11,
    {...},
    {...}
  ],
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr12",
      "Name": "NameStr13",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "PageCount": 1,
    "DateCreated": "2019-04-30T07:10:47.7667219-04:00",
    "DateLastModified": "2019-04-30T07:10:47.7667219-04:00",
    "DocumentDate": "2019-04-30T07:10:47.7667219-04:00",
    "ReceivedDate": "2019-04-30T07:10:47.7667219-04:00",
    "Deleted": true,
    "DocumentTypeId": 7,
    "Attributes": [
      {
        "DisplayName": "DisplayNameStr8",
        "Name": "NameStr9",
        "Value": {...},
        "AttributeType": "atInt"
      },
      {...},
      {...}
    ],
    "EffectivePermissions": 11,
    "Id": 12,
    "Description": "DescriptionStr13"
  },
  {...},
  {...}
]
```

Error codes:		
400		The document structure is invalid.
602		The attribute is not allowed.
600		The attribute was not found.
601		The attribute is required.
603		The attribute value is invalid.

GetDocumentById: API call to get a document by ID.

Method:	GET				
URL:	api/documents/{docId}				
Parameters:					
	<table><tr><td>Name</td><td>docId</td></tr><tr><td>Description</td><td>The document ID.</td></tr></table>	Name	docId	Description	The document ID.
Name	docId				
Description	The document ID.				
Request:	api/documents/1				
Response:	200 OK				

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "PageCount": 1,
  "DateCreated": "2019-04-30T07:10:47.7677219-04:00",
  "DateLastModified": "2019-04-30T07:10:47.7677219-04:00",
  "DocumentDate": "2019-04-30T07:10:47.7677219-04:00",
  "ReceivedDate": "2019-04-30T07:10:47.7677219-04:00",
  "Deleted": true,
  "DocumentTypeId": 7,
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr8",
      "Name": "NameStr9",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ],
  "EffectivePermissions": 11,
  "Id": 12,
  "Description": "DescriptionStr13"
}
```

Error codes:		
	2	You do not have access to do this operation.
	17	An unexpected error occurred. Please try again later.
	403	Object is not a document.
	10	The item requested was not found.
	907	Permission was denied for this operation.
	401	You do not have sufficient permissions to perform this action.

MoveDocument: API call to move a document or set of documents to a different location.

Method:	POST	
URL:	api/documents/move	
Parameters:		
	Name	documentMoveModel
	Description	The information on what documents to move. The list of document IDs and target parent ID are required. If the target type conversion ID is not specified, the document will remain the same type. The type ID path is an optional parameter to copy the documents to the end of a specified type path beginning at the specified parent.
Request:	api/documents/move	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "DocumentIds": [
    1,
    {...},
    {...}
  ],
  "TargetParentId": 2,
  "TargetTypeConversionId": 3,
  "TypeIdPath": [
    4,
    {...},
    {...}
  ]
}
```

Response: 200 OK

```
{
  "DocumentIdMap": [
    {
      "Key": 1,
      "Value": 2
    },
    {...},
    {...}
  ],
  "PageIdMap": [
    {
      "Key": 3,
      "Value": 4
    },
    {...},
    {...}
  ],
  "FailedDocumentMoves": [
    5,
    {...},
    {...}
  ],
  "Error": "ErrorStr6"
}
```

Error codes:	400	The document structure is invalid.
	701	The page was not found.
	401	You do not have sufficient permissions to perform this action.
	10	The item requested was not found.
	402	Cannot move documents from different source files.

UpdateProperties: API call to update the properties of a document.

Method:	POST
URL:	api/documents/{docId}/properties
Parameters:	
Name	docId

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Description	The document ID of the document to be updated.
Name	documentProperties
Description	The properties of the document to be updated.
Request:	api/documents/1/properties
	<pre>{ "Description": "DescriptionStr2", "DocumentDate": "2019-04-30T07:10:47.7727219-04:00", "ReceivedDate": "2019-04-30T07:10:47.7727219-04:00", "DocTypeId": 5, "NewAttributes": [{ "DisplayName": "DisplayNameStr6", "Name": "NameStr7", "Value": {...}, "AttributeType": "atInt" }, {...}, {...}], "AttributesToChange": [{ "DisplayName": "DisplayNameStr9", "Name": "NameStr10", "Value": {...}, "AttributeType": "atInt" }, {...}, {...}], "AttributesToRemove": [12, {...}, {...}] }</pre>
Response:	200 OK
Error codes:	
600	The attribute was not found.
602	The attribute is not allowed.
601	The attribute is required.
603	The attribute value is invalid.
400	The document structure is invalid.
401	You do not have sufficient permissions to perform this action.
1002	The type was not found.
901	The data is not valid for the template.

DocumentsV2

Documents object allows you to create, update, delete and retrieve list of Documents.

CopyDocumentV2: API call to copy a document or set of documents to a different location.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Method: POST

URL: api/v2/documents/copy

Parameters:

Name	Description
model	The information on what documents to copy. The list of document IDs, target parent ID and copy option are required. If the target type conversion ID is not specified, the document will remain the same type. The type ID path is an optional parameter to copy the documents to the end of a specified type path beginning at the specified parent.

Request: api/v2/documents/copy

```
{
  "DocumentIds": [
    1,
    {...},
    {...}
  ],
  "TargetId": 2,
  "TargetTypeId": 3,
  "CopyOption": "Logical",
  "PathToCreate": [
    {
      "TypeId": 4,
      "Description": "DescriptionStr5"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "PageIdMap": [
    {
      "Key": 1,
      "Value": 2
    },
    {...},
    {...}
  ],
  "DocumentIdMap": [
    {
      "Key": 3,
      "Value": 4
    },
    {...},
    {...}
  ],
  "FailedObjects": [
    {
      "Key": 5,
      "Value": {
        "Message": "MessageStr6",
        "Code": 7
      }
    },
    {...},
    {...}
  ],
  "CreatedPath": [
    {
      "Name": "NameStr8",
      "TypeId": 9,
      "TypeDescription": "TypeDescriptionStr10",
      "EffectivePermissions": 11,
      "ObjectClass": "RootContainer",
      "DateCreated": "2019-04-30T07:10:47.7787219-04:00",
      "DateLastModified": "2019-04-30T07:10:47.7787219-04:00",
      "Id": 14,
      "Description": "DescriptionStr15"
    },
    {...},
    {...}
  ]
}
```

Error codes:		
27		Target for this operation is invalid.
28		Target for this operation was not found.
402		Cannot move documents from different source files.
407		Document type is not valid for the template.

DeleteDocument: Deletes a document.

Method:	DELETE
URL:	api/v2/documents/{documentId}?force={force}
Parameters:	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Name	documentId
Description	The document id.
Name	force
Description	If true, deletes a document with associated tasks. Optional , default value is False .
Request:	api/v2/documents/1?force=true
Response:	200 OK
<pre>{ "Problems": [{ "Key": 1, "Value": { "Message": "MessageStr2", "Code": 3 } }, {...}, {...}] }</pre>	
Error codes:	
403	Object is not a document.
410	The document cannot be deleted because it has tasks associated with them.

MoveDocumentV2: API call to move a document or set of documents to a different location.

Method:	POST
URL:	api/v2/documents/move
Parameters:	
Name	model
Description	The information on what documents to move. The list of document IDs and target parent ID are required. If the target type conversion ID is not specified, the document will remain the same type. The type ID path is an optional parameter to copy the documents to the end of a specified type path beginning at the specified parent.
Request:	api/v2/documents/move

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "DocumentIds": [
    1,
    {...},
    {...}
  ],
  "TargetId": 2,
  "TargetTypeId": 3,
  "CopyOption": "Logical",
  "PathToCreate": [
    {
      "TypeId": 4,
      "Description": "DescriptionStr5"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "PageIdMap": [
    {
      "Key": 1,
      "Value": 2
    },
    {...},
    {...}
  ],
  "DocumentIdMap": [
    {
      "Key": 3,
      "Value": 4
    },
    {...},
    {...}
  ],
  "FailedObjects": [
    {
      "Key": 5,
      "Value": {
        "Message": "MessageStr6",
        "Code": 7
      }
    },
    {...},
    {...}
  ],
  "CreatedPath": [
    {
      "Name": "NameStr8",
      "TypeId": 9,
      "TypeDescription": "TypeDescriptionStr10",
      "EffectivePermissions": 11,
      "ObjectClass": "RootContainer",
      "DateCreated": "2019-04-30T07:10:47.7837219-04:00",
      "DateLastModified": "2019-04-30T07:10:47.7837219-04:00",
      "Id": 14,
      "Description": "DescriptionStr15"
    },
    {...},
    {...}
  ]
}
```

Error codes:		
27		Target for this operation is invalid.
28		Target for this operation was not found.
402		Cannot move documents from different source files.
407		Document type is not valid for the template.

Drawers

Drawers object allows you to retrieve Drawers.

GetDrawerById: Retrieve a drawer by drawer id.

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Method: GET
URL: api/drawers/{drawerId}
Parameters:

Name	drawerId
Description	The id of the drawer to retrieve.

Request: api/drawers/1
Response: 200 OK

```
{
  "Id": 1,
  "DrawerTypeId": 2,
  "ParentId": 3,
  "Name": "NameStr4",
  "Description": "DescriptionStr5",
  "Created": "2019-04-30T07:10:47.8017219-04:00",
  "LastModified": "2019-04-30T07:10:47.8017219-04:00"
}
```

Error codes:

400	The document structure is invalid.
------------	------------------------------------

GetDrawers: Retrieve all drawers in the system.

Method: GET
URL: api/drawers
Request: api/drawers
Response: 200 OK

```
[
  {
    "Id": 1,
    "DrawerTypeId": 2,
    "ParentId": 3,
    "Name": "NameStr4",
    "Description": "DescriptionStr5",
    "Created": "2019-04-30T07:10:47.8027219-04:00",
    "LastModified": "2019-04-30T07:10:47.8027219-04:00"
  },
  {...},
  {...}
]
```

Error codes:

400	The document structure is invalid.
------------	------------------------------------

GetDrawersByName: Retrieve drawers with a drawer name.

Method: GET
URL: api/drawers/{name}
Parameters:

Name	name
-------------	------

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description	The name of the drawer to search for.
Request:	api/drawers/ <i>nameStr1</i>
Response:	200 OK

```
[
  {
    "Id": 1,
    "DrawerTypeId": 2,
    "ParentId": 3,
    "Name": "NameStr4",
    "Description": "DescriptionStr5",
    "Created": "2019-04-30T07:10:47.8027219-04:00",
    "LastModified": "2019-04-30T07:10:47.8027219-04:00"
  },
  {...},
  {...}
]
```

Error codes:	
400	The document structure is invalid.

GetDrawersInContainer: Retrieve all drawers in a location/container.

Method:	GET
URL:	api/containers/{ <i>containerId</i> }/drawers
Parameters:	
Name	<i>containerId</i>
Description	The location/container ID.
Request:	api/containers/ <i>1</i> /drawers
Response:	200 OK

```
[
  {
    "Id": 1,
    "DrawerTypeId": 2,
    "ParentId": 3,
    "Name": "NameStr4",
    "Description": "DescriptionStr5",
    "Created": "2019-04-30T07:10:47.8027219-04:00",
    "LastModified": "2019-04-30T07:10:47.8027219-04:00"
  },
  {...},
  {...}
]
```

Error codes:	
400	The document structure is invalid.

GetDrawersInContainerByName: Retrieve all drawers inside a specified location/container filtered by name.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Method: GET
URL: api/containers/{containerId}/drawers/{name}
Parameters:

Name	containerId
Description	The location/container ID.
Name	name
Description	The name to filter by.

Request: api/containers/1/drawers/nameStr2
Response: 200 OK

```
[
  {
    "Id": 1,
    "DrawerTypeId": 2,
    "ParentId": 3,
    "Name": "NameStr4",
    "Description": "DescriptionStr5",
    "Created": "2019-04-30T07:10:47.8027219-04:00",
    "LastModified": "2019-04-30T07:10:47.8027219-04:00"
  },
  {...},
  {...}
]
```

Error codes:

400	The document structure is invalid.
------------	------------------------------------

Encryption
Controller only accessible by administrators that will return a scheudle that will run to encrypt/decrypt attributes and do the encryption/decryption action when asked to

DecryptAttributes: No Help Available

Method: GET
URL: api/encryption/attributes/decrypt
Request: api/encryption/attributes/decrypt
Response: 200 OK

EncryptAttributes: No Help Available

Method: GET
URL: api/encryption/attributes/encrypt
Request: api/encryption/attributes/encrypt
Response: 200 OK

GetSchedule: No Help Available

Method: GET

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

URL: api/encryption/schedule
Request: api/encryption/schedule
Response: 200 OK

RotateEncryptionKey: No Help Available

Method: GET
URL: api/encryption/rotatekey
Request: api/encryption/rotatekey
Response: 200 OK

Files

Files object allows you to create and retrieve Files.

CreateFile: API call to create a file.

Method: POST
URL: api/files
Parameters:

Name	fileDataInfo
Description	The object to create the file from. File numbers and attributes are optional.

Request: api/files

```
{
  "FileTypeId": 1,
  "ParentId": 2,
  "Name": "NameStr3",
  "FileNumberPart1": "FileNumberPart1Str4",
  "FileNumberPart2": "FileNumberPart2Str5",
  "FileNumberPart3": "FileNumberPart3Str6",
  "IsTemporary": true,
  "CreatedByApplication": "CreatedByApplicationStr8",
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr9",
      "Name": "NameStr10",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ]
}
```

Response: 201 Created

1

Error codes:

500	File already exists.
400	The document structure is invalid.

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

602	The attribute is not allowed.
600	The attribute was not found.
601	The attribute is required.
603	The attribute value is invalid.
501	FileNumber is not valid for the FileType Number Format.

FindFiles: Search for files.

Method: POST
URL: api/files/find
Parameters:

Name
Description

fileSearchObject
The search parameters for files. All parameters are optional. When specifying an attribute to search for, if the ID is provided, the name is ignored.

Request: api/files/find

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "FileNumberPart1": "FileNumberPart1Str1",
  "FileNumberPart2": "FileNumberPart2Str2",
  "FileNumberPart3": "FileNumberPart3Str3",
  "FileName": "FileNameStr4",
  "IsTemporary": true,
  "IsDeleted": true,
  "ParentId": 7,
  "FileTypeIds": [
    8,
    {...},
    {...}
  ],
  "MarkIds": [
    9,
    {...},
    {...}
  ],
  "CreatedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7327219-04:00",
    "EndDate": "2019-04-30T07:10:47.7327219-04:00"
  },
  "LastModifiedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7327219-04:00",
    "EndDate": "2019-04-30T07:10:47.7327219-04:00"
  },
  "LastOpenedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7327219-04:00",
    "EndDate": "2019-04-30T07:10:47.7327219-04:00"
  },
  "Attributes": [
    {
      "Id": 16,
      "DisplayName": "DisplayNameStr17",
      "Name": "NameStr18",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Id": 1,
    "FileTypeId": 2,
    "FileName": "FileNameStr3",
    "FileTypeDescription": "FileTypeDescriptionStr4",
    "DrawerId": 5,
    "DrawerName": "DrawerNameStr6",
    "DrawerDescription": "DrawerDescriptionStr7",
    "HasNotes": true,
    "NotesId": 9,
    "Description": "DescriptionStr10",
    "FileNumberPart1": "FileNumberPart1Str11",
    "FileNumberPart2": "FileNumberPart2Str12",
    "FileNumberPart3": "FileNumberPart3Str13",
    "IsTemporary": true,
    "IsDeleted": true,
    "DateLastOpened": "2019-04-30T07:10:47.7347219-04:00",
    "LastModified": "2019-04-30T07:10:47.7347219-04:00",
    "DateCreated": "2019-04-30T07:10:47.7347219-04:00",
    "Attributes": [
      {
        "DisplayName": "DisplayNameStr19",
        "Name": "NameStr20",
        "Value": {...},
        "AttributeType": "atInt"
      },
      {...},
      {...}
    ],
    "EffectivePermissions": 22,
    "IsFrozen": true
  },
  {...},
  {...}
]
```

Error codes:		
	500	File already exists.
	400	The document structure is invalid.
	602	The attribute is not allowed.
	600	The attribute was not found.
	601	The attribute is required.
	603	The attribute value is invalid.
	501	FileNumber is not valid for the FileType Number Format.

GetFileById: Returns File based on file Id

Method:	GET	
URL:	api/files/{fileId}?includeHasNotes={includeHasNotes}	
Parameters:	Name	fileId
	Description	Id of the file
	Name	includeHasNotes

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description	Whether to include if the file has notes or not in the return. Optional , default value is True .
Request:	api/files/1?includeHasNotes=true
Response:	200 OK
<pre>{ "Id": 1, "FileTypeId": 2, "FileName": "FileNameStr3", "FileTypeDescription": "FileTypeDescriptionStr4", "DrawerId": 5, "DrawerName": "DrawerNameStr6", "DrawerDescription": "DrawerDescriptionStr7", "HasNotes": true, "NotesId": 9, "Description": "DescriptionStr10", "FileNumberPart1": "FileNumberPart1Str11", "FileNumberPart2": "FileNumberPart2Str12", "FileNumberPart3": "FileNumberPart3Str13", "IsTemporary": true, "IsDeleted": true, "DateLastOpened": "2019-04-30T07:10:47.7357219-04:00", "LastModified": "2019-04-30T07:10:47.7357219-04:00", "DateCreated": "2019-04-30T07:10:47.7357219-04:00", "Attributes": [{ "DisplayName": "DisplayNameStr19", "Name": "NameStr20", "Value": {...}, "AttributeType": "atInt" }, {...}, {...}], "EffectivePermissions": 22, "IsFrozen": true }</pre>	
Error codes:	
10	The item requested was not found.

GetRelatedFiles: Returns list of related files for the specified file.

Method:	GET
URL:	api/files/{fileId}/related
Parameters:	
Name	field
Description	The file id.
Request:	api/files/1/related
Response:	200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Id": 1,
    "FileTypeId": 2,
    "FileName": "FileNameStr3",
    "FileTypeDescription": "FileTypeDescriptionStr4",
    "DrawerId": 5,
    "DrawerName": "DrawerNameStr6",
    "DrawerDescription": "DrawerDescriptionStr7",
    "HasNotes": true,
    "NotesId": 9,
    "Description": "DescriptionStr10",
    "FileNumberPart1": "FileNumberPart1Str11",
    "FileNumberPart2": "FileNumberPart2Str12",
    "FileNumberPart3": "FileNumberPart3Str13",
    "IsTemporary": true,
    "IsDeleted": true,
    "DateLastOpened": "2019-04-30T07:10:47.7357219-04:00",
    "LastModified": "2019-04-30T07:10:47.7357219-04:00",
    "DateCreated": "2019-04-30T07:10:47.7357219-04:00",
    "Attributes": [
      {
        "DisplayName": "DisplayNameStr19",
        "Name": "NameStr20",
        "Value": {...},
        "AttributeType": "atInt"
      },
      {...},
      {...}
    ],
    "EffectivePermissions": 22,
    "IsFrozen": true
  },
  {...},
  {...}
]
```

Error codes:	10	The item requested was not found.
	502	Object is not a file.

Folders

Folders object allows you to create and retrieve Folders.

CreateFolder: Create a folder.

Method:	POST				
URL:	api/folders				
Parameters:					
	<table><tr><td>Name</td><td>folderDataInfo</td></tr><tr><td>Description</td><td>The object to create a folder with. Attributes is optional.</td></tr></table>	Name	folderDataInfo	Description	The object to create a folder with. Attributes is optional.
Name	folderDataInfo				
Description	The object to create a folder with. Attributes is optional.				
Request:	api/folders				

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FolderTypeId": 1,
  "ParentId": 2,
  "Description": "DescriptionStr3",
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr4",
      "Name": "NameStr5",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ]
}
```

Response: 201 Created

1

Error codes:

400	The document structure is invalid.
602	The attribute is not allowed.
600	The attribute was not found.
601	The attribute is required.
603	The attribute value is invalid.

DeleteFolder: Deletes a folder.

Method: DELETE
URL: `api/folders/{folderId}?force={force}`

Parameters:

Name	folderId
Description	The folder id.
Name	force
Description	If true, deletes a folder with associated tasks. Optional , default value is False .

Request: `api/folders/1?force=true`

Response: 200 OK

```
{
  "Problems": [
    {
      "Key": 1,
      "Value": {
        "Message": "MessageStr2",
        "Code": 3
      }
    },
    {...},
    {...}
  ]
}
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Error codes:	1102	Object is not a folder.
	1103	Folder cannot be deleted because it has associated tasks.

FindFolders: Search for folders.

Method:	POST
URL:	api/folders/find
Parameters:	
Name	folderSearchObject
Description	The search parameters. Filed is the only required property.
Request:	api/folders/find

```
{
  "Description": "DescriptionStr1",
  "IsDeleted": true,
  "FileId": 3,
  "ParentId": 4,
  "FolderTypeIds": [
    5,
    {...},
    {...}
  ],
  "CreatedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7937219-04:00",
    "EndDate": "2019-04-30T07:10:47.7937219-04:00"
  },
  "LastModifiedDateRange": {
    "StartDate": "2019-04-30T07:10:47.7937219-04:00",
    "EndDate": "2019-04-30T07:10:47.7937219-04:00"
  },
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr10",
      "Name": "NameStr11",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "FileId": 1,
    "ParentFolderId": 2,
    "FolderTypeId": 3,
    "HasNotes": true,
    "IsDeleted": true,
    "LastModified": "2019-04-30T07:10:47.7947219-04:00",
    "DateCreated": "2019-04-30T07:10:47.7947219-04:00",
    "Attributes": [
      {
        "DisplayName": "DisplayNameStr8",
        "Name": "NameStr9",
        "Value": {...},
        "AttributeType": "atInt"
      },
      {...},
      {...}
    ],
    "EffectivePermissions": 11,
    "Id": 12,
    "Description": "DescriptionStr13"
  },
  {...},
  {...}
]
```

Error codes:		
602		The attribute is not allowed.
600		The attribute was not found.
601		The attribute is required.
603		The attribute value is invalid.

GetFolderById: Returns Folder based on Folder Id

Method:	GET	
URL:	api/folders/{folderId}?includeHasNotes={includeHasNotes}	
Parameters:	Name	folderId
	Description	Folder Id
	Name	includeHasNotes
	Description	Whether if the folder contains notes should be included in the response. Optional , default value is True .
Request:	api/folders/1?includeHasNotes=true	
Response:	200 OK	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FileId": 1,
  "ParentFolderId": 2,
  "FolderTypeId": 3,
  "HasNotes": true,
  "IsDeleted": true,
  "LastModified": "2019-04-30T07:10:47.7957219-04:00",
  "DateCreated": "2019-04-30T07:10:47.7957219-04:00",
  "Attributes": [
    {
      "DisplayName": "DisplayNameStr8",
      "Name": "NameStr9",
      "Value": {...},
      "AttributeType": "atInt"
    },
    {...},
    {...}
  ],
  "EffectivePermissions": 11,
  "Id": 12,
  "Description": "DescriptionStr13"
}
```

Error codes:

10	The item requested was not found.
-----------	-----------------------------------

MoveFolder: API call to move a folder or set of folders to a different location.

Method:	POST
URL:	api/folders/move
Parameters:	
Name	model
Description	The information on what folders to move. The list of folder IDs and target parent ID are required. If the target type conversion ID is not specified, the folder will remain the same type. The type ID path is an optional parameter to copy the folders to the end of a specified type path beginning at the specified parent.
Request:	api/folders/move

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "FolderIds": [
    1,
    {...},
    {...}
  ],
  "TargetId": 2,
  "TargetTypeId": 3,
  "CopyOption": "Logical",
  "PathToCreate": [
    {
      "TypeId": 4,
      "Description": "DescriptionStr5"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "PageIdMap": [
    {
      "Key": 1,
      "Value": 2
    },
    {...},
    {...}
  ],
  "DocumentIdMap": [
    {
      "Key": 3,
      "Value": 4
    },
    {...},
    {...}
  ],
  "FolderIdMap": [
    {
      "Key": 5,
      "Value": 6
    },
    {...},
    {...}
  ],
  "FailedObjects": [
    {
      "Key": 7,
      "Value": {
        "Message": "MessageStr8",
        "Code": 9
      }
    },
    {...},
    {...}
  ],
  "CreatedPath": [
    {
      "Name": "NameStr10",
      "TypeId": 11,
      "TypeDescription": "TypeDescriptionStr12",
      "EffectivePermissions": 13,
      "ObjectClass": "RootContainer",
      "DateCreated": "2019-04-30T07:10:47.7967219-04:00",
      "DateLastModified": "2019-04-30T07:10:47.7967219-04:00",
      "Id": 16,
      "Description": "DescriptionStr17"
    },
    {...},
    {...}
  ]
}
```

Error codes:	27	Target for this operation is invalid.
	28	Target for this operation was not found.
	402	Cannot move documents from different source files.
	407	Document type is not valid for the template.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Images

Images object allows you to get images.

GetImageById: API call to get a stream of an image's data by image id and version number.

Method: GET
URL: `api/images/{imageId}?version={version}`
Parameters:

Name	imageId
Description	The image ID to get the stream data from.
Name	version
Description	The version of that image to get the data from.

Request: `api/images/1?version=2`
Response: 200 OK

"Image goes here (stream)"

StreamMedia: API call to stream an audio or video file. See `AllowedStreamFormats` for list of supported extensions

Method: GET
URL: `api/images/streamMedia/{imageId}?version={version}`
Parameters:

Name	imageId
Description	The image ID to get the stream data from.
Name	version
Description	The version of that image to get the data from. Optional , default value is -1 .

Request: `api/images/streamMedia/1?version=2`
Response: 200 OK

"Image goes here (stream)"

ImagesV2

Images object allows you to get images.

GetImageByIdV2: API call to get a stream of an image's data by image id and version number.

Method: GET
URL: `api/v2/pages/{pageId}/images/{imageId}/{version}`
Parameters:

Name	pageId
Description	The id of the page the image belongs to.
Name	imageId
Description	The image ID to get the stream data from.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Name	version
Description	The version of that image to get the data from.

Request: api/v2/pages/1/images/2/3

Response: 200 OK

"Image goes here (stream)"

StreamMediaV2: API call to stream an audio or video file. See `AllowedStreamFormats` for list of supported extensions

Method:	GET				
URL:	api/v2/pages/{pageId}/images/{imageId}/{version}/streamMedia				
Parameters:					
<table><tr><td>Name</td><td>pageId</td></tr><tr><td>Description</td><td>The id of the page the image belongs to.</td></tr></table>	Name	pageId	Description	The id of the page the image belongs to.	
Name	pageId				
Description	The id of the page the image belongs to.				
<table><tr><td>Name</td><td>imageId</td></tr><tr><td>Description</td><td>The image ID to get the stream data from.</td></tr></table>	Name	imageId	Description	The image ID to get the stream data from.	
Name	imageId				
Description	The image ID to get the stream data from.				
<table><tr><td>Name</td><td>version</td></tr><tr><td>Description</td><td>The version of that image to get the data from. Optional, default value is -1.</td></tr></table>	Name	version	Description	The version of that image to get the data from. Optional , default value is -1.	
Name	version				
Description	The version of that image to get the data from. Optional , default value is -1.				

Request: api/v2/pages/1/images/2/3/streamMedia

Response: 200 OK

"Image goes here (stream)"

Instances

Instances object allows you to retrieve list of children instances of object.

CreateInstances: Creates folders and documents in the specified parent.

Method:	POST				
URL:	api/instances/{id}/children				
Parameters:					
<table><tr><td>Name</td><td>id</td></tr><tr><td>Description</td><td>The parent ID.</td></tr></table>	Name	id	Description	The parent ID.	
Name	id				
Description	The parent ID.				
<table><tr><td>Name</td><td>objectsToCreate</td></tr><tr><td>Description</td><td>The data for objects creating.</td></tr></table>	Name	objectsToCreate	Description	The data for objects creating.	
Name	objectsToCreate				
Description	The data for objects creating.				

Request: api/instances/1/children

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "TypeId": 2,
    "Description": "DescriptionStr3"
  },
  {...},
  {...}
]
```

Response: 200 OK

```
[
  {
    "Name": "NameStr1",
    "TypeId": 2,
    "TypeDescription": "TypeDescriptionStr3",
    "EffectivePermissions": 4,
    "ObjectClass": "RootContainer",
    "DateCreated": "2019-04-30T07:10:47.8207219-04:00",
    "DateLastModified": "2019-04-30T07:10:47.8207219-04:00",
    "Id": 7,
    "Description": "DescriptionStr8"
  },
  {...},
  {...}
]
```

Error codes:		
10		The item requested was not found.
1100		A non-repeatable folder has been already created.
1101		Folder type is not valid for the template.
407		Document type is not valid for the template.
903		Object class is unsupported for this operation.
1002		The type was not found.
28		Target for this operation was not found.

GetInstanceChildren: Returns a list of children instances of files/folders/documents.

Method:	GET		
URL:	api/instances/{id}/children		
Parameters:	Name	id	
	Description	The ID of the file or folder.	
Request:	api/instances/1/children		
Response:	200 OK		

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
[
  {
    "Id": 1,
    "Description": "DescriptionStr2"
  },
  {...},
  {...}
]
```

Error codes:

10

The item requested was not found.

GetParentPath: Returns the path to the file of the object passed representing each ancestor. The first element is the lowest level container (in the case of passing in a page, the first element will be the document; in the case of passing in a document, folder, or file, the first element will be the object that was passed in). The last element is the file. Each object has an associated ObjectClass number. The ObjectClasses correspond to the following negative numbers (ignoring the positive number in the generated response sample):
File: -3
Folder: -2
Document: -1

Method: GET

URL: `api/instances/getparentpath?id={id}&isContainer={isContainer}`

Parameters:

Name id

Description The ID of the object to retrieve the path to the file for.

Name isContainer

Description Whether the object is a container (a file, folder, or document) or not a container (a page). This is required as they share different keysets.

Request: `api/instances/getparentpath?id=1&isContainer=true`

Response: 200 OK

```
[
  {
    "ObjectId": 1,
    "ObjectClass": 2,
    "Description": "DescriptionStr3"
  },
  {...},
  {...}
]
```

Error codes:

10

The item requested was not found.

GetPathToObject: Returns the path to the file of the object passed representing each ancestor. The first element is the lowest level container (in the case of passing in a page, the first element will be the document; in the case of passing in a document, folder, or file, the first element will be the object that was passed in). The last element is the file. Each object has an associated ObjectClass number. The ObjectClasses correspond to the following negative numbers (ignoring the positive number in the generated response sample):
File: -3
Folder: -2
Document: -1

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Method: GET
URL: api/instances/GetPathToObject?id={id}&isPage={isPage}
Parameters:

Name	
id	
Description	The ID of the object to retrieve the path to the file for.

Name	
isPage	
Description	Whether the object is a container (a file, folder, or document) or not a container (a page). This is required as they share different keysets.

Request: api/instances/GetPathToObject?id=1&isPage=true

Response: 200 OK

```
[
  {
    "ObjectId": 1,
    "ObjectType": "ObjectTypeStr2",
    "Description": "DescriptionStr3"
  },
  {...},
  {...}
]
```

Error codes:

10	The item requested was not found.
----	-----------------------------------

Instrumentation

Provides enterprise level functionality and data for application server instances

GetEnterpriseId: Retrieves the enterprise id from the application server instance.

Method: GET
URL: api/enterpriseid
Request: api/enterpriseid
Response: 200 OK

```
"98f73ab2-8a13-41a2-ae20-da42903a0c8e"
```

Error codes:

2	You do not have access to do this operation.
---	--

Integration

IntegrationController allows you to get files and folders by attribute value.

FindVapTasks: Provides functionality to Tasks retrieval and filtering

Method: POST
URL: api/integration/vap/tasks/find?skip={skip}&top={top}
Parameters:

Name	
filter	

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Description	Filter object
Name	skip
Description	Skip value for paging, default = null Optional , default value is null (empty) .
Name	top
Description	Top value for paging, default = null Optional , default value is null (empty) .

Request: `api/integration/vap/tasks/find?skip=22&top=23`

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Tasks": [
    1,
    {...},
    {...}
  ],
  "AvailableDateStart": "2019-04-30T07:10:47.7187219-04:00",
  "AvailableDateEnd": "2019-04-30T07:10:47.7187219-04:00",
  "Flows": [
    4,
    {...},
    {...}
  ],
  "ExcludeFlows": true,
  "Steps": [
    6,
    {...},
    {...}
  ],
  "ExcludeSteps": true,
  "AssignedTo": [
    8,
    {...},
    {...}
  ],
  "AgeModelSet": [
    {
      "Id": 9,
      "From": 10,
      "To": 11
    },
    {...},
    {...}
  ],
  "AgeCalculationAlgorithm": "DateInitiated",
  "TaskStatus": [
    12,
    {...},
    {...}
  ],
  "ExcludeStatus": true,
  "Lockable": true,
  "OrderBy": "OrderByStr15",
  "LockedBy": [
    16,
    {...},
    {...}
  ],
  "Debug": true,
  "FileTypes": [
    18,
    {...},
    {...}
  ],
  "ExcludeFileTypes": true,
  "Drawers": [
    20,
    {...},
    {...}
  ]
}
```

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

```
{...}  
},  
"ExcludeDrawers": true  
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "TaskModel": {
    "Id": 176,
    "Description": "DescriptionStr177",
    "Priority": 178,
    "FileId": 179,
    "FileName": "FileNameStr180",
    "FileNumber": "FileNumberStr181",
    "FlowId": 182,
    "FlowName": "FlowNameStr183",
    "StepId": 184,
    "StepName": "StepNameStr185",
    "SenderStep": 186,
    "SuperTaskId": 187,
    "RendezvousStepId": 188,
    "SubTaskIsRequired": true,
    "NoteId": 190,
    "AssignedTo": {
      "Id": 191,
      "ExternalId": "ExternalIdStr192",
      "Name": "NameStr193",
      "FriendlyName": "FriendlyNameStr194",
      "Type": "user",
      "Enabled": true,
      "Description": "DescriptionStr196"
    },
    "Sender": {
      "Id": 197,
      "ExternalId": "ExternalIdStr198",
      "Name": "NameStr199",
      "FriendlyName": "FriendlyNameStr200",
      "Type": "user",
      "Enabled": true,
      "Description": "DescriptionStr202"
    },
    "LockedBy": {
      "Id": 203,
      "ExternalId": "ExternalIdStr204",
      "Name": "NameStr205",
      "FriendlyName": "FriendlyNameStr206",
      "Type": "user",
      "Enabled": true,
      "Description": "DescriptionStr208"
    },
    "AvailableDate": "2019-04-30T07:10:47.7207219-04:00",
    "StartDate": "2019-04-30T07:10:47.7207219-04:00",
    "LockExpiration": "2019-04-30T07:10:47.7207219-04:00",
    "UndoExpires": "2019-04-30T07:10:47.7207219-04:00",
    "DeadLine": "2019-04-30T07:10:47.7207219-04:00",
    "DateInitiated": "2019-04-30T07:10:47.7207219-04:00",
    "Debug": true,
    "StackLevel": 216,
    "ErrorCode": "None",
    "ErrorMessage": "ErrorMessageStr217",
    "StatusDetails": "None",
    "Status": "Ready",
    "PageNumber": 218,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
"ObjectId": 219,
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr220",
    "Name": "NameStr221",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr223",
    "Name": "NameStr224",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr226",
    "Name": "NameStr227",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 229,
"FileTypeId": 230,
"DrawerId": 231,
"ObjectTypeId": 232,
"ObjectClass": "RootContainer"
},
"ClientId": "ClientIdStr233",
"PolicyId": "PolicyIdStr234"
},
{...},
{...}
]
```

FindVapTasksByClientId: Retrieves Tasks by client id via Http Post Request

Method:	POST
URL:	api/integration/vap/client/{clientId}/tasks/find?skip={skip}&top={top}&includeDeletedFiles={includeDeletedFiles}
Parameters:	
Name	filter
Description	Filter object
Name	clientId
Description	Client ID to filter by
Name	skip

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Description	Skip value for paging, default = null Optional , default value is null (empty) .
Name	top
Description	Top value for paging, default = null Optional , default value is null (empty) .
Name	includeDeletedFiles
Description	Whether or not deleted files should be included. Defaults to false. Optional , default value is False .

Request: `api/integration/vap/client/clientIdStr22/tasks/find?skip=23&top=24&includeDeletedFiles=true`

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "Tasks": [
    1,
    {...},
    {...}
  ],
  "AvailableDateStart": "2019-04-30T07:10:47.7217219-04:00",
  "AvailableDateEnd": "2019-04-30T07:10:47.7217219-04:00",
  "Flows": [
    4,
    {...},
    {...}
  ],
  "ExcludeFlows": true,
  "Steps": [
    6,
    {...},
    {...}
  ],
  "ExcludeSteps": true,
  "AssignedTo": [
    8,
    {...},
    {...}
  ],
  "AgeModelSet": [
    {
      "Id": 9,
      "From": 10,
      "To": 11
    },
    {...},
    {...}
  ],
  "AgeCalculationAlgorithm": "DateInitiated",
  "TaskStatus": [
    12,
    {...},
    {...}
  ],
  "ExcludeStatus": true,
  "Lockable": true,
  "OrderBy": "OrderByStr15",
  "LockedBy": [
    16,
    {...},
    {...}
  ],
  "Debug": true,
  "FileTypes": [
    18,
    {...},
    {...}
  ],
  "ExcludeFileTypes": true,
  "Drawers": [
    20,
    {...},
    {...}
  ]
}
```

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

```
{...}  
},  
"ExcludeDrawers": true  
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "TaskModel": {
    "Id": 176,
    "Description": "DescriptionStr177",
    "Priority": 178,
    "FileId": 179,
    "FileName": "FileNameStr180",
    "FileNumber": "FileNumberStr181",
    "FlowId": 182,
    "FlowName": "FlowNameStr183",
    "StepId": 184,
    "StepName": "StepNameStr185",
    "SenderStep": 186,
    "SuperTaskId": 187,
    "RendezvousStepId": 188,
    "SubTaskIsRequired": true,
    "NoteId": 190,
    "AssignedTo": {
      "Id": 191,
      "ExternalId": "ExternalIdStr192",
      "Name": "NameStr193",
      "FriendlyName": "FriendlyNameStr194",
      "Type": "user",
      "Enabled": true,
      "Description": "DescriptionStr196"
    },
    "Sender": {
      "Id": 197,
      "ExternalId": "ExternalIdStr198",
      "Name": "NameStr199",
      "FriendlyName": "FriendlyNameStr200",
      "Type": "user",
      "Enabled": true,
      "Description": "DescriptionStr202"
    },
    "LockedBy": {
      "Id": 203,
      "ExternalId": "ExternalIdStr204",
      "Name": "NameStr205",
      "FriendlyName": "FriendlyNameStr206",
      "Type": "user",
      "Enabled": true,
      "Description": "DescriptionStr208"
    },
    "AvailableDate": "2019-04-30T07:10:47.7227219-04:00",
    "StartDate": "2019-04-30T07:10:47.7227219-04:00",
    "LockExpiration": "2019-04-30T07:10:47.7227219-04:00",
    "UndoExpires": "2019-04-30T07:10:47.7227219-04:00",
    "DeadLine": "2019-04-30T07:10:47.7227219-04:00",
    "DateInitiated": "2019-04-30T07:10:47.7227219-04:00",
    "Debug": true,
    "StackLevel": 216,
    "ErrorCode": "None",
    "ErrorMessage": "ErrorMessageStr217",
    "StatusDetails": "None",
    "Status": "Ready",
    "PageNumber": 218,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
"ObjectId": 219,
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr220",
    "Name": "NameStr221",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr223",
    "Name": "NameStr224",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr226",
    "Name": "NameStr227",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 229,
"FileTypeId": 230,
"DrawerId": 231,
"ObjectTypeId": 232,
"ObjectClass": "RootContainer"
},
"ClientId": "ClientIdStr233",
"PolicyId": "PolicyIdStr234"
},
{...},
{...}
]
```

GetClientFiles: Returns a list of files associated with specified client.

Method:	GET
URL:	api/integration/vap/files/client/{value}?includeDeletedFiles={includeDeletedFiles}
Parameters:	
Name	value
Description	The clientID attribute value.
Name	includeDeletedFiles
Description	Whether or not deleted files should be included. Defaults to false. Optional , default value is False .
Request:	api/integration/vap/files/client/valueStr1?includeDeletedFiles=true

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 200 OK

```
[
  {
    "FileId": 1,
    "FileNumber": "FileNumberStr2",
    "FileName": "FileNameStr3",
    "FileType": {
      "Id": 4,
      "Name": "NameStr5"
    },
    "Drawer": {
      "Id": 6,
      "Name": "NameStr7"
    }
  },
  {...},
  {...}
]
```

GetPolicyFolder: Returns fileId and folderId associated with specified policy.

Method: GET
URL: api/integration/vap/files/policy/{value}?includeDeletedFiles={includeDeletedFiles}

Parameters:

Name	value
Description	The policy attribute value.

Name	includeDeletedFiles
Description	Whether or not deleted files should be included. Defaults to false. Optional , default value is False .

Request: api/integration/vap/files/policy/valueStr1?includeDeletedFiles=true

Response: 200 OK

```
[
  {
    "FileId": 1,
    "TargetId": 2
  },
  {...},
  {...}
]
```

GetVersion: Returns the version of the ImageRight Application Server.

Method: GET
URL: api/integration/version
Request: api/integration/version

Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "Major": 0,
  "Minor": 0,
  "Build": -1,
  "Revision": -1,
  "MajorRevision": -1,
  "MinorRevision": -1
}
```

Licensing

The Licensing object allows you login/logout, send a still alive heartbeat and retrieve the current count of concurrent users logged into the system.

GetConcurrentLoginData: This fetches data for users currently logged into the system. If there are none, this returns an empty array (NOT NULL).

Method: GET

URL: api/licensing/{featureName}/currentlogindata

Parameters:

Name featureName

Description The id of a feature for which logged in user data should be returned users

Request: api/licensing/featureNameStr1/currentlogindata

Response: 200 OK

```
{
  "LicenseFeature": [
    {
      "Name": "NameStr31",
      "LicenseValidatorData": [
        {
          "Name": "NameStr36",
          "CountParameter": 37,
          "DateParameter": "2019-04-30T07:10:47.8297219-04:00",
          "AccumulatorParameter": 39
        },
        {...},
        {...}
      ],
      "CurrentAccounts": [
        {
          "SecurityId": {
            "Id": -1,
            "HasId": true
          },
          "ProgramId": "ProgramIdStr40"
        },
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ]
}
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetMessages: This fetches any pending messages. If there are none, this returns an empty array (NOT NULL). After this is called, the server discards the pending messages. Thus, calling this won't return the same thing twice in a row. Save the return for processing or process immediately!

Method: GET
URL: api/licensing/{referenceKey}
Parameters:

Name	referenceKey
Description	The last valid ID assigned from the server

Request: api/licensing/referenceKeyStr1
Response: 200 OK

```
[
  {
    "Message": "The license install date has expired.",
    "Parameters": [
      {...},
      {...},
      {...}
    ],
    "CapturedDate": 636922050478307219,
    "MessageSource": "Str1",
    "IsError": true,
    "MessageKey": 2
  },
  {...},
  {...}
]
```

Login: This is invoked to check the license for logging in. It returns the server-assigned ID to allow aliveness notifications.

Method: POST
URL: api/licensing/{programId}/{featureName}/login
Parameters:

Name	programId
Description	The id of a program used to log in

Name	featureName
Description	Not Available

Request: api/licensing/programIdStr1/featureNameStr2/login
Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "LoginId": "LoginIdStr1",
  "IsLicensed": true,
  "LicenseMessages": [
    {
      "Message": "The purchased license count has been exceeded.",
      "Parameters": [
        {...},
        {...},
        {...}
      ],
      "CapturedDate": 636922050478317219,
      "MessageSource": "Str3",
      "IsError": true,
      "MessageKey": 4
    },
    {...},
    {...}
  ]
}
```

Logout: This is used to notify the server that it can discard the session. If this isn't called, the server will detect the lack of polling and drop the connection on its own.

Method: POST
URL: `api/licensing/{loginId}/{featureName}/logout`
Parameters:

Name	loginId
Description	Value returned from Login() or StillAlive()
Name	featureName
Description	The id of a feature used to log in

Request: `api/licensing/loginIdStr1/featureNameStr2/logout`
Response: 200 OK

StillAlive: This is invoked by the lease renewal polling thread to notify the server that it's still alive and in use. Since the server could have crashed from the time the id was created and this call, this can re-set the ID.

Method: POST
URL: `api/licensing/{referenceKey}/{programId}/{featureName}/stillalive`
Parameters:

Name	referenceKey
Description	The last valid ID assigned from the server
Name	programId
Description	The id of a program used to log in
Name	featureName
Description	The name of the feature

Request: `api/licensing/referenceKeyStr1/programIdStr2/featureNameStr3/stillalive`

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Response: 200 OK

```
{
  "self": "selfStr1",
  "pendingMessageCount": 2
}
```

ValidateLicense: Verifies whether the specified feature is licensed or not. Valid features to ask for are OCR, FTS, Retention, and WebClient_Workflow.

Method: GET

URL: api/licensing/ValidateLicense/{feature}

Parameters:

Name	feature
Description	Specified feature to validate.

Request: api/licensing/ValidateLicense/featureStr1

Response: 200 OK

```
{
  "LoginId": "LoginIdStr1",
  "IsLicensed": true,
  "LicenseMessages": [
    {
      "Message": "The purchased license count has been exceeded.",
      "Parameters": [
        {...},
        {...},
        {...}
      ],
      "CapturedDate": 636922050478337219,
      "MessageSource": "Str3",
      "IsError": true,
      "MessageKey": 4
    },
    {...},
    {...}
  ]
}
```

Marks

Marks object allows you to retrieve marks.

GetFileMarkDefinitions: API call to get all FileMarkDefinitions.

Method: GET

URL: api/marks

Request: api/marks

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "markId": 1,
    "fileTypeId": 2,
    "description": "descriptionStr3",
    "programmaticname": "programmaticnameStr4",
    "color": 5,
    "binaryData": [
      64,
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

GetFileMarkDefinitionsByFileTypeId: API call to get FileMarkDefinitions by FileTypeId.

Method: GET
URL: api/marks?fileTypeId={fileTypeId}
Parameters:

Name	fileTypeId
Description	The file type ID to get mark definitions for.

Request: api/marks?fileTypeId=1
Response: 200 OK

```
[
  {
    "markId": 1,
    "fileTypeId": 2,
    "description": "descriptionStr3",
    "programmaticname": "programmaticnameStr4",
    "color": 5,
    "binaryData": [
      64,
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

MarksV2

Marks object allows you to retrieve marks.

GetFileMarkDefinitionsByFileTypeIdV2: API call to get FileMarkDefinitions by FileTypeId.

Method: GET
URL: api/v2/objecttypes/{fileTypeId}/filemarks
Parameters:

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Name	fileTypeId
Description	The file type ID to get mark definitions for.

Request: api/v2/objecttypes/1/filemarks

Response: 200 OK

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

GetFileMarkDefinitionsV2: API call to get all FileMarkDefinitions.

Method: GET

URL: api/v2/filemarks

Request: api/v2/filemarks

Response: 200 OK

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

GetImageDataForFileMark: API call to get image data for a given file mark id.

Method: GET

URL: api/v2/filemarks/{markId}/image

Parameters:

Name	markId
Description	Not Available

Request: api/v2/filemarks/1/image

Response: 200 OK

```
"Image goes here (stream)"
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

GetImageDataForPageMark: API call to get image data for a given page mark id.

Method: GET
URL: api/v2/pagemarks/{markId}/Image
Parameters:

Name	markId
Description	Not Available

Request: api/v2/pagemarks/1/Image
Response: 200 OK

```
"Image goes here (stream)"
```

GetMarksForFile: API call to get marks for a given file id.

Method: GET
URL: api/v2/files/{fileId}/marks
Parameters:

Name	fileId
Description	Not Available

Request: api/v2/files/1/marks
Response: 200 OK

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

GetMarksForPage: API call to get marks for a given page id.

Method: GET
URL: api/v2/pages/{pageId}/marks
Parameters:

Name	pageId
Description	Not Available

Request: api/v2/pages/1/marks
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

GetPageMarkDefinitions: API call to get all PageMarkDefinitions.

Method: GET
URL: api/v2/pagemarks
Request: api/v2/pagemarks
Response: 200 OK

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

GetPageMarkDefinitionsByFileTypeId: API call to get PageMarkDefinitions by FileTypeId.

Method: GET
URL: api/v2/objecttypes/{*fileTypeId*}/pagemarks
Parameters:

Name	fileTypeId
Description	The file type ID to get mark definitions for.

Request: api/v2/objecttypes/1/pagemarks
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

GetPageMarksForFile: API call to get marks for a given file id.

Method: GET
URL: `api/v2/files/{fileId}/pagemarks`
Parameters:

Name	fileId
Description	Not Available

Request: `api/v2/files/1/pagemarks`
Response: 200 OK

```
[
  {
    "Color": 1,
    "Description": "DescriptionStr2",
    "FileTypeId": 3,
    "MarkId": 4,
    "ProgrammaticName": "ProgrammaticNameStr5"
  },
  {...},
  {...}
]
```

Notes

The notes object allows you to create, update, delete and retrieve file or folder regular or/and calculated notes.

AddNoteContainer: Adds a new empty note container.

Method: POST
URL: `api/objects/{objectId}/notes/{category}`
Parameters:

Name	objectId
Description	The file or folder id.
Name	category
Description	The note category to be created.
Values	Regular, Calculated

Request: `api/objects/1/notes/regular`

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Response: 201 Created

```
{
  "Id": 11,
  "Version": 12,
  "Category": "Regular",
  "IsDeleted": true,
  "Notes": [
    {
      "NoteId": 9,
      "IsDefault": true,
      "Status": "Visible",
      "Items": [
        {
          "ItemId": 1,
          "Replaced": 2,
          "TimeStamp": "2019-04-30T07:10:47.7837219-04:00",
          "Text": "TextStr4",
          "Drawer": "DrawerStr5",
          "FileNumber": "FileNumberStr6",
          "IsHidden": true,
          "UserName": "UserNameStr8"
        },
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ]
}
```

Error codes:

120	The note container already exists.
110	File or folder template does not support regular notes.
111	File or folder template does not support calculated notes.

AddNoteItem: Adds a note item. Requires a lock call before and an unlock call after it.

Method:	POST
URL:	api/objects/{objectId}/notes
Parameters:	
Name	objectId
Description	The file or folder id.
Name	model
Description	The note model to be added.
Request:	api/objects/1/notes

```
{
  "Category": "Regular",
  "Text": "TextStr2",
  "Amount": 3.0
}
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 201 Created

```
{
  "ItemId": 1,
  "Replaced": 2,
  "TimeStamp": "2019-04-30T07:10:47.7867219-04:00",
  "Text": "TextStr4",
  "Drawer": "DrawerStr5",
  "FileNumber": "FileNumberStr6",
  "IsHidden": true,
  "UserName": "UserNameStr8"
}
```

Error codes:

100	You are trying to lock a note locked by another user.
102	The note is not locked.
110	File or folder template does not support regular notes.
111	File or folder template does not support calculated notes.

GetNotes: Returns file or folder notes. If there are no notes returns an empty container with Id = -1

Method:	GET
URL:	api/objects/{objectId}/notes/{category}?status={status}&includeHidden={includeHidden}&includeDeleted={includeDeleted}
Parameters:	
Name	objectId
Description	The file or folder id.
Name	category
Description	Notes type. Optional , default value is null (empty) .
Values	Regular, Calculated
Name	status
Description	Notes status. Optional , default value is Visible .
Values	Visible, Deleted, Cut, All
Name	includeHidden
Description	Specifies do we need to include hidden note items. Optional , default value is False .
Name	includeDeleted
Description	Specifies do we need to include a deleted note container. Optional , default value is False .
Request:	api/objects/1/notes/regular?status=visible&includeHidden=true&includeDeleted=true
Response:	200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Id": 11,
    "Version": 12,
    "Category": "Regular",
    "IsDeleted": true,
    "Notes": [
      {
        "NoteId": 9,
        "IsDefault": true,
        "Status": "Visible",
        "Items": [
          {
            "ItemId": 1,
            "Replaced": 2,
            "TimeStamp": "2019-04-30T07:10:47.7867219-04:00",
            "Text": "TextStr4",
            "Drawer": "DrawerStr5",
            "FileNumber": "FileNumberStr6",
            "IsHidden": true,
            "UserName": "UserNameStr8"
          },
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

Error codes:

121	The note container was not found.
-----	-----------------------------------

LockNote: Locks a note.

Method:	POST
URL:	api/objects/{objectId}/notes/{category}/lock?version={version}
Parameters:	
Name	objectId
Description	The file or folder id.
Name	category
Description	The note category.
Values	Regular, Calculated
Name	version
Description	The note version.
Request:	api/objects/1/notes/regular/lock?version=2
Response:	200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Error codes:

100	You are trying to lock a note locked by another user.
101	The provided note version is not correct.

SetNoteAsDefault: Sets a note collection as default. Requires a lock call before and an unlock call after it.

Method: POST

URL: api/objects/{objectId}/notes/{category}/{collectionId}/setasdefault

Parameters:

Name	objectId
Description	The file or folder id.
Name	collectionId
Description	The collection identifier.
Name	category
Description	The note category.
Values	Regular, Calculated

Request: api/objects/1/notes/regular/2/setasdefault

Response: 200 OK

Error codes:

100	You are trying to lock a note locked by another user.
102	The note is not locked.
130	The provided note collection number is invalid.

UnlockNote: Unlocks a note.

Method: POST

URL: api/objects/{objectId}/notes/{category}/unlock

Parameters:

Name	objectId
Description	The file or folder id.
Name	category
Description	The note category.
Values	Regular, Calculated

Request: api/objects/1/notes/regular/unlock

Response: 200 OK

Error codes:

100	You are trying to lock a note locked by another user.
-----	---

ObjectTypes

Controller supports operations related to Object Types

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

AssignPermissionsForType: Assign permissions to an existing storage type for a specific ImageRight user, group or role. Storage types are created in the ImageRight Management Console and consist of locations, drawers, files, folders and document types.

Method: POST

URL: `api/objecttypes/{objectTypeId}/assignpermissions`

Parameters:

Name	objectTypeId
Description	The id of the type
Name	objectPermissionModel
Description	Implementation of

Request: `api/objecttypes/1/assignpermissions`

```
{
  "AccountId": 2,
  "Permissions": "read"
}
```

Response: 200 OK

Error codes:

11	The account was not found.
1002	The type was not found.
29	The permission was not found.

DenyPermissionsForType: Deny permissions to an existing storage type for a specific ImageRight user, group or role. Storage types are created in the ImageRight Management Console and consist of locations, drawers, files, folders and document types.

Method: POST

URL: `api/objecttypes/{objectTypeId}/denypermissions`

Parameters:

Name	objectTypeId
Description	The id of the type
Name	objectPermissionModel
Description	Implementation of

Request: `api/objecttypes/1/denypermissions`

```
{
  "AccountId": 2,
  "Permissions": "read"
}
```

Response: 200 OK

Error codes:

11	The account was not found.
1002	The type was not found.
29	The permission was not found.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetAllowedTypes: Retrieves allowed child types for a given type ID. The type ID must be a drawer type ID or a location type ID.

Method: GET

URL: `api/objecttypes/allowedtypes?typeId={typeId}`

Parameters:

Name typeId

Description The drawer/location type ID to get child types for.

Request: `api/objecttypes/allowedtypes?typeId=1`

Response: 200 OK

```
[
  {
    "ClassId": 1,
    "Id": 2,
    "Name": "NameStr3",
    "Description": "DescriptionStr4",
    "OverlayId": 5,
    "AutomationId": "AutomationIdStr6",
    "DateLastModified": "2019-04-30T07:10:47.8537219-04:00",
    "EffectivePermissions": 8
  },
  {...},
  {...}
]
```

Error codes:

1001

The type is not a drawer type nor a location type.

GetAllowedTypesForContainer: Retrieves allowed child types for a given container ID(Root,Location,Drawer,File,Folder).

Method: GET

URL: `api/containers/{objectId}/allowedtypes`

Parameters:

Name objectId

Description The container ID to get allowed child types for.

Request: `api/containers/1/allowedtypes`

Response: 200 OK

```
[
  {
    "ClassId": 1,
    "Id": 2,
    "Name": "NameStr3",
    "Description": "DescriptionStr4",
    "OverlayId": 5,
    "AutomationId": "AutomationIdStr6",
    "DateLastModified": "2019-04-30T07:10:47.8547219-04:00",
    "EffectivePermissions": 8
  },
  {...},
  {...}
]
```

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

Error codes: 903 Object class is unsupported for this operation.

GetAttributeDefinitionsForType: Retrieves the attribute definitions for a given type ID.

Method: GET
URL: `api/objecttypes/{objectTypeId}/attributes`
Parameters:

Name	
Description	The type ID to get attribute definitions for.

Request: `api/objecttypes/1/attributes`
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "Id": 1,
  "Type": "atInt",
  "DisplayName": "DisplayNameStr2",
  "Name": "NameStr3",
  "Description": "DescriptionStr4",
  "Mandatory": true,
  "DefaultValue": {...},
  "HasDefaultValue": true,
  "Disabled": true,
  "Visibility": true,
  "ValidationRule": {
    "ControlType": "CheckBox",
    "MinValue": 10.0,
    "MaxValue": 11.0,
    "Mask": "MaskStr12",
    "Length": 13,
    "Choices": [
      {
        "Value": {...},
        "Description": "DescriptionStr15",
        "Tag": "TagStr16"
      },
      {...},
      {...}
    ],
    "Validators": [
      {
        "Mask": "MaskStr17",
        "Name": "NameStr18",
        "Type": "TypeStr19",
        "ValidatorItemList": [
          "Str20",
          {...},
          {...}
        ],
        "ValidatorRange": [
          {
            "From": "FromStr21",
            "To": "ToStr22"
          },
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ],
    "Readonly": true,
    "AllowNew": true,
    "NameVisible": true,
    "DescrVisible": true,
    "Separator": "SeparatorStr27",
    "Order": 28,
    "Flags": 29,
    "ShouldSort": true,
    "IsDefaultChoices": true
  }
}
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
    },
    "IsEncrypted": true
  },
  {...},
  {...}
]
```

Error codes:

600	The attribute was not found.
------------	------------------------------

GetFileTypeExtensions: Retrieves a template tree of allowed types for a specified file type ID.

Method: GET
URL: `api/objecttypes/{fileTypeId}/extensions`

Parameters:

Name	fileTypeId
Description	The file type ID to get the template tree for.

Request: `api/objecttypes/1/extensions`

Response: 200 OK

```
{
  "FilePromptModel": {
    "FileNumberPrompt": "FileNumberPromptStr22",
    "FileNamePrompt": "FileNamePromptStr23",
    "FileNameDescription": "FileNameDescriptionStr24"
  },
  "FileNumberMaskModel": {
    "Fields": [
      {
        "MinimumFieldLength": 34,
        "MaximumFieldLength": 35,
        "Enabled": true,
        "ZeroPadding": "None"
      },
      {...},
      {...}
    ]
  }
}
```

Error codes:

1000	The type is not a file type.
-------------	------------------------------

GetFileTypeTemplate: Retrieves a template tree of allowed types for a specified file type ID.

Method: GET
URL: `api/objecttypes/{fileTypeId}/template`

Parameters:

Name	fileTypeId
Description	The file type ID to get the template tree for.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Request: api/objecttypes/1/template

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Id": 1,
  "NotesAllowed": true,
  "CalculateNotesAllowed": true,
  "EffectivePermissions": 4,
  "FolderTypeDataResults": [
    {
      "IsRepeatable": true,
      "NotesAllowed": true,
      "CalculateNotesAllowed": true,
      "FolderTypeDataResults": [
        {
          "IsRepeatable": true,
          "NotesAllowed": true,
          "CalculateNotesAllowed": true,
          "FolderTypeDataResults": [
            {
              "IsRepeatable": true,
              "NotesAllowed": true,
              "CalculateNotesAllowed": true,
              "FolderTypeDataResults": [],
              "DocumentTypeDataResults": [
                {
                  "ClassId": 14,
                  "Id": 15,
                  "Name": "NameStr16",
                  "Description": "DescriptionStr17",
                  "OverlayId": 18,
                  "AutomationId": "AutomationIdStr19",
                  "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
                  "EffectivePermissions": 21
                },
                {...},
                {...}
              ],
              "ClassId": 22,
              "Id": 23,
              "Name": "NameStr24",
              "Description": "DescriptionStr25",
              "OverlayId": 26,
              "AutomationId": "AutomationIdStr27",
              "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
              "EffectivePermissions": 29
            },
            {...},
            {...}
          ],
          "DocumentTypeDataResults": [
            {
              "ClassId": 30,
              "Id": 31,
              "Name": "NameStr32",
              "Description": "DescriptionStr33",
              "OverlayId": 34,
              "AutomationId": "AutomationIdStr35",
              "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
              "EffectivePermissions": 37
            },
            {...},
            {...}
          ]
        },
        {...},
        {...}
      ],
      "DocumentTypeDataResults": [
        {
          "ClassId": 30,
          "Id": 31,
          "Name": "NameStr32",
          "Description": "DescriptionStr33",
          "OverlayId": 34,
          "AutomationId": "AutomationIdStr35",
          "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
          "EffectivePermissions": 37
        },
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ],
  "DocumentTypeDataResults": [
    {
      "ClassId": 30,
      "Id": 31,
      "Name": "NameStr32",
      "Description": "DescriptionStr33",
      "OverlayId": 34,
      "AutomationId": "AutomationIdStr35",
      "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
      "EffectivePermissions": 37
    },
    {...},
    {...}
  ]
}
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{...},
{...}
],
"ClassId": 38,
"Id": 39,
"Name": "NameStr40",
"Description": "DescriptionStr41",
"OverlayId": 42,
"AutomationId": "AutomationIdStr43",
"DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
"EffectivePermissions": 45
},
{...},
{...}
],
"DocumentTypeDataResults": [
  {
    "ClassId": 46,
    "Id": 47,
    "Name": "NameStr48",
    "Description": "DescriptionStr49",
    "OverlayId": 50,
    "AutomationId": "AutomationIdStr51",
    "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
    "EffectivePermissions": 53
  },
  {...},
  {...}
],
"ClassId": 54,
"Id": 55,
"Name": "NameStr56",
"Description": "DescriptionStr57",
"OverlayId": 58,
"AutomationId": "AutomationIdStr59",
"DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
"EffectivePermissions": 61
},
{...},
{...}
],
"DocumentTypeDataResults": [
  {
    "ClassId": 62,
    "Id": 63,
    "Name": "NameStr64",
    "Description": "DescriptionStr65",
    "OverlayId": 66,
    "AutomationId": "AutomationIdStr67",
    "DateLastModified": "2019-04-30T07:10:47.8597219-04:00",
    "EffectivePermissions": 69
  },
  {...},
  {...}
]
}
```

Error codes:

1000

The type is not a file type.

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

GetFileTypeTemplateList: Retrieves an array of template trees of allowed types for an array of specified file type IDs.

Method: GET
URL: `api/objecttypes/templatelist?typeIds[0]={typeIds[0]}&typeIds[1]={typeIds[1]}`
Parameters:

Name	typeIds
Description	The file type ID(s) to get the template tree(s) for.

Request: `api/objecttypes/templatelist?typeIds=system.int64[]`
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "Id": 1,
    "NotesAllowed": true,
    "CalculateNotesAllowed": true,
    "EffectivePermissions": 4,
    "FolderTypeDataResults": [
      {
        "IsRepeatable": true,
        "NotesAllowed": true,
        "CalculateNotesAllowed": true,
        "FolderTypeDataResults": [
          {
            "IsRepeatable": true,
            "NotesAllowed": true,
            "CalculateNotesAllowed": true,
            "FolderTypeDataResults": [
              {
                "IsRepeatable": true,
                "NotesAllowed": true,
                "CalculateNotesAllowed": true,
                "FolderTypeDataResults": [],
                "DocumentTypeDataResults": [
                  {
                    "ClassId": 14,
                    "Id": 15,
                    "Name": "NameStr16",
                    "Description": "DescriptionStr17",
                    "OverlayId": 18,
                    "AutomationId": "AutomationIdStr19",
                    "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
                    "EffectivePermissions": 21
                  },
                  {...},
                  {...}
                ],
                "ClassId": 22,
                "Id": 23,
                "Name": "NameStr24",
                "Description": "DescriptionStr25",
                "OverlayId": 26,
                "AutomationId": "AutomationIdStr27",
                "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
                "EffectivePermissions": 29
              },
              {...},
              {...}
            ],
            "DocumentTypeDataResults": [
              {
                "ClassId": 30,
                "Id": 31,
                "Name": "NameStr32",
                "Description": "DescriptionStr33",
                "OverlayId": 34,
                "AutomationId": "AutomationIdStr35",
                "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
                "EffectivePermissions": 37
              }
            ]
          }
        ]
      }
    ]
  }
]
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
    },
    {...},
    {...}
  ],
  "ClassId": 38,
  "Id": 39,
  "Name": "NameStr40",
  "Description": "DescriptionStr41",
  "OverlayId": 42,
  "AutomationId": "AutomationIdStr43",
  "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
  "EffectivePermissions": 45
},
{...},
{...}
],
"DocumentTypeDataResults": [
  {
    "ClassId": 46,
    "Id": 47,
    "Name": "NameStr48",
    "Description": "DescriptionStr49",
    "OverlayId": 50,
    "AutomationId": "AutomationIdStr51",
    "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
    "EffectivePermissions": 53
  },
  {...},
  {...}
],
"ClassId": 54,
"Id": 55,
"Name": "NameStr56",
"Description": "DescriptionStr57",
"OverlayId": 58,
"AutomationId": "AutomationIdStr59",
"DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
"EffectivePermissions": 61
},
{...},
{...}
],
"DocumentTypeDataResults": [
  {
    "ClassId": 62,
    "Id": 63,
    "Name": "NameStr64",
    "Description": "DescriptionStr65",
    "OverlayId": 66,
    "AutomationId": "AutomationIdStr67",
    "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",
    "EffectivePermissions": 69
  },
  {...},
  {...}
]
},
{...},
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{...}  
]
```

Error codes:

1000	The type is not a file type.
-------------	------------------------------

GetObjectType: Retrieves the type data for a particular object type ID.

Method: GET
URL: `api/objecttypes/{objectTypeId}`
Parameters:

Name	objectTypeId
Description	The type ID to get data for.

Request: `api/objecttypes/1`
Response: 200 OK

```
{  
  "ClassId": 1,  
  "Id": 2,  
  "Name": "NameStr3",  
  "Description": "DescriptionStr4",  
  "OverlayId": 5,  
  "AutomationId": "AutomationIdStr6",  
  "DateLastModified": "2019-04-30T07:10:47.8607219-04:00",  
  "EffectivePermissions": 8  
}
```

GetSortOptionsForType: Retrieves the sorting options for a given type ID. The type ID must be a file or folder type ID.

Method: GET
URL: `api/objecttypes/{objectTypeId}/sortoptions`
Parameters:

Name	objectTypeId
Description	The file/folder type to get sorting options for.

Request: `api/objecttypes/1/sortoptions`
Response: 200 OK

```
{  
  "SortOptions": "SortOptionsStr1",  
  "RepeatableFolderNaturalSort": true,  
  "RepeatableFolderSortAscending": true  
}
```

Error codes:

1004	The type is not a file nor a folder type.
1002	The type was not found.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetTypesForClass: Retrieves all the object type data for a particular class.

Method: GET
URL: `api/objecttypes/{standardObjectClass}`

Parameters:

Name	standardObjectClass
Description	Enumeration that represents the Object Class. Valid terms are Drawer, File, Folder, Document, FileContainer (which is a location).
Values	RootContainer, FileContainer, Drawer, File, Folder, Document

Request: `api/objecttypes/rootcontainer`

Response: 200 OK

```
[
  {
    "ClassId": 1,
    "Id": 2,
    "Name": "NameStr3",
    "Description": "DescriptionStr4",
    "OverlayId": 5,
    "AutomationId": "AutomationIdStr6",
    "DateLastModified": "2019-04-30T07:10:47.8617219-04:00",
    "EffectivePermissions": 8
  },
  {...},
  {...}
]
```

OcrDataSets

Ocr DataSets object allows you to create and retrieve DataSets.

CreateOcrDataSet: Creates a new dataset for a form recognition result.

Method: POST
URL: `api/pages/{pageId}/ocrdatasets`

Parameters:

Name	pageId
Description	The page id.

Name	model
Description	The OcrDataSet model to be created.

Request: `api/pages/1/ocrdatasets`

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Id": 2,
  "PageId": 3,
  "FormVersionId": 4,
  "Created": "2019-04-30T07:10:47.8037219-04:00",
  "OcrDataPages": [
    {
      "Id": 6,
      "FormPageNumber": 7,
      "Confidence": 8,
      "OcrData": [
        {
          "Id": 10,
          "FieldName": "FieldNameStr11",
          "Value": "ValueStr12",
          "Confidence": 13,
          "OcrFormFieldId": 14
        },
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ]
}
```

Response: 201 Created

1

GetOcrDataSetByDocumentId: Returns a list of ocr datasets by document id.

Method:	GET
URL:	api/documents/{documentId}/ocrdatasets
Parameters:	
Name	documentId
Description	The document id.
Request:	api/documents/1/ocrdatasets
Response:	200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Id": 1,
    "PageId": 2,
    "FormVersionId": 3,
    "Created": "2019-04-30T07:10:47.8047219-04:00",
    "OcrDataPages": [
      {
        "Id": 5,
        "FormPageNumber": 6,
        "Confidence": 7,
        "OcrData": [
          {
            "Id": 9,
            "FieldName": "FieldNameStr10",
            "Value": "ValueStr11",
            "Confidence": 12,
            "OcrFormFieldId": 13
          },
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

GetOcrDataSetByPageId: Returns the ocr dataset by page id.

Method:	GET
URL:	api/pages/{pageId}/ocrdatasets
Parameters:	
Name	pageId
Description	The page id.
Request:	api/pages/1/ocrdatasets
Response:	200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Id": 1,
  "PageId": 2,
  "FormVersionId": 3,
  "Created": "2019-04-30T07:10:47.8047219-04:00",
  "OcrDataPages": [
    {
      "Id": 5,
      "FormPageNumber": 6,
      "Confidence": 7,
      "OcrData": [
        {
          "Id": 9,
          "FieldName": "FieldNameStr10",
          "Value": "ValueStr11",
          "Confidence": 12,
          "OcrFormFieldId": 13
        },
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ]
}
```

OcrFields

OcrFields object allows you to retrieve OCR fields and set them as encrypted.

CreateOcrFields: Creates a new OCR field.

Method:	POST
URL:	api/ocrfields
Parameters:	
Name	ocrFieldModel
Description	The OCR field model

Request: api/ocrfields

```
{
  "Id": 1,
  "FieldName": "FieldNameStr2",
  "Encrypted": true
}
```

Response: 200 OK

```
{
  "Id": 1,
  "FieldName": "FieldNameStr2",
  "Encrypted": true
}
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetOcrFields: Returns a list of all OCR fields.

Method: GET
URL: api/ocrfields
Request: api/ocrfields
Response: 200 OK

```
[
  {
    "Id": 1,
    "FieldName": "FieldNameStr2",
    "Encrypted": true
  },
  {...},
  {...}
]
```

SetOcrFieldAsEncrypted: Mark a field as encrypted.

Method: PUT
URL: api/ocrfields/{fieldId}/pii
Parameters:

Name	Description
fieldId	The ID of the OCR field.
isEncrypted	Not Available

Request: api/ocrfields/1/pii

```
true
```

Response: 200 OK

OcrFolders

OcrFolders object allows you to create, retrieve, delete and rename OCR Folders.

CreateOcrFolder: Creates a new OCR folder.

Method: POST
URL: api/ocrfolders
Parameters:

Name	Description
folderModel	The model of the folder.

Request: api/ocrfolders

```
{
  "Id": 1,
  "FolderName": "FolderNameStr2"
}
```

Response: 201 Created

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Id": 1,
  "FolderName": "FolderNameStr2"
}
```

GetOcrFolders: Returns a list of OCR folders.

Method: GET
URL: api/ocrfolders
Request: api/ocrfolders
Response: 200 OK

```
[
  {
    "Id": 1,
    "FolderName": "FolderNameStr2"
  },
  {...},
  {...}
]
```

RemoveOcrFolder: Deletes the OCR folder.

Method: DELETE
URL: api/ocrfolders
Parameters:

Name	folderId
Description	The ID of the OCR folder

Request: api/ocrfolders

1

Response: 200 OK

RenameOcrFolder: Renames the OCR folder.

Method: PUT
URL: api/ocrfolders/{folderId}/name
Parameters:

Name	folderId
Description	The ID of the OCR folder.
Name	newFolderName
Description	The name of the OCR folder.

Request: api/ocrfolders/1/name

"Str2"

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 200 OK

OcrForms

OcrForms object allows you to manage the OCR Forms.

CalculateOcrFormsSize: Returns total size of the selected OCR form docs.

Method: POST
URL: api/ocrforms/size
Parameters:

Name	
Description	formIds OCR form ids to calculate

Request: api/ocrforms/size

```
[
  1,
  {...},
  {...}
]
```

Response: 200 OK

```
1
```

CreateOcrForm: Creates a new OCR form.

Method: POST
URL: api/ocrforms
Parameters:

Name	
Description	formCreateModel The OcrForm model to create

Request: api/ocrforms

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "FormName": "FormNameStr1",
  "FolderId": 2,
  "FormVersions": [
    {
      "FormFields": [
        {
          "Id": 3,
          "FormVersionId": 4,
          "FieldId": 5,
          "FormPage": 6,
          "FieldType": 7,
          "MinFieldConfidence": 8,
          "X": 9,
          "Y": 10,
          "Width": 11,
          "Height": 12,
          "TextType": 13,
          "Dropout": 14,
          "Sensitivity": 15,
          "Frame": 16
        },
        {...},
        {...}
      ],
      "AreasOfInterest": [
        {
          "Id": 17,
          "FormVersionId": 18,
          "FormPage": 19,
          "X": 20,
          "Y": 21,
          "Width": 22,
          "Height": 23
        },
        {...},
        {...}
      ],
      "Id": 24,
      "FormId": 25,
      "FormVersion": 26,
      "Created": "2019-04-30T07:10:47.7887219-04:00",
      "CreatedBy": 28,
      "MinFormConfidence": 29,
      "Description": "DescriptionStr30",
      "DocTypeId": 31,
      "TaskCreationLevel": 32,
      "StepRootId": 33,
      "AssignedTo": 34,
      "Priority": 35,
      "FileNumberField": "FileNumberFieldStr36",
      "FileNameField": "FileNameFieldStr37"
    },
    {...},
    {...}
  ],
  "FormDoc": [
    64,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{...},
{...}
]
```

Response: 201 Created

```
{
  "Id": 1,
  "FormName": "FormNameStr2",
  "FolderId": 3,
  "LiveVersionId": 4,
  "Deleted": "2019-04-30T07:10:47.7917219-04:00",
  "DeletedBy": 6,
  "Hidden": true,
  "Locked": "2019-04-30T07:10:47.7917219-04:00",
  "LockedBy": 9,
  "LockExpiration": "2019-04-30T07:10:47.7917219-04:00"
}
```

DeleteOcrForm: Deletes the OCR form permanently.

Method: DELETE
URL: api/ocrforms/{formId}
Parameters:

Name	formId
Description	The ID of the OCR form.

Request: api/ocrforms/1
Response: 200 OK

GetOcrForm: Returns the OCR form.

Method: GET
URL: api/ocrforms/{formId}
Parameters:

Name	formId
Description	The ID of the OCR form.

Request: api/ocrforms/1
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Id": 1,
  "FormName": "FormNameStr2",
  "FolderId": 3,
  "LiveVersionId": 4,
  "Deleted": "2019-04-30T07:10:47.7927219-04:00",
  "DeletedBy": 6,
  "Hidden": true,
  "Locked": "2019-04-30T07:10:47.7927219-04:00",
  "LockedBy": 9,
  "LockExpiration": "2019-04-30T07:10:47.7927219-04:00"
}
```

GetOcrFormDoc: Returns the OCR form document.

Method: GET
URL: api/ocrforms/{formId}/doc
Parameters:

Name	formId
Description	The ID of the OCR form.

Request: api/ocrforms/1/doc
Response: 200 OK

```
[
  64,
  {...},
  {...}
]
```

GetOcrForms: Returns an array of the OCR forms.

Method: GET
URL: api/ocrforms?includeDeleted={includeDeleted}
Parameters:

Name	includeDeleted
Description	If true, all OCR forms including the forms in the deleted state will be returned. Optional , default value is False .

Request: api/ocrforms?includeDeleted=true
Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Id": 1,
    "FormName": "FormNameStr2",
    "FolderId": 3,
    "LiveVersionId": 4,
    "Deleted": "2019-04-30T07:10:47.7927219-04:00",
    "DeletedBy": 6,
    "Hidden": true,
    "Locked": "2019-04-30T07:10:47.7927219-04:00",
    "LockedBy": 9,
    "LockExpiration": "2019-04-30T07:10:47.7927219-04:00"
  },
  {...},
  {...}
]
```

LockOcrForm: Locks the OCR form.

Method:	POST				
URL:	api/ocrforms/lock				
Parameters:					
	<table><tr><td>Name</td><td>formId</td></tr><tr><td>Description</td><td>The ID of the OCR form.</td></tr></table>	Name	formId	Description	The ID of the OCR form.
Name	formId				
Description	The ID of the OCR form.				
Request:	api/ocrforms/lock				
	1				
Response:	200 OK				

MarkOcrFormAsDeleted: Deletes the OCR form.

Method:	POST				
URL:	api/ocrforms/markdeleted				
Parameters:					
	<table><tr><td>Name</td><td>formId</td></tr><tr><td>Description</td><td>The ID of the OCR form.</td></tr></table>	Name	formId	Description	The ID of the OCR form.
Name	formId				
Description	The ID of the OCR form.				
Request:	api/ocrforms/markdeleted				
	1				
Response:	200 OK				

MoveOcrForm: Moves the OCR form to the certain folder.

Method:	PUT		
URL:	api/ocrforms/{formId}/{folderId}		
Parameters:			
	<table><tr><td>Name</td><td>formId</td></tr></table>	Name	formId
Name	formId		

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

Push Notifications

Images

The ID of the OCR form.

folderId

The ID of the OCR folder. **Optional**, default value is **null (empty)**.

Response: 200 OK

RenameOcrForm: Renames the OCR form.

PUT

api/ocrforms/{formId}/name

Name

formId

The ID of the OCR form.

name

The name of the OCR form.

"Str2"

Response: 200 OK

RestoreOcrForm: Restores the OCR form

PUT

api/ocrforms/{formId}/restore

Name

formId

The ID of the OCR form.

folderId

The Folder ID to restore into

2

Response: 200 OK

UnlockOcrForm: Unlocks the OCR form.

POST

api/ocrforms/unlock

Name

formId

The ID of the OCR form.

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Request: api/ocrforms/unlock

1

Response: 200 OK

OcrFormVersions

OcrFormVersions object allows you to create, update, retrieve and set as active OCR form Versions.

AddVersion: Creates a new OCR form version.

Method:	POST
URL:	api/ocrforms/{formId}/versions
Parameters:	
Name	formId
Description	The ID of the OCR form.
Name	formVersionModel
Description	The model of the OCR form version.
Request:	api/ocrforms/1/versions

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "FormFields": [
    {
      "Id": 2,
      "FormVersionId": 3,
      "FieldId": 4,
      "FormPage": 5,
      "FieldType": 6,
      "MinFieldConfidence": 7,
      "X": 8,
      "Y": 9,
      "Width": 10,
      "Height": 11,
      "TextType": 12,
      "Dropout": 13,
      "Sensitivity": 14,
      "Frame": 15
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 16,
      "FormVersionId": 17,
      "FormPage": 18,
      "X": 19,
      "Y": 20,
      "Width": 21,
      "Height": 22
    },
    {...},
    {...}
  ],
  "Id": 23,
  "FormId": 24,
  "FormVersion": 25,
  "Created": "2019-04-30T07:10:47.8347219-04:00",
  "CreatedBy": 27,
  "MinFormConfidence": 28,
  "Description": "DescriptionStr29",
  "DocTypeId": 30,
  "TaskCreationLevel": 31,
  "StepRootId": 32,
  "AssignedTo": 33,
  "Priority": 34,
  "FileNumberField": "FileNumberFieldStr35",
  "FileNameField": "FileNameFieldStr36"
}
```

Response: 201 Created

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FormFields": [
    {
      "Id": 1,
      "FormVersionId": 2,
      "FieldId": 3,
      "FormPage": 4,
      "FieldType": 5,
      "MinFieldConfidence": 6,
      "X": 7,
      "Y": 8,
      "Width": 9,
      "Height": 10,
      "TextType": 11,
      "Dropout": 12,
      "Sensitivity": 13,
      "Frame": 14
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 15,
      "FormVersionId": 16,
      "FormPage": 17,
      "X": 18,
      "Y": 19,
      "Width": 20,
      "Height": 21
    },
    {...},
    {...}
  ],
  "Id": 22,
  "FormId": 23,
  "FormVersion": 24,
  "Created": "2019-04-30T07:10:47.8347219-04:00",
  "CreatedBy": 26,
  "MinFormConfidence": 27,
  "Description": "DescriptionStr28",
  "DocTypeId": 29,
  "TaskCreationLevel": 30,
  "StepRootId": 31,
  "AssignedTo": 32,
  "Priority": 33,
  "FileNumberField": "FileNumberFieldStr34",
  "FileNameField": "FileNameFieldStr35"
}
```

ChangeActiveOcrVersion: Changes the active version for the OCR form.

Method:	PUT
URL:	api/ocrforms/{formId}/active
Parameters:	
Name	formId

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

	Description	The ID of the OCR form.
	Name	newActiveFormVersionId
	Description	The ID of the new active OCR form version.
Request:	api/ocrforms/1/active	
	2	
Response:	200 OK	

GetActiveOcrVersion: Returns the active version of the OCR form.

Method:	GET	
URL:	api/ocrforms/{formId}/versions/active	
Parameters:	Name	formId
	Description	The ID of the OCR form.
Request:	api/ocrforms/1/versions/active	
Response:	200 OK	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FormFields": [
    {
      "Id": 1,
      "FormVersionId": 2,
      "FieldId": 3,
      "FormPage": 4,
      "FieldType": 5,
      "MinFieldConfidence": 6,
      "X": 7,
      "Y": 8,
      "Width": 9,
      "Height": 10,
      "TextType": 11,
      "Dropout": 12,
      "Sensitivity": 13,
      "Frame": 14
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 15,
      "FormVersionId": 16,
      "FormPage": 17,
      "X": 18,
      "Y": 19,
      "Width": 20,
      "Height": 21
    },
    {...},
    {...}
  ],
  "Id": 22,
  "FormId": 23,
  "FormVersion": 24,
  "Created": "2019-04-30T07:10:47.8347219-04:00",
  "CreatedBy": 26,
  "MinFormConfidence": 27,
  "Description": "DescriptionStr28",
  "DocTypeId": 29,
  "TaskCreationLevel": 30,
  "StepRootId": 31,
  "AssignedTo": 32,
  "Priority": 33,
  "FileNumberField": "FileNumberFieldStr34",
  "FileNameField": "FileNameFieldStr35"
}
```

GetLatestOcrVersion: Returns the latest OCR form version for the certain OCR form.

Method:	GET
URL:	api/ocrforms/{formId}/versions/latest
Parameters:	
Name	formId

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description The ID of the OCR form.

Request: api/ocrforms/1/versions/latest

Response: 200 OK

```
{
  "FormFields": [
    {
      "Id": 1,
      "FormVersionId": 2,
      "FieldId": 3,
      "FormPage": 4,
      "FieldType": 5,
      "MinFieldConfidence": 6,
      "X": 7,
      "Y": 8,
      "Width": 9,
      "Height": 10,
      "TextType": 11,
      "Dropout": 12,
      "Sensitivity": 13,
      "Frame": 14
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 15,
      "FormVersionId": 16,
      "FormPage": 17,
      "X": 18,
      "Y": 19,
      "Width": 20,
      "Height": 21
    },
    {...},
    {...}
  ],
  "Id": 22,
  "FormId": 23,
  "FormVersion": 24,
  "Created": "2019-04-30T07:10:47.8347219-04:00",
  "CreatedBy": 26,
  "MinFormConfidence": 27,
  "Description": "DescriptionStr28",
  "DocTypeId": 29,
  "TaskCreationLevel": 30,
  "StepRootId": 31,
  "AssignedTo": 32,
  "Priority": 33,
  "FileNumberField": "FileNumberFieldStr34",
  "FileNameField": "FileNameFieldStr35"
}
```

GetOcrVersion: Returns the OCR form version.

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Method: GET
URL: api/ocrforms/{formId}/versions/{verId}
Parameters:

Name	formId
Description	The ID of the OCR form.
Name	verId
Description	The ID of the OCR form version.

Request: api/ocrforms/1/versions/2
Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FormFields": [
    {
      "Id": 1,
      "FormVersionId": 2,
      "FieldId": 3,
      "FormPage": 4,
      "FieldType": 5,
      "MinFieldConfidence": 6,
      "X": 7,
      "Y": 8,
      "Width": 9,
      "Height": 10,
      "TextType": 11,
      "Dropout": 12,
      "Sensitivity": 13,
      "Frame": 14
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 15,
      "FormVersionId": 16,
      "FormPage": 17,
      "X": 18,
      "Y": 19,
      "Width": 20,
      "Height": 21
    },
    {...},
    {...}
  ],
  "Id": 22,
  "FormId": 23,
  "FormVersion": 24,
  "Created": "2019-04-30T07:10:47.8347219-04:00",
  "CreatedBy": 26,
  "MinFormConfidence": 27,
  "Description": "DescriptionStr28",
  "DocTypeId": 29,
  "TaskCreationLevel": 30,
  "StepRootId": 31,
  "AssignedTo": 32,
  "Priority": 33,
  "FileNumberField": "FileNumberFieldStr34",
  "FileNameField": "FileNameFieldStr35"
}
```

GetOcrVersions: Returns a list of versions for the certain form.

Method:	GET
URL:	api/ocrforms/{formId}/versions
Parameters:	
Name	formId

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description	The ID of the OCR form.
-------------	-------------------------

Request: api/ocrforms/1/versions

Response: 200 OK

```
[
  {
    "Id": 1,
    "FormId": 2,
    "FormVersion": 3,
    "Created": "2019-04-30T07:10:47.8357219-04:00",
    "CreatedBy": 5,
    "MinFormConfidence": 6,
    "Description": "DescriptionStr7",
    "DocTypeId": 8,
    "TaskCreationLevel": 9,
    "StepRootId": 10,
    "AssignedTo": 11,
    "Priority": 12,
    "FileNumberField": "FileNumberFieldStr13",
    "FileNameField": "FileNameFieldStr14"
  },
  {...},
  {...}
]
```

UpdateVersion: Updates the OCR form version.

Method:	POST
URL:	api/ocrforms/{formId}/versions/{verId}
Parameters:	
Name	formId
Description	The ID of the OCR form.
Name	verId
Description	The ID of the OCR form version.
Name	formVersionModel
Description	The model of the OCR form version.
Request:	api/ocrforms/1/versions/2

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "FormFields": [
    {
      "Id": 3,
      "FormVersionId": 4,
      "FieldId": 5,
      "FormPage": 6,
      "FieldType": 7,
      "MinFieldConfidence": 8,
      "X": 9,
      "Y": 10,
      "Width": 11,
      "Height": 12,
      "TextType": 13,
      "Dropout": 14,
      "Sensitivity": 15,
      "Frame": 16
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 17,
      "FormVersionId": 18,
      "FormPage": 19,
      "X": 20,
      "Y": 21,
      "Width": 22,
      "Height": 23
    },
    {...},
    {...}
  ],
  "Id": 24,
  "FormId": 25,
  "FormVersion": 26,
  "Created": "2019-04-30T07:10:47.8357219-04:00",
  "CreatedBy": 28,
  "MinFormConfidence": 29,
  "Description": "DescriptionStr30",
  "DocTypeId": 31,
  "TaskCreationLevel": 32,
  "StepRootId": 33,
  "AssignedTo": 34,
  "Priority": 35,
  "FileNumberField": "FileNumberFieldStr36",
  "FileNameField": "FileNameFieldStr37"
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FormFields": [
    {
      "Id": 1,
      "FormVersionId": 2,
      "FieldId": 3,
      "FormPage": 4,
      "FieldType": 5,
      "MinFieldConfidence": 6,
      "X": 7,
      "Y": 8,
      "Width": 9,
      "Height": 10,
      "TextType": 11,
      "Dropout": 12,
      "Sensitivity": 13,
      "Frame": 14
    },
    {...},
    {...}
  ],
  "AreasOfInterest": [
    {
      "Id": 15,
      "FormVersionId": 16,
      "FormPage": 17,
      "X": 18,
      "Y": 19,
      "Width": 20,
      "Height": 21
    },
    {...},
    {...}
  ],
  "Id": 22,
  "FormId": 23,
  "FormVersion": 24,
  "Created": "2019-04-30T07:10:47.8357219-04:00",
  "CreatedBy": 26,
  "MinFormConfidence": 27,
  "Description": "DescriptionStr28",
  "DocTypeId": 29,
  "TaskCreationLevel": 30,
  "StepRootId": 31,
  "AssignedTo": 32,
  "Priority": 33,
  "FileNumberField": "FileNumberFieldStr34",
  "FileNameField": "FileNameFieldStr35"
}
```

Overlays

Overlays object allows you to create, update, delete and retrieve list of Documents.

GetDocumentOverlayInfo: API call to get overlay information for a document

Method: GET

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

URL: api/documents/{docId}/overlayInfo

Parameters:

Name	docId
Description	The document ID.

Request: api/documents/1/overlayInfo

Response: 200 OK

```
{
  "EffectiveDate": "2019-04-30T07:10:47.7597219-04:00",
  "ExpirationDate": "2019-04-30T07:10:47.7597219-04:00",
  "FontStyle": "Regular",
  "OverlayId": 3,
  "OverlayImageId": 4,
  "PointSize": 5.1,
  "XOffset": 6.1,
  "YOffset": 7.1
}
```

Error codes:

2	You do not have access to do this operation.
17	An unexpected error occurred. Please try again later.
403	Object is not a document.
10	The item requested was not found.
907	Permission was denied for this operation.
401	You do not have sufficient permissions to perform this action.

GetOverlayImage: API call to get an overlay image

Method: GET

URL: api/overlays/{overlayId}?overlayImageId={overlayImageId}

Parameters:

Name	overlayId
Description	The overlay ID.
Name	overlayImageId
Description	The overlay image ID.

Request: api/overlays/1?overlayImageId=2

Response: 200 OK

```
"Image goes here (stream)"
```

Error codes:

2	You do not have access to do this operation.
17	An unexpected error occurred. Please try again later.
904	Unable to load image. May be due to IOException in reading from file store.
10	The item requested was not found.
907	Permission was denied for this operation.
401	You do not have sufficient permissions to perform this action.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Pages

Pages object allows you to create, update, delete and retrieve list of Pages.

CheckReadPermissions: API call to check if the current users read permissions for a page by ID.

Method: GET
URL: `api/pages/{pageId}/readpermissions`
Parameters:

Name	Description
pageId	The page ID.

Request: `api/pages/1/readpermissions`
Response: 200 OK
Error codes:

907	Permission was denied for this operation.
701	The page was not found.

CreatePage: API call to create a Page.

This call is a POST request with multipart/form-data. Please refer to existing documentation on how to submit multipart/form-data requests in your programming language. An example of what a multipart/form-data request looks like is available at <http://www.asp.net/web-api/overview/advanced/sending-html-form-data,-part-2> .

Note that the first section in the request body should have the lines

Content-Type: application/json
Content-Disposition: form-data; name="PageCreateData"

followed by the JSON object describing the page (properties allowed are docId, batchId, description). The second section in the request body should have the line

Content-Disposition: form-data; name="image0"; filename="name.extension"

followed by the binary bytes of the file. The content-disposition names must match exactly "PageCreateData" and "image0" for the request to be accepted.

A full example is provided below:

Request Headers:
User-Agent: Fiddler
Host: localhost:8093
Content-Length: 270401
Authorization: AccessToken L8QAhF2rPs ...
Content-Type: multipart/form-data; boundary=-----abcdef123456789

Request Body:
-----abcdef123456789
Content-Type: application/json
Content-Disposition: form-data; name="PageCreateData"

{"docId":1234}

-----abcdef123456789
Content-Disposition: form-data; name="image0"; filename="name.extension"

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

[INSERT BINARY DATA OF FILE HERE]
-----abcdef123456789--

The request example provided below in the next section of the documentation shows what the object will be bound to in the system, but does not reflect the request headers and request body that should be constructed in order to make this request.

Method:	POST
URL:	api/pages
Parameters:	
Name	pageCreationData
Description	Not Available

Request: api/pages

```
{
  "DocId": 1,
  "BatchId": 2,
  "Description": "DescriptionStr3",
  "ListImages": [
    {
      "Id": 4,
      "ImageType": 5,
      "ContentType": 6,
      "Version": 7,
      "Size": 8,
      "Extension": "ExtensionStr9",
      "ImageData": [
        64,
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ]
}
```

Response: 201 Created

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "DocumentId": 1,
  "Id": 2,
  "Description": "DescriptionStr3",
  "PageNumber": 4,
  "Version": 5,
  "Deleted": true,
  "DeleteOpType": 7,
  "LatestImages": {
    "Rotation": 8,
    "PreRotation": 9,
    "ImageMetadata": [
      {
        "Id": 10,
        "ImageType": 11,
        "ContentType": 12,
        "Version": 13,
        "Size": 14,
        "Extension": "ExtensionStr15",
        "ImageData": [
          64,
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  },
  "Marks": [
    {
      "BinaryData": [
        64,
        {...},
        {...}
      ],
      "Color": 18,
      "Description": "DescriptionStr19",
      "FileTypeId": 20,
      "MarkId": 21,
      "ProgrammaticName": "ProgrammaticNameStr22"
    },
    {...},
    {...}
  ]
}
```

Error codes:

400	The document structure is invalid.
------------	------------------------------------

GetAllPagesFromDocument: API call to get all pages from a document.

Method:	GET
URL:	api/documents/{docId}/pages
Parameters:	
Name	docId

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Description

The ID of the document.

Request: api/documents/1/pages

Response: 200 OK

```
[
  {
    "DocumentId": 1,
    "Id": 2,
    "Description": "DescriptionStr3",
    "PageNumber": 4,
    "Version": 5,
    "Deleted": true,
    "DeleteOpType": 7,
    "LatestImages": {
      "Rotation": 8,
      "PreRotation": 9,
      "ImageMetadata": [
        {
          "Id": 10,
          "ImageType": 11,
          "ContentType": 12,
          "Version": 13,
          "Size": 14,
          "Extension": "ExtensionStr15",
          "ImageData": [
            64,
            {...},
            {...}
          ]
        },
        {...},
        {...}
      ]
    },
    "Marks": [
      {
        "BinaryData": [
          64,
          {...},
          {...}
        ],
        "Color": 18,
        "Description": "DescriptionStr19",
        "FileTypeId": 20,
        "MarkId": 21,
        "ProgrammaticName": "ProgrammaticNameStr22"
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

Error codes:

400

The document structure is invalid.

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

GetPageById: API call to get page by pageid

Method:	GET				
URL:	api/pages/{pageId}				
Parameters:					
	<table><tr><th>Name</th><td>pageId</td></tr><tr><th>Description</th><td>Not Available</td></tr></table>	Name	pageId	Description	Not Available
Name	pageId				
Description	Not Available				
Request:	api/pages/1				
Response:	200 OK				

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "DocumentId": 1,
  "Id": 2,
  "Description": "DescriptionStr3",
  "PageNumber": 4,
  "Version": 5,
  "Deleted": true,
  "DeleteOpType": 7,
  "LatestImages": {
    "Rotation": 8,
    "PreRotation": 9,
    "ImageMetadata": [
      {
        "Id": 10,
        "ImageType": 11,
        "ContentType": 12,
        "Version": 13,
        "Size": 14,
        "Extension": "ExtensionStr15",
        "ImageData": [
          64,
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  },
  "Marks": [
    {
      "BinaryData": [
        64,
        {...},
        {...}
      ],
      "Color": 18,
      "Description": "DescriptionStr19",
      "FileTypeId": 20,
      "MarkId": 21,
      "ProgrammaticName": "ProgrammaticNameStr22"
    },
    {...},
    {...}
  ]
}
```

Error codes:	400	The document structure is invalid.
	701	The page was not found.

GetPageImageMetadata: API call to get the image metadata information associated with a page.

Method:	GET
URL:	api/pages/{pageId}/imagemetadata
Parameters:	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Name	pageId
Description	The page ID.

Request: api/pages/1/imagemetadata

Response: 200 OK

```
{
  "CurrentVersion": 1,
  "Images": {
    "Rotation": 2,
    "PreRotation": 3,
    "ImageMetadata": [
      {
        "Id": 4,
        "ImageType": 5,
        "ContentType": 6,
        "Version": 7,
        "Size": 8,
        "Extension": "ExtensionStr9",
        "ImageData": [
          64,
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  }
}
```

Error codes:	400	The document structure is invalid.
	701	The page was not found.

LockPage: API call to lock a page. When you lock a page, ImageRight guarantees your lock for 30 seconds, after which you still hold the lock and the right to edit the page but another user can take the lock if it's not refreshed. That means that this call must be made from your application every 10 seconds to maintain the lock.

Method:	GET
URL:	api/pages/{pageId}/lock
Parameters:	
Name	pageId
Description	The ID of the page to lock.

Request: api/pages/1/lock

Response: 200 OK

Error codes:	700	Page versions are mismatched.
---------------------	------------	-------------------------------

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

MovePage: API call to move a page or set of pages to a destination. Returns a dictionary page ID mappings for any changed page IDs. If moving the pages within the same file, page IDs should not change and thus the dictionary will be empty. If moving to a different file, the pages will have new IDs. The key is the old ID and the value is the new ID.

Method: POST
URL: api/pages/move

Parameters:

Name	Description
pageMoveModel	The information on what pages to move. The list of PageIds and target document ID are required. The target Page Id to place the pages at and the boolean After to specify whether it should be placed before or after that page are optional and will default to the beginning of the document if not specified.

Request: api/pages/move

```
{
  "PageIds": [
    1,
    {...},
    {...}
  ],
  "TargetDocumentId": 2,
  "TargetPageId": 3,
  "After": true
}
```

Response: 200 OK

```
[
  {
    "Key": 1,
    "Value": 2
  },
  {...},
  {...}
]
```

Error codes:

400	The document structure is invalid.
701	The page was not found.
401	You do not have sufficient permissions to perform this action.
10	The item requested was not found.

RotatePage: API call to rotate a page.

Method: POST
URL: api/pages/{pageId}/rotate?rotationAngle={rotationAngle}

Parameters:

Name	pageId
Description	The ID of the page to rotate.
Name	rotationAngle
Description	The new orientation of the page. Valid values are 0, 90, 180, and 270.

Request: api/pages/1/rotate?rotationAngle=2

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Error codes:	400	The document structure is invalid.
	902	Object is not locked.

SearchPages: API call to get SearchResults from FullTextSearch service

Method:	POST
URL:	api/pages/searchpages
Parameters:	
Name	searchData
Description	Not Available

Request: api/pages/searchpages

```
{
  "SearchTerm": "SearchTermStr1",
  "FileIds": [
    2,
    {...},
    {...}
  ],
  "FileTypeIds": [
    3,
    {...},
    {...}
  ],
  "FolderTypeIds": [
    4,
    {...},
    {...}
  ],
  "DocumentTypeIds": [
    5,
    {...},
    {...}
  ],
  "DocumentId": 6,
  "FolderId": 7,
  "PageId": 8,
  "FileId": 9
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "ImageRightPages": [
    {
      "TenantId": "TenantIdStr1",
      "PgId": "PgIdStr2",
      "Path": "PathStr3",
      "FileNumber": "FileNumberStr4",
      "PageId": "PageIdStr5",
      "FileId": "FileIdStr6",
      "FileTypeId": "FileTypeIdStr7",
      "Folders": [
        {
          "FolderId": "FolderIdStr8",
          "FolderTypeId": "FolderTypeIdStr9"
        },
        {...},
        {...}
      ],
      "DocumentId": "DocumentIdStr10",
      "DocTypeId": "DocTypeIdStr11",
      "Description": "DescriptionStr12",
      "Text": "TextStr13",
      "IndexDate": "2019-04-30T07:10:47.8657219-04:00",
      "DeletedCounter": 15,
      "PageHighlight": {
        "Text": [
          "Str16",
          {...},
          {...}
        ]
      }
    },
    {...},
    {...}
  ],
  "ImageRightNotes": [
    {
      "TenantId": "TenantIdStr17",
      "NtId": "NtIdStr18",
      "ObjectId": 19,
      "NoteId": "NoteIdStr20",
      "Path": "PathStr21",
      "FileNumber": "FileNumberStr22",
      "FileId": "FileIdStr23",
      "FileTypeId": "FileTypeIdStr24",
      "Folders": [
        {
          "FolderId": "FolderIdStr25",
          "FolderTypeId": "FolderTypeIdStr26"
        },
        {...},
        {...}
      ],
      "Text": "TextStr27",
      "IndexDate": "2019-04-30T07:10:47.8667219-04:00",
      "DeletionCounter": 29,
      "NoteHighlight": {
        "Text": [
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
        "Str30",
        {...},
        {...}
    ],
    },
    "IsFileNote": true
  },
  {...},
  {...}
],
"IsSuccess": true,
"Error": "ErrorStr33"
}
```

Error codes:

2033	An error occured executing full text search request
-------------	---

UnlockPage: API call to unlock a page.

Method:	GET
URL:	api/pages/{pageId}/unlock
Parameters:	
Name	pageId
Description	The ID of the page to lock.
Request:	api/pages/1/unlock
Response:	200 OK
Error codes:	
700	Page versions are mismatched.

UpdatePageContent: API call to update a page.

Method:	POST
URL:	api/pages/{pageId}/content
Parameters:	
Name	pageId
Description	The ID of the page to update.
Name	pageContentModel
Description	This call is a POST request with multipart/form-data. The difference between this call and creating a page is the first section name and properties. One section must be called PageUpdateData and contain a JSON object with the document ID. Do not include the list of images in that section. The other sections must be named "image0", "image1", etc... depending on how many images are associated with that page. Each one of those sections requires the extension and the byte array of image data. ID and Version will be automatically generated by the server.
Request:	api/pages/1/content

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "ListImages": [
    {
      "Id": 2,
      "ImageType": 3,
      "ContentType": 4,
      "Version": 5,
      "Size": 6,
      "Extension": "ExtensionStr7",
      "ImageData": [
        64,
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

1

Error codes:

400	The document structure is invalid.
902	Object is not locked.

UpdatePageProperties: API call to update page properties.

Method: POST
URL: api/pages/{pageId}/properties

Parameters:

Name	pageId
Description	The ID of the page to update.
Name	pagePropertiesModel
Description	The model for properties to be updated on the page.

Request: api/pages/1/properties

{
 "Description": "DescriptionStr2"
}

Response: 200 OK

1

Error codes:

701	The page was not found.
401	You do not have sufficient permissions to perform this action.
400	The document structure is invalid.
704	The requested page does not belong to the document.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

PagesV2

Pages object allows you to create, update, delete and retrieve list of Pages.

CopyPageV2: API call to copy a page or set of pages to a destination. Returns a dictionary page ID mappings for any changed page IDs. The key is the old ID and the value is the new ID.

Method: POST

URL: api/v2/pages/copy

Parameters:

Name

model

Description

The information on what pages to copy. The list of PageIds, target document ID and CopyOption are required. The target Page Id to place the pages at and the boolean After to specify whether it should be placed before or after that page are optional and will default to the beginning of the document if not specified.

Request: api/v2/pages/copy

```
{
  "PageIds": [
    1,
    {...},
    {...}
  ],
  "TargetId": 2,
  "TargetPageId": 3,
  "After": true,
  "CopyOption": "Logical",
  "PathToCreate": [
    {
      "TypeId": 5,
      "Description": "DescriptionStr6"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FailedObjects": [
    {
      "Key": 1,
      "Value": {
        "Message": "MessageStr2",
        "Code": 3
      }
    },
    {...},
    {...}
  ],
  "PageIdMap": [
    {
      "Key": 4,
      "Value": 5
    },
    {...},
    {...}
  ],
  "CreatedPath": [
    {
      "Name": "NameStr6",
      "TypeId": 7,
      "TypeDescription": "TypeDescriptionStr8",
      "EffectivePermissions": 9,
      "ObjectClass": "RootContainer",
      "DateCreated": "2019-04-30T07:10:47.8367219-04:00",
      "DateLastModified": "2019-04-30T07:10:47.8367219-04:00",
      "Id": 12,
      "Description": "DescriptionStr13"
    },
    {...},
    {...}
  ]
}
```

Error codes:		
27		Target for this operation is invalid.
28		Target for this operation was not found.
402		Cannot move documents from different source files.
407		Document type is not valid for the template.

CreatePageV2: API call to create a Page.
If TargetPageId is null and Before is false(true), the new page will be inserted as the last(first) page of the document. If TargetPageId is specified, the new page will be inserted before/after this page.

This call is a POST request with multipart/form-data. Please refer to existing documentation on how to submit multipart/form-data requests in your programming language. An example of what a multipart/form-data request looks like is available at <http://www.asp.net/web-api/overview/advanced/sending-html-form-data,-part-2> .

Note that the first section in the request body should have the lines

Content-Type: application/json
Content-Disposition: form-data; name="PageCreateData"

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

followed by the JSON object describing the page (properties allowed are docId, batchId, description). The second section in the request body should have the line

Content-Disposition: form-data; name="Image"; filename="name.extension"

followed by the binary bytes of the file. The content-disposition names must match exactly "PageCreateData" and "Image" for the request to be accepted. You may also optionally specify a third section with the line

Content-Disposition: form-data; name="Annotation"; filename="name.extension"

followed by the binary data bytes of the annotation file.

A full example is provided below:

Request Headers:
User-Agent: Fiddler
Host: localhost:8093
Content-Length: 270401
Authorization: AccessToken L8QAhF2rPs ...
Content-Type: multipart/form-data; boundary=-----abcdef123456789

Request Body:
-----abcdef123456789
Content-Type: application/json
Content-Disposition: form-data; name="PageCreateData"

{ "docId": 1234 }
-----abcdef123456789
Content-Disposition: form-data; name="Image"; filename="name.extension"

[INSERT BINARY DATA OF FILE HERE]
-----abcdef123456789--

The request example provided below in the next section of the documentation shows what the object will be bound to in the system, but does not reflect the request headers and request body that should be constructed in order to make this request.

Method:

POST

URL:

api/v2/pages?targetPageId={targetPageId}&before={before}

Parameters:

Name

pageCreationData

Description

Not Available

Name

targetPageId

Description

Not Available **Optional**, default value is **null (empty)**.

Name

before

Description

Not Available **Optional**, default value is **False**.

Request:

api/v2/pages?targetPageId=8&before=true

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "DocId": 1,
  "BatchId": 2,
  "Description": "DescriptionStr3",
  "Image": {
    "Extension": "ExtensionStr4",
    "Data": [
      64,
      {...},
      {...}
    ],
    "IsRedacted": false
  },
  "Annotation": {
    "Extension": "ExtensionStr6",
    "Data": [
      64,
      {...},
      {...}
    ],
    "IsRedacted": false
  }
}
```

Response: 201 Created

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "DocumentId": 1,
  "Id": 2,
  "Description": "DescriptionStr3",
  "PageNumber": 4,
  "Version": 5,
  "Deleted": true,
  "DeleteOpType": 7,
  "LatestImages": {
    "Rotation": 8,
    "PreRotation": 9,
    "ImageMetadata": [
      {
        "Id": 10,
        "ImageType": 11,
        "ContentType": 12,
        "Version": 13,
        "Size": 14,
        "Extension": "ExtensionStr15",
        "ImageData": [
          64,
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ]
  },
  "Marks": [
    {
      "BinaryData": [
        64,
        {...},
        {...}
      ],
      "Color": 18,
      "Description": "DescriptionStr19",
      "FileTypeId": 20,
      "MarkId": 21,
      "ProgrammaticName": "ProgrammaticNameStr22"
    },
    {...},
    {...}
  ]
}
```

Error codes:		
	400	The document structure is invalid.
	411	The document was not found.
	701	The page was not found.
	914	The image is empty.

MergeToDocument: API call to merge pages and pages from documents to a destination. Returns a dictionary page ID mappings only for passed page IDs. The key is the old ID and the value is the new ID.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Method:

POST

URL:

api/v2/pages/merge

Parameters:

Name	Description
model	The information on what pages to process. The list of PageIds and DocumentIds, target document ID are required. The target Page Id to place the pages at and the boolean After to specify whether it should be placed before or after that page are optional and will default to the beginning of the document if not specified. The initial items can be cut depends on the boolean Cut parameter.

Request:

api/v2/pages/merge

```
{
  "DocumentIds": [
    1,
    {...},
    {...}
  ],
  "Cut": true,
  "PageIds": [
    3,
    {...},
    {...}
  ],
  "TargetId": 4,
  "TargetPageId": 5,
  "After": true,
  "CopyOption": "Logical",
  "PathToCreate": [
    {
      "TypeId": 7,
      "Description": "DescriptionStr8"
    },
    {...},
    {...}
  ]
}
```

Response:

200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FailedPages": [
    {
      "Key": 1,
      "Value": {
        "Message": "MessageStr2",
        "Code": 3
      }
    },
    {...},
    {...}
  ],
  "FailedDocuments": [
    {
      "Key": 4,
      "Value": {
        "Message": "MessageStr5",
        "Code": 6
      }
    },
    {...},
    {...}
  ],
  "PageIdMap": [
    {
      "Key": 7,
      "Value": 8
    },
    {...},
    {...}
  ],
  "CreatedPath": [
    {
      "Name": "NameStr9",
      "TypeId": 10,
      "TypeDescription": "TypeDescriptionStr11",
      "EffectivePermissions": 12,
      "ObjectClass": "RootContainer",
      "DateCreated": "2019-04-30T07:10:47.8407219-04:00",
      "DateLastModified": "2019-04-30T07:10:47.8407219-04:00",
      "Id": 15,
      "Description": "DescriptionStr16"
    },
    {...},
    {...}
  ]
}
```

Error codes:	
27	Target for this operation is invalid.
28	Target for this operation was not found.
402	Cannot move documents from different source files.
407	Document type is not valid for the template.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

MovePageV2: API call to move a page or set of pages to a destination. Returns a dictionary page ID mappings for any changed page IDs. If moving the pages within the same file, page IDs should not change and thus the dictionary will be empty. If moving to a different file, the pages will have new IDs. The key is the old ID and the value is the new ID.

Method: POST
URL: api/v2/pages/move
Parameters:

Name	model
Description	The information on what pages to move. The list of PageIds and target document ID are required. The target Page Id to place the pages at and the boolean After to specify whether it should be placed before or after that page are optional and will default to the beginning of the document if not specified.

Request: api/v2/pages/move

```
{
  "PageIds": [
    1,
    {...},
    {...}
  ],
  "TargetId": 2,
  "TargetPageId": 3,
  "After": true,
  "CopyOption": "Logical",
  "PathToCreate": [
    {
      "TypeId": 5,
      "Description": "DescriptionStr6"
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "FailedObjects": [
    {
      "Key": 1,
      "Value": {
        "Message": "MessageStr2",
        "Code": 3
      }
    },
    {...},
    {...}
  ],
  "PageIdMap": [
    {
      "Key": 4,
      "Value": 5
    },
    {...},
    {...}
  ],
  "CreatedPath": [
    {
      "Name": "NameStr6",
      "TypeId": 7,
      "TypeDescription": "TypeDescriptionStr8",
      "EffectivePermissions": 9,
      "ObjectClass": "RootContainer",
      "DateCreated": "2019-04-30T07:10:47.8407219-04:00",
      "DateLastModified": "2019-04-30T07:10:47.8407219-04:00",
      "Id": 12,
      "Description": "DescriptionStr13"
    },
    {...},
    {...}
  ]
}
```

Error codes:	27	Target for this operation is invalid.
	28	Target for this operation was not found.
	402	Cannot move documents from different source files.
	407	Document type is not valid for the template.

UpdatePageContentV2: API call to update a page.

Method:	POST
URL:	api/v2/pages/{pageId}/content
Parameters:	
Name	pageId
Description	The ID of the page to update.
Name	pageContentModel
Description	This call is a POST request with multipart/form-data. The difference between this call and creating a page is the first section name and properties. One section must be called PageUpdateData and contain a JSON object with the document

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

ID. Do not include the list of images in that section. The other sections must be named "image" or "annotation" and contain the data.

Request: api/v2/pages/1/content

```
{
  "Image": {
    "Extension": "ExtensionStr2",
    "Data": [
      64,
      {...},
      {...}
    ],
    "IsRedacted": false
  },
  "Annotation": {
    "Extension": "ExtensionStr4",
    "Data": [
      64,
      {...},
      {...}
    ],
    "IsRedacted": false
  }
}
```

Response: 200 OK

```
1
```

Error codes:	
400	The document structure is invalid.
902	Object is not locked.

RedactionRuleSets

RedactionRuleSetsController

AddRuleSet: Creates a new Redaction Rule Set.

Method:	POST
URL:	api/redactionrulesets
Parameters:	
Name	newRedactionRuleSet
Description	The RedactionRuleSet model to create
Request:	api/redactionrulesets

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Name": "NameStr1",
  "Rules": [
    {
      "Id": 2,
      "Type": "Numeric",
      "Value": "ValueStr3",
      "RuleSetId": 4
    },
    {...},
    {...}
  ]
}
```

Response: 201 Created

```
{
  "Id": 1,
  "Name": "NameStr2",
  "Created": "2019-04-30T07:10:47.7757219-04:00",
  "CreatedBy": 4,
  "LockedBy": 5,
  "LockExpiration": "2019-04-30T07:10:47.7757219-04:00",
  "LastModified": "2019-04-30T07:10:47.7757219-04:00",
  "LastModifiedBy": 8,
  "Deleted": "2019-04-30T07:10:47.7757219-04:00",
  "DeletedBy": 10,
  "Rules": [
    {
      "Id": 11,
      "Type": "Numeric",
      "Value": "ValueStr12",
      "RuleSetId": 13
    },
    {...},
    {...}
  ]
}
```

GetAllRuleSets: Retuns an array of the Redaction Rules.

Method: GET
URL: api/redactionrulesets
Request: api/redactionrulesets
Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
[
  {
    "Id": 1,
    "Name": "NameStr2",
    "Created": "2019-04-30T07:10:47.7767219-04:00",
    "CreatedBy": 4,
    "LockedBy": 5,
    "LockExpiration": "2019-04-30T07:10:47.7767219-04:00",
    "LastModified": "2019-04-30T07:10:47.7767219-04:00",
    "LastModifiedBy": 8,
    "Deleted": "2019-04-30T07:10:47.7767219-04:00",
    "DeletedBy": 10,
    "Rules": [
      {
        "Id": 11,
        "Type": "Numeric",
        "Value": "ValueStr12",
        "RuleSetId": 13
      },
      {...},
      {...}
    ]
  },
  {...},
  {...}
]
```

GetRuleSet: Returns the Redaction Rule Set.

Method:	GET		
URL:	api/redactionrulesets/{ruleSetId}		
Parameters:	Name	ruleSetId	
	Description	Not Available	
Request:	api/redactionrulesets/1		
Response:	200 OK		

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Id": 1,
  "Name": "NameStr2",
  "Created": "2019-04-30T07:10:47.7767219-04:00",
  "CreatedBy": 4,
  "LockedBy": 5,
  "LockExpiration": "2019-04-30T07:10:47.7767219-04:00",
  "LastModified": "2019-04-30T07:10:47.7767219-04:00",
  "LastModifiedBy": 8,
  "Deleted": "2019-04-30T07:10:47.7767219-04:00",
  "DeletedBy": 10,
  "Rules": [
    {
      "Id": 11,
      "Type": "Numeric",
      "Value": "ValueStr12",
      "RuleSetId": 13
    },
    {...},
    {...}
  ]
}
```

LockRuleSet: Locks the Rule Set.

Method:	POST				
URL:	api/redactionrulesets/lock				
Parameters:					
	<table><tr><td>Name</td><td>ruleSetId</td></tr><tr><td>Description</td><td>The ID of the RuleSet.</td></tr></table>	Name	ruleSetId	Description	The ID of the RuleSet.
Name	ruleSetId				
Description	The ID of the RuleSet.				
Request:	api/redactionrulesets/lock				
	1				
Response:	200 OK				

RemoveRuleSet: Deletes the Rule Set.

Method:	DELETE				
URL:	api/redactionrulesets/{ruleSetId}				
Parameters:					
	<table><tr><td>Name</td><td>ruleSetId</td></tr><tr><td>Description</td><td>The ID of the RuleSet.</td></tr></table>	Name	ruleSetId	Description	The ID of the RuleSet.
Name	ruleSetId				
Description	The ID of the RuleSet.				
Request:	api/redactionrulesets/1				
Response:	200 OK				

UnlockRuleSet: Unlocks the Rule Set.

Method:	POST
----------------	------

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

URL: api/redactionrulesets/unlock

Parameters:

Name	
ruleSetId	
Description	The ID of the RuleSet.

Request: api/redactionrulesets/unlock

1

Response: 200 OK

UpdateRuleSet: Updates the Rule Set.

Method: POST

URL: api/redactionrulesets/{ruleSetId}

Parameters:

Name	
ruleSetId	
Description	The ID of the RuleSet.

Name	
newRuleSet	
Description	The RedactionRuleSet model.

Request: api/redactionrulesets/1

```
{
  "Id": 2,
  "Name": "NameStr3",
  "Created": "2019-04-30T07:10:47.7767219-04:00",
  "CreatedBy": 5,
  "LockedBy": 6,
  "LockExpiration": "2019-04-30T07:10:47.7767219-04:00",
  "LastModified": "2019-04-30T07:10:47.7767219-04:00",
  "LastModifiedBy": 9,
  "Deleted": "2019-04-30T07:10:47.7767219-04:00",
  "DeletedBy": 11,
  "Rules": [
    {
      "Id": 12,
      "Type": "Numeric",
      "Value": "ValueStr13",
      "RuleSetId": 14
    },
    {...},
    {...}
  ]
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Id": 1,
  "Name": "NameStr2",
  "Created": "2019-04-30T07:10:47.7767219-04:00",
  "CreatedBy": 4,
  "LockedBy": 5,
  "LockExpiration": "2019-04-30T07:10:47.7767219-04:00",
  "LastModified": "2019-04-30T07:10:47.7767219-04:00",
  "LastModifiedBy": 8,
  "Deleted": "2019-04-30T07:10:47.7767219-04:00",
  "DeletedBy": 10,
  "Rules": [
    {
      "Id": 11,
      "Type": "Numeric",
      "Value": "ValueStr12",
      "RuleSetId": 13
    },
    {...},
    {...}
  ]
}
```

ReportParameters

GetGroups: Gets the groups with batches.

Method: GET
URL: api/reports/parameters/groups
Request: api/reports/parameters/groups
Response: 200 OK

GetProgramSources: Gets the program sources.

Method: GET
URL: api/reports/parameters/programs
Request: api/reports/parameters/programs
Response: 200 OK

GetRoles: Gets the roles with batches.

Method: GET
URL: api/reports/parameters/roles
Request: api/reports/parameters/roles
Response: 200 OK

GetUsers: Gets the users with batches.

Method: GET

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

URL:

api/reports/parameters/users

Request:

api/reports/parameters/users

Response:

200 OK

Reports

Reports controller allows you to perform CRUD operations on reports.

CreateReport: Creates the report.

Method:

POST

URL:

api/reports

Parameters:

Name	
Description	The model.

Request:

api/reports

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "ReportType": "ProgramSource",
  "ReportStatus": "Pending",
  "DateStart": "2019-04-30T07:10:47.8257219-04:00",
  "DateEnd": "2019-04-30T07:10:47.8257219-04:00",
  "DateSubmitted": "2019-04-30T07:10:47.8257219-04:00",
  "Accountid": 4,
  "Reportid": 5,
  "Templateid": 6,
  "ReportName": "ReportNameStr7",
  "Programs": [
    {
      "Id": 8,
      "Name": "NameStr9"
    },
    {...},
    {...}
  ],
  "Users": [
    {
      "Id": 10,
      "Name": "NameStr11"
    },
    {...},
    {...}
  ],
  "Drawers": [
    {
      "Id": 12,
      "Name": "NameStr13"
    },
    {...},
    {...}
  ],
  "FileTypes": [
    {
      "Id": 14,
      "Name": "NameStr15"
    },
    {...},
    {...}
  ],
  "FolderTypes": [
    {
      "Id": 16,
      "Name": "NameStr17"
    },
    {...},
    {...}
  ],
  "DocTypes": [
    {
      "Id": 18,
      "Description": "DescriptionStr19"
    },
    {...},
    {...}
  ],
}
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
"FunctionalityRightNames": [
  "Str20",
  {...},
  {...}
],
"ReportSubtypes": [
  "NoSubtype",
  {...},
  {...}
],
"ShowAdvanced": true,
"Workflows": [
  {
    "Steps": [
      {
        "Id": 22,
        "Name": "NameStr23"
      },
      {...},
      {...}
    ],
    "Id": 24,
    "Name": "NameStr25"
  },
  {...},
  {...}
],
"DateOption": "DateOptionStr26",
"IncludeDeactivatedUsers": true,
"Actions": [
  {
    "Category": "CategoryStr28",
    "Id": 29,
    "Name": "NameStr30"
  },
  {...},
  {...}
],
"Files": [
  {
    "Id": 31,
    "Name": "NameStr32"
  },
  {...},
  {...}
],
"FilesOption": "FilesOptionStr33"
}
```

Response: 200 OK

DeleteAllReportsForUser: Deletes all reports created by the specified user.

Method: DELETE
URL: api/reports/delete-all
Request: api/reports/delete-all

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 200 OK

DeleteReport: Deletes a specified report.

Method: DELETE
URL: api/reports/{reportid}
Parameters:
Name reportid
Description Not Available
Request: api/reports/1
Response: 200 OK

GetMetaDataForReports: Gets report meta data for an array of reports

Method: GET
URL: api/reports/get-metadata?reportsid[0]={reportsid[0]}&reportsid[1]={reportsid[1]}
Parameters:
Name reportsid
Description The array accounts Ids.
Request: api/reports/get-metadata?reportsid=system.collections.generic.list`1[System.Int32]
Response: 200 OK

GetReportChunk: Gets JSON report chunk as compressed binary data.

Method: GET
URL: api/reports/{reportid}/chunks/{chunkNumber}
Parameters:
Name reportid
Description The id of the report to be downloaded.
Name chunkNumber
Description The chunk number.
Request: api/reports/1/chunks/2
Response: 200 OK

GetReportChunkCount: Gets the number of chunks a specific report is saved as in the db.

Method: GET
URL: api/reports/{reportid}/chunk-count
Parameters:
Name reportid
Description The report id
Request: api/reports/1/chunk-count

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

Response: 200 OK

GetReportMetaDataForUser: Gets all reports sorted from most recent to least recent.

Method: GET
URL: api/reports
Request: api/reports
Response: 200 OK

RerunReport: Reruns a specified report.

Method: POST
URL: api/reports/rerun/{reportid}
Parameters:

Name	reportid
Description	Not Available

Request: api/reports/rerun/1
Response: 200 OK

ReturnReportParams: Returns Report params

Method: GET
URL: api/reports/params/{reportid}
Parameters:

Name	reportid
Description	Not Available

Request: api/reports/params/1
Response: 200 OK

ReportTemplates

CreateTemplate: Creates the template.This function checks if the name is valid first then checks against system requirements before it allows the user to name as the name must be unique.

Method: POST
URL: api/reports/templates
Parameters:

Name	model
Description	The model.

Request: api/reports/templates

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "TemplateId": 1,
  "AccountId": 2,
  "ReportType": "ProgramSource",
  "TemplateName": "TemplateNameStr3",
  "Description": "DescriptionStr4",
  "Parameters": {
    "ReportType": "ProgramSource",
    "ReportStatus": "Pending",
    "DateStart": "2019-04-30T07:10:47.8107219-04:00",
    "DateEnd": "2019-04-30T07:10:47.8107219-04:00",
    "DateSubmitted": "2019-04-30T07:10:47.8107219-04:00",
    "Accountid": 8,
    "Reportid": 9,
    "Templateid": 10,
    "ReportName": "ReportNameStr11",
    "Programs": [
      {
        "Id": 12,
        "Name": "NameStr13"
      },
      {...},
      {...}
    ],
    "Users": [
      {
        "Id": 14,
        "Name": "NameStr15"
      },
      {...},
      {...}
    ],
    "Drawers": [
      {
        "Id": 16,
        "Name": "NameStr17"
      },
      {...},
      {...}
    ],
    "FileTypes": [
      {
        "Id": 18,
        "Name": "NameStr19"
      },
      {...},
      {...}
    ],
    "FolderTypes": [
      {
        "Id": 20,
        "Name": "NameStr21"
      },
      {...},
      {...}
    ],
    "DocTypes": [
      {
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
      "Id": 22,
      "Description": "DescriptionStr23"
    },
    {...},
    {...}
  ],
  "FunctionalityRightNames": [
    "Str24",
    {...},
    {...}
  ],
  "ReportSubtypes": [
    "NoSubtype",
    {...},
    {...}
  ],
  "ShowAdvanced": true,
  "Workflows": [
    {
      "Steps": [
        {
          "Id": 26,
          "Name": "NameStr27"
        },
        {...},
        {...}
      ],
      "Id": 28,
      "Name": "NameStr29"
    },
    {...},
    {...}
  ],
  "DateOption": "DateOptionStr30",
  "IncludeDeactivatedUsers": true,
  "Actions": [
    {
      "Category": "CategoryStr32",
      "Id": 33,
      "Name": "NameStr34"
    },
    {...},
    {...}
  ],
  "Files": [
    {
      "Id": 35,
      "Name": "NameStr36"
    },
    {...},
    {...}
  ],
  "FilesOption": "FilesOptionStr37"
},
"JsonParameters": "JsonParametersStr38",
"LastRun": "2019-04-30T07:10:47.8137219-04:00",
"Created": "2019-04-30T07:10:47.8137219-04:00"
}
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 200 OK

DeleteAllTemplates: Deletes the templates.

Method: DELETE
URL: api/reports/templates
Request: api/reports/templates
Response: 200 OK

DeleteTemplate: Deletes the template.

Method: DELETE
URL: api/reports/templates/{*templateId*}

Parameters:	Name	templateId
	Description	The template identifier.

Request: api/reports/templates/1
Response: 200 OK

GetTemplateParameters: Gets the template parameters.

Method: GET
URL: api/reports/templates/parameters/{*templateId*}

Parameters:	Name	templateId
	Description	The template identifier.

Request: api/reports/templates/parameters/1
Response: 200 OK

GetTemplates: Gets the templates.

Method: GET
URL: api/reports/templates
Request: api/reports/templates
Response: 200 OK

RenameTemplate: Renames the template. This function checks if the name is valid first then checks against system requirements before it allows the user to rename as the name must be unique.

Method: PUT
URL: api/reports/templates/rename

Parameters:	Name	model
	Description	The model.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Request: api/reports/templates/rename

```
{
  "TemplateId": 1,
  "AccountId": 2,
  "ReportType": "ProgramSource",
  "ReportTypeName": "ReportTypeNameStr3",
  "TemplateName": "TemplateNameStr4",
  "Description": "DescriptionStr5",
  "LastRun": "2019-04-30T07:10:47.8197219-04:00",
  "Created": "2019-04-30T07:10:47.8197219-04:00"
}
```

Response: 200 OK

RunTemplateReport: Runs the template report.

Method: GET

URL: api/reports/templates/{*templateId*}

Parameters:

Name	templateId
Description	The template identifier.

Request: api/reports/templates/1

Response: 200 OK

SaveTemplate: Saves the template.

Method: PUT

URL: api/reports/templates

Parameters:

Name	model
Description	The model.

Request: api/reports/templates

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "TemplateId": 1,
  "AccountId": 2,
  "ReportType": "ProgramSource",
  "TemplateName": "TemplateNameStr3",
  "Description": "DescriptionStr4",
  "Parameters": {
    "ReportType": "ProgramSource",
    "ReportStatus": "Pending",
    "DateStart": "2019-04-30T07:10:47.8197219-04:00",
    "DateEnd": "2019-04-30T07:10:47.8197219-04:00",
    "DateSubmitted": "2019-04-30T07:10:47.8197219-04:00",
    "Accountid": 8,
    "Reportid": 9,
    "Templateid": 10,
    "ReportName": "ReportNameStr11",
    "Programs": [
      {
        "Id": 12,
        "Name": "NameStr13"
      },
      {...},
      {...}
    ],
    "Users": [
      {
        "Id": 14,
        "Name": "NameStr15"
      },
      {...},
      {...}
    ],
    "Drawers": [
      {
        "Id": 16,
        "Name": "NameStr17"
      },
      {...},
      {...}
    ],
    "FileTypes": [
      {
        "Id": 18,
        "Name": "NameStr19"
      },
      {...},
      {...}
    ],
    "FolderTypes": [
      {
        "Id": 20,
        "Name": "NameStr21"
      },
      {...},
      {...}
    ],
    "DocTypes": [
      {
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
      "Id": 22,
      "Description": "DescriptionStr23"
    },
    {...},
    {...}
  ],
  "FunctionalityRightNames": [
    "Str24",
    {...},
    {...}
  ],
  "ReportSubtypes": [
    "NoSubtype",
    {...},
    {...}
  ],
  "ShowAdvanced": true,
  "Workflows": [
    {
      "Steps": [
        {
          "Id": 26,
          "Name": "NameStr27"
        },
        {...},
        {...}
      ],
      "Id": 28,
      "Name": "NameStr29"
    },
    {...},
    {...}
  ],
  "DateOption": "DateOptionStr30",
  "IncludeDeactivatedUsers": true,
  "Actions": [
    {
      "Category": "CategoryStr32",
      "Id": 33,
      "Name": "NameStr34"
    },
    {...},
    {...}
  ],
  "Files": [
    {
      "Id": 35,
      "Name": "NameStr36"
    },
    {...},
    {...}
  ],
  "FilesOption": "FilesOptionStr37"
},
"JsonParameters": "JsonParametersStr38",
"LastRun": "2019-04-30T07:10:47.8197219-04:00",
"Created": "2019-04-30T07:10:47.8197219-04:00"
}
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 200 OK

ServiceDiscovery

ServiceDiscovery object allows you to determine if a service is configured.

IsEventStoreConfigured: Gets the current configuration

Method: GET
URL: api/servicediscovery
Request: api/servicediscovery
Response: 200 OK

```
[
  {
    "IsEventStoreConfigured": true
  },
  {...},
  {...}
]
```

Sla

An SLA object allows you to create, update, delete and retrieve SLAs.

CreateSla: Creates an SLA.

Method: POST
URL: api/slas
Parameters:

Name	model
Description	The SLA model to be created.

Request: api/slas

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

```
{
  "Name": "NameStr1",
  "Algorithm": "TaskCreationTime",
  "TimeUnit": "Days",
  "AgeModelSets": [
    {
      "From": 2,
      "To": 3,
      "Level": 4
    },
    {...},
    {...}
  ],
  "Filter": {
    "AssignedTo": [
      5,
      {...},
      {...}
    ],
    "DrawersAndFileTypes": [
      {
        "Key": 6,
        "Value": [
          7
        ]
      },
      {...},
      {...}
    ],
    "WorkflowsAndSteps": [
      {
        "Key": 8,
        "Value": [
          9
        ]
      },
      {...},
      {...}
    ]
  ]
}
```

Response: 201 Created

1

DeleteSLA: Deletes an SLA with lastModified header check.

Method:	DELETE
URL:	api/slas/{id}
Parameters:	
Name	id
Description	The id of the SLA.
Request:	api/slas/1

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Response: 200 OK

GetAllSlas: Retrieves all SLAs with max value in lastModified header.

Method: GET

URL: api/slas

Request: api/slas

Response: 200 OK

```
[
  {
    "Id": 4,
    "Name": "NameStr5",
    "Algorithm": "TaskCreationTime",
    "TimeUnit": "Days",
    "AgeModelSets": [
      {
        "From": 1,
        "To": 2,
        "Level": 3
      },
      {...},
      {...}
    ],
    "Filter": {
      "AssignedTo": [
        6,
        {...},
        {...}
      ],
      "DrawersAndFileTypes": [
        {
          "Key": 7,
          "Value": [
            8
          ]
        },
        {...},
        {...}
      ],
      "WorkflowsAndSteps": [
        {
          "Key": 9,
          "Value": [
            10
          ]
        },
        {...},
        {...}
      ]
    }
  },
  {...},
  {...}
]
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetSla: Retrieves an SLA with lastModified header.

Method: GET
URL: api/slas/{id}

Parameters:

Name	id
Description	The id of the SLA.

Request: api/slas/1

Response: 200 OK

```
{
  "Name": "NameStr1",
  "Algorithm": "TaskCreationTime",
  "TimeUnit": "Days",
  "AgeModelSets": [
    {
      "From": 2,
      "To": 3,
      "Level": 4
    },
    {...},
    {...}
  ],
  "Filter": {
    "AssignedTo": [
      5,
      {...},
      {...}
    ],
    "DrawersAndFileTypes": [
      {
        "Key": 6,
        "Value": [
          7
        ]
      },
      {...},
      {...}
    ],
    "WorkflowsAndSteps": [
      {
        "Key": 8,
        "Value": [
          9
        ]
      },
      {...},
      {...}
    ]
  }
}
```

UpdateSla: Updates the SLA with lastModified header check.

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Method: PUT
URL: api/slas/{id}
Parameters:

Name	id
Description	The id of the SLA to be updated.

Name	model
Description	The SLA model to be updated.

Request: api/slas/1

```
{
  "Name": "NameStr2",
  "Algorithm": "TaskCreationTime",
  "TimeUnit": "Days",
  "AgeModelSets": [
    {
      "From": 3,
      "To": 4,
      "Level": 5
    },
    {...},
    {...}
  ],
  "Filter": {
    "AssignedTo": [
      6,
      {...},
      {...}
    ],
    "DrawersAndFileTypes": [
      {
        "Key": 7,
        "Value": [
          8
        ]
      },
      {...},
      {...}
    ],
    "WorkflowsAndSteps": [
      {
        "Key": 9,
        "Value": [
          10
        ]
      },
      {...},
      {...}
    ]
  }
}
```

Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

TaskActions object provides lock/unlock, release, route, kill task functionality.

KillTask: Kills a task.

Method: POST
URL: `api/tasks/{taskId}/cancel`
Parameters:

Name	
Description	The task id.

Request: `api/tasks/1/cancel`

Response: 200 OK

Error codes:

200	The task was not found.
202	The task is not locked.
201	Task locked by another user.
205	The task is already completed.

LockTask: Locks a task on the specified workflow step, if the step specified. Returns fresh task info if flag specified.

Method: POST
URL: `api/tasks/{taskId}/lock?stepId={stepId}&returnTask={returnTask}`
Parameters:

Name	taskId
Description	The task id.
Name	stepId
Description	Workflow step id. Optional , default value is null (empty) .
Name	returnTask
Description	If true, returns fresh task info. Optional , default value is True .

Request: `api/tasks/1/lock?stepId=2&returnTask=true`

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Id": 1,
  "Description": "DescriptionStr2",
  "Priority": 3,
  "FileId": 4,
  "FileName": "FileNameStr5",
  "FileNumber": "FileNumberStr6",
  "FlowId": 7,
  "FlowName": "FlowNameStr8",
  "StepId": 9,
  "StepName": "StepNameStr10",
  "SenderStep": 11,
  "SuperTaskId": 12,
  "RendezvousStepId": 13,
  "SubTaskIsRequired": true,
  "NoteId": 15,
  "AssignedTo": {
    "Id": 16,
    "ExternalId": "ExternalIdStr17",
    "Name": "NameStr18",
    "FriendlyName": "FriendlyNameStr19",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr21"
  },
  "Sender": {
    "Id": 22,
    "ExternalId": "ExternalIdStr23",
    "Name": "NameStr24",
    "FriendlyName": "FriendlyNameStr25",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr27"
  },
  "LockedBy": {
    "Id": 28,
    "ExternalId": "ExternalIdStr29",
    "Name": "NameStr30",
    "FriendlyName": "FriendlyNameStr31",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr33"
  },
  "AvailableDate": "2019-04-30T07:10:47.8007219-04:00",
  "StartDate": "2019-04-30T07:10:47.8007219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.8007219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.8007219-04:00",
  "DeadLine": "2019-04-30T07:10:47.8007219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.8007219-04:00",
  "Debug": true,
  "StackLevel": 41,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr42",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 43,
  "ObjectId": 44,
  "DocumentAttributes": [
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "DisplayName": "DisplayNameStr45",
  "Name": "NameStr46",
  "Value": {...},
  "AttributeType": "atInt"
},
{...},
{...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr48",
    "Name": "NameStr49",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr51",
    "Name": "NameStr52",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 54,
"FileTypeId": 55,
"DrawerId": 56,
"ObjectTypeId": 57,
"ObjectClass": "RootContainer"
}
```

Error codes:		
200		The task was not found.
201		Task locked by another user.
205		The task is already completed.

RefreshTaskLock: Refreshes the lock of a task. The task should be locked by current user.

Method:	POST	
URL:	api/tasks/{taskId}/refresh	
Parameters:	Name	taskId
	Description	The task id.
Request:	api/tasks/1/refresh	
Response:	200 OK	
Error codes:		
	200	The task was not found.
	201	Task locked by another user.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

	202	The task is not locked.
	205	The task is already completed.
ReleaseTask: Releases a task from a single/multi link step.		
Method:	POST	
URL:	api/tasks/{taskId}/release?destStepId={destStepId}&commit={commit}	
Parameters:		
	Name	taskId
	Description	The task id.
	Name	destStepId
	Description	The destination step id. Optional , default value is null (empty) .
	Name	commit
	Description	Commit release action if true. Optional , default value is True .
Request:	api/tasks/1/release?destStepId=2&commit=true	
Response:	200 OK	
Error codes:		
	200	The task was not found.
	202	The task is not locked.
	201	Task locked by another user.
	204	The task is in an error state.
	212	A mandatory attribute on a task was not set.
	320	The link to destination step was not found.
	220	Supertask has required subtasks that have not been completed.
	322	You try to release not split step. Use method for release single/multi link steps.
	324	Step code check failed. The task could not be released.
ReleaseTaskByAnchor: Releases a task by step link anchor. If userId specified, then attempts to assign the task to the user.		
Method:	POST	
URL:	api/tasks/{taskId}/release?anchor={anchor}&userId={userId}&commit={commit}	
Parameters:		
	Name	taskId
	Description	The task id.
	Name	anchor
	Description	The step link anchor.
	Name	userId
	Description	The internal user id to assign. Optional , default value is null (empty) .
	Name	commit
	Description	Commit release action if true. Optional , default value is True .
Request:	api/tasks/1/release?anchor=2&userId=3&commit=true	

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response:	200 OK
Error codes:	
200	The task was not found.
202	The task is not locked.
201	Task locked by another user.
204	The task is in an error state.
212	A mandatory attribute on a task was not set.
320	The link to destination step was not found.
220	Supertask has required subtasks that have not been completed.
322	You try to release not split step. Use method for release single/multi link steps.
324	Step code check failed. The task could not be released.
221	User id parameter is invalid.

ReleaseTaskSplitStep: Releases a task from a split step.

Method:	POST
URL:	api/tasks/{taskId}/split?commit={commit}
Parameters:	
Name	taskId
Description	The task id.
Name	splitParams
Description	The split link parameters.
Name	commit
Description	Commit release action if true. Optional , default value is True .
Request:	api/tasks/1/split?commit=true

```
[
  {
    "LinkAnchor": 5,
    "RequiredToComplete": true,
    "Independent": true
  },
  {...},
  {...}
]
```

Response:	200 OK
Error codes:	
200	The task was not found.
202	The task is not locked.
201	Task locked by another user.
204	The task is in an error state.
212	A mandatory attribute on a task was not set.
320	The link to destination step was not found.
220	Supertask has required subtasks that have not been completed.
322	You try to release not split step. Use method for release single/multi link steps.
223	Split link parameters are empty.
224	Split link parameters are invalid.
324	Step code check failed. The task could not be released.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

205 The task is already completed.

RouteTask: Routes a task to a new workflow step and then attempts to assign the task to the specified user, if specified.

Method: POST	
URL: api/tasks/{taskId}/route?stepId={stepId}&availableDate={availableDate}&userId={userId}&extUserId={extUserId}&commit={commit}	
Parameters:	
Name	taskId
Description	The task id.
Name	stepId
Description	The new step id.
Name	availableDate
Description	The new available date. Optional , default value is null (empty) .
Name	userId
Description	The internal user id to assign. Optional , default value is null (empty) .
Name	extUserId
Description	The external user id to assign. Optional , default value is null (empty) .
Name	commit
Description	Commit route action if true. Optional , default value is True .
Request: api/tasks/1/route?stepId=2&availableDate=4/30/2019 7:10:47 am&userId=4&extUserId=extUserIdStr5&commit=true	
Response: 200 OK	
Error codes:	
200	The task was not found.
300	The step was not found.
202	The task is not locked.
201	Task locked by another user.
221	User id parameter is invalid.
205	The task is already completed.

UnlockTask: Unlocks a task by the task id, only if the task was locked by current user.

Method: POST	
URL: api/tasks/{taskId}/unlock	
Parameters:	
Name	taskId
Description	The task id.
Request: api/tasks/1/unlock	
Response: 200 OK	
Error codes:	
200	The task was not found.
201	Task locked by another user.

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

TaskAttributes

TaskAttributes object allows you to get/set task attributes.

GetTaskAttributeById: Returns existing task attribute by id.

Method:	GET
URL:	api/tasks/{taskId}/attributes/{attrId}
Parameters:	
Name	taskId
Description	The task id.
Name	attrId
Description	The attribute id.
Request:	api/tasks/1/attributes/2
Response:	200 OK

```
{
  "Id": 1,
  "AttrType": "atInt",
  "Name": "NameStr2",
  "Value": {...}
}
```

Error codes:	
200	The task was not found.
210	A task attribute was not found.
213	A task attribute value was not found.

GetTaskAttributeByName: Returns existing task attribute by name.

Method:	GET
URL:	api/tasks/{taskId}/attributes/{name}
Parameters:	
Name	taskId
Description	The task id.
Name	name
Description	The attribute name.
Request:	api/tasks/1/attributes/nameStr2
Response:	200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Id": 1,
  "AttrType": "atInt",
  "Name": "NameStr2",
  "Value": {...}
}
```

Error codes:		
200		The task was not found.
210		A task attribute was not found.
213		A task attribute value was not found.

GetTaskAttributes: Returns a list of existing task attributes.

Method:	GET
URL:	api/tasks/{taskId}/attributes
Parameters:	
Name	taskId
Description	The task id.
Request:	api/tasks/1/attributes
Response:	200 OK

```
[
  {
    "Id": 1,
    "AttrType": "atInt",
    "Name": "NameStr2",
    "Value": {...}
  },
  {...},
  {...}
]
```

Error codes:		
200		The task was not found.
210		A task attribute was not found.

SetTaskAttributeById: Sets task attribute by id. Attribute value should use JSON format. For example: int - 12, float - 1.2, bool - true, string - "qwerty", date - "2012-04-23T18:25:43.511Z", user can be set by internal long id - 12 or by external string id(guid in string format) - "5e07d9a641229f439ae7b33b7d00f6ae".

Method:	POST
URL:	api/tasks/{taskId}/attributes/{attrId}
Parameters:	
Name	taskId
Description	The task id.
Name	attrId
Description	The attribute id.
Name	value

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Description

The task attribute value to be set.

Request:

api/tasks/1/attributes/2

{...}

Response:

200 OK

Error codes:

200

The task was not found.

210

A task attribute was not found.

201

Task locked by another user.

211

A task attribute value type is invalid.

221

User id parameter is invalid.

SetTaskAttributeByName: Sets task attribute by name. Attribute value should use JSON format. For example: int - 12, float - 1.2, bool - true, string - "qwerty", date - "2012-04-23T18:25:43.511Z", user can be set by internal long id - 12 or by external string id(guid in string format) - "5e07d9a641229f439ae7b33b7d00f6ae".

Method:

POST

URL:

api/tasks/{taskId}/attributes/{name}

Parameters:

Name

taskId

Description

The task id.

Name

name

Description

The attribute name.

Name

value

Description

The task attribute attribute to be set.

Request:

api/tasks/1/attributes/nameStr2

{...}

Response:

200 OK

Error codes:

200

The task was not found.

210

A task attribute was not found.

201

Task locked by another user.

211

A task attribute value type is invalid.

221

User id parameter is invalid.

Tasks

Tasks object allows you to create, update and retrieve list of Tasks.

CreateTask: Creates a new Task or Diary(if step id equals the DiaryManualStep value)

Method:

POST

URL:

api/tasks

Parameters:

Name

dataModel

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description

Not Available

Request:

api/tasks

```
{
  "ObjectId": 1,
  "StepId": 2,
  "Priority": 3,
  "Description": "DescriptionStr4",
  "PageNumber": 5,
  "Debug": true,
  "NoteId": 7,
  "Note": "NoteStr8",
  "UserId": 9,
  "ExtUserId": "ExtUserIdStr10",
  "Attributes": [
    {
      "Id": 11,
      "Value": {...}
    },
    {...},
    {...}
  ],
  "AvailableDate": "2019-04-30T07:10:47.5307219-04:00",
  "DeadLine": "2019-04-30T07:10:47.5307219-04:00"
}
```

Response:

200 OK

GetPostTasks: Provides functionality to Tasks retrieval and filtering.
See **GetTasks** method for filters clarifying.

Method:

POST

URL:

api/tasks/find?skip={skip}&top={top}

Parameters:

Name

Description

filter

Filter object

Name

Description

skip

Skip value for paging. **Optional**, default value is **null (empty)**.

Name

Description

top

Top value for paging. **Optional**, default value is **null (empty)**.

Request:

api/tasks/find?skip=22&top=23

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Tasks": [
    1,
    {...},
    {...}
  ],
  "AvailableDateStart": "2019-04-30T07:10:47.5337219-04:00",
  "AvailableDateEnd": "2019-04-30T07:10:47.5337219-04:00",
  "Flows": [
    4,
    {...},
    {...}
  ],
  "ExcludeFlows": true,
  "Steps": [
    6,
    {...},
    {...}
  ],
  "ExcludeSteps": true,
  "AssignedTo": [
    8,
    {...},
    {...}
  ],
  "AgeModelSet": [
    {
      "Id": 9,
      "From": 10,
      "To": 11
    },
    {...},
    {...}
  ],
  "AgeCalculationAlgorithm": "DateInitiated",
  "TaskStatus": [
    12,
    {...},
    {...}
  ],
  "ExcludeStatus": true,
  "Lockable": true,
  "OrderBy": "OrderByStr15",
  "LockedBy": [
    16,
    {...},
    {...}
  ],
  "Debug": true,
  "FileTypes": [
    18,
    {...},
    {...}
  ],
  "ExcludeFileTypes": true,
  "Drawers": [
    20,
    {...},
```

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

```
{...}  
},  
"ExcludeDrawers": true  
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "Id": 60,
  "Description": "DescriptionStr61",
  "Priority": 62,
  "FileId": 63,
  "FileName": "FileNameStr64",
  "FileNumber": "FileNumberStr65",
  "FlowId": 66,
  "FlowName": "FlowNameStr67",
  "StepId": 68,
  "StepName": "StepNameStr69",
  "SenderStep": 70,
  "SuperTaskId": 71,
  "RendezvousStepId": 72,
  "SubTaskIsRequired": true,
  "NoteId": 74,
  "AssignedTo": {
    "Id": 75,
    "ExternalId": "ExternalIdStr76",
    "Name": "NameStr77",
    "FriendlyName": "FriendlyNameStr78",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr80"
  },
  "Sender": {
    "Id": 81,
    "ExternalId": "ExternalIdStr82",
    "Name": "NameStr83",
    "FriendlyName": "FriendlyNameStr84",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr86"
  },
  "LockedBy": {
    "Id": 87,
    "ExternalId": "ExternalIdStr88",
    "Name": "NameStr89",
    "FriendlyName": "FriendlyNameStr90",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr92"
  },
  "AvailableDate": "2019-04-30T07:10:47.5557219-04:00",
  "StartDate": "2019-04-30T07:10:47.5557219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.5557219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.5557219-04:00",
  "DeadLine": "2019-04-30T07:10:47.5557219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.5557219-04:00",
  "Debug": true,
  "StackLevel": 100,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr101",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 102,
  "ObjectId": 103,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr104",
    "Name": "NameStr105",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr107",
    "Name": "NameStr108",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr110",
    "Name": "NameStr111",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 113,
"FileTypeId": 114,
"DrawerId": 115,
"ObjectTypeId": 116,
"ObjectClass": "RootContainer"
},
{...},
{...}
]
```

GetPostTasksBatch: Provides functionality to Tasks batch retrieval and filtering.
See **GetTasksBatch** method for filters clarifying.

Method:	POST
URL:	api/tasks/batch/find?rowNumStartFrom={rowNumStartFrom}&batchSize={batchSize}
Parameters:	
Name	filter
Description	Not Available
Name	rowNumStartFrom
Description	Row number to start batch from Optional , default value is -1.
Name	batchSize
Description	Amount of tasks at a batch Optional , default value is null (empty) .

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Request: `api/tasks/batch/find?rowNumStartFrom=27&batchSize=28`

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Direction": "Forward",
  "DrawerFileTypesTree": [
    {
      "Id": 1,
      "Children": [
        2,
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ],
  "FlowStepsTree": [
    {
      "Id": 3,
      "Children": [
        4,
        {...},
        {...}
      ]
    },
    {...},
    {...}
  ],
  "StepDurationSteps": [
    5,
    {...},
    {...}
  ],
  "Tasks": [
    6,
    {...},
    {...}
  ],
  "AvailableDateStart": "2019-04-30T07:10:47.5647219-04:00",
  "AvailableDateEnd": "2019-04-30T07:10:47.5647219-04:00",
  "Flows": [
    9,
    {...},
    {...}
  ],
  "ExcludeFlows": true,
  "Steps": [
    11,
    {...},
    {...}
  ],
  "ExcludeSteps": true,
  "AssignedTo": [
    13,
    {...},
    {...}
  ],
  "AgeModelSet": [
    {
      "Id": 14,
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
    "From": 15,
    "To": 16
  },
  {...},
  {...}
],
"AgeCalculationAlgorithm": "DateInitiated",
"TaskStatus": [
  17,
  {...},
  {...}
],
"ExcludeStatus": true,
"Lockable": true,
"OrderBy": "OrderByStr20",
"LockedBy": [
  21,
  {...},
  {...}
],
"Debug": true,
"FileTypes": [
  23,
  {...},
  {...}
],
"ExcludeFileTypes": true,
"Drawers": [
  25,
  {...},
  {...}
],
"ExcludeDrawers": true
}
```

Response: 200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
{
  "Id": 60,
  "Description": "DescriptionStr61",
  "Priority": 62,
  "FileId": 63,
  "FileName": "FileNameStr64",
  "FileNumber": "FileNumberStr65",
  "FlowId": 66,
  "FlowName": "FlowNameStr67",
  "StepId": 68,
  "StepName": "StepNameStr69",
  "SenderStep": 70,
  "SuperTaskId": 71,
  "RendezvousStepId": 72,
  "SubTaskIsRequired": true,
  "NoteId": 74,
  "AssignedTo": {
    "Id": 75,
    "ExternalId": "ExternalIdStr76",
    "Name": "NameStr77",
    "FriendlyName": "FriendlyNameStr78",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr80"
  },
  "Sender": {
    "Id": 81,
    "ExternalId": "ExternalIdStr82",
    "Name": "NameStr83",
    "FriendlyName": "FriendlyNameStr84",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr86"
  },
  "LockedBy": {
    "Id": 87,
    "ExternalId": "ExternalIdStr88",
    "Name": "NameStr89",
    "FriendlyName": "FriendlyNameStr90",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr92"
  },
  "AvailableDate": "2019-04-30T07:10:47.5657219-04:00",
  "StartDate": "2019-04-30T07:10:47.5657219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.5657219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.5657219-04:00",
  "DeadLine": "2019-04-30T07:10:47.5657219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.5657219-04:00",
  "Debug": true,
  "StackLevel": 100,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr101",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 102,
  "ObjectId": 103,
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr104",
    "Name": "NameStr105",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr107",
    "Name": "NameStr108",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr110",
    "Name": "NameStr111",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 113,
"FileTypeId": 114,
"DrawerId": 115,
"ObjectTypeId": 116,
"ObjectClass": "RootContainer"
},
{...},
{...}
]
```

GetTasks: Provides functionality to Tasks retrieval and filtering.

Filters should be passed as a query string using the standard ?key1=value1&key2=value2&etc syntax.

TaskId filter:

Filter by an array of task ids

e.g. /api/tasks?tasks=1&tasks=2

Available date filter - contains 2 parts:

Filter by DateTime since a task is available

e.g. /api/tasks?availableDateStart=2015-04-14T14:58:44.717

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Filter by DateTime before a task is available
e.g. /api/tasks?availableDateEnd=2015-04-14T14:58:44.717

Flow filter:

Filter by an array of flow ids
e.g. /api/tasks?flows=1&flows=2

ExcludeFlow:

Flag denotes, if ids specified in Flows are to be excluded. Default value = false
e.g. /api/tasks?flows=1&excludeFlows=true

Step filter:

Filters by an array of step ids
e.g. /api/tasks?steps=1&steps=2

ExcludeStep:

Flag denotes, if ids in specified Steps are to be excluded. Default value = false
e.g. /api/tasks?steps=1&excludeSteps=true

Assignment filter:

Filters by the assignment. Takes an array of ids.
e.g. /api/tasks?assignedTo=1&assignedTo=2
if you need an unassigned tasks you need to pass empty value
e.g. api/tasks?assignedTo - return only unassigned tasks.
api/tasks?assignedTo&assignedTo=1 returns unassigned tasks and tasks assigned to the user/group/role with id=1

Status filter:

Filters By Task Status, array of integers
e.g. /api/tasks?taskStatus=1&taskStatus=2

Statuses are:

0 = Ready
1 = Hold
2 = Locked (internal status! it does not relate to the lock/unlock functionality)
3 = Uncommitted or Waiting
4 = Completed
5 = Error
6 = Un-doable
7 = One Step

Exclude status:

Flag denotes, if ids in specified TaskStatus are to be excluded. Default value = false
e.g. /api/tasks?taskStatus=1&excludeStatus=true

Lockable filter:

If true, returns filtered list of tasks according the user's effective permissions to work with task. Default value = false
e.g. /api/tasks?lockable=true

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Ordering:

OrderBy enables ordering by a number of task properties.

Currently available properties: Priority, AvailableDate, DateInitiated, FlowName, FileNumber, TaskId

e.g. /api/tasks?orderBy=Priority

Locked by filter:

Filters tasks locked by the user with specified id.

e.g. /api/tasks?lockedBy=1&lockedBy=2

If you need an unlocked tasks you need to pass empty value.

e.g. api/tasks?lockedBy - return only unlocked tasks.

api/tasks?lockedBy&lockedBy=1 returns unlocked tasks and tasks locked by the user id=1

Debug filter:

If true, returns filtered list of debug tasks. Default = false

e.g. /api/tasks?debug=true

Method: GET

URL: api/tasks?skip={skip}&top={top}

Parameters:

Name	filter
------	--------

Description	See above
-------------	-----------

Name	skip
------	------

Description	Skip value for paging. Optional , default value is null (empty) .
-------------	---

Name	top
------	-----

Description	Top value for paging. Optional , default value is null (empty) .
-------------	--

Request: api/tasks?skip=22&top=23&filter=*imageright.api.models.tasks.taskfilter*

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "Id": 60,
  "Description": "DescriptionStr61",
  "Priority": 62,
  "FileId": 63,
  "FileName": "FileNameStr64",
  "FileNumber": "FileNumberStr65",
  "FlowId": 66,
  "FlowName": "FlowNameStr67",
  "StepId": 68,
  "StepName": "StepNameStr69",
  "SenderStep": 70,
  "SuperTaskId": 71,
  "RendezvousStepId": 72,
  "SubTaskIsRequired": true,
  "NoteId": 74,
  "AssignedTo": {
    "Id": 75,
    "ExternalId": "ExternalIdStr76",
    "Name": "NameStr77",
    "FriendlyName": "FriendlyNameStr78",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr80"
  },
  "Sender": {
    "Id": 81,
    "ExternalId": "ExternalIdStr82",
    "Name": "NameStr83",
    "FriendlyName": "FriendlyNameStr84",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr86"
  },
  "LockedBy": {
    "Id": 87,
    "ExternalId": "ExternalIdStr88",
    "Name": "NameStr89",
    "FriendlyName": "FriendlyNameStr90",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr92"
  },
  "AvailableDate": "2019-04-30T07:10:47.6147219-04:00",
  "StartDate": "2019-04-30T07:10:47.6147219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.6147219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.6147219-04:00",
  "DeadLine": "2019-04-30T07:10:47.6147219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.6147219-04:00",
  "Debug": true,
  "StackLevel": 100,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr101",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 102,
  "ObjectId": 103,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr104",
    "Name": "NameStr105",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr107",
    "Name": "NameStr108",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr110",
    "Name": "NameStr111",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 113,
"FileTypeId": 114,
"DrawerId": 115,
"ObjectTypeId": 116,
"ObjectClass": "RootContainer"
},
{...},
{...}
]
```

GetTasksBatch: Provides functionality to Tasks batch retrieval and filtering.

Method:	GET
URL:	api/tasks/batch?rowNumStartFrom={rowNumStartFrom}&batchSize={batchSize}
Parameters:	
Name	filter
Description	Filtering options. See GetTasks method for filters clarifying. Additionally bacth filter supports direction options: Batch direction option Specifies the direction of the batch Forward/Backward e.g. /api/tasks/batch?taskIdStartFrom=1&batchSize=1000&direction=0 - returns tasks start from taskIdStartFrom and next

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

or
/api/tasks/batch?taskIdStartFrom=1&batchSize=1000&direction=1 - returns tasks start from taskIdStartFrom and previous

Name	rowNumStartFrom
Description	Row number to start batch from Optional , default value is -1.

Name	batchSize
Description	Amount of tasks at a batch Optional , default value is null (empty) .

Request: api/tasks/batch?rowNumStartFrom=27&batchSize=28&filter=*imageright.api.models.tasks.batchtaskfilter*

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "Id": 60,
  "Description": "DescriptionStr61",
  "Priority": 62,
  "FileId": 63,
  "FileName": "FileNameStr64",
  "FileNumber": "FileNumberStr65",
  "FlowId": 66,
  "FlowName": "FlowNameStr67",
  "StepId": 68,
  "StepName": "StepNameStr69",
  "SenderStep": 70,
  "SuperTaskId": 71,
  "RendezvousStepId": 72,
  "SubTaskIsRequired": true,
  "NoteId": 74,
  "AssignedTo": {
    "Id": 75,
    "ExternalId": "ExternalIdStr76",
    "Name": "NameStr77",
    "FriendlyName": "FriendlyNameStr78",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr80"
  },
  "Sender": {
    "Id": 81,
    "ExternalId": "ExternalIdStr82",
    "Name": "NameStr83",
    "FriendlyName": "FriendlyNameStr84",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr86"
  },
  "LockedBy": {
    "Id": 87,
    "ExternalId": "ExternalIdStr88",
    "Name": "NameStr89",
    "FriendlyName": "FriendlyNameStr90",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr92"
  },
  "AvailableDate": "2019-04-30T07:10:47.6157219-04:00",
  "StartDate": "2019-04-30T07:10:47.6157219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.6157219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.6157219-04:00",
  "DeadLine": "2019-04-30T07:10:47.6157219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.6157219-04:00",
  "Debug": true,
  "StackLevel": 100,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr101",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 102,
  "ObjectId": 103,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr104",
    "Name": "NameStr105",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr107",
    "Name": "NameStr108",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr110",
    "Name": "NameStr111",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 113,
"FileTypeId": 114,
"DrawerId": 115,
"ObjectId": 116,
"ObjectClass": "RootContainer"
},
{...},
{...}
]
```

GetTasksByFileId: Retrieves Tasks by file id via Http Get Request.
See **GetTasks** method for filters clarifying.

Method:	GET
URL:	api/files/{fileId}/tasks?skip={skip}&top={top}
Parameters:	
Name	filter
Description	Filter object
Name	fileId
Description	File ID to filter by
Name	skip
Description	Skip value for paging. Optional , default value is null (empty) .

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Name	top
Description	Top value for paging. Optional , default value is null (empty) .

Request: `api/files/22/tasks?skip=23&top=24&filter=imageright.api.models.tasks.taskfilter`

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "Id": 60,
  "Description": "DescriptionStr61",
  "Priority": 62,
  "FileId": 63,
  "FileName": "FileNameStr64",
  "FileNumber": "FileNumberStr65",
  "FlowId": 66,
  "FlowName": "FlowNameStr67",
  "StepId": 68,
  "StepName": "StepNameStr69",
  "SenderStep": 70,
  "SuperTaskId": 71,
  "RendezvousStepId": 72,
  "SubTaskIsRequired": true,
  "NoteId": 74,
  "AssignedTo": {
    "Id": 75,
    "ExternalId": "ExternalIdStr76",
    "Name": "NameStr77",
    "FriendlyName": "FriendlyNameStr78",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr80"
  },
  "Sender": {
    "Id": 81,
    "ExternalId": "ExternalIdStr82",
    "Name": "NameStr83",
    "FriendlyName": "FriendlyNameStr84",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr86"
  },
  "LockedBy": {
    "Id": 87,
    "ExternalId": "ExternalIdStr88",
    "Name": "NameStr89",
    "FriendlyName": "FriendlyNameStr90",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr92"
  },
  "AvailableDate": "2019-04-30T07:10:47.6157219-04:00",
  "StartDate": "2019-04-30T07:10:47.6157219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.6157219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.6157219-04:00",
  "DeadLine": "2019-04-30T07:10:47.6157219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.6157219-04:00",
  "Debug": true,
  "StackLevel": 100,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr101",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 102,
  "ObjectId": 103,
```

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr104",
    "Name": "NameStr105",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr107",
    "Name": "NameStr108",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr110",
    "Name": "NameStr111",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 113,
"FileTypeId": 114,
"DrawerId": 115,
"ObjectTypeId": 116,
"ObjectClass": "RootContainer"
},
{...},
{...}
]
```

GetTasksByFileIdPost: Retrieves Tasks by file id via Http Post Request.
See **GetTasks** method for filters clarifying.

Method:	POST
URL:	api/files/{fileId}/tasks/find?skip={skip}&top={top}
Parameters:	
Name	filter
Description	Filter object
Name	fileId
Description	File ID to filter by
Name	skip
Description	Skip value for paging. Optional , default value is null (empty) .

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

Name	top
Description	Top value for paging. Optional , default value is null (empty) .

Request: `api/files/22/tasks/find?skip=23&top=24`

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "Tasks": [
    1,
    {...},
    {...}
  ],
  "AvailableDateStart": "2019-04-30T07:10:47.6167219-04:00",
  "AvailableDateEnd": "2019-04-30T07:10:47.6167219-04:00",
  "Flows": [
    4,
    {...},
    {...}
  ],
  "ExcludeFlows": true,
  "Steps": [
    6,
    {...},
    {...}
  ],
  "ExcludeSteps": true,
  "AssignedTo": [
    8,
    {...},
    {...}
  ],
  "AgeModelSet": [
    {
      "Id": 9,
      "From": 10,
      "To": 11
    },
    {...},
    {...}
  ],
  "AgeCalculationAlgorithm": "DateInitiated",
  "TaskStatus": [
    12,
    {...},
    {...}
  ],
  "ExcludeStatus": true,
  "Lockable": true,
  "OrderBy": "OrderByStr15",
  "LockedBy": [
    16,
    {...},
    {...}
  ],
  "Debug": true,
  "FileTypes": [
    18,
    {...},
    {...}
  ],
  "ExcludeFileTypes": true,
  "Drawers": [
    20,
    {...},
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{...}  
},  
"ExcludeDrawers": true  
}
```

Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
{
  "Id": 60,
  "Description": "DescriptionStr61",
  "Priority": 62,
  "FileId": 63,
  "FileName": "FileNameStr64",
  "FileNumber": "FileNumberStr65",
  "FlowId": 66,
  "FlowName": "FlowNameStr67",
  "StepId": 68,
  "StepName": "StepNameStr69",
  "SenderStep": 70,
  "SuperTaskId": 71,
  "RendezvousStepId": 72,
  "SubTaskIsRequired": true,
  "NoteId": 74,
  "AssignedTo": {
    "Id": 75,
    "ExternalId": "ExternalIdStr76",
    "Name": "NameStr77",
    "FriendlyName": "FriendlyNameStr78",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr80"
  },
  "Sender": {
    "Id": 81,
    "ExternalId": "ExternalIdStr82",
    "Name": "NameStr83",
    "FriendlyName": "FriendlyNameStr84",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr86"
  },
  "LockedBy": {
    "Id": 87,
    "ExternalId": "ExternalIdStr88",
    "Name": "NameStr89",
    "FriendlyName": "FriendlyNameStr90",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr92"
  },
  "AvailableDate": "2019-04-30T07:10:47.6167219-04:00",
  "StartDate": "2019-04-30T07:10:47.6167219-04:00",
  "LockExpiration": "2019-04-30T07:10:47.6167219-04:00",
  "UndoExpires": "2019-04-30T07:10:47.6167219-04:00",
  "DeadLine": "2019-04-30T07:10:47.6167219-04:00",
  "DateInitiated": "2019-04-30T07:10:47.6167219-04:00",
  "Debug": true,
  "StackLevel": 100,
  "ErrorCode": "None",
  "ErrorMessage": "ErrorMessageStr101",
  "StatusDetails": "None",
  "Status": "Ready",
  "PageNumber": 102,
  "ObjectId": 103,
```

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
"DocumentAttributes": [
  {
    "DisplayName": "DisplayNameStr104",
    "Name": "NameStr105",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FolderAttributes": [
  {
    "DisplayName": "DisplayNameStr107",
    "Name": "NameStr108",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"FileAttributes": [
  {
    "DisplayName": "DisplayNameStr110",
    "Name": "NameStr111",
    "Value": {...},
    "AttributeType": "atInt"
  },
  {...},
  {...}
],
"RowNum": 113,
"FileTypeId": 114,
"DrawerId": 115,
"ObjectTypeId": 116,
"ObjectClass": "RootContainer"
},
{...},
{...}
]
```

UpdateTask: Updates the task.

Method:	POST
URL:	api/tasks/{taskId}
Parameters:	
Name	updateModel
Description	The task update model.
Name	taskId
Description	Task id
Request:	api/tasks/6

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "DateAvailable": "2019-04-30T07:10:47.7187219-04:00",
  "Description": "DescriptionStr2",
  "Priority": 3,
  "UserId": 4,
  "ExtUserId": "ExtUserIdStr5"
}
```

Response:	200 OK
Error codes:	
221	User id parameter is invalid.
12	The model is empty.
200	The task was not found.
201	Task locked by another user.
202	The task is not locked.

Users

API for operations on user accounts

AssignUserToRole: Allows for an ImageRight user to be added to an existing ImageRight role. The specified user will inherit the permissions assigned to the role.

Method:	POST
URL:	api/users/{userId}/assignrole/{roleId}
Parameters:	
Name	userId
Description	The id of the user account.
Name	roleId
Description	The id of the role.
Request:	api/users/1/assignrole/2
Response:	200 OK
Error codes:	
1304	The user is not found.
1402	The role is not found.
1401	The role already exists.

ChangeUserPassword: Allows for the ability to change an ImageRight user's password.

This method is available anonymously.

Method:	POST
URL:	api/users/changepassword
Parameters:	
Name	passwordChangeModel
Description	The data for changing password.
Request:	api/users/changepassword

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "UserName": "UserNameStr1",
  "OldPassword": "OldPasswordStr2",
  "NewPassword": "NewPasswordStr3"
}
```

Response: 200 OK

Error codes:

1	The username and/or the password is incorrect.
3	Account is locked. Try again later.
6	Password doesn't comply with the password policy.
5	Password has been previously used. Choose another password.
7	You are not allowed change your password.

CreateUser: Creates a new ImageRight user and enter specific information about that user in the ImageRight system.

Method: POST

URL: api/users

Parameters:

Name	userModel
Description	The data for creating a new user account.

Request: api/users

```
{
  "DepartmentId": 1,
  "ExternalId": "ExternalIdStr2",
  "EmployeeId": "EmployeeIdStr3",
  "Address": "AddressStr4",
  "WorkPhone": "WorkPhoneStr5",
  "HomePhone": "HomePhoneStr6",
  "Email": "EmailStr7",
  "Zipcode": "ZipcodeStr8",
  "StateProvince": "StateProvinceStr9",
  "Country": "CountryStr10",
  "FullName": "FullNameStr11",
  "ProfileName": "ProfileNameStr12",
  "Name": "NameStr13",
  "Description": "DescriptionStr14",
  "Password": "PasswordStr15",
  "UserFlags": "CanChangePassword",
  "AccountExpires": "2019-04-30T07:10:47.7267219-04:00",
  "AccountActivates": "2019-04-30T07:10:47.7267219-04:00",
  "PasswordExpires": "2019-04-30T07:10:47.7267219-04:00"
}
```

Response: 201 Created

```
1
```

Error codes:

1301	The user name already exists.
1300	Profile not found.
6	Password doesn't comply with the password policy.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetCurrentUser: Returns the accountId of the current ImageRight user.

Method: GET
URL: api/users/currentuser
Request: api/users/currentuser
Response: 200 OK

```
1
```

GetCurrentUserData: Returns account information about the current user in the ImageRight system.

Method: GET
URL: api/users/currentuser/userdata
Request: api/users/currentuser/userdata
Response: 200 OK

```
{
  "DepartmentId": 1,
  "EmployeeId": "EmployeeIdStr2",
  "Address": "AddressStr3",
  "WorkPhone": "WorkPhoneStr4",
  "HomePhone": "HomePhoneStr5",
  "Email": "EmailStr6",
  "Zipcode": "ZipcodeStr7",
  "StateProvince": "StateProvinceStr8",
  "Country": "CountryStr9",
  "FullName": "FullNameStr10",
  "ProfileName": "ProfileNameStr11",
  "Name": "NameStr12",
  "UserFlags": "CanChangePassword"
}
```

GetGroupsForUser: Returns a list of groups that a specified user account is a member of.

Method: GET
URL: api/users/{userid}/groups
Parameters:

Name	userid
Description	Not Available

Request: api/users/1/groups
Response: 200 OK

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

GetUser: Returns a specified user account from the ImageRight system.

Method: GET
URL: api/users/{userid}
Parameters:

Name	userid
Description	Not Available

Request: api/users/1
Response: 200 OK

```
{
  "Id": 1,
  "ExternalId": "ExternalIdStr2",
  "Name": "NameStr3",
  "FriendlyName": "FriendlyNameStr4",
  "Type": "user",
  "Enabled": true,
  "Description": "DescriptionStr6"
}
```

GetUserData: Returns account information about the specified user in the ImageRight system.

Method: GET
URL: api/users/{userId}/userdata
Parameters:

Name	userId
Description	The Id of the user account.

Request: api/users/1/userdata
Response: 200 OK

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

```
{
  "DepartmentId": 1,
  "EmployeeId": "EmployeeIdStr2",
  "Address": "AddressStr3",
  "WorkPhone": "WorkPhoneStr4",
  "HomePhone": "HomePhoneStr5",
  "Email": "EmailStr6",
  "Zipcode": "ZipcodeStr7",
  "StateProvince": "StateProvinceStr8",
  "Country": "CountryStr9",
  "FullName": "FullNameStr10",
  "ProfileName": "ProfileNameStr11",
  "Name": "NameStr12",
  "UserFlags": "CanChangePassword"
}
```

GetUsers: Returns all user accounts in the ImageRight system.

Method: GET
URL: api/users
Request: api/users
Response: 200 OK

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

MapUserToVssoAccount: No Help Available

Method: POST
URL: api/users/{userid}/mapexternalid
Parameters:

Name	userid
Description	Not Available
Name	vimUserId
Description	Not Available

Request: api/users/1/mapexternalid

2

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Response: 200 OK

true

Error codes:

11	The account was not found.
1800	The vsso migration state is invalid for this operation.
1801	The vsso account is already mapped.

SetUserAccountDisabled: Disables the specified ImageRight user account from being used within the ImageRight system.

Method: POST
URL: api/users/{userId}/disable

Parameters:

Name	userId
Description	The Id of the user account.

Request: api/users/1/disable

Response: 200 OK

Error codes:

1302	Unable to disable/enable admin user account.
1303	Unable to disable/enable IR server user account.

SetUserAccountEnabled: Enables the specified ImageRight user account to be used within the ImageRight system.

Method: POST
URL: api/users/{userId}/enable

Parameters:

Name	userId
Description	The Id of the user account.

Request: api/users/1/enable

Response: 200 OK

Error codes:

1302	Unable to disable/enable admin user account.
1303	Unable to disable/enable IR server user account.

UnassignUserFromRole: Un-assigns the specified user from an ImageRight role. This will also remove the inherited permissions that were previously granted through the assigned role.

Method: POST
URL: api/users/{userId}/unassignrole/{roleId}

Parameters:

Name	userId
Description	The Id of the user account.
Name	roleId

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Description

The Id of the role.

Request:

api/users/1/unassignrole/2

Response:

200 OK

Error codes:

1304

The user is not found.

1402

The role is not found.

UpdatePassword: Allows the user to reset the password.

This method is available anonymously.

Method:

POST

URL:

api/users/newpasswordrequest

Parameters:

Name

saveNewPasswordModel

Description

Not Available

Request:

api/users/newpasswordrequest

```
{
  "Token": "TokenStr1",
  "NewPassword": "NewPasswordStr2"
}
```

Response:

200 OK

UpdateUserData: Updates the data releated to a specified ImageRight user account.

Method:

POST

URL:

api/users/{userid}

Parameters:

Name

userid

Description

The userid for the user that needs to be updated

Name

updateUserModel

Description

The user data for the specified user account.

Request:

api/users/1

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
{
  "Name": "NameStr2",
  "Description": "DescriptionStr3",
  "UserDataModel": {
    "DepartmentId": 4,
    "EmployeeId": "EmployeeIdStr5",
    "Address": "AddressStr6",
    "WorkPhone": "WorkPhoneStr7",
    "HomePhone": "HomePhoneStr8",
    "Email": "EmailStr9",
    "Zipcode": "ZipcodeStr10",
    "StateProvince": "StateProvinceStr11",
    "Country": "CountryStr12",
    "FullName": "FullNameStr13",
    "ProfileName": "ProfileNameStr14",
    "Name": "NameStr15",
    "UserFlags": "CanChangePassword"
  }
}
```

Response: 200 OK

Error codes:

11	The account was not found.
----	----------------------------

Workflow

Workflow object provides workflow and step metadata retrieval.

GetDefaultStepLink: Returns a default step link.

Method:	GET
URL:	api/steps/{stepId}/links/default?isDefault={isDefault}&status={status}
Parameters:	
Name	stepId
Description	The source step id.
Name	isDefault
Description	The isDefault parameter. Not used now. Optional , default value is True .
Name	status
Description	The link status. Optional , default value is All .
Values	Original, New, ToBeDeleted, All
Request:	api/steps/1/links/default?isDefault=true&status=original
Response:	200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
[
  {
    "Anchor": 1,
    "Description": "DescriptionStr2",
    "IsAnchorSet": true,
    "IsDebug": true,
    "IsDefault": true,
    "IsDeleted": true,
    "Label": "LabelStr7",
    "OneWay": true,
    "SourceStepId": 9,
    "TargetStepId": 10
  },
  {...},
  {...}
]
```

Error codes:		
300		The step was not found.
311		The step links was not found.

GetFlowById: Returns a flow data by the flow id.

Method:	GET				
URL:	api/workflows/{flowId}				
Parameters:					
	<table><tr><td>Name</td><td>flowId</td></tr><tr><td>Description</td><td>The flow id.</td></tr></table>	Name	flowId	Description	The flow id.
Name	flowId				
Description	The flow id.				
Request:	api/workflows/1				
Response:	200 OK				

```
{
  "Id": 1,
  "Name": "NameStr2",
  "TypeGuid": "TypeGuidStr3",
  "Status": "Normal",
  "EffectivePermissions": "None",
  "Debug": true,
  "IsStart": true,
  "IsValidation": true,
  "ProgName": "ProgNameStr7",
  "FlowId": 8
}
```

GetPriorityList: Returns a list of priorities the user can apply at the given step.

Method:	GET				
URL:	api/steps/{stepId}/priorities				
Parameters:					
	<table><tr><td>Name</td><td>stepId</td></tr><tr><td>Description</td><td>The step id.</td></tr></table>	Name	stepId	Description	The step id.
Name	stepId				
Description	The step id.				

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

Request: api/steps/1/priorities

Response: 200 OK

```
[
  1,
  {...},
  {...}
]
```

Error codes:
300 The step was not found.

GetSplitLinkParameters: Returns split link parameters for a split step.

Method: GET
URL: api/steps/{stepId}/splitparams?isDebug={isDebug}
Parameters:

Name	stepId
Description	The split step id.
Name	isDebug
Description	True to get debug step definition, false to get production step definition. Optional , default value is False .

Request: api/steps/1/splitparams?isDebug=true

Response: 200 OK

```
[
  {
    "LinkAnchor": 4,
    "RequiredToComplete": true,
    "Independent": true
  },
  {...},
  {...}
]
```

Error codes:
300 The step was not found.
225 Split link parameters of this step were not found.

GetStepAttributeById: Returns a step attribute by id.

Method: GET
URL: api/steps/{stepId}/attributes/{attrId}
Parameters:

Name	stepId
Description	The step id.
Name	attrId
Description	The attribute id.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

Request: api/steps/1/attributes/2

Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
{
  "Id": 1,
  "Type": "atInt",
  "DisplayName": "DisplayNameStr2",
  "Name": "NameStr3",
  "Description": "DescriptionStr4",
  "Mandatory": true,
  "DefaultValue": {...},
  "HasDefaultValue": true,
  "Disabled": true,
  "Visibility": true,
  "RequirementsRule": "No",
  "ValidationRule": {
    "ControlType": "CheckBox",
    "MinValue": 10.0,
    "MaxValue": 11.0,
    "Mask": "MaskStr12",
    "Length": 13,
    "Choices": [
      {
        "Value": {...},
        "Description": "DescriptionStr15",
        "Tag": "TagStr16"
      },
      {...},
      {...}
    ],
    "Validators": [
      {
        "Mask": "MaskStr17",
        "Name": "NameStr18",
        "Type": "TypeStr19",
        "ValidatorItemList": [
          "Str20",
          {...},
          {...}
        ],
        "ValidatorRange": [
          {
            "From": "FromStr21",
            "To": "ToStr22"
          },
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ],
    "ReadOnly": true,
    "AllowNew": true,
    "NameVisible": true,
    "DescrVisible": true,
    "Separator": "SeparatorStr27",
    "Order": 28,
    "Flags": 29,
    "ShouldSort": true,
    "IsDefaultChoices": true
  }
}
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
}  
}
```

Error codes:	300	The step was not found.
	310	A step attribute was not found.

GetStepAttributeByName: Returns a step attribute by name.

Method:	GET
URL:	api/steps/{stepId}/attributes/{name}
Parameters:	
Name	stepId
Description	The step id.
Name	name
Description	The attribute name.
Request:	api/steps/1/attributes/nameStr2
Response:	200 OK

API

- Introduction
- Authentication
- Status and error codes
- Push Notifications

METHODS

- Accounts
- Attributes
- Authentication
- Batches
- ClientActions
- Containers
- Dashboard
- DashboardViews
- DataValidation
- DocumentFilter
- Documents
- DocumentsV2
- Drawers
- Encryption
- Files
- Folders
- Images

```
{
  "Id": 1,
  "Type": "atInt",
  "DisplayName": "DisplayNameStr2",
  "Name": "NameStr3",
  "Description": "DescriptionStr4",
  "Mandatory": true,
  "DefaultValue": {...},
  "HasDefaultValue": true,
  "Disabled": true,
  "Visibility": true,
  "RequirementsRule": "No",
  "ValidationRule": {
    "ControlType": "CheckBox",
    "MinValue": 10.0,
    "MaxValue": 11.0,
    "Mask": "MaskStr12",
    "Length": 13,
    "Choices": [
      {
        "Value": {...},
        "Description": "DescriptionStr15",
        "Tag": "TagStr16"
      },
      {...},
      {...}
    ],
    "Validators": [
      {
        "Mask": "MaskStr17",
        "Name": "NameStr18",
        "Type": "TypeStr19",
        "ValidatorItemList": [
          "Str20",
          {...},
          {...}
        ],
        "ValidatorRange": [
          {
            "From": "FromStr21",
            "To": "ToStr22"
          },
          {...},
          {...}
        ]
      },
      {...},
      {...}
    ],
    "Readonly": true,
    "AllowNew": true,
    "NameVisible": true,
    "DescrVisible": true,
    "Separator": "SeparatorStr27",
    "Order": 28,
    "Flags": 29,
    "ShouldSort": true,
    "IsDefaultChoices": true
  }
}
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
}  
}
```

Error codes:	300	The step was not found.
	310	A step attribute was not found.

GetStepAttributes: Returns a list of all step attributes.

Method:	GET				
URL:	api/steps/{stepId}/attributes				
Parameters:					
	<table><tr><th>Name</th><td>stepId</td></tr><tr><th>Description</th><td>The step id.</td></tr></table>	Name	stepId	Description	The step id.
Name	stepId				
Description	The step id.				
Request:	api/steps/1/attributes				
Response:	200 OK				

API
Introduction
Authentication
Status and error codes
Push Notifications
METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
[
  {
    "Id": 1,
    "Type": "atInt",
    "DisplayName": "DisplayNameStr2",
    "Name": "NameStr3",
    "Description": "DescriptionStr4",
    "Mandatory": true,
    "DefaultValue": {...},
    "HasDefaultValue": true,
    "Disabled": true,
    "Visibility": true,
    "RequirementsRule": "No",
    "ValidationRule": {
      "ControlType": "CheckBox",
      "MinValue": 10.0,
      "MaxValue": 11.0,
      "Mask": "MaskStr12",
      "Length": 13,
      "Choices": [
        {
          "Value": {...},
          "Description": "DescriptionStr15",
          "Tag": "TagStr16"
        },
        {...},
        {...}
      ],
      "Validators": [
        {
          "Mask": "MaskStr17",
          "Name": "NameStr18",
          "Type": "TypeStr19",
          "ValidatorItemList": [
            "Str20",
            {...},
            {...}
          ],
          "ValidatorRange": [
            {
              "From": "FromStr21",
              "To": "ToStr22"
            },
            {...},
            {...}
          ]
        },
        {...},
        {...}
      ],
      "Readonly": true,
      "AllowNew": true,
      "NameVisible": true,
      "DescrVisible": true,
      "Separator": "SeparatorStr27",
      "Order": 28,
      "Flags": 29,
      "ShouldSort": true,
```

API
Introduction
Authentication
Status and error codes
Push Notifications

METHODS
Accounts
Attributes
Authentication
Batches
ClientActions
Containers
Dashboard
DashboardViews
DataValidation
DocumentFilter
Documents
DocumentsV2
Drawers
Encryption
Files
Folders
Images

```
    "IsDefaultChoices": true
  }
},
{...},
{...}
]
```

Error codes:	
300	The step was not found.
310	A step attribute was not found.

GetStepById: Returns a step data by the step id.

Method:	GET
URL:	api/steps/{stepId}
Parameters:	
Name	stepId
Description	The step id.
Request:	api/steps/1
Response:	200 OK

```
{
  "Id": 1,
  "Name": "NameStr2",
  "TypeGuid": "TypeGuidStr3",
  "Status": "Normal",
  "EffectivePermissions": "None",
  "Debug": true,
  "IsStart": true,
  "IsValidation": true,
  "ProgName": "ProgNameStr7",
  "FlowId": 8
}
```

GetStepByProgName: Returns a step data by the step programmatic name.

Method:	GET
URL:	api/workflows/{flowProgName}/{stepProgName}
Parameters:	
Name	flowProgName
Description	The flow programmatic name.
Name	stepProgName
Description	The step programmatic name.
Request:	api/workflows/flowProgNameStr1/stepProgNameStr2
Response:	200 OK

API	
Introduction	
Authentication	
Status and error codes	
Push Notifications	
METHODS	
Accounts	
Attributes	
Authentication	
Batches	
ClientActions	
Containers	
Dashboard	
DashboardViews	
DataValidation	
DocumentFilter	
Documents	
DocumentsV2	
Drawers	
Encryption	
Files	
Folders	
Images	

```
{
  "Id": 1,
  "Name": "NameStr2",
  "TypeGuid": "TypeGuidStr3",
  "Status": "Normal",
  "EffectivePermissions": "None",
  "Debug": true,
  "IsStart": true,
  "IsValidation": true,
  "ProgName": "ProgNameStr7",
  "FlowId": 8
}
```

GetStepLinks: Returns a list of all step links or a link between source and target steps.

Method:	GET
URL:	api/steps/{srcStepId}/links?targetStepId={targetStepId}&status={status}
Parameters:	
Name	srcStepId
Description	The source step id.
Name	targetStepId
Description	The target step id. Optional , default value is null (empty) .
Name	status
Description	The link status. Optional , default value is All .
Values	Original, New, ToBeDeleted, All
Request:	api/steps/1/links?targetStepId=2&status=original
Response:	200 OK

```
[
  {
    "Anchor": 1,
    "Description": "DescriptionStr2",
    "IsAnchorSet": true,
    "IsDebug": true,
    "IsDefault": true,
    "IsDeleted": true,
    "Label": "LabelStr7",
    "OneWay": true,
    "SourceStepId": 9,
    "TargetStepId": 10
  },
  {...},
  {...}
]
```

Error codes:	
300	The step was not found.
311	The step links was not found.

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

GetSteps: Returns a list of workflow steps with current user effective permissions.

Method: GET

URL: `api/workflows/{flowId}/steps?includeBuddies={includeBuddies}&flag={flag}`

Parameters:

Name flowId

Description The workflow id.

Name includeBuddies

Description Not Available **Optional**, default value is **False**.

Name flag

Description The step request flag. You can use two or more flags together using OR bitwise operation. If you don't add the LoadDetails flag, you won't get the StepDef RoundRobins populated, among other things. **Optional**, default value is **Production**.

Values Production, Debug, Deleted, Both, LoadDetails

Request: `api/workflows/1/steps?includeBuddies=true&flag=production`

Response: 200 OK

```
[
  {
    "Id": 1,
    "Name": "NameStr2",
    "TypeGuid": "TypeGuidStr3",
    "Status": "Normal",
    "EffectivePermissions": "None",
    "Debug": true,
    "IsStart": true,
    "IsValid": true,
    "ProgName": "ProgNameStr7",
    "FlowId": 8
  },
  {...},
  {...}
]
```

Error codes:

14 The workflow was not found.

GetUsersToAssign: Returns a list of users which can be assign on this step.

Method: GET

URL: `api/steps/{stepId}/users`

Parameters:

Name stepId

Description The step id.

Request: `api/steps/1/users`

Response: 200 OK

API

Introduction

Authentication

Status and error codes

Push Notifications

METHODS

Accounts

Attributes

Authentication

Batches

ClientActions

Containers

Dashboard

DashboardViews

DataValidation

DocumentFilter

Documents

DocumentsV2

Drawers

Encryption

Files

Folders

Images

```
[
  {
    "Id": 1,
    "ExternalId": "ExternalIdStr2",
    "Name": "NameStr3",
    "FriendlyName": "FriendlyNameStr4",
    "Type": "user",
    "Enabled": true,
    "Description": "DescriptionStr6"
  },
  {...},
  {...}
]
```

Error codes:

300 The step was not found.

GetWorkflows: Returns a list of workflows with current user effective permissions.

Method: GET

URL: api/workflows?includeBuddies={includeBuddies}

Parameters:

Name	includeBuddies
Description	Not Available Optional , default value is False .

Request: api/workflows?includeBuddies=true

Response: 200 OK

```
[
  {
    "Id": 1,
    "Name": "NameStr2",
    "Status": "Active",
    "EffectivePermissions": "Read",
    "FlowProgName": "FlowProgNameStr3"
  },
  {...},
  {...}
]
```