BFGUI Api Documentation

# BFGUI Api Documentation

Version: v0.39.3

BasePath: https://bfgui-dev.webdev.banksvcs.net/bfgui

## Access

## Methods

[ Jump to [Models](#X08a735672bda25f91812485a33b810177d042f8) ]

### Table of Contents

#### [AuditEventController](#AuditEventController)

* [post /api/news](#getAuditEvents)
* [get /api/news/event-criteria-data](#getEventCriteriaData)

#### [AuthController](#AuthController)

* [post /api/auth/reauth](#continueSession)
* [post /api/auth/signin](#signIn)
* [post /api/auth/sso](#signInSso)

#### [CertificateController](#CertificateController)

* [post /api/certificates](#createTrustedCertificate)
* [delete /api/certificates/pending/{id}](#deletePendingCertificates)
* [delete /api/certificates/{id}](#deleteTrustedCertificate)
* [get /api/certificates/{id}](#getCertificateById)
* [get /api/certificates](#getCertificates)
* [get /api/certificates/pending/{id}](#getChangeControlCertificate)
* [get /api/certificates/pending](#getPendingCertificates)
* [get /api/certificates/existence](#isExistingCertificateName)
* [post /api/certificates/pending](#postPendingCertificates)
* [put /api/certificates/pending/{id}](#updatePendingCertificate)
* [post /api/certificates/upload](#uploadFile)
* [get /api/certificates/validate/{id}](#validateCertificateById)
* [get /api/certificates/validate/pending/{id}](#validatePendingCertificateById)

#### [EntityController](#EntityController)

* [post /api/entities](#createEntity)
* [delete /api/entities/{id}](#deleteEntity)
* [delete /api/entities/pending/{id}](#deletePendingEntity)
* [get /api/entities/changeControl/{id}](#getChangeControlById)
* [get /api/entities](#getEntities)
* [get /api/entities/{id}](#getEntityById)
* [get /api/entities/file-type](#getFileType)
* [get /api/entities/inbound-request-type](#getInboundRequestType)
* [get /api/entities/inbound-service](#getInboundService)
* [get /api/entities/mq-details](#getMqDetails)
* [get /api/entities/participants](#getParticipants)
* [get /api/entities/pending](#getPendingEntities)
* [get /api/entities/pending/{id}](#getPendingEntityById)
* [get /api/entities/existence/route-attributes](#getRequestType)
* [get /api/entities/swift-service](#getSwiftService)
* [get /api/entities/existence/entity-service](#isExistingEntity)
* [get /api/entities/existence/mailbox](#isExistingMailboxPathOut)
* [get /api/entities/existence/queue](#isExistingMqQueueOut)
* [post /api/entities/pending](#postPendingEntities)
* [post /api/entities/transmit](#transmit)
* [put /api/entities/{id}](#updateEntity)
* [put /api/entities/pending/{id}](#updatePendingEntity)

#### [FileSearchController](#FileSearchController)

* [get /api/files/sepa/export-report](#exportFilesReport)
* [get /api/files/{fileId}/transactions/export-report](#exportTransactionsReport)
* [get /api/files/document-content](#getDocumentContent1)
* [get /api/files/error/{errorCode}](#getErrorDetailsByCode)
* [get /api/files/{id}](#getFileById)
* [get /api/files/file-criteria-data](#getFileCriteriaData)
* [get /api/files/file-monitor](#getFileMonitor)
* [post /api/files](#getFiles)
* [post /api/files/sepa](#getSEPAFiles)
* [get /api/files/{fileId}/transactions](#getTransactionsForFiles)

#### [SctTransactionSearchController](#SctTransactionSearchController)

* [post /api/transactions](#getSCTTransactions)
* [get /api/transactions/{id}](#getTransactionById)
* [get /api/transactions/transaction-criteria-data](#getTransactionCriteriaData)

#### [StatisticsController](#StatisticsController)

* [get /api/statistics/sct-alerts](#getSCTAlerts)
* [get /api/statistics/sct-traffic](#getSCTTrafficSummary)
* [get /api/statistics/sepa-traffic](#getSepaTrafficSummary)
* [get /api/statistics/system-errors](#getSystemErrorsStatistics)

#### [VersionController](#VersionController)

* [get /api/version](#getApplicationVersion)
* [get /api/version/f5](#getF5Link)
* [get /api/version/sepa-visibility](#isSepaDashboardVisible)

#### [WorkflowController](#WorkflowController)

* [get /api/workflow/bp-details](#getBPDetails)
* [get /api/workflow/bp-header](#getBPHeader)
* [get /api/workflow/document-content](#getDocumentContent)
* [get /api/workflow/wfcId/{wfcId}](#getWfIdAndVersion)
* [get /api/workflow/{id}/steps](#getWorkflowSteps)

# AuditEventController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/news

(getAuditEvents)

Returns the admin audit events to populate the event logs on the Home page.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [AuditEventCriteria](#AuditEventCriteria) (optional)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

array[[AuditEvent](#AuditEvent)]

### Example data

Content-Type: application/json

[ {  
 "actor" : "actor",  
 "actionType" : "actionType",  
 "objectActedOn" : "objectActedOn",  
 "created" : "2000-01-23T04:56:07.000+00:00",  
 "action" : "action",  
 "id" : "id",  
 "eventType" : "eventType",  
 "eventContext" : "eventContext"  
}, {  
 "actor" : "actor",  
 "actionType" : "actionType",  
 "objectActedOn" : "objectActedOn",  
 "created" : "2000-01-23T04:56:07.000+00:00",  
 "action" : "action",  
 "id" : "id",  
 "eventType" : "eventType",  
 "eventContext" : "eventContext"  
} ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/news/event-criteria-data

(getEventCriteriaData)

Returns the admin audit events criteria data to filter the event logs on the Home page.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, Object]

### Example data

Content-Type: application/json

{  
 "key" : { }  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

# AuthController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/auth/reauth

(continueSession)

Returns JWT token to continue session.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [LoginRequest](#LoginRequest) (required)

Body Parameter

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/auth/signin

(signIn)

Returns JWT, authenticates the user.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [LoginRequest](#LoginRequest) (required)

Body Parameter

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/auth/sso

(signInSso)

Returns JWT token, authenticates the user.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [LoginSsoRequest](#LoginSsoRequest) (required)

Body Parameter

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 403

Forbidden

#### 404

Not Found

# CertificateController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/certificates

(createTrustedCertificate)

Creates new trusted certificate and returns data of created trusted certificate.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [api\_certificates\_body](#api_certificates_body) (optional)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

name (required)

Query Parameter

comments (required)

Query Parameter

### Return type

[TrustedCertificate](#TrustedCertificate)

### Example data

Content-Type: application/json

{  
 "changerComments" : "changerComments",  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "changeID" : "changeID",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificate](#TrustedCertificate)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

delete /api/certificates/pending/{id}

(deletePendingCertificates)

Deletes pending change of trusted certificate by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[ChangeControlCert](#ChangeControlCert)

### Example data

Content-Type: application/json

{  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "trustedCertificateLog" : {  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "certificateLogId" : "certificateLogId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [ChangeControlCert](#ChangeControlCert)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

delete /api/certificates/{id}

(deleteTrustedCertificate)

Deletes trusted certificate by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

changerComments (optional)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates/{id}

(getCertificateById)

Returns trusted certificate data by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[TrustedCertificate](#TrustedCertificate)

### Example data

Content-Type: application/json

{  
 "changerComments" : "changerComments",  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "changeID" : "changeID",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificate](#TrustedCertificate)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates

(getCertificates)

Returns the page with persisted and pending trusted certificates filtered by partition matched name and thumbprint.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

certName (optional)

Query Parameter

thumbprint (optional)

Query Parameter

page (optional)

Query Parameter — default: 0 format: int32

size (optional)

Query Parameter — default: 10 format: int32

### Return type

[PageCertType](#PageCertType)

### Example data

Content-Type: application/json

{  
 "number" : 5,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 5,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "changerComments" : "changerComments",  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "changeID" : "changeID",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageCertType](#PageCertType)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates/pending/{id}

(getChangeControlCertificate)

Returns pending change control of trusted certificate by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[ChangeControlCert](#ChangeControlCert)

### Example data

Content-Type: application/json

{  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "trustedCertificateLog" : {  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "certificateLogId" : "certificateLogId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [ChangeControlCert](#ChangeControlCert)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates/pending

(getPendingCertificates)

Returns the page with pending trusted certificates filtered by partition matched name and thumbprint.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

certName (optional)

Query Parameter

thumbprint (optional)

Query Parameter

page (optional)

Query Parameter — default: 0 format: int32

size (optional)

Query Parameter — default: 10 format: int32

### Return type

[PageChangeControlCert](#PageChangeControlCert)

### Example data

Content-Type: application/json

{  
 "number" : 5,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 5,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "trustedCertificateLog" : {  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "certificateLogId" : "certificateLogId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
 }, {  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "trustedCertificateLog" : {  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "certificateLogId" : "certificateLogId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageChangeControlCert](#PageChangeControlCert)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates/existence

(isExistingCertificateName)

Checks if trusted certificate name exists.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

name (required)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/certificates/pending

(postPendingCertificates)

Approves or rejects trusted certificate

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [map](#map) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[TrustedCertificate](#TrustedCertificate)

### Example data

Content-Type: application/json

{  
 "changerComments" : "changerComments",  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "changeID" : "changeID",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificate](#TrustedCertificate)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

put /api/certificates/pending/{id}

(updatePendingCertificate)

Updates pending change of trusted certificate

### Path parameters

id (required)

Path Parameter

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [map](#map) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[TrustedCertificate](#TrustedCertificate)

### Example data

Content-Type: application/json

{  
 "changerComments" : "changerComments",  
 "serialNumber" : "serialNumber",  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "certificateId" : "certificateId",  
 "thumbprint" : "thumbprint",  
 "certificateName" : "certificateName",  
 "changeID" : "changeID",  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificate](#TrustedCertificate)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/certificates/upload

(uploadFile)

Returns trusted certificate data with validation of uploaded certificate file.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [certificates\_upload\_body](#certificates_upload_body) (optional)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[TrustedCertificateDetails](#TrustedCertificateDetails)

### Example data

Content-Type: application/json

{  
 "valid" : true,  
 "serialNumber" : "serialNumber",  
 "certificateErrors" : [ {  
 "key" : [ "certificateErrors", "certificateErrors" ]  
 }, {  
 "key" : [ "certificateErrors", "certificateErrors" ]  
 } ],  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "thumbprint" : "thumbprint",  
 "certificateWarnings" : [ {  
 "key" : { }  
 }, {  
 "key" : { }  
 } ],  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificateDetails](#TrustedCertificateDetails)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates/validate/{id}

(validateCertificateById)

Returns trusted certificate validation data by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[TrustedCertificateDetails](#TrustedCertificateDetails)

### Example data

Content-Type: application/json

{  
 "valid" : true,  
 "serialNumber" : "serialNumber",  
 "certificateErrors" : [ {  
 "key" : [ "certificateErrors", "certificateErrors" ]  
 }, {  
 "key" : [ "certificateErrors", "certificateErrors" ]  
 } ],  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "thumbprint" : "thumbprint",  
 "certificateWarnings" : [ {  
 "key" : { }  
 }, {  
 "key" : { }  
 } ],  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificateDetails](#TrustedCertificateDetails)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/certificates/validate/pending/{id}

(validatePendingCertificateById)

Returns pending change of trusted certificate validation data by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[TrustedCertificateDetails](#TrustedCertificateDetails)

### Example data

Content-Type: application/json

{  
 "valid" : true,  
 "serialNumber" : "serialNumber",  
 "certificateErrors" : [ {  
 "key" : [ "certificateErrors", "certificateErrors" ]  
 }, {  
 "key" : [ "certificateErrors", "certificateErrors" ]  
 } ],  
 "endDate" : "endDate",  
 "subject" : {  
 "key" : [ "subject", "subject" ]  
 },  
 "thumbprint" : "thumbprint",  
 "certificateWarnings" : [ {  
 "key" : { }  
 }, {  
 "key" : { }  
 } ],  
 "thumbprint256" : "thumbprint256",  
 "startDate" : "startDate",  
 "issuer" : {  
 "key" : [ "issuer", "issuer" ]  
 },  
 "authChainReport" : [ {  
 "key" : "authChainReport"  
 }, {  
 "key" : "authChainReport"  
 } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [TrustedCertificateDetails](#TrustedCertificateDetails)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

# EntityController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/entities

(createEntity)

Creates new entity and returns data of created entity.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [Entity](#Entity) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[Entity](#Entity)

### Example data

Content-Type: application/json

{  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "changerComments" : "changerComments",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "changeID" : "changeID",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "mailboxPathIn" : "mailboxPathIn",  
 "snF" : true,  
 "mqHeader" : "mqHeader",  
 "routingRules" : {  
 "response" : [ {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 }, {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 } ],  
 "warnings" : [ {  
 "key" : "warnings"  
 }, {  
 "key" : "warnings"  
 } ],  
 "errors" : [ {  
 "key" : "errors"  
 }, {  
 "key" : "errors"  
 } ]  
 },  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "mqChannel" : "mqChannel",  
 "deleted" : true,  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Entity](#Entity)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

delete /api/entities/{id}

(deleteEntity)

Deletes entity by ID.

### Path parameters

id (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

changerComments (optional)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

delete /api/entities/pending/{id}

(deletePendingEntity)

Deletes pending change of entity by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/changeControl/{id}

(getChangeControlById)

Returns JWT token to continue session.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[ChangeControl](#ChangeControl)

### Example data

Content-Type: application/json

{  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "entityLog" : {  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "entityLogId" : "entityLogId",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "inboundRoutingRule" : true,  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "inboundDir" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "snF" : true,  
 "mailboxPathIn" : "mailboxPathIn",  
 "mqHeader" : "mqHeader",  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "deleted" : true,  
 "mqChannel" : "mqChannel",  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [ChangeControl](#ChangeControl)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities

(getEntities)

Returns the page with persisted and pending entities filtered by partition matched name and thumbprint.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

entity (optional)

Query Parameter

service (optional)

Query Parameter

swiftDN (optional)

Query Parameter

page (optional)

Query Parameter — default: 0 format: int32

size (optional)

Query Parameter — default: 10 format: int32

### Return type

[PageEntityType](#PageEntityType)

### Example data

Content-Type: application/json

{  
 "number" : 5,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 5,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "changerComments" : "changerComments",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "changeID" : "changeID",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "mailboxPathIn" : "mailboxPathIn",  
 "snF" : true,  
 "mqHeader" : "mqHeader",  
 "routingRules" : {  
 "response" : [ {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 }, {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 } ],  
 "warnings" : [ {  
 "key" : "warnings"  
 }, {  
 "key" : "warnings"  
 } ],  
 "errors" : [ {  
 "key" : "errors"  
 }, {  
 "key" : "errors"  
 } ]  
 },  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "mqChannel" : "mqChannel",  
 "deleted" : true,  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageEntityType](#PageEntityType)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/{id}

(getEntityById)

Returns entity data by ID.

### Path parameters

id (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[Entity](#Entity)

### Example data

Content-Type: application/json

{  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "changerComments" : "changerComments",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "changeID" : "changeID",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "mailboxPathIn" : "mailboxPathIn",  
 "snF" : true,  
 "mqHeader" : "mqHeader",  
 "routingRules" : {  
 "response" : [ {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 }, {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 } ],  
 "warnings" : [ {  
 "key" : "warnings"  
 }, {  
 "key" : "warnings"  
 } ],  
 "errors" : [ {  
 "key" : "errors"  
 }, {  
 "key" : "errors"  
 } ]  
 },  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "mqChannel" : "mqChannel",  
 "deleted" : true,  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Entity](#Entity)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/file-type

(getFileType)

Return list of file type strings for entity schedules.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

array[String]

### Example data

Content-Type: application/json

[ "", "" ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/inbound-request-type

(getInboundRequestType)

Return map of entity inbound request types.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, String]

### Example data

Content-Type: application/json

{  
 "key" : ""  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/inbound-service

(getInboundService)

Return inbound service of entity.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/mq-details

(getMqDetails)

Return map of entity mq-details.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, array[String]]

### Example data

Content-Type: application/json

{  
 "key" : [ "", "" ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/participants

(getParticipants)

Returns list of entity participants by entity id.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

id (optional)

Query Parameter — format: int32

### Return type

array[String]

### Example data

Content-Type: application/json

[ "", "" ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/pending

(getPendingEntities)

Returns the page with pending entities filtered by partition matched name and thumbprint.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

entity (optional)

Query Parameter

service (optional)

Query Parameter

swiftDN (optional)

Query Parameter

page (optional)

Query Parameter — default: 0 format: int32

size (optional)

Query Parameter — default: 10 format: int32

### Return type

[PageChangeControl](#PageChangeControl)

### Example data

Content-Type: application/json

{  
 "number" : 5,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 5,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "entityLog" : {  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "entityLogId" : "entityLogId",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "inboundRoutingRule" : true,  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "inboundDir" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "snF" : true,  
 "mailboxPathIn" : "mailboxPathIn",  
 "mqHeader" : "mqHeader",  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "deleted" : true,  
 "mqChannel" : "mqChannel",  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
 }, {  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "entityLog" : {  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "entityLogId" : "entityLogId",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "inboundRoutingRule" : true,  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "inboundDir" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "snF" : true,  
 "mailboxPathIn" : "mailboxPathIn",  
 "mqHeader" : "mqHeader",  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "deleted" : true,  
 "mqChannel" : "mqChannel",  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageChangeControl](#PageChangeControl)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/pending/{id}

(getPendingEntityById)

Returns pending change control of entity by ID.

### Path parameters

id (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[Entity](#Entity)

### Example data

Content-Type: application/json

{  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "changerComments" : "changerComments",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "changeID" : "changeID",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "mailboxPathIn" : "mailboxPathIn",  
 "snF" : true,  
 "mqHeader" : "mqHeader",  
 "routingRules" : {  
 "response" : [ {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 }, {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 } ],  
 "warnings" : [ {  
 "key" : "warnings"  
 }, {  
 "key" : "warnings"  
 } ],  
 "errors" : [ {  
 "key" : "errors"  
 }, {  
 "key" : "errors"  
 } ]  
 },  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "mqChannel" : "mqChannel",  
 "deleted" : true,  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Entity](#Entity)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/existence/route-attributes

(getRequestType)

Checks if entity route attributes exist.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

inboundRequestorDN (required)

Query Parameter

inboundResponderDN (required)

Query Parameter

inboundService (required)

Query Parameter

inboundRequestType (required)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/swift-service

(getSwiftService)

Return swift service of entity.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

String

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [String](#String)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/existence/entity-service

(isExistingEntity)

Checks if entity name exists with selected service.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

service (required)

Query Parameter

entity (required)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/existence/mailbox

(isExistingMailboxPathOut)

Checks if entity mailboxPathOut exist.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

mailboxPathOut (required)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/entities/existence/queue

(isExistingMqQueueOut)

Checks if entity mqQueueOut exist.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

mqQueueOut (required)

Query Parameter

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/entities/pending

(postPendingEntities)

Approves or rejects entity pending change.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [map](#map) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[Entity](#Entity)

### Example data

Content-Type: application/json

{  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "changerComments" : "changerComments",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "changeID" : "changeID",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "mailboxPathIn" : "mailboxPathIn",  
 "snF" : true,  
 "mqHeader" : "mqHeader",  
 "routingRules" : {  
 "response" : [ {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 }, {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 } ],  
 "warnings" : [ {  
 "key" : "warnings"  
 }, {  
 "key" : "warnings"  
 } ],  
 "errors" : [ {  
 "key" : "errors"  
 }, {  
 "key" : "errors"  
 } ]  
 },  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "mqChannel" : "mqChannel",  
 "deleted" : true,  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Entity](#Entity)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/entities/transmit

(transmit)

Runs transmit action of entity.

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [Transmittal](#Transmittal) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, Object]

### Example data

Content-Type: application/json

{  
 "key" : { }  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

put /api/entities/{id}

(updateEntity)

Updates entity by id.

### Path parameters

id (required)

Path Parameter — format: int32

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [Entity](#Entity) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[Entity](#Entity)

### Example data

Content-Type: application/json

{  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "changerComments" : "changerComments",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "changeID" : "changeID",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "mailboxPathIn" : "mailboxPathIn",  
 "snF" : true,  
 "mqHeader" : "mqHeader",  
 "routingRules" : {  
 "response" : [ {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 }, {  
 "updateAction" : "updateAction",  
 "routingRuleName" : "routingRuleName",  
 "code" : 2,  
 "failCause" : "failCause"  
 } ],  
 "warnings" : [ {  
 "key" : "warnings"  
 }, {  
 "key" : "warnings"  
 } ],  
 "errors" : [ {  
 "key" : "errors"  
 }, {  
 "key" : "errors"  
 } ]  
 },  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "mqChannel" : "mqChannel",  
 "deleted" : true,  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Entity](#Entity)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

put /api/entities/pending/{id}

(updatePendingEntity)

Updates pending change of entity.

### Path parameters

id (required)

Path Parameter

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [Entity](#Entity) (required)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[ChangeControl](#ChangeControl)

### Example data

Content-Type: application/json

{  
 "dateChanged" : "2000-01-23T04:56:07.000+00:00",  
 "approver" : "approver",  
 "changerComments" : "changerComments",  
 "approverComments" : "approverComments",  
 "pending" : true,  
 "resultMeta2" : "resultMeta2",  
 "changer" : "changer",  
 "resultMeta3" : "resultMeta3",  
 "changeID" : "changeID",  
 "objectType" : "objectType",  
 "dateApproved" : "2000-01-23T04:56:07.000+00:00",  
 "resultMeta1" : "resultMeta1",  
 "statusText" : "statusText",  
 "entityLog" : {  
 "psrWTOMsgId" : "psrWTOMsgId",  
 "entityLogId" : "entityLogId",  
 "fileDesc" : "fileDesc",  
 "transferInfo" : "transferInfo",  
 "inboundRoutingRule" : true,  
 "irishStep2" : true,  
 "e2eSigning" : "e2eSigning",  
 "deliveryNotification" : true,  
 "cdNode" : "cdNode",  
 "deliveryNotifRT" : "deliveryNotifRT",  
 "cdfWTOMsgId" : "cdfWTOMsgId",  
 "mqSSLOptions" : "mqSSLOptions",  
 "mqPort" : 5,  
 "mqQueueIn" : "mqQueueIn",  
 "idfWTOMsgId" : "idfWTOMsgId",  
 "rsfWTOMsgId" : "rsfWTOMsgId",  
 "mqSessionTimeout" : 5,  
 "mqQueueBinding" : "mqQueueBinding",  
 "mqQueueContext" : "mqQueueContext",  
 "nonRepudiation" : true,  
 "inboundDir" : true,  
 "requestorDN" : "requestorDN",  
 "entityParticipantType" : "entityParticipantType",  
 "inboundService" : "inboundService",  
 "transferDesc" : "transferDesc",  
 "mqQueueOut" : "mqQueueOut",  
 "mqSSLCaCert" : "mqSSLCaCert",  
 "requestRef" : "requestRef",  
 "routeOutbound" : true,  
 "mqHost" : "mqHost",  
 "sdfWTOMsgId" : "sdfWTOMsgId",  
 "compression" : true,  
 "directParticipant" : "directParticipant",  
 "msrWTOMsgId" : "msrWTOMsgId",  
 "mqSSLKeyCert" : "mqSSLKeyCert",  
 "responderDN" : "responderDN",  
 "inboundResponderDN" : "inboundResponderDN",  
 "mbpWTOMsgId" : "mbpWTOMsgId",  
 "mqDebug" : "mqDebug",  
 "snF" : true,  
 "mailboxPathIn" : "mailboxPathIn",  
 "mqHeader" : "mqHeader",  
 "startOfDay" : "startOfDay",  
 "trace" : true,  
 "routeInbound" : true,  
 "schedules" : [ {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "isWindow" : true,  
 "timeStart" : "timeStart",  
 "lastRun" : "2000-01-23T04:56:07.000+00:00",  
 "windowInterval" : "windowInterval",  
 "active" : true,  
 "windowEnd" : "windowEnd",  
 "transThreshold" : 9,  
 "scheduleId" : 7,  
 "fileType" : "fileType",  
 "nextRun" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "maxTransfersPerBulk" : 6,  
 "rtfWTOMsgId" : "rtfWTOMsgId",  
 "mqSSLCiphers" : "mqSSLCiphers",  
 "dvfWTOMsgId" : "dvfWTOMsgId",  
 "pauseInbound" : true,  
 "dnfWTOMsgId" : "dnfWTOMsgId",  
 "mailboxPathOut" : "mailboxPathOut",  
 "endOfDay" : "endOfDay",  
 "inboundRequestorDN" : "inboundRequestorDN",  
 "pauseOutbound" : true,  
 "inboundRequestType" : [ "inboundRequestType", "inboundRequestType" ],  
 "requestType" : "requestType",  
 "fileInfo" : "fileInfo",  
 "entityId" : 0,  
 "serviceName" : "serviceName",  
 "deliveryNotifDN" : "deliveryNotifDN",  
 "deleted" : true,  
 "mqChannel" : "mqChannel",  
 "mqQueueName" : "mqQueueName",  
 "service" : "service",  
 "maxBulksPerFile" : 1,  
 "drrWTOMsgId" : "drrWTOMsgId",  
 "mqQManager" : "mqQManager",  
 "entity" : "entity"  
 },  
 "operation" : "CREATE",  
 "status" : "PENDING"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [ChangeControl](#ChangeControl)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

# FileSearchController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/sepa/export-report

(exportFilesReport)

Returns PDF/EXCEL reports of files selected with datetime period.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

from (optional)

Query Parameter

to (optional)

Query Parameter

type (required)

Query Parameter

### Return type

byte[]

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [byte[]](#byte%5B%5D)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/{fileId}/transactions/export-report

(exportTransactionsReport)

Returns PDF/EXCEL reports of Transactions selected with datetime period.

### Path parameters

fileId (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

fileName (required)

Query Parameter

size (optional)

Query Parameter — default: 100000 format: int32

type (required)

Query Parameter

### Return type

byte[]

### Example data

Content-Type: application/json

""

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [byte[]](#byte%5B%5D)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/document-content

(getDocumentContent1)

Returns Document Content of some file by filename

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

id (optional)

Query Parameter

messageId (optional)

Query Parameter — format: int32

### Return type

map[String, String]

### Example data

Content-Type: application/json

{  
 "key" : ""  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/error/{errorCode}

(getErrorDetailsByCode)

Returns File Error info by error code

### Path parameters

errorCode (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[ErrorDetail](#ErrorDetail)

### Example data

Content-Type: application/json

{  
 "code" : "code",  
 "name" : "name",  
 "description" : "description"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [ErrorDetail](#ErrorDetail)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/{id}

(getFileById)

Returns File info by File ID

### Path parameters

id (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[File](#File)

### Example data

Content-Type: application/json

{  
 "settleAmountTotal" : 2.3021358869347655,  
 "docID" : "docID",  
 "messageID" : 7,  
 "errorCode" : "errorCode",  
 "type" : "type",  
 "reference" : "reference",  
 "filename" : "filename",  
 "service" : "service",  
 "id" : 5,  
 "transactionTotal" : 5,  
 "workflowID" : 3,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00",  
 "status" : 9  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [File](#File)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/file-criteria-data

(getFileCriteriaData)

Returns File criteria data for searching files.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

service (optional)

Query Parameter

outbound (optional)

Query Parameter

### Return type

map[String, array[Object]]

### Example data

Content-Type: application/json

{  
 "key" : [ { }, { } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/file-monitor

(getFileMonitor)

Returns List of Files with last 10 files

### Request headers

Authorization (required)

Bearer accessToken

### Return type

array[[File](#File)]

### Example data

Content-Type: application/json

[ {  
 "settleAmountTotal" : 2.3021358869347655,  
 "docID" : "docID",  
 "messageID" : 7,  
 "errorCode" : "errorCode",  
 "type" : "type",  
 "reference" : "reference",  
 "filename" : "filename",  
 "service" : "service",  
 "id" : 5,  
 "transactionTotal" : 5,  
 "workflowID" : 3,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00",  
 "status" : 9  
}, {  
 "settleAmountTotal" : 2.3021358869347655,  
 "docID" : "docID",  
 "messageID" : 7,  
 "errorCode" : "errorCode",  
 "type" : "type",  
 "reference" : "reference",  
 "filename" : "filename",  
 "service" : "service",  
 "id" : 5,  
 "transactionTotal" : 5,  
 "workflowID" : 3,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00",  
 "status" : 9  
} ]

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/files

(getFiles)

Returns the page with Files filtered by search fields

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [FileSearchCriteria](#FileSearchCriteria) (optional)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[PageFile](#PageFile)

### Example data

Content-Type: application/json

{  
 "number" : 2,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 4,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "settleAmountTotal" : 2.3021358869347655,  
 "docID" : "docID",  
 "messageID" : 7,  
 "errorCode" : "errorCode",  
 "type" : "type",  
 "reference" : "reference",  
 "filename" : "filename",  
 "service" : "service",  
 "id" : 5,  
 "transactionTotal" : 5,  
 "workflowID" : 3,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00",  
 "status" : 9  
 }, {  
 "settleAmountTotal" : 2.3021358869347655,  
 "docID" : "docID",  
 "messageID" : 7,  
 "errorCode" : "errorCode",  
 "type" : "type",  
 "reference" : "reference",  
 "filename" : "filename",  
 "service" : "service",  
 "id" : 5,  
 "transactionTotal" : 5,  
 "workflowID" : 3,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00",  
 "status" : 9  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageFile](#PageFile)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/files/sepa

(getSEPAFiles)

Returns the page with SEPA Files filtered by search fields

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [FileSearchCriteria](#FileSearchCriteria) (optional)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[PageSEPAFile](#PageSEPAFile)

### Example data

Content-Type: application/json

{  
 "number" : 9,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 3,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "settleAmountTotal" : 2.3021358869347655,  
 "filename" : "filename",  
 "docID" : "docID",  
 "messageID" : 7,  
 "id" : 5,  
 "type" : "type",  
 "transactionTotal" : 5,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "settleAmountTotal" : 2.3021358869347655,  
 "filename" : "filename",  
 "docID" : "docID",  
 "messageID" : 7,  
 "id" : 5,  
 "type" : "type",  
 "transactionTotal" : 5,  
 "direction" : "direction",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageSEPAFile](#PageSEPAFile)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/files/{fileId}/transactions

(getTransactionsForFiles)

Returns the page with SCT Transaction by File ID.

### Path parameters

fileId (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

page (optional)

Query Parameter — default: 0 format: int32

size (optional)

Query Parameter — default: 10 format: int32

### Return type

[PageTransaction](#PageTransaction)

### Example data

Content-Type: application/json

{  
 "number" : 9,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 7,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "payaway" : true,  
 "docID" : "docID",  
 "settleAmount" : 2.3021358869347655,  
 "settleDate" : "2000-01-23T04:56:07.000+00:00",  
 "id" : 5,  
 "type" : "type",  
 "transactionID" : "transactionID",  
 "workflowID" : 7,  
 "status" : 5,  
 "direction" : "direction",  
 "fileID" : "fileID",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "payaway" : true,  
 "docID" : "docID",  
 "settleAmount" : 2.3021358869347655,  
 "settleDate" : "2000-01-23T04:56:07.000+00:00",  
 "id" : 5,  
 "type" : "type",  
 "transactionID" : "transactionID",  
 "workflowID" : 7,  
 "status" : 5,  
 "direction" : "direction",  
 "fileID" : "fileID",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageTransaction](#PageTransaction)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

# SctTransactionSearchController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

post /api/transactions

(getSCTTransactions)

Returns the page with SCT Transactions filtered by search fields

### Consumes

This API call consumes the following media types via the Content-Type request header:

* application/json

### Request body

body [TransactionSearchCriteria](#TransactionSearchCriteria) (optional)

Body Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[PageTransaction](#PageTransaction)

### Example data

Content-Type: application/json

{  
 "number" : 9,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 7,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "payaway" : true,  
 "docID" : "docID",  
 "settleAmount" : 2.3021358869347655,  
 "settleDate" : "2000-01-23T04:56:07.000+00:00",  
 "id" : 5,  
 "type" : "type",  
 "transactionID" : "transactionID",  
 "workflowID" : 7,  
 "status" : 5,  
 "direction" : "direction",  
 "fileID" : "fileID",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
 }, {  
 "payaway" : true,  
 "docID" : "docID",  
 "settleAmount" : 2.3021358869347655,  
 "settleDate" : "2000-01-23T04:56:07.000+00:00",  
 "id" : 5,  
 "type" : "type",  
 "transactionID" : "transactionID",  
 "workflowID" : 7,  
 "status" : 5,  
 "direction" : "direction",  
 "fileID" : "fileID",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageTransaction](#PageTransaction)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/transactions/{id}

(getTransactionById)

Returns SCT Transaction data by ID.

### Path parameters

id (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[Transaction](#Transaction)

### Example data

Content-Type: application/json

{  
 "payaway" : true,  
 "docID" : "docID",  
 "settleAmount" : 2.3021358869347655,  
 "settleDate" : "2000-01-23T04:56:07.000+00:00",  
 "id" : 5,  
 "type" : "type",  
 "transactionID" : "transactionID",  
 "workflowID" : 7,  
 "status" : 5,  
 "direction" : "direction",  
 "fileID" : "fileID",  
 "timestamp" : "2000-01-23T04:56:07.000+00:00"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Transaction](#Transaction)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/transactions/transaction-criteria-data

(getTransactionCriteriaData)

Returns SCT transaction criteria data for searching sct transactions.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

direction (optional)

Query Parameter

### Return type

map[String, array[Object]]

### Example data

Content-Type: application/json

{  
 "key" : [ { }, { } ]  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

# StatisticsController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/statistics/sct-alerts

(getSCTAlerts)

Returns SCT Alerts info for the Home page.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, Integer]

### Example data

Content-Type: application/json

{  
 "key" : 0  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/statistics/sct-traffic

(getSCTTrafficSummary)

Returns SCT Traffic for the Home page.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, map[String, Integer]]

### Example data

Content-Type: application/json

{  
 "key" : {  
 "key" : 0  
 }  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/statistics/sepa-traffic

(getSepaTrafficSummary)

Returns SEPA Traffic with list of files for the past 1 hour.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, Integer]

### Example data

Content-Type: application/json

{  
 "key" : 0  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/statistics/system-errors

(getSystemErrorsStatistics)

Returns number of System Errors on the Home page.

### Request headers

Authorization (required)

Bearer accessToken

### Return type

[StatisticalData](#StatisticalData)

### Example data

Content-Type: application/json

{  
 "count" : 0,  
 "type" : "type"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [StatisticalData](#StatisticalData)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

# VersionController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/version

(getApplicationVersion)

Returns version of BFGUI Application, SEPA visibility property param and login description info

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/version/f5

(getF5Link)

Returns List of contacts for sharing details of F5 issue.

### Return type

Object

### Example data

Content-Type: application/json

{ }

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Object](#Object)

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/version/sepa-visibility

(isSepaDashboardVisible)

Returns SEPA visibility property param

### Return type

Boolean

### Example data

Content-Type: application/json

true

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Boolean](#Boolean)

#### 403

Forbidden

#### 404

Not Found

# WorkflowController

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/workflow/bp-details

(getBPDetails)

Returns Business Process details by Business Process header bpRef param.

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

identifier (required)

Query Parameter

### Return type

[BPDetails](#BPDetails)

### Example data

Content-Type: application/json

{  
 "persistenceLevel" : "persistenceLevel",  
 "expedite" : "expedite",  
 "secondNotificationMinutes" : "secondNotificationMinutes",  
 "description" : "description",  
 "secondNotificationHours" : "secondNotificationHours",  
 "documentStorage" : "documentStorage",  
 "firstNotificationHours" : "firstNotificationHours",  
 "recoveryLevel" : "recoveryLevel",  
 "deadlineHours" : "deadlineHours",  
 "firstNotificationMinutes" : "firstNotificationMinutes",  
 "eventReportingLevel" : "eventReportingLevel",  
 "softstopRecoveryLevel" : "softstopRecoveryLevel",  
 "wfdVersion" : "wfdVersion",  
 "lifespanDays" : "lifespanDays",  
 "deadlineMinutes" : "deadlineMinutes",  
 "onfaultProcessing" : "onfaultProcessing",  
 "node" : "node",  
 "businessProcess" : "businessProcess",  
 "lifespanHours" : "lifespanHours",  
 "startMode" : "startMode",  
 "nodePreference" : "nodePreference",  
 "name" : "name",  
 "enableTransaction" : "enableTransaction",  
 "documentTracking" : "documentTracking",  
 "removalMethod" : "removalMethod",  
 "queue" : "queue",  
 "lifespanType" : "lifespanType"  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [BPDetails](#BPDetails)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/workflow/bp-header

(getBPHeader)

Returns Business Process header by Workflow Id and Version

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

wfdVersion (required)

Query Parameter — format: int32

wfdID (required)

Query Parameter — format: int32

### Return type

map[String, String]

### Example data

Content-Type: application/json

{  
 "key" : ""  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/workflow/document-content

(getDocumentContent)

Returns Document Content of some workflow step by document id

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

id (required)

Query Parameter

### Return type

[Document](#Document)

### Example data

Content-Type: application/json

{  
 "storageType" : "storageType",  
 "documentPayload" : "documentPayload",  
 "documentId" : "documentId",  
 "documentName" : "documentName",  
 "workflowId" : 0  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [Document](#Document)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/workflow/wfcId/{wfcId}

(getWfIdAndVersion)

Returns Workflow ID with Version by Business Process name

### Path parameters

wfcId (required)

Path Parameter

### Request headers

Authorization (required)

Bearer accessToken

### Return type

map[String, Integer]

### Example data

Content-Type: application/json

{  
 "key" : 0  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

[Up](#X19a5892cf6363802f9e7770722aeedc8638de9d)

get /api/workflow/{id}/steps

(getWorkflowSteps)

Returns the page with Workflow steps by workflow ID

### Path parameters

id (required)

Path Parameter — format: int32

### Request headers

Authorization (required)

Bearer accessToken

### Query parameters

page (optional)

Query Parameter — default: 0 format: int32

size (optional)

Query Parameter — default: 10 format: int32

### Return type

[PageWorkflowStep](#PageWorkflowStep)

### Example data

Content-Type: application/json

{  
 "number" : 7,  
 "size" : 1,  
 "last" : true,  
 "numberOfElements" : 9,  
 "totalPages" : 0,  
 "pageable" : {  
 "paged" : true,  
 "pageNumber" : 2,  
 "offset" : 3,  
 "pageSize" : 4,  
 "unpaged" : true  
 },  
 "sort" : {  
 "unsorted" : true,  
 "sorted" : true,  
 "empty" : true  
 },  
 "content" : [ {  
 "docId" : "docId",  
 "inlineInvocation" : true,  
 "stepId" : 5,  
 "wfdVersion" : 2,  
 "serviceName" : "serviceName",  
 "wfdId" : 5,  
 "advStatus" : "advStatus",  
 "wfcId" : "wfcId",  
 "startTime" : "2000-01-23T04:56:07.000+00:00",  
 "endTime" : "2000-01-23T04:56:07.000+00:00",  
 "exeState" : "exeState",  
 "nodeExecuted" : "nodeExecuted",  
 "statusRpt" : "statusRpt"  
 }, {  
 "docId" : "docId",  
 "inlineInvocation" : true,  
 "stepId" : 5,  
 "wfdVersion" : 2,  
 "serviceName" : "serviceName",  
 "wfdId" : 5,  
 "advStatus" : "advStatus",  
 "wfcId" : "wfcId",  
 "startTime" : "2000-01-23T04:56:07.000+00:00",  
 "endTime" : "2000-01-23T04:56:07.000+00:00",  
 "exeState" : "exeState",  
 "nodeExecuted" : "nodeExecuted",  
 "statusRpt" : "statusRpt"  
 } ],  
 "first" : true,  
 "totalElements" : 6,  
 "empty" : true  
}

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

* \*/\*

### Responses

#### 200

OK [PageWorkflowStep](#PageWorkflowStep)

#### 401

Unauthorized

#### 403

Forbidden

#### 404

Not Found

## Models

[ Jump to [Methods](#X19a5892cf6363802f9e7770722aeedc8638de9d) ]

### Table of Contents

1. [AuditEvent](#AuditEvent)
2. [AuditEventCriteria](#AuditEventCriteria)
3. [BPDetails](#BPDetails)
4. [CertType](#CertType)
5. [ChangeControl](#ChangeControl)
6. [ChangeControlCert](#ChangeControlCert)
7. [Document](#Document)
8. [Entity](#Entity)
9. [EntityLog](#EntityLog)
10. [EntityType](#EntityType)
11. [ErrorDetail](#ErrorDetail)
12. [File](#File)
13. [FileSearchCriteria](#FileSearchCriteria)
14. [LoginRequest](#LoginRequest)
15. [LoginSsoRequest](#LoginSsoRequest)
16. [PageCertType](#PageCertType)
17. [PageChangeControl](#PageChangeControl)
18. [PageChangeControlCert](#PageChangeControlCert)
19. [PageEntityType](#PageEntityType)
20. [PageFile](#PageFile)
21. [PageSEPAFile](#PageSEPAFile)
22. [PageTransaction](#PageTransaction)
23. [PageWorkflowStep](#PageWorkflowStep)
24. [Pageable](#Pageable)
25. [SEPAFile](#SEPAFile)
26. [SWIFTNetRoutingRuleBfgUiRestResponse](#SWIFTNetRoutingRuleBfgUiRestResponse)
27. [SWIFTNetRoutingRuleServiceResponse](#SWIFTNetRoutingRuleServiceResponse)
28. [Schedule](#Schedule)
29. [Sort](#Sort)
30. [StatisticalData](#StatisticalData)
31. [Transaction](#Transaction)
32. [TransactionSearchCriteria](#TransactionSearchCriteria)
33. [Transmittal](#Transmittal)
34. [TrustedCertificate](#TrustedCertificate)
35. [TrustedCertificateDetails](#TrustedCertificateDetails)
36. [TrustedCertificateLog](#TrustedCertificateLog)
37. [WorkflowStep](#WorkflowStep)
38. [api\_certificates\_body](#api_certificates_body)
39. [certificates\_upload\_body](#certificates_upload_body)

### AuditEvent [Up](#X08a735672bda25f91812485a33b810177d042f8)

id (optional)

[String](#string)

actor (optional)

[String](#string)

actionType (optional)

[String](#string)

eventContext (optional)

[String](#string)

eventType (optional)

[String](#string)

objectActedOn (optional)

[String](#string)

created (optional)

[Date](#DateTime) format: date-time

action (optional)

[String](#string)

### AuditEventCriteria [Up](#X08a735672bda25f91812485a33b810177d042f8)

id (optional)

[String](#string)

size (optional)

[Integer](#integer) format: int32

action (optional)

[String](#string)

Enum:

approved

rejected

requested

request-edited

request-cancelled

event-type (optional)

[String](#string)

username (optional)

[String](#string)

object-acted-on (optional)

[String](#string)

action-type (optional)

[String](#string)

Enum:

create

update

delete

### BPDetails [Up](#X08a735672bda25f91812485a33b810177d042f8)

name (optional)

[String](#string)

startMode (optional)

[String](#string)

documentTracking (optional)

[String](#string)

queue (optional)

[String](#string)

recoveryLevel (optional)

[String](#string)

softstopRecoveryLevel (optional)

[String](#string)

lifespanDays (optional)

[String](#string)

lifespanHours (optional)

[String](#string)

lifespanType (optional)

[String](#string)

removalMethod (optional)

[String](#string)

eventReportingLevel (optional)

[String](#string)

wfdVersion (optional)

[String](#string)

onfaultProcessing (optional)

[String](#string)

enableTransaction (optional)

[String](#string)

persistenceLevel (optional)

[String](#string)

documentStorage (optional)

[String](#string)

expedite (optional)

[String](#string)

deadlineHours (optional)

[String](#string)

deadlineMinutes (optional)

[String](#string)

firstNotificationHours (optional)

[String](#string)

firstNotificationMinutes (optional)

[String](#string)

secondNotificationHours (optional)

[String](#string)

secondNotificationMinutes (optional)

[String](#string)

description (optional)

[String](#string)

node (optional)

[String](#string)

nodePreference (optional)

[String](#string)

businessProcess (optional)

[String](#string)

### CertType [Up](#X08a735672bda25f91812485a33b810177d042f8)

### ChangeControl [Up](#X08a735672bda25f91812485a33b810177d042f8)

changeID

[String](#string)

operation

[String](#string)

Enum:

CREATE

UPDATE

DELETE

status

[String](#string)

Enum:

PENDING

REJECTED

ACCEPTED

FAILED

UNKNOWN

changer

[String](#string)

dateChanged (optional)

[Date](#DateTime) format: date-time

approver (optional)

[String](#string)

dateApproved (optional)

[Date](#DateTime) format: date-time

changerComments (optional)

[String](#string)

approverComments (optional)

[String](#string)

resultMeta1 (optional)

[String](#string)

resultMeta2 (optional)

[String](#string)

resultMeta3 (optional)

[String](#string)

entityLog (optional)

[EntityLog](#EntityLog)

objectType (optional)

[String](#string)

pending (optional)

[Boolean](#boolean)

statusText (optional)

[String](#string)

### ChangeControlCert [Up](#X08a735672bda25f91812485a33b810177d042f8)

changeID

[String](#string)

operation

[String](#string)

Enum:

CREATE

UPDATE

DELETE

status

[String](#string)

Enum:

PENDING

REJECTED

ACCEPTED

FAILED

UNKNOWN

changer

[String](#string)

dateChanged (optional)

[Date](#DateTime) format: date-time

approver (optional)

[String](#string)

dateApproved (optional)

[Date](#DateTime) format: date-time

changerComments (optional)

[String](#string)

approverComments (optional)

[String](#string)

resultMeta1 (optional)

[String](#string)

resultMeta2 (optional)

[String](#string)

resultMeta3 (optional)

[String](#string)

trustedCertificateLog (optional)

[TrustedCertificateLog](#TrustedCertificateLog)

objectType (optional)

[String](#string)

pending (optional)

[Boolean](#boolean)

statusText (optional)

[String](#string)

### Document [Up](#X08a735672bda25f91812485a33b810177d042f8)

workflowId (optional)

[Integer](#integer) format: int32

documentName (optional)

[String](#string)

storageType (optional)

[String](#string)

documentId (optional)

[String](#string)

documentPayload (optional)

[String](#string)

### Entity [Up](#X08a735672bda25f91812485a33b810177d042f8)

entityId (optional)

[Integer](#integer) format: int32

entity

[String](#string)

service

[String](#string)

requestorDN

[String](#string)

responderDN

[String](#string)

serviceName

[String](#string)

requestType (optional)

[String](#string)

trace

[Boolean](#boolean)

deliveryNotification

[Boolean](#boolean)

deliveryNotifDN (optional)

[String](#string)

deliveryNotifRT (optional)

[String](#string)

requestRef (optional)

[String](#string)

fileDesc (optional)

[String](#string)

fileInfo (optional)

[String](#string)

transferDesc (optional)

[String](#string)

transferInfo (optional)

[String](#string)

compression

[Boolean](#boolean)

mailboxPathIn

[String](#string)

mailboxPathOut

[String](#string)

mqQueueIn (optional)

[String](#string)

mqQueueOut (optional)

[String](#string)

entityParticipantType (optional)

[String](#string)

directParticipant (optional)

[String](#string)

maxTransfersPerBulk

[Integer](#integer) format: int32

maxBulksPerFile

[Integer](#integer) format: int32

startOfDay

[String](#string)

endOfDay

[String](#string)

cdNode (optional)

[String](#string)

idfWTOMsgId (optional)

[String](#string)

cdfWTOMsgId (optional)

[String](#string)

sdfWTOMsgId (optional)

[String](#string)

rsfWTOMsgId (optional)

[String](#string)

dnfWTOMsgId (optional)

[String](#string)

dvfWTOMsgId (optional)

[String](#string)

msrWTOMsgId (optional)

[String](#string)

psrWTOMsgId (optional)

[String](#string)

drrWTOMsgId (optional)

[String](#string)

rtfWTOMsgId (optional)

[String](#string)

mbpWTOMsgId (optional)

[String](#string)

mqHost (optional)

[String](#string)

mqPort (optional)

[Integer](#integer) format: int32

mqQManager (optional)

[String](#string)

mqChannel (optional)

[String](#string)

mqQueueName (optional)

[String](#string)

mqQueueBinding (optional)

[String](#string)

mqQueueContext (optional)

[String](#string)

mqDebug (optional)

[String](#string)

mqSSLOptions (optional)

[String](#string)

mqSSLCiphers (optional)

[String](#string)

mqSSLKeyCert (optional)

[String](#string)

mqSSLCaCert (optional)

[String](#string)

mqHeader (optional)

[String](#string)

mqSessionTimeout (optional)

[Integer](#integer) format: int32

routeOutbound (optional)

[Boolean](#boolean)

routingRules (optional)

[SWIFTNetRoutingRuleServiceResponse](#SWIFTNetRoutingRuleServiceResponse)

routeInbound (optional)

[Boolean](#boolean)

inboundRequestorDN (optional)

[String](#string)

inboundResponderDN (optional)

[String](#string)

inboundService (optional)

[String](#string)

inboundRequestType (optional)

[array[String]](#string)

nonRepudiation

[Boolean](#boolean)

pauseInbound

[Boolean](#boolean)

pauseOutbound

[Boolean](#boolean)

deleted (optional)

[Boolean](#boolean)

changeID (optional)

[String](#string)

changerComments (optional)

[String](#string)

irishStep2 (optional)

[Boolean](#boolean)

e2eSigning (optional)

[String](#string)

schedules (optional)

[array[Schedule]](#Schedule)

snF (optional)

[Boolean](#boolean)

### EntityLog [Up](#X08a735672bda25f91812485a33b810177d042f8)

entityLogId (optional)

[String](#string)

entityId (optional)

[Integer](#integer) format: int32

entity

[String](#string)

service

[String](#string)

mailboxPathOut

[String](#string)

mqQueueOut (optional)

[String](#string)

routeInbound (optional)

[Boolean](#boolean)

routeOutbound (optional)

[Boolean](#boolean)

inboundDir (optional)

[Boolean](#boolean)

inboundRoutingRule (optional)

[Boolean](#boolean)

deleted (optional)

[Boolean](#boolean)

requestorDN (optional)

[String](#string)

responderDN (optional)

[String](#string)

serviceName (optional)

[String](#string)

requestType (optional)

[String](#string)

snF

[Boolean](#boolean)

trace

[Boolean](#boolean)

deliveryNotification

[Boolean](#boolean)

deliveryNotifDN (optional)

[String](#string)

deliveryNotifRT (optional)

[String](#string)

requestRef (optional)

[String](#string)

fileDesc (optional)

[String](#string)

fileInfo (optional)

[String](#string)

transferDesc (optional)

[String](#string)

transferInfo (optional)

[String](#string)

compression

[Boolean](#boolean)

mailboxPathIn

[String](#string)

mqQueueIn (optional)

[String](#string)

entityParticipantType (optional)

[String](#string)

directParticipant (optional)

[String](#string)

maxTransfersPerBulk

[Integer](#integer) format: int32

maxBulksPerFile

[Integer](#integer) format: int32

startOfDay

[String](#string)

endOfDay

[String](#string)

cdNode (optional)

[String](#string)

idfWTOMsgId (optional)

[String](#string)

cdfWTOMsgId (optional)

[String](#string)

sdfWTOMsgId (optional)

[String](#string)

rsfWTOMsgId (optional)

[String](#string)

dnfWTOMsgId (optional)

[String](#string)

dvfWTOMsgId (optional)

[String](#string)

msrWTOMsgId (optional)

[String](#string)

psrWTOMsgId (optional)

[String](#string)

drrWTOMsgId (optional)

[String](#string)

rtfWTOMsgId (optional)

[String](#string)

mbpWTOMsgId (optional)

[String](#string)

mqHost (optional)

[String](#string)

mqPort (optional)

[Integer](#integer) format: int32

mqQManager (optional)

[String](#string)

mqChannel (optional)

[String](#string)

mqQueueName (optional)

[String](#string)

mqQueueBinding (optional)

[String](#string)

mqQueueContext (optional)

[String](#string)

mqDebug (optional)

[String](#string)

mqSSLOptions (optional)

[String](#string)

mqSSLCiphers (optional)

[String](#string)

mqSSLKeyCert (optional)

[String](#string)

mqSSLCaCert (optional)

[String](#string)

mqHeader (optional)

[String](#string)

mqSessionTimeout (optional)

[Integer](#integer) format: int32

inboundRequestorDN (optional)

[String](#string)

inboundResponderDN (optional)

[String](#string)

inboundService (optional)

[String](#string)

nonRepudiation

[Boolean](#boolean)

pauseInbound

[Boolean](#boolean)

pauseOutbound

[Boolean](#boolean)

e2eSigning (optional)

[String](#string)

inboundRequestType (optional)

[array[String]](#string)

irishStep2 (optional)

[Boolean](#boolean)

schedules (optional)

[array[Schedule]](#Schedule)

### EntityType [Up](#X08a735672bda25f91812485a33b810177d042f8)

### ErrorDetail [Up](#X08a735672bda25f91812485a33b810177d042f8)

code (optional)

[String](#string)

name (optional)

[String](#string)

description (optional)

[String](#string)

### File [Up](#X08a735672bda25f91812485a33b810177d042f8)

id (optional)

[Integer](#integer) format: int32

filename (optional)

[String](#string)

type (optional)

[String](#string)

docID (optional)

[String](#string)

transactionTotal (optional)

[Integer](#integer) format: int32

direction (optional)

[String](#string)

settleAmountTotal (optional)

[Double](#double) format: double

timestamp (optional)

[Date](#DateTime) format: date-time

messageID (optional)

[Integer](#integer) format: int32

status (optional)

[Integer](#integer) format: int32

errorCode (optional)

[String](#string)

reference (optional)

[String](#string)

service (optional)

[String](#string)

workflowID (optional)

[Integer](#integer) format: int32

### FileSearchCriteria [Up](#X08a735672bda25f91812485a33b810177d042f8)

type (optional)

[String](#string)

direction (optional)

[String](#string)

service (optional)

[String](#string)

reference (optional)

[String](#string)

wfid (optional)

[Integer](#integer) format: int32

status (optional)

[Integer](#integer) format: int32

page (optional)

[Integer](#integer) format: int32

start (optional)

[Integer](#integer) format: int32

id (optional)

[Integer](#integer) format: int32

messageid (optional)

[Long](#long) format: int64

entityid (optional)

[Integer](#integer) format: int32

override (optional)

[Boolean](#boolean)

filename (optional)

[String](#string)

from (optional)

[String](#string)

to (optional)

[String](#string)

rows (optional)

[Integer](#integer) format: int32

is-error (optional)

[Boolean](#boolean)

exclude-archive (optional)

[Boolean](#boolean)

bp-state (optional)

[String](#string)

### LoginRequest [Up](#X08a735672bda25f91812485a33b810177d042f8)

login (optional)

[String](#string)

password (optional)

[String](#string)

### LoginSsoRequest [Up](#X08a735672bda25f91812485a33b810177d042f8)

login (optional)

[String](#string)

nodeName (optional)

[String](#string)

dlssoToken (optional)

[String](#string)

### PageCertType [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[CertType]](#CertType)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageChangeControl [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[ChangeControl]](#ChangeControl)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageChangeControlCert [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[ChangeControlCert]](#ChangeControlCert)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageEntityType [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[EntityType]](#EntityType)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageFile [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[File]](#File)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageSEPAFile [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[SEPAFile]](#SEPAFile)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageTransaction [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[Transaction]](#Transaction)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### PageWorkflowStep [Up](#X08a735672bda25f91812485a33b810177d042f8)

totalPages (optional)

[Integer](#integer) format: int32

totalElements (optional)

[Long](#long) format: int64

size (optional)

[Integer](#integer) format: int32

content (optional)

[array[WorkflowStep]](#WorkflowStep)

number (optional)

[Integer](#integer) format: int32

first (optional)

[Boolean](#boolean)

sort (optional)

[Sort](#Sort)

pageable (optional)

[Pageable](#Pageable)

last (optional)

[Boolean](#boolean)

numberOfElements (optional)

[Integer](#integer) format: int32

empty (optional)

[Boolean](#boolean)

### Pageable [Up](#X08a735672bda25f91812485a33b810177d042f8)

offset (optional)

[Long](#long) format: int64

sort (optional)

[Sort](#Sort)

paged (optional)

[Boolean](#boolean)

unpaged (optional)

[Boolean](#boolean)

pageNumber (optional)

[Integer](#integer) format: int32

pageSize (optional)

[Integer](#integer) format: int32

### SEPAFile [Up](#X08a735672bda25f91812485a33b810177d042f8)

id (optional)

[Integer](#integer) format: int32

filename (optional)

[String](#string)

type (optional)

[String](#string)

docID (optional)

[String](#string)

transactionTotal (optional)

[Integer](#integer) format: int32

direction (optional)

[String](#string)

settleAmountTotal (optional)

[Double](#double) format: double

timestamp (optional)

[Date](#DateTime) format: date-time

messageID (optional)

[Integer](#integer) format: int32

### SWIFTNetRoutingRuleBfgUiRestResponse [Up](#X08a735672bda25f91812485a33b810177d042f8)

routingRuleName (optional)

[String](#string)

updateAction (optional)

[String](#string)

failCause (optional)

[String](#string)

code (optional)

[Integer](#integer) format: int32

### SWIFTNetRoutingRuleServiceResponse [Up](#X08a735672bda25f91812485a33b810177d042f8)

response (optional)

[array[SWIFTNetRoutingRuleBfgUiRestResponse]](#SWIFTNetRoutingRuleBfgUiRestResponse)

errors (optional)

[array[map[String, String]]](#map)

warnings (optional)

[array[map[String, String]]](#map)

### Schedule [Up](#X08a735672bda25f91812485a33b810177d042f8)

scheduleId (optional)

[Long](#long) format: int64

entity (optional)

[Entity](#Entity)

isWindow (optional)

[Boolean](#boolean)

timeStart (optional)

[String](#string)

windowEnd (optional)

[String](#string)

windowInterval (optional)

[String](#string)

transThreshold (optional)

[Integer](#integer) format: int32

active

[Boolean](#boolean)

nextRun (optional)

[Date](#DateTime) format: date-time

lastRun (optional)

[Date](#DateTime) format: date-time

fileType

[String](#string)

### Sort [Up](#X08a735672bda25f91812485a33b810177d042f8)

unsorted (optional)

[Boolean](#boolean)

sorted (optional)

[Boolean](#boolean)

empty (optional)

[Boolean](#boolean)

### StatisticalData [Up](#X08a735672bda25f91812485a33b810177d042f8)

count (optional)

[Integer](#integer) format: int32

type (optional)

[String](#string)

### Transaction [Up](#X08a735672bda25f91812485a33b810177d042f8)

id (optional)

[Integer](#integer) format: int32

status (optional)

[Integer](#integer) format: int32

transactionID (optional)

[String](#string)

type (optional)

[String](#string)

settleDate (optional)

[Date](#DateTime) format: date-time

settleAmount (optional)

[Double](#double) format: double

workflowID (optional)

[Integer](#integer) format: int32

direction (optional)

[String](#string)

fileID (optional)

[String](#string)

docID (optional)

[String](#string)

timestamp (optional)

[Date](#DateTime) format: date-time

payaway (optional)

[Boolean](#boolean)

### TransactionSearchCriteria [Up](#X08a735672bda25f91812485a33b810177d042f8)

type (optional)

[String](#string)

direction (optional)

[String](#string)

service (optional)

[String](#string)

reference (optional)

[String](#string)

wfid (optional)

[Integer](#integer) format: int32

status (optional)

[Integer](#integer) format: int32

page (optional)

[Integer](#integer) format: int32

start (optional)

[Integer](#integer) format: int32

entity (optional)

[String](#string)

paymentbic (optional)

[String](#string)

transactionid (optional)

[String](#string)

payaway (optional)

[Boolean](#boolean)

rows (optional)

[Integer](#integer) format: int32

is-error (optional)

[Boolean](#boolean)

exclude-archive (optional)

[Boolean](#boolean)

settlement-from (optional)

[String](#string)

settlement-to (optional)

[String](#string)

ptimestamp-from (optional)

[String](#string)

ptimestamp-to (optional)

[String](#string)

has-message-id (optional)

[Boolean](#boolean)

transaction-type (optional)

[String](#string)

Enum:

transaction

payaway

### Transmittal [Up](#X08a735672bda25f91812485a33b810177d042f8)

username (optional)

[String](#string)

password (optional)

[String](#string)

entityID (optional)

[Integer](#integer) format: int32

fileType (optional)

[String](#string)

### TrustedCertificate [Up](#X08a735672bda25f91812485a33b810177d042f8)

certificateId (optional)

[String](#string)

certificateName

[String](#string)

serialNumber (optional)

[String](#string)

thumbprint (optional)

[String](#string)

thumbprint256 (optional)

[String](#string)

startDate (optional)

[String](#string)

endDate (optional)

[String](#string)

issuer (optional)

[map[String, array[String]]](#array)

subject (optional)

[map[String, array[String]]](#array)

authChainReport (optional)

[array[map[String, String]]](#map)

changerComments (optional)

[String](#string)

changeID (optional)

[String](#string)

### TrustedCertificateDetails [Up](#X08a735672bda25f91812485a33b810177d042f8)

serialNumber (optional)

[String](#string)

thumbprint (optional)

[String](#string)

thumbprint256 (optional)

[String](#string)

startDate (optional)

[String](#string)

endDate (optional)

[String](#string)

issuer (optional)

[map[String, array[String]]](#array)

subject (optional)

[map[String, array[String]]](#array)

authChainReport (optional)

[array[map[String, String]]](#map)

valid (optional)

[Boolean](#boolean)

certificateErrors (optional)

[array[map[String, array[String]]]](#map)

certificateWarnings (optional)

[array[map[String, Object]]](#map)

### TrustedCertificateLog [Up](#X08a735672bda25f91812485a33b810177d042f8)

certificateLogId (optional)

[String](#string)

certificateId (optional)

[String](#string)

certificateName (optional)

[String](#string)

serialNumber (optional)

[String](#string)

thumbprint (optional)

[String](#string)

thumbprint256 (optional)

[String](#string)

startDate (optional)

[String](#string)

endDate (optional)

[String](#string)

issuer (optional)

[map[String, array[String]]](#array)

subject (optional)

[map[String, array[String]]](#array)

authChainReport (optional)

[array[map[String, String]]](#map)

controlCert (optional)

[ChangeControlCert](#ChangeControlCert)

### WorkflowStep [Up](#X08a735672bda25f91812485a33b810177d042f8)

stepId (optional)

[Integer](#integer) format: int32

serviceName (optional)

[String](#string)

exeState (optional)

[String](#string)

advStatus (optional)

[String](#string)

startTime (optional)

[Date](#DateTime) format: date-time

endTime (optional)

[Date](#DateTime) format: date-time

nodeExecuted (optional)

[String](#string)

statusRpt (optional)

[String](#string)

docId (optional)

[String](#string)

wfcId (optional)

[String](#string)

wfdId (optional)

[Integer](#integer) format: int32

wfdVersion (optional)

[Integer](#integer) format: int32

inlineInvocation (optional)

[Boolean](#boolean)

### api\_certificates\_body [Up](#X08a735672bda25f91812485a33b810177d042f8)

file (optional)

[byte[]](#binary) format: binary

### certificates\_upload\_body [Up](#X08a735672bda25f91812485a33b810177d042f8)

file (optional)

[byte[]](#binary) format: binary