CUT API

The information on this site describes the CUT REST APIs.

Information system usage may be monitored, recorded, and subject to audit.

Unauthorized use of the information system is prohibited and subject to criminal and civil penalties. Use of the information system indicates consent to monitoring and recording.

If you are not an authorized user, exit immediately.

Version: 8.0.1 (R8A)

Auth

POST /auth/login

Implementation Notes

Login to get access token

Response Class (Status 200)

If login is successful an access token shall be returned else error message

Model | Example Value

LoginSuccess {
  success (boolean),
  token (string, *optional*): return access token,
  message (string, *optional*): return message,
  userLoginInfo (Inline Model 1, *optional*): info about the user's previous login
}
Inline Model 1 {
  timestamp (number): timestamp of last previous login,
  userAgentInfo (string): user agent used for the previous login,
  ipAddress (string): ip address from which the previous login occurred,
  geoLocation (string): geo location data of the user for their previous login
}

```
{
  "success": true,
  "token": "string",
  "message": "string",
  "userLoginInfo": {
    "timestamp": 0,
    "userAgentInfo": "string",
    "ipAddress": "string",
    "geoLocation": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

body		body	Model | Example Value
			UserLogin {
			username (string): login accout,
			password (string): login password
			}
			`{ "username": "string", "password": "string" }`

Response Messages

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /auth/logout

Implementation Notes

logout

Response Class (Status 200)

Successful response. User is logged out.

Model | Example Value

Success {
  success (boolean)
}

```
{
  "success": true
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /auth/token

Implementation Notes

Login to get access token

Response Class (Status 200)

If login is successfull an access token shall be retuned else error message

Model | Example Value

Success {
  success (boolean)
}

```
{
  "success": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`authCode`	Azure Active Directory authorization code	query	string
`redirectUri`	One of the redirect_uris of the application, registered in the Azure Portal and used when receiving the authorization_code during the first stage of authorization.	query	string

ActionSchedules

GET /actionSchedules/schedulations

Implementation Notes

Get all action schedules

Response Class (Status 200)

Successful response. Action schedules are returned.

Model | Example Value

actionScheduleArray {
  actionSchedules (Array[actionSchedule], *optional*): Action schedules
}
actionSchedule {
  id (string): Id Action schedule,
  actionId (string): Action/macros which scheduled,
  deviceId (string): Identifier of device which actionSchedule belongs to,
  serviceId (string): Identifier of service which actionSchedule belongs to,
  recurrence (recurrence): recurrence,
  meta (object): Information about user who scheduled by action/macro
}
recurrence {
  executionTime (string): execution time,
  frequency (string): frequency,
  endBy (string,null): end by
}

```
{
  "actionSchedules": [
    {
      "id": "d68436dd-ac04-41c6-bd59-705054643444",
      "actionId": "a4227938-47fe-42d6-ae7e-261fc4ada6ed",
      "serviceId": "48b22a9c-93ae-40b1-9462-1d9aabfe4a4a",
      "deviceId": "",
      "recurrence": {
        "executionTime": "2019-05-04T12:51:59.000Z",
        "frequency": "DAILY",
        "endBy": "2019-06-04T12:51:59.000Z"
      },
      "meta": {
        "userId": "01600a21-a101-40fb-add5-06931b7ed2ff",
        "organizationId": "373f636a-5a33-4c82-996a-5d461ddb64bc",
        "userName": "fullname"
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationIds`	Filter on all action schedulations that have a specific organization (based on organizationId) Multiple fields can be specified separated by a comma (,)	query	Array[string]

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

POST /actionSchedules/schedulations

Implementation Notes

Create a new action schedule. Time format must be converted to UTC+0 time zone

Response Class (Status 201)

Successful response. Actions schedules are returned.

Model | Example Value

actionSchedule {
  id (string): Id Action schedule,
  actionId (string): Action/macros which scheduled,
  deviceId (string): Identifier of device which actionSchedule belongs to,
  serviceId (string): Identifier of service which actionSchedule belongs to,
  recurrence (recurrence): recurrence,
  meta (object): Information about user who scheduled by action/macro
}
recurrence {
  executionTime (string): execution time,
  frequency (string): frequency,
  endBy (string,null): end by
}

```
{
  "id": "d68436dd-ac04-41c6-bd59-705054643444",
  "actionId": "a4227938-47fe-42d6-ae7e-261fc4ada6ed",
  "serviceId": "48b22a9c-93ae-40b1-9462-1d9aabfe4a4a",
  "deviceId": "",
  "recurrence": {
    "executionTime": "2019-05-04T12:51:59.000Z",
    "frequency": "DAILY",
    "endBy": "2019-06-04T12:51:59.000Z"
  },
  "meta": {
    "userId": "01600a21-a101-40fb-add5-06931b7ed2ff",
    "organizationId": "373f636a-5a33-4c82-996a-5d461ddb64bc",
    "userName": "user name"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

<table>
<tr><th>Parameter</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr>
<tr><td><code>actionSchedulesData</code></td><td></td><td>body</td><td>Model | Example Value

<b>actionSchedulePost {</b>
<b>actionId</b> (string): Action/macros which scheduled,
<b>deviceId</b> (string): Identifier of device which actionSchedule belongs to,
<b>recurrence</b> (recurrence): recurrence
<b>}</b>
<b>recurrence {</b>
<b>executionTime</b> (string): execution time,
<b>frequency</b> (string): frequency,
<b>endBy</b> (string,null): end by
<b>}</b>
<pre>
{
  "actionId": "a4227938-47fe-42d6-ae7e-261fc4ada6ed",
  "deviceId": "",
  "recurrence": {
    "executionTime": "2019-05-04T12:51:59.000Z",
    "frequency": "DAILY",
    "endBy": "2019-07-04T12:51:59.000Z"
  }
}
</pre></td></tr>
</table>

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers
422	Server Error	Model | Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

DELETE /actionSchedules/schedulations/{schedulationId}

Implementation Notes

Delete a specific action schedule

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`schedulationId`		path	string
`deleteHistory`	Delete action schedule	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

GET /actionSchedules/schedulations/{schedulationId}

Implementation Notes

Get a specific action schedule

Response Class (Status 200)

Successful response. Action schedule is returned.

Model | Example Value

actionSchedule {
  id (string): Id Action schedule,
  actionId (string): Action/macros which scheduled,
  deviceId (string): Identifier of device which actionSchedule belongs to,
  serviceId (string): Identifier of service which actionSchedule belongs to,
  recurrence (recurrence): recurrence,
  meta (object): Information about user who scheduled by action/macro
}
recurrence {
  executionTime (string): execution time,
  frequency (string): frequency,
  endBy (string,null): end by
}

```
{
  "id": "d68436dd-ac04-41c6-bd59-705054643444",
  "actionId": "a4227938-47fe-42d6-ae7e-261fc4ada6ed",
  "serviceId": "48b22a9c-93ae-40b1-9462-1d9aabfe4a4a",
  "deviceId": "",
  "recurrence": {
    "executionTime": "2019-05-04T12:51:59.000Z",
    "frequency": "DAILY",
    "endBy": "2019-06-04T12:51:59.000Z"
  },
  "meta": {
    "userId": "01600a21-a101-40fb-add5-06931b7ed2ff",
    "organizationId": "373f636a-5a33-4c82-996a-5d461ddb64bc",
    "userName": "user name"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`schedulationId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /actionSchedules/schedulations/{schedulationId}

Implementation Notes

Update the action schedule. Time format must be converted to UTC+0 time zone

Response Class (Status 201)

Successful response. Action schedule is returned.

Model | Example Value

actionSchedule {
  id (string): Id Action schedule,
  actionId (string): Action/macros which scheduled,
  deviceId (string): Identifier of device which actionSchedule belongs to,
  serviceId (string): Identifier of service which actionSchedule belongs to,
  recurrence (recurrence): recurrence,
  meta (object): Information about user who scheduled by action/macro
}
recurrence {
  executionTime (string): execution time,
  frequency (string): frequency,
  endBy (string,null): end by
}

```
{
  "id": "d68436dd-ac04-41c6-bd59-705054643444",
  "actionId": "a4227938-47fe-42d6-ae7e-261fc4ada6ed",
  "serviceId": "48b22a9c-93ae-40b1-9462-1d9aabfe4a4a",
  "deviceId": "",
  "recurrence": {
    "executionTime": "2019-05-04T12:51:59.000Z",
    "frequency": "DAILY",
    "endBy": "2019-06-04T12:51:59.000Z"
  },
  "meta": {
    "userId": "01600a21-a101-40fb-add5-06931b7ed2ff",
    "organizationId": "373f636a-5a33-4c82-996a-5d461ddb64bc",
    "userName": "user name"
  }
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
schedulationId		path	string

Parameter	Description	Parameter Type	Data Type

`actionScheduleRecurrence`		body	Model Example Value
			recurrence {
			executionTime (string): execution time,
			frequency (string): frequency,
			endBy (string,null): end by
			}

```
{
  "executionTime": "2019-05-04T12:51:59.000Z",
  "frequency": "DAILY",
  "endBy": "2019-06-04T12:51:59.000Z"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /actionSchedules/history

Implementation Notes

Get all action schedules history. Time format must be converted to UTC+0 time zone

Response Class (Status 200)

Successful response. Action schedules history are returned.

Model | Example Value

actionScheduleHistoryArray {
  actionSchedulesHistory (Array[actionScheduleHistory], *optional*): Action schedules history
}
actionScheduleHistory {
  id (string): Unique action schedule history identifier,
  actionSchedulationId (string): Executed id action schedule,
  executionTime (string): Date when action schedule execute,
  status (string): Status after action schedule execute = ['OK', 'Failed'],
  meta (object, *optional*): Information about action schedule
}

```
{
  "actionSchedulesHistory": [
    {
      "id": "094111c0-49a4-436f-ac26-e0229d488205",
      "actionSchedulationId": "d68436dd-ac04-41c6-bd59-705054643444",
      "executionTime": "2018-05-04T12:51:59.000Z",
      "status": "OK",
      "meta": {
        "actionId": "a4227938-47fe-42d6-ae7e-261fc4ada6ed",
        "serviceId": "c1a00239-5bb0-4e96-be9c-bf9260978867",
        "deviceId": "",
        "userId": "userId",
        "userName": "fullname"
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`schedulationId`		query	string
`timestampTo`	Get all action schedule history that are raised before the timestamp	query	date-time
`timestampFrom`	get all action schedule history that are raised after the timestamp	query	date-time

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /actionSchedules/history/{schedulationId}

Implementation Notes

Delete history by schedulation Id. Time format must be converted to UTC+0 time zone

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`schedulationId`	**Schedulation id**	path	string
`timestampTo`	Delete all action schedule history that are raised before the timestamp.	query	date-time

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>204</td><td>Successful response.</td><td></td><td></td></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b><pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b><pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

Actions

GET /actions

Implementation Notes

Get all Action list. Actions are activities that can be executed either triggered from a Rule or directly by the end-user.

Response Class (Status 200)

Successful response. The Action list is returned.

Model | Example Value

Actions {
  actions (Array[action], *optional*): all of the actions
}
action {
  serviceId (string): serviceId of Action.,
  deviceTypeKey (number, *optional*): List of all device type keys that are in relation with the correspondindg device category.,
  name (string): The name of this Action,
  type (string): The type of this Action,
  description (string, *optional*),
  needConfirmation (boolean): Whether we need confirmation,
  runtime (string): The runtime of this action.,
  meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).
}

```
{
  "actions": [
    {
      "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
      "deviceTypeKey": 127001,
      "name": "Action Name",
      "description": "Action Description",
      "needConfirmation": false,
      "runtime": "API",
      "type": "testType",
      "meta": {}
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceTypeKey`	Filter all actions based on deviceTypeKey. Multiple fields can be specified separated by a comma (,)	query	Array[double]

Parameter	Description	Parameter Type	Data Type
`serviceId`	Filter all actions based on serviceId. Multiple fields can be specified separated by a comma (,)	query	Array[string]
`type`	Filter all actions based on type. Multiple fields can be specified separated by a comma (,)	query	Array[string]

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers
500	Internal Server Error	Model Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /actions

Implementation Notes

Create Action

Response Class (Status 201)

Successful response. Returns created Action.

Model | Example Value

action {
serviceId (string): serviceId of Action.,
deviceTypeKey (number, *optional*): List of all device type keys that are in relation with the correspondindg device category.,
name (string): The name of this Action,
type (string): The type of this Action,
description (string, *optional*),
needConfirmation (boolean): Whether we need confirmation,
runtime (string): The runtime of this action.,
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).
}

```
{
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 127001,
  "name": "Action Name",
  "description": "Action Description",
  "needConfirmation": false,
  "runtime": "API",
  "type": "testType",
  "meta": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Parameter	Description	Parameter Type	Data Type
body	**Action to be added to the database.**	body	Model Example Value **action {** **serviceId** (string): serviceId of Action., **deviceTypeKey** (number, *optional*): List of all device type keys that are in relation with the correspondindg device category., **name** (string): The name of this Action, **type** (string): The type of this Action, **description** (string, *optional*), **needConfirmation** (boolean): Whether we need confirmation, **runtime** (string): The runtime of this action., **meta** (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on). **}**

```
{
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 127001,
  "name": "Action Name",
  "description": "Action Description",
  "needConfirmation": false,
  "runtime": "API",
  "type": "testType",
  "meta": {}
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /actions/{id}

Implementation Notes

Delete Action

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**id of the action to be deleted**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response.		

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /actions/{id}

Implementation Notes

Get Action based on id

Response Class (Status 200)

Successful response. Action is returned.

Model | Example Value

action {

serviceId (string): serviceId of Action.,
deviceTypeKey (number, *optional*): List of all device type keys that are in relation with the correspondindg device category.,
name (string): The name of this Action,
type (string): The type of this Action,
description (string, *optional*),
needConfirmation (boolean): Whether we need confirmation,
runtime (string): The runtime of this action.,
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).

}

```
{
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 127001,
  "name": "Action Name",
  "description": "Action Description",
  "needConfirmation": false,
  "runtime": "API",
  "type": "testType",
  "meta": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id	**id of the action to get**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

PUT /actions/{id}

Implementation Notes

Update the Action

Response Class (Status 201)

Successful response. Action is returned.

Model | Example Value

action {

serviceId (string): serviceId of Action.,
deviceTypeKey (number, *optional*): List of all device type keys that are in relation with the correspondindg device category.,
name (string): The name of this Action,
type (string): The type of this Action,
description (string, *optional*),
needConfirmation (boolean): Whether we need confirmation,
runtime (string): The runtime of this action.,
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).

}

```
{
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 127001,
  "name": "Action Name",
  "description": "Action Description",
  "needConfirmation": false,
  "runtime": "API",
  "type": "testType",
  "meta": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**id of the action to update**	path	string

Parameter	Description	Parameter Type	Data Type
`action`	**New action object that will replace the old one.**	body	Model Example Value

action {

serviceId (string): serviceId of Action.,
deviceTypeKey (number, *optional*): List of all device type keys that are in relation with the correspondindg device category.,
name (string): The name of this Action,
type (string): The type of this Action,
description (string, *optional*),
needConfirmation (boolean): Whether we need confirmation,
runtime (string): The runtime of this action.,
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).

}

```
{
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 127001,
  "name": "Action Name",
  "description": "Action Description",
  "needConfirmation": false,
  "runtime": "API",
  "type": "testType",
  "meta": {}
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model Example Value	

Errors {

errors (Array[Error])

}

Error {

code (integer): Four-digits error code,
message (string, *optional*): Error description

}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Internal Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /actions/trigger/{id}

Implementation Notes

Trigger the Action

Parameters

Parameter	Description	Parameter Type	Data Type
id	**id of the action to trigger**	path	string
x-access-token	User token.	header	string
deviceId	Device id the action is triggered for.	query	string

body	New action meta to override the old one.	body	Model Example Value
			ActionPostBody {
			meta (object, *optional*): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).
			}

```
{
  "meta": {}
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
202	Successful response.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

AlarmRecords

GET /alarmRecords

Implementation Notes

Get all alarm records

Response Class (Status 200)

Successful response. Returns all alarm records array.

Model | Example Value

alarmRecordsResponse {
alarmRecords (Array[alarmRecord], *optional*): DeviceId of the alarm record
}
alarmRecord {
id (string): Id of the alarm record,
deviceId (string, *optional*): DeviceId of the alarm record,
serviceId (string): ServiceId of the alarm record,
timestamp (string): Timestamp of the alarm record,
severity (number): Severity level of the alarm record,
alarmCode (string): Alarm code of the alarm record,
name (string): Alarm record name,
fromSync (boolean): Is the alarm record from Sync,
meta (object): Meta object of the alarm record,
history (Array[alarmRecordHistory], *optional*): History of the acknowledgement events
}
alarmRecordHistory {
id (string): Alarm record history id,
status (string): Acknowledged or Unacknowledged status record = ['acknowledged', 'unacknowledged'],
timestamp (string): Timestamp of the alarm record history,
userName (string): Name of the user who made a record,
userEmail (string): Email of the user who made a record
}

```
{
  "alarmRecords": [
    {
      "id": "string",
      "deviceId": "string",
      "serviceId": "string",
      "timestamp": "2019-10-03T07:46:34.174Z",
      "severity": 0,
      "alarmCode": "string",
      "name": "string",
      "fromSync": true,
      "meta": {},
      "history": [
        {
          "id": "string",
          "status": "acknowledged",
          "timestamp": "2019-10-03T07:46:34.174Z",
          "userName": "string",
          "userEmail": "string"
        }
      ]
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`alarmRecordId`	Get only alarmRecords with specific ids Multiple fields can be specified separated by a comma (,)	query	Array[string]
`timestampFrom`	**Get all alarmRecords that are raised after the timestamp.**	query	date-time
`timestampTo`	Get all alarmRecords that are raised before the timestamp.	query	date-time
`deviceId`	Filter on a specific deviceId Multiple fields can be specified separated by a comma (,)	query	Array[string]
`severity`	Filter on a specific severity - Critical: 1 - Major: 2 - Minor: 3 - Warning: 4	query	Array[integer]
`sort`	Sort the response by specified fields. - prefix imply a descending sort order. Sort can't be used with raised.	query	Array[string]
`search`	Searching alarmRecords by fields: alarmCode, name, deviceName, serviceName	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /alarmRecords

Implementation Notes

Add specific alarm record. Time format must be converted to UTC+0 time zone.

Response Class (Status 201)

Successful response. Returns created alarm records array.

Model | Example Value

alarmRecordsResponse {
  alarmRecords (Array[alarmRecord], *optional*): DeviceId of the alarm record
}
alarmRecord {
  id (string): Id of the alarm record,
  deviceId (string, *optional*): DeviceId of the alarm record,
  serviceId (string): ServiceId of the alarm record,
  timestamp (string): Timestamp of the alarm record,
  severity (number): Severity level of the alarm record,
  alarmCode (string): Alarm code of the alarm record,
  name (string): Alarm record name,
  fromSync (boolean): Is the alarm record from Sync,
  meta (object): Meta object of the alarm record,
  history (Array[alarmRecordHistory], *optional*): History of the acknowledgement events
}
alarmRecordHistory {
  id (string): Alarm record history id,
  status (string): Acknowledged or Unacknowledged status record = ['acknowledged', 'unacknowledged'],
  timestamp (string): Timestamp of the alarm record history,
  userName (string): Name of the user who made a record,
  userEmail (string): Email of the user who made a record
}

```
{
  "alarmRecords": [
    {
      "id": "string",
      "deviceId": "string",
      "serviceId": "string",
      "timestamp": "2019-10-03T07:46:34.190Z",
      "severity": 0,
      "alarmCode": "string",
      "name": "string",
      "fromSync": true,
      "meta": {},
      "history": [
        {
          "id": "string",
          "status": "acknowledged",
          "timestamp": "2019-10-03T07:46:34.190Z",
          "userName": "string",
          "userEmail": "string"
        }
      ]
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

body	**Array of alarm records that would be added to the database.**	body	Model | Example Value
			alarmRecordsInput [
			alarmRecordWithoutId
			]
			alarmRecordWithoutId {
			deviceId (string, *optional*): DeviceId of the alarm record
			,
			serviceId (string): ServiceId of the alarm record,
			timestamp (string, *optional*): Timestamp of the alarm record,
			severity (number): Severity level of the alarm record,
			alarmCode (string): Alarm code of the alarm record,
			name (string): Alarm record name,
			fromSync (boolean, *optional*): Is the alarm record from Sync,
			meta (object, *optional*): Meta object of the alarm record
			}
			[
			{
			"deviceId": "string",
			"serviceId": "string",
			"timestamp": "2019-10-03T07:46:34.194Z",
			"severity": 0,
			"alarmCode": "string",
			"name": "string",
			"fromSync": true,
			"meta": {}
			}
			]

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

GET /alarmRecords/{alarmRecordId}

Implementation Notes

Get a specific alarm records

Response Class (Status 200)

Successful response. Returns specific alarm records.

Model | Example Value

alarmRecord {
  id (string): Id of the alarm record,
  deviceId (string, *optional*): DeviceId of the alarm record,
  serviceId (string): ServiceId of the alarm record,
  timestamp (string): Timestamp of the alarm record,
  severity (number): Severity level of the alarm record,
  alarmCode (string): Alarm code of the alarm record,
  name (string): Alarm record name,
  fromSync (boolean): Is the alarm record from Sync,
  meta (object): Meta object of the alarm record,
  history (Array[alarmRecordHistory], *optional*): History of the acknowledgement events
}
alarmRecordHistory {
  id (string): Alarm record history id,
  status (string): Acknowledged or Unacknowledged status record = ['acknowledged', 'unacknowledged'],
  timestamp (string): Timestamp of the alarm record history,
  userName (string): Name of the user who made a record,
  userEmail (string): Email of the user who made a record
}

```
{
  "id": "string",
  "deviceId": "string",
  "serviceId": "string",
  "timestamp": "2019-10-03T07:46:34.203Z",
  "severity": 0,
  "alarmCode": "string",
  "name": "string",
  "fromSync": true,
  "meta": {},
  "history": [
    {
      "id": "string",
      "status": "acknowledged",
      "timestamp": "2019-10-03T07:46:34.203Z",
      "userName": "string",
      "userEmail": "string"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`alarmRecordId`		path	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}<pre>{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}<pre>{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

Alarms

GET /alarms

Implementation Notes

Get all alarms

Response Class (Status 200)

Successful response. Alarms are returned.

Model | Example Value

alarmArray {
alarms (Array[alarmObject], *optional*)
}
alarmObject {
id (string): Id of occured alarm,
deviceId (string, *optional*): Id of the device,
serviceId (string): ServiceId of of the alarm,
raiseTimestamp (string): Raise time stamp,
clearTimestamp (string, *optional*): Clear time stamp (if the alarm is cleared),
created (string, *optional*): Time stamp when the alarm is created,
severity (number): Level of the alarm. Cleared: 0, Critical: 1,Major: 2,Minor: 3,Warning: 4,
alarmCode (string): Alarm code,
name (string, *optional*): Name of the alarm,
alarmRecordIds (Array[string]): Alarm Records linked to this alarm,
clearedFromSync (boolean, *optional*): Is alarm cleared from a Sync,
raisedFromSync (boolean, *optional*): Is alarm raised from a Sync,
acknowledged (boolean, *optional*): Is alarm acknowledged,
cleared (boolean, *optional*): Is alarm cleared,
meta (object, *optional*): Meta info
}

```
{
  "alarms": [
    {
      "id": "string",
      "deviceId": "string",
      "serviceId": "string",
      "raiseTimestamp": "2019-10-03T07:46:34.213Z",
      "clearTimestamp": "2019-10-03T07:46:34.213Z",
      "created": "2019-10-03T07:46:34.213Z",
      "severity": 0,
      "alarmCode": "string",
      "name": "string",
      "alarmRecordIds": [
        "string"
      ],
      "clearedFromSync": true,
      "raisedFromSync": true,
      "acknowledged": true,
      "cleared": true,
      "meta": {}
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceId`	Filter on a specific serviceId Multiple fields can be specified separated by a comma (,)	query	Array[string]
`deviceId`	Filter on a specific deviceId Multiple fields can be specified separated by a comma (,)	query	Array[string]
`alarmRecordId`	Filter on a specific alarmRecordId Multiple fields can be specified separated by a comma (,)	query	Array[string]
`severity`	Filter on a specific severity • Critical: 1 • Major: 2 • Minor: 3 • Warning: 4	query	Array[integer]
`timestampFrom`	Get all alarms that are raised after the timestamp. Alarm does not have to be cleared. Timestamp must be in ISO 8601 format.	query	date-time
`timestampTo`	Get all alarms that are raised before the timestamp. Alarm does not have to be cleared. Timestamp must be in ISO 8601 format.	query	date-time
`sort`	Sort the response by specified fields. - prefix imply a descending sort order. Sort can't be used with raised.	query	Array[string]
`raised`	filter on the last unique alarms that are raised and not cleared.	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

400	Bad request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /alarms/{alarmId}

Implementation Notes

Get a specific alarm

Response Class (Status 200)

Successful response. Alarm is returned.

Model | Example Value

alarmObject {
id (string): Id of occured alarm,
deviceId (string, *optional*): Id of the device,
serviceId (string): ServiceId of of the alarm,
raiseTimestamp (string): Raise time stamp,
clearTimestamp (string, *optional*): Clear time stamp (if the alarm is cleared),
created (string, *optional*): Time stamp when the alarm is created,
severity (number): Level of the alarm. Cleared: 0, Critical: 1,Major: 2,Minor: 3,Warning: 4,
alarmCode (string): Alarm code,
name (string, *optional*): Name of the alarm,
alarmRecordIds (Array[string]): Alarm Records linked to this alarm,
clearedFromSync (boolean, *optional*): Is alarm cleared from a Sync,
raisedFromSync (boolean, *optional*): Is alarm raised from a Sync,
acknowledged (boolean, *optional*): Is alarm acknowledged,
cleared (boolean, *optional*): Is alarm cleared,
meta (object, *optional*): Meta info
}

```
{
  "id": "string",
  "deviceId": "string",
  "serviceId": "string",
  "raiseTimestamp": "2019-10-03T07:46:34.237Z",
  "clearTimestamp": "2019-10-03T07:46:34.237Z",
  "created": "2019-10-03T07:46:34.237Z",
  "severity": 0,
  "alarmCode": "string",
  "name": "string",
  "alarmRecordIds": [
    "string"
  ],
  "clearedFromSync": true,
  "raisedFromSync": true,
  "acknowledged": true,
  "cleared": true,
  "meta": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
`alarmId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

404	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /alarms/{alarmId}/acknowledge

Implementation Notes

Acknowledge a specific alarm

Response Class (Status 200)

Successful response. Alarm is acknowledged.

Model | Example Value

alarmWithAcknowledge {
id (string): Id of occured alarm,
deviceId (string, *optional*): Id of the device,
serviceId (string): ServiceId of of the alarm,
raiseTimestamp (string): Raise time stamp,
clearTimestamp (string, *optional*): Clear time stamp (if the alarm is cleared),
created (string, *optional*): Time stamp when the alarm is created,
severity (number): Level of the alarm. Cleared: 0, Critical: 1,Major: 2,Minor: 3,Warning: 4,
alarmCode (string): Alarm code,
name (string, *optional*): Name of the alarm,
alarmRecordIds (Array[string]): Alarm Records linked to this alarm,
clearedFromSync (boolean, *optional*): Is alarm cleared from a Sync,
raisedFromSync (boolean, *optional*): Is alarm raised from a Sync,
acknowledged (boolean, *optional*): Is alarm acknowledged,
cleared (boolean, *optional*): Is alarm cleared,
meta (object, *optional*): Meta info,
acknowledge (object): Alarm acknowledge
}

```
{
  "id": "string",
  "deviceId": "string",
  "serviceId": "string",
  "raiseTimestamp": "2019-10-03T07:46:34.248Z",
  "clearTimestamp": "2019-10-03T07:46:34.248Z",
  "created": "2019-10-03T07:46:34.248Z",
  "severity": 0,
  "alarmCode": "string",
  "name": "string",
  "alarmRecordIds": [
    "string"
  ],
  "clearedFromSync": true,
  "raisedFromSync": true,
  "acknowledged": true,
  "cleared": true,
  "meta": {},
  "acknowledge": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
alarmId		path	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /alarms/{alarmId}/unacknowledge

Implementation Notes

Unacknowledge a specific alarm

Response Class (Status 200)

Successful response. Alarm is unacknowledged.

Model | Example Value

alarmObject {
id (string): Id of occured alarm,
deviceId (string, *optional*): Id of the device,
serviceId (string): ServiceId of of the alarm,
raiseTimestamp (string): Raise time stamp,
clearTimestamp (string, *optional*): Clear time stamp (if the alarm is cleared),
created (string, *optional*): Time stamp when the alarm is created,
severity (number): Level of the alarm. Cleared: 0, Critical: 1,Major: 2,Minor: 3,Warning: 4,
alarmCode (string): Alarm code,
name (string, *optional*): Name of the alarm,
alarmRecordIds (Array[string]): Alarm Records linked to this alarm,
clearedFromSync (boolean, *optional*): Is alarm cleared from a Sync,
raisedFromSync (boolean, *optional*): Is alarm raised from a Sync,
acknowledged (boolean, *optional*): Is alarm acknowledged,
cleared (boolean, *optional*): Is alarm cleared,
meta (object, *optional*): Meta info
}

```
{
  "id": "string",
  "deviceId": "string",
  "serviceId": "string",
  "raiseTimestamp": "2019-10-03T07:46:34.260Z",
  "clearTimestamp": "2019-10-03T07:46:34.260Z",
  "created": "2019-10-03T07:46:34.260Z",
  "severity": 0,
  "alarmCode": "string",
  "name": "string",
  "alarmRecordIds": [
    "string"
  ],
  "clearedFromSync": true,
  "raisedFromSync": true,
  "acknowledged": true,
  "cleared": true,
  "meta": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
`alarmId`		path	string

Response Messages

<table>
<thead>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
</thead>
<tbody>
<tr><td>403</td><td>Permission Error</td><td>
Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre>
</td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>
Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre>
</td><td></td></tr>
</tbody>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model │ Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

Api Information

GET /apiInfo

Implementation Notes

Get api build information

Response Class (Status 200)

Successful response. Build information is returned.

Model | Example Value

apiInfo {
apiName (string): name of api,
apiVersion (string): version of api
}

```
{
  "apiName": "string",
  "apiVersion": "string"
}
```

Response Content Type application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

Attachments

GET /attachments

Implementation Notes

Get all attachments based on filter

Response Class (Status 200)

Successful response. Return attachment list.

Model | Example Value

AttachmentListResponse {
  attachments (Array[AttachmentModel], *optional*): attachment array
}
AttachmentModel {
  id (string): Unique identifier representing this attachment,
  url (string): the access url of attachment,
  name (string): the name of attachment,
  type (string): the type of attachment,
  correlationId (string, *optional*): the Id of the data record in the DB which this attachment corresponding to. For example, the id of an article which own this attachment,
  size (number): the size of attachment, the unit is byte,
  uploadedAt (string): the time when this attachment is uploaded,
  userId (string): the user id of uploader,
  organizationId (string): the organization id of the uploader,
  isTemporary (boolean, *optional*): whether this attachment is a temporary file or not. Temporary file will be deleted by a daily schedule task(scheduledTemporaryAttachmentRemove)
}

```
{
  "attachments": [
    {
      "id": "string",
      "url": "string",
      "name": "string",
      "type": "string",
      "correlationId": "string",
      "size": 0,
      "uploadedAt": "2019-10-03T07:46:34.279Z",
      "userId": "string",
      "organizationId": "string",
      "isTemporary": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
`correlationId`	the Id of the data record in the DB which this attachment corresponding to. For example, the id of an article which own this attachment	query	string
`name`	the name of attachment	query	string
`isTemporary`	whether this attachment is a temporary file or not.Temporary file will be deleted by a daily schedule task(scheduledTemporaryAttachmentRemove)	query	boolean
`type`	the type of attachment	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /attachments

Implementation Notes

Post a file to storage, and save the upload file record into database. Files with specific names and extensions cannot be uploaded. (Forbidden names: crossdomain.xml, .htpasswd, etc.; Forbidden extensions: asp, php, css, swf, xhtml, etc.)

Response Class (Status 201)

Successful response. Return information of uploaded file.

Model | Example Value

AttachmentListResponse {
  attachments (Array[AttachmentModel], *optional*): attachment array
}
AttachmentModel {
  id (string): Unique identifier representing this attachment,
  url (string): the access url of attachment,
  name (string): the name of attachment,
  type (string): the type of attachment,
  correlationId (string, *optional*): the Id of the data record in the DB which this attachment corresponding to. For example, the id of an article which own this attachment,
  size (number): the size of attachment, the unit is byte,
  uploadedAt (string): the time when this attachment is uploaded,
  userId (string): the user id of uploader,
  organizationId (string): the organization id of the uploader,
  isTemporary (boolean, *optional*): whether this attachment is a temporary file or not. Temporary file will be deleted by a daily schedule task(scheduledTemporaryAttachmentRemove)
}

```
{
  "attachments": [
    {
      "id": "string",
      "url": "string",
      "name": "string",
      "type": "string",
      "correlationId": "string",
      "size": 0,
      "uploadedAt": "2019-10-03T07:46:34.295Z",
      "userId": "string",
      "organizationId": "string",
      "isTemporary": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Parameter	Description	Parameter Type	Data Type
attachment	**The file to upload.**	formData	file
correlationId	the Id of the data record in the DB which this attachment corresponding to. For example, the id of an article which own this attachment.	formData	string
isTemporary	whether this attachment is a temporary file or not. Temporary file will be deleted by a daily schedule task(scheduledTemporaryAttachmentRemove)	formData	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /attachments/{attachmentId}/download

Implementation Notes

Download the specific attachment by id

Note that the this API check the permission via the x-access-token in the cookies.

Parameters

Parameter	Description	Parameter Type	Data Type
`attachmentId`	**Id of the attachment to delete**	path	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>302</td><td>Redirect to download location</td><td></td><td>Location string</td></tr>
<tr><td>403</td><td>User does not have permission to do the operation</td><td>Model | Example Value
<b>Errors {</b>
   <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
   <b>code</b> (integer): Four-digits error code,
   <b>message</b> (string, <i>optional</i>): Error description
<b>}</b><pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model | Example Value
<b>Errors {</b>
   <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
   <b>code</b> (integer): Four-digits error code,
   <b>message</b> (string, <i>optional</i>): Error description
<b>}</b><pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

DELETE /attachments/{attachmentId}

Implementation Notes

Delete the specific attachment

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`attachmentId`	**Id of the attachment to delete**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The attachment is deleted.		

403	User does not have permission to do the operation	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Audit Events

POST /auditEvents

Implementation Notes

Send audit event

Response Class (Status 201)

Successful response.

Model | Example Value

auditEvents [
auditEvent
]
auditEvent {
eventId (string): Id of occured event,
userId (string, *optional*): User Id,
timestamp (string): Timestamp of occured event,
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).
}

```
[
  {
    "eventId": "string",
    "userId": "string",
    "timestamp": "2019-10-03T07:46:34.332Z",
    "meta": {}
  }
]
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Parameter	Description	Parameter Type	Data Type
body		body	Model Example Value

Array[**auditEvent {**

eventId (string): Id of occured event,
userId (string, *optional*): User Id,
timestamp (string): Timestamp of occured event,
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).

}]

```
[
  {
    "eventId": "string",
    "userId": "string",
    "timestamp": "2019-10-03T07:46:34.340Z",
    "meta": {}
  }
]
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
422	Server Error	Model Example Value	

Error {

code (integer): Four-digits error code,
message (string, *optional*): Error description

}

```
{
  "code": 1000,
  "message": "Unknown error with API"
}
```

Command

POST /command/{serviceId}

Implementation Notes

execute command

Response Class (Status 201)

Successful response

Model | Example Value

CommandResponse {
  command (string, *optional*): command,
  result (string, *optional*): return message,
  statusCode (number, *optional*): statusCode
}

```
{
  "statusCode": 201,
  "command": "moveup",
  "response": {
    "result": "success"
  }
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceId`	**service Id**	path	string
body	command body	body	Model | Example Value **CommandRequest {** **permissionDomain** (string): permissionDomain, **command** (string): command, **metas** (object, *optional*) **}**

```
{
  "permissionDomain": "movecamera",
  "command": "moveup",
  "metas": {
    "device": {
      "id": 1,
      "type": "camera"
    }
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

401	Not authenticated	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

Configuration

GET /configuration

Implementation Notes

Get configuration

Response Class (Status 200)

Successful response. Settings have been committed.

Model | Example Value

Configuration {
core (object, *optional*): core configuration,
measurements (object, *optional*): devices measurements configuration,
devices (object, *optional*): layers bar configuration,
search (object, *optional*): global search configuration,
rules (object, *optional*): Rules form and Device form configuration,
localization (object, *optional*): localization module configuration,
theme (string, *optional*): themeProvider module configuration,
fileSize (object, *optional*): uploaded file size limit,
externalLinks (Array[externalLink], *optional*): Links component on ui configuration,
infoText (string, *optional*): InfoComponent on ui configuration,
contactLink (string, *optional*): Contact us link in UI navigation under Help dropdown,
allowedDeviceImageTypes (Array[imageFileType], *optional*): allowed image file types for user defined device,
deviceTypesWithoutCustomMainImage (Array[string], *optional*),
userDefinedDevice (object, *optional*): user defined device configuration,
legalDisclaimer (string, *optional*): InfoComponent on ui configuration,
azure (object, *optional*): audit module configuration,
layers (object, *optional*): layer control configuration,
audit (object, *optional*): audit module configuration,
services (object, *optional*): core services configuration,
map (object, *optional*): map on ui configuration,
deviceFilters (Array[deviceFilterNode], *optional*): filters for devices on map,
alarmRecords (object, *optional*): alarmRecords intervals configuration,
rulesMode (string, *optional*): api mode of rules
}
externalLink {
id (string, *optional*): id of external link,
type (string, *optional*): type of external link,
name (string, *optional*): name of external link,
url (string, *optional*): url of external link
}
imageFileType {
type (string, *optional*): file type, for example, image/gif, image/jpeg or image/png,
name (string, *optional*): display name of the file type. it is used in the hint text display
}
deviceFilterNode {
id (string, *optional*): item id,
key (string, *optional*): item key,
level (string, *optional*): parent or child,
name (string, *optional*): node label,
selected (boolean, *optional*): is item selected,
partiallySelected (boolean, *optional*): is item partially selected,
nodes (Array[Inline Model 1], *optional*)

}

Inline Model 1 {}

```
{
  "core": {},
  "measurements": {},
  "devices": {},
  "search": {},
  "rules": {},
  "localization": {},
  "theme": "string",
  "fileSize": {},
  "externalLinks": [
    {
      "id": "string",
      "type": "string",
      "name": "string",
      "url": "string"
    }
  ],
  "infoText": "string",
  "contactLink": "string",
  "allowedDeviceImageTypes": [
    {
      "type": "string",
      "name": "string"
    }
  ],
  "deviceTypesWithoutCustomMainImage": [
    "string"
  ],
  "userDefinedDevice": {},
  "legalDisclaimer": "string",
  "azure": {},
  "layers": {},
  "audit": {},
  "services": {},
  "map": {},
  "deviceFilters": [
    {
      "id": "string",
      "key": "string",
      "level": "string",
      "name": "string",
      "selected": true,
      "partiallySelected": true,
      "nodes": [
        {}
      ]
    }
  ],
  "alarmRecords": {},
  "rulesMode": "string"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

Device Attachments

GET /deviceAttachments

Implementation Notes

Get attachments associated with given device id. Latest uploaded first.

Response Class (Status 200)

Success response

Model | Example Value

DeviceAttachments {
  deviceAttachments (Array[DeviceAttachment]): deviceAttachment array
}
DeviceAttachment {
  id (string): The id of the attachment,
  name (string): The name of the attachment,
  deviceIds (Array[string]),
  uploadedAt (object): The time the device attachment is uploaded,
  uploader (string, *optional*): The name of the user who uploaded the attachment,
  size (number): The size in bytes of the attachment,
  url (string, *optional*): The Azure Blob sharable url of attachment,
  deviceImage (boolean, *optional*): The flag to indidate whether this attachment is the main image of device that the attachment belongs to
}

```
{
  "deviceAttachments": [
    {
      "id": "string",
      "name": "string",
      "deviceIds": [
        "string"
      ],
      "uploadedAt": {},
      "uploader": "string",
      "size": 0,
      "url": "string",
      "deviceImage": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
deviceId		query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /deviceAttachments

Implementation Notes

Upload an attachment and associate it with a device. Files with specific names cannot be uploaded, like: crossdomain, htpasswd, etc. Files with specific extensions cannot be uploaded, like: asp, php, css, swf, xhtml, etc.

Response Class (Status 201)

Successful response

Model | Example Value

DeviceAttachment {
- **id** (string): The id of the attachment,
- **name** (string): The name of the attachment,
- **deviceIds** (Array[string]),
- **uploadedAt** (object): The time the device attachment is uploaded,
- **uploader** (string, *optional*): The name of the user who uploaded the attachment,
- **size** (number): The size in bytes of the attachment,
- **url** (string, *optional*): The Azure Blob sharable url of attachment,
- **deviceImage** (boolean, *optional*): The flag to indidate whether this attachment is the main image of device that the attachment belongs to

}

```
{
  "id": "string",
  "name": "string",
  "deviceIds": [
    "string"
  ],
  "uploadedAt": {},
  "uploader": "string",
  "size": 0,
  "url": "string",
  "deviceImage": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`name`	The name of the attachment	formData	string
`deviceId`	The id of the device the attachment is associated with	formData	string
`deviceImage`	The type of the device the attachment is associated with	formData	boolean
`attachment`	The file to upload	formData	file

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /deviceAttachments/{attachmentId}

Implementation Notes

update attachment infomaiton

Response Class (Status 201)

Successful response

Model | Example Value

DeviceAttachment {
id (string): The id of the attachment,
name (string): The name of the attachment,
deviceIds (Array[string]),
uploadedAt (object): The time the device attachment is uploaded,
uploader (string, *optional*): The name of the user who uploaded the attachment,
size (number): The size in bytes of the attachment,
url (string, *optional*): The Azure Blob sharable url of attachment,
deviceImage (boolean, *optional*): The flag to indidate whether this attachment is the main image of device that the attachment belongs to
}

```
{
  "id": "string",
  "name": "string",
  "deviceIds": [
    "string"
  ],
  "uploadedAt": {},
  "uploader": "string",
  "size": 0,
  "url": "string",
  "deviceImage": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`attachmentId`		path	string

`deviceIds`		body	Model | Example Value
			DeviceIdArray {
			deviceIds (Array[string], *optional*)
			}
			`{ "deviceIds": [ "string" ] }`

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /deviceAttachments/{attachmentId}/{deviceId}

Implementation Notes

Delete the attachment data and file in DB and Azure blob

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`attachmentId`		path	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Success response		

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model Example Value
<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model Example Value
<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

POST /deviceImage/{deviceId}/unset

Implementation Notes

Unset current device image

Response Class (Status 200)

Success response

Model | Example Value

EmptyObject {}

```
{}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>404</td><td>Item was not found</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

DELETE /deviceImage/{deviceId}

Implementation Notes

Delete device image data and file in DB and Azure blob

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Success response		

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		  "errors": [
		    {
		      "code": 1000,
		      "message": "Unknown error with API"
		    }
		  ]
		}	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		  "errors": [
		    {
		      "code": 1000,
		      "message": "Unknown error with API"
		    }
		  ]
		}	

GET /deviceImage/{deviceId}

Implementation Notes

Get device image of device

Response Class (Status 200)

Success response

Model | Example Value

DeviceImage {
  deviceImage (DeviceAttachment)
}
DeviceAttachment {
  id (string): The id of the attachment,
  name (string): The name of the attachment,
  deviceIds (Array[string]),
  uploadedAt (object): The time the device attachment is uploaded,
  uploader (string, *optional*): The name of the user who uploaded the attachment,
  size (number): The size in bytes of the attachment,
  url (string, *optional*): The Azure Blob sharable url of attachment,
  deviceImage (boolean, *optional*): The flag to indidate whether this attachment is the main image of device that the attachment belongs to
}

```
{
  "deviceImage": {
    "id": "string",
    "name": "string",
    "deviceIds": [
      "string"
    ],
    "uploadedAt": {},
    "uploader": "string",
    "size": 0,
    "url": "string",
    "deviceImage": true
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /deviceImage/{deviceId}

Implementation Notes

Set new device image to one of the images that are set as device attachment

Response Class (Status 200)

Success response

Model | Example Value

DeviceImage {
  deviceImage (DeviceAttachment)
}
DeviceAttachment {
  id (string): The id of the attachment,
  name (string): The name of the attachment,
  deviceIds (Array[string]),
  uploadedAt (object): The time the device attachment is uploaded,
  uploader (string, *optional*): The name of the user who uploaded the attachment,
  size (number): The size in bytes of the attachment,
  url (string, *optional*): The Azure Blob sharable url of attachment,
  deviceImage (boolean, *optional*): The flag to indidate whether this attachment is the main image of device that the attachment belongs to
}

```
{
  "deviceImage": {
    "id": "string",
    "name": "string",
    "deviceIds": [
      "string"
    ],
    "uploadedAt": {},
    "uploader": "string",
    "size": 0,
    "url": "string",
    "deviceImage": true
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string
`attachmentId`		formData	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /deviceAttachments/{attachmentId}/{deviceId}/url

Implementation Notes

Get Azure Blob sharable url that can be used to download given attachment

Response Class (Status 200)

Success response

Model | Example Value

Url {
  url (string,null): Azure Blob sharable url
}

```
{}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`attachmentId`		path	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

GET /deviceImageUrl/{deviceId}

Implementation Notes

Get Azure Blob sharable url of device image

Response Class (Status 200)

Success response

Model | Example Value

Url {
  url (string,null): Azure Blob sharable url
}

```
{}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

DeviceCategories

GET /deviceCategories

Implementation Notes

Get all device categories with device types, smartObject types and resource types

Response Class (Status 200)

Successful response. An array of device categories is returned.

Model | Example Value

DeviceCategoriesModel {
deviceCategories (Array[DeviceCategoryFull], *optional*): List of device categories with device types in the system,
smartObjectTypes (Array[SmartObjectTypeDeviceCategory], *optional*): List of smart object types in the system,
resourceTypes (Array[ResourceTypeDeviceCategory], *optional*): List of resource types in the system
}
DeviceCategoryFull {
key (number, *optional*): Predefined key for the device category,
name (string, *optional*): Predefined name of a device category,
iconName (string, *optional*): Image name used for marker icons on the map,
description (string, *optional*): Short description of device category,
deviceTypeKeys (Array[number], *optional*): List of all device type keys that are in relation with the correspondindg device category,
deviceTypes (Array[DeviceTypeWithSmartObjectIds], *optional*): List of device types included in device category
}
SmartObjectTypeDeviceCategory {
smartObjectTypeKey (number, *optional*): Predefined key for the smart object type,
name (string, *optional*): Predefined name of a smart object type,
multipleInstances (boolean, *optional*): Flag to define if a device is able to have multiple instances of this specific smart object type,
resourceTypeKeys (Array[number], *optional*): List of resource type keys that belongs to a specific smart object type
}
ResourceTypeDeviceCategory {
resourceTypeKey (number, *optional*): Predefined key for the resource type,
name (string, *optional*): Predefined name of a resource type,
dataType (string, *optional*): Type of data that is stored on a specific resource (examples: 'string', 'integer', 'boolean')
}
DeviceTypeWithSmartObjectIds {
name (string, *optional*): Name of a device type,
key (number, *optional*): Predefined key of a device type,
discovery (boolean, *optional*): Flag to define if we are able to add custom smart objects on a device on the fly, or we just use those predefined smart object types in the device type definition,
smartObjectTypeKeys (Array[number], *optional*): List of smart object type keys that belongs to a specific device type
}

```
{
  "deviceCategories": [
    {
      "key": 0,
      "name": "string",
      "iconName": "string",
      "description": "string",
      "deviceTypeKeys": [
        0
      ],
      "deviceTypes": [
        {
          "name": "string",
          "key": 0,
          "discovery": true,
          "smartObjectTypeKeys": [
            0
          ]
        }
      ]
    }
  ],
  "smartObjectTypes": [
    {
      "smartObjectTypeKey": 0,
      "name": "string",
      "multipleInstances": true,
      "resourceTypeKeys": [
        0
      ]
    }
  ],
  "resourceTypes": [
    {
      "resourceTypeKey": 0,
      "name": "string",
      "dataType": "string"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`refresh`	Refreshing list of categories	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

Device Groups

GET /deviceGroups

Response Class (Status 200)

Successful response

Model | Example Value

DeviceGroupsModel {
deviceGroups (Array[DeviceGroupFromServer], *optional*),
hasRoleEditPermission (boolean, *optional*): whether the current user has role edit permission
}
DeviceGroupFromServer {
id (string): Unique device group identifier,
created (string): Time stamp when the device group is created,
name (string): Name of device group,
ownerType (string, *optional*): Owner type of device group = ['user', 'role', 'org'],
ownerId (string, *optional*): The id of owner of device group,
deviceIds (Array[string]): The id array of devices in the device group
}

```
{
  "deviceGroups": [
    {
      "id": "string",
      "created": "2019-10-03T07:46:34.485Z",
      "name": "string",
      "ownerType": "user",
      "ownerId": "string",
      "deviceIds": [
        "string"
      ]
    }
  ],
  "hasRoleEditPermission": true
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

`400`	Bad Request (Parameter not correct)	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

`401`	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

`403`	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>500</td><td>Not able to process the request</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

POST /deviceGroups

Response Class (Status 201)

Successful response

Model | Example Value

DeviceGroupFromServer {
id (string): Unique device group identifier,
created (string): Time stamp when the device group is created,
name (string): Name of device group,
ownerType (string, *optional*): Owner type of device group = ['user', 'role', 'org'],
ownerId (string, *optional*): The id of owner of device group,
deviceIds (Array[string]): The id array of devices in the device group
}

```
{
  "id": "string",
  "created": "2019-10-03T07:46:34.493Z",
  "name": "string",
  "ownerType": "user",
  "ownerId": "string",
  "deviceIds": [
    "string"
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

body		body	Model | Example Value
			DeviceGroupToServer {
			name (string): Name of device group,
			ownerType (string, *optional*): Owner type of device group = ['user', 'role', 'org'],
			ownerId (string, *optional*): The id of owner of device group,
			deviceIds (Array[string]): The id array of devices in the device group
			}
			`{`
			`"name": "string",`
			`"ownerType": "user",`
			`"ownerId": "string",`
			`"deviceIds": [`
			`"string"`
			`]`
			`}`

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

400	Bad Request (Parameter not correct)	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

401	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

DELETE /deviceGroups/{id}

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**The id of the resource**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	The device group was deleted successfully		

400	Bad Request (Parameter not correct)	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

401	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

GET /deviceGroups/{id}

Response Class (Status 200)

Successful response

Model | Example Value

DeviceGroupFromServer {
id (string): Unique device group identifier,
created (string): Time stamp when the device group is created,
name (string): Name of device group,
ownerType (string, *optional*): Owner type of device group = ['user', 'role', 'org'],
ownerId (string, *optional*): The id of owner of device group,
deviceIds (Array[string]): The id array of devices in the device group
}

```
{
  "id": "string",
  "created": "2019-10-03T07:46:34.513Z",
  "name": "string",
  "ownerType": "user",
  "ownerId": "string",
  "deviceIds": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**The id of the resource**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

400	Bad Request (Parameter not correct)	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

401	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

PUT /deviceGroups/{id}

Response Class (Status 201)

Successful response

Model | Example Value

DeviceGroupFromServer {
id (string): Unique device group identifier,
created (string): Time stamp when the device group is created,
name (string): Name of device group,
ownerType (string, *optional*): Owner type of device group = ['user', 'role', 'org'],
ownerId (string, *optional*): The id of owner of device group,
deviceIds (Array[string]): The id array of devices in the device group
}

```
{
  "id": "string",
  "created": "2019-10-03T07:46:34.530Z",
  "name": "string",
  "ownerType": "user",
  "ownerId": "string",
  "deviceIds": [
    "string"
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id	**The id of the resource**	path	string
body		body	Model | Example Value **DeviceGroupToServer {** **name** (string): Name of device group, **ownerType** (string, *optional*): Owner type of device group = ['user', 'role', 'org'], **ownerId** (string, *optional*): The id of owner of device group, **deviceIds** (Array[string]): The id array of devices in the device group **}**

```
{
  "name": "string",
  "ownerType": "user",
  "ownerId": "string",
  "deviceIds": [
    "string"
  ]
}
```

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>400</td><td>Bad Request (Parameter not correct)</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>401</td><td>API key is missing or invalid</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>403</td><td>User does not have permission to do the operation</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Device Notes

GET /deviceNotes/{deviceId}

Implementation Notes

Get an array of wiki article ids associated with the device

Response Class (Status 200)

Success response

Model | Example Value

deviceNotesArray {

data (Array[string], *optional*): Device note Id array that associated with the given device. Empty array is returned if no device note has been associated with the device.

}

```
{
  "data": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

`401`	Not logged in	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

DeviceTypes

GET /deviceTypes

Implementation Notes

Get deviceTypes based on filter

Response Class (Status 200)

Successful response. All deviceTypes are returned

Model | Example Value

DeviceTypeArray {
deviceTypes (Array[DeviceTypeEnrich], *optional*): DeviceType array
}
DeviceTypeEnrich {
deviceTypeKey (number): DeviceType deviceTypeKey,
name (string): DeviceType name,
discovery (boolean): DeviceType discovery,
smartObjects (Array[SmartObjectType]): DeviceType smartObjectsKeys
}
SmartObjectType {
name (string): Name of the Smart Object Type,
smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
resources (Array[ResourceType]): List of resource types keys,
id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "deviceTypes": [
    {
      "deviceTypeKey": 12345654321,
      "name": "Example deviceType",
      "discovery": false,
      "smartObjects": [
        26312
      ]
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>

Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre>
</td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>

Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre>
</td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /deviceTypes

Implementation Notes

Add specific DeviceType

Response Class (Status 201)

Successful response

Model | Example Value

DeviceTypeCtc {
deviceTypeKey (number): DeviceType deviceTypeKey,
name (string): DeviceType name,
discovery (boolean): DeviceType discovery,
smartObjects (Array[SmartObjectType]): DeviceType smartObjectsKeys,
id (string, *optional*): Id of the object
}
SmartObjectType {
name (string): Name of the Smart Object Type,
smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
resources (Array[ResourceType]): List of resource types keys,
id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "deviceTypeKey": 12345654321,
  "name": "Example deviceType",
  "discovery": false,
  "smartObjects": [
    26312
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type

<table>
<tr><th>Parameter</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr>
<tr><td><b>body</b></td><td><b>DeviceType to be added to the database.</b></td><td>body</td><td>Model | Example Value
<b>DeviceTypeBase {</b>
<b>deviceTypeKey</b> (number): DeviceType deviceTypeKey,
<b>name</b> (string): DeviceType name,
<b>discovery</b> (boolean): DeviceType discovery,
<b>smartObjects</b> (Array[number]): DeviceType smartObjectsKeys
<b>}</b>
<pre>
{
  "deviceTypeKey": 12345654321,
  "name": "Example deviceType",
  "discovery": false,
  "smartObjects": [
    26312
  ]
}
</pre></td></tr>
<tr><td>x-access-token</td><td>User token.</td><td>header</td><td>string</td></tr>
</table>

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown Server Error</td><td>Model | Example Value

<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

DELETE /deviceTypes/{deviceTypeId}

Implementation Notes

Delete the specific DeviceType

Parameters

Parameter	Description	Parameter Type	Data Type
`deviceTypeId`	**deviceTypeId of the DeviceType to be deleted**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The DeviceType is deleted.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /deviceTypes/{deviceTypeId}

Implementation Notes

Get the specific DeviceType

Response Class (Status 200)

Successful response. The selected DeviceType is returned

Model | Example Value

DeviceTypeCtc {
deviceTypeKey (number): DeviceType deviceTypeKey,
name (string): DeviceType name,
discovery (boolean): DeviceType discovery,
smartObjects (Array[SmartObjectType]): DeviceType smartObjectsKeys,
id (string, *optional*): Id of the object
}
SmartObjectType {
name (string): Name of the Smart Object Type,
smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
resources (Array[ResourceType]): List of resource types keys,
id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "deviceTypeKey": 12345654321,
  "name": "Example deviceType",
  "discovery": false,
  "smartObjects": [
    26312
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`deviceTypeId`	**deviceTypeId of the DeviceType to get**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

PUT /deviceTypes/{deviceTypeId}

Implementation Notes

Update the specific DeviceType

Response Class (Status 201)

Successful response. The updated DeviceType is returned

Model | Example Value

DeviceTypeCtc {
  deviceTypeKey (number): DeviceType deviceTypeKey,
  name (string): DeviceType name,
  discovery (boolean): DeviceType discovery,
  smartObjects (Array[SmartObjectType]): DeviceType smartObjectsKeys,
  id (string, *optional*): Id of the object
}
SmartObjectType {
  name (string): Name of the Smart Object Type,
  smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
  multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
  resources (Array[ResourceType]): List of resource types keys,
  id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
  id (string, *optional*): Id of the object,
  name (string): ResourceType name,
  resourceTypeKey (number): ResourceType identifier,
  resourceDataType (string): Type of the data that resourceType provides,
  units (string, *optional*): Units of the ResourceType
}

```
{
  "deviceTypeKey": 12345654321,
  "name": "Example deviceType",
  "discovery": false,
  "smartObjects": [
    26312
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`deviceTypeId`	**deviceTypeId of the DeviceType to update**	path	string

Parameter	Description	Parameter Type	Data Type

body	**New DeviceType object that will replace the old one. if password field is not there it will not change**	body	Model Example Value
			DeviceTypeBase {
			deviceTypeKey (number): DeviceType deviceTypeKey,
			name (string): DeviceType name,
			discovery (boolean): DeviceType discovery,
			smartObjects (Array[number]): DeviceType smartObjectsKeys
			}
			`{`
			`"deviceTypeKey": 12345654321,`
			`"name": "Example deviceType",`
			`"discovery": false,`
			`"smartObjects": [`
			`26312`
			`]`
			`}`

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Devices

GET /devices

Implementation Notes

Get the devices that could be filtered by device type, Service Id, geographical boundaries and fields.

Search: With the search parameter it possible to search for devices based on:

- Name
- Device Identifier

The search is case incentive

Pagination: To enable pagination **page** or **pageSize** parameter must be included in the request. The default pageSize is 200. (base on system wide configuration) Pagination start with page 0. If only the pageSize is included page 0 shall be returned. Pagination works with serviceId and devicetypeId filter and search

Response Class (Status 200)

Successful response. An array of devices is returned.

Model | Example Value

```
DevicesModel {
  devices (Array[OutputDevice], optional): List of available devices,
  filters (object, optional): Listing input fileters if used,
  pagination (PaginationDevice, optional)
}
OutputDevice {
  id (string): Unique device identifier,
  deviceType (DeviceType): Object defining device type for a device and its category,
  externalId (string, optional): Additional custom identifier for a device (doesn't have to be unique),
  deviceIdentifier (string): Additional custom identifier for a device (has to be unique),
  name (string, optional): User defined device name,
  serviceId (string): Identifier of service which device belongs to.,
  smartObjects (Array[OutputSmartObject], optional): List of smart objects belonging to a device,
  metas (object, optional): Static properties for a device,
  location (object,null, optional): Standard GeoJSON format object for defining location of a device,
  alarms (Array[Alarm], optional): List of alarms related to a device,
  notifications (Array[Notification], optional): List of notification related to a device,
  actions (Array[string], optional): List of actions related to a device,
  attachments (Array[Attachment], optional): List of atachments related to a device,
  timestamp (string, optional): Latest measurement received,
  status (object,null, optional),
  state (any,null, optional): Output property for state of a device, values are extracted from state smart object,
  healthScore (integer,null, optional): Output property for health of a device, values are extracted from health score smart object
}
PaginationDevice {
  page (number, optional): Page Number,
  pageSize (number, optional): Size of a Page,
  nbrOfPages (number, optional): Number of pages,
  nbrOfDevices (number, optional): Number of Devices in total
}
DeviceType {
  key (number, optional): Predefined key for a device type,
  name (string, optional): User defined device type name,
```

category (DeviceCategoryObject, *optional*): Related device category for a device type

}

OutputSmartObject {

instanceId (number, *optional*): Identifier of an instance of a smart object unique per device and device type,

smartObjectTypeKey (number): Predefined smart object key of a smart object instance on a device,

name (string, *optional*): Predefine name of a smart object type,

resources (Array[OutputResource], *optional*): List of resources related on a specific device type

}

Alarm {}

Notification {

id (string),

type (string): The types of notification. i.e = ['broadcastNotification', 'toast', 'notification', 'actionNotification'],

userId (string): user id,

title (string): The notification(messages) of title.,

validUntil (string): The time of notification when it is expired .,

timestamp (string): The time of notification.,

persisted (boolean): The notification is persisted or not.,

severity (number): The severity of notification. = ['1', '2'],

meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).,

dismissed (boolean): whether the notification is dismissed.

}

Attachment {}

DeviceCategoryObject {

key (number, *optional*): Predefined key for a device category,

name (string, *optional*): Name of a device category,

iconName (string, *optional*): Image name used for marker icons on the map,

description (string, *optional*): Short description of device category

}

OutputResource {

url (string, *optional*): Unique identifier of a resource,

resourceTypeKey (number): Predefined key defining resource type,

name (string, *optional*): Predefined name of a resource,

timestamp (string, *optional*): Timestamp realated with a resource value,

value (any,null, *optional*): Value of a resource (string, boolean,or integer)

}

```
{
  "devices": [
    {
      "id": "string",
      "deviceType": {
        "key": 0,
        "name": "string",
        "category": {
          "key": 0,
          "name": "string",
          "iconName": "string",
          "description": "string"
        }
      },
      "externalId": "string",
      "deviceIdentifier": "string",
      "name": "string",
      "serviceId": "string",
      "smartObjects": [
        {
          "instanceId": 0,
          "smartObjectTypeKey": 0,
          "name": "string",
          "resources": [
            {
              "url": "<deviceIdentifier>/<smartObjectTypeKey>/<instanceId>/<resourceTypeKey>",
              "resourceTypeKey": 0,
              "name": "string",
              "timestamp": "2019-10-03T07:46:34.649Z",
              "value": "string"
            }
          ]
        }
      ],
      "metas": {},
      "alarms": [
        {}
      ],
      "notifications": [
        {
          "id": "string",
          "type": "broadcastNotification",
          "userId": "string",
          "title": "string",
          "validUntil": "2019-10-03T07:46:34.649Z",
          "timestamp": "2019-10-03T07:46:34.649Z",
          "persisted": true,
          "severity": 1,
          "meta": {},
          "dismissed": true
        }
      ],
      "actions": [
        "string"
      ],
      "attachments": [
        {}
      ],
      "timestamp": "2019-10-03T07:46:34.649Z",
      "state": "",
      "healthScore": "7"
    }
  ],
  "filters": {},
  "pagination": {
    "page": 0,
    "pageSize": 0,
    "nbrOfPages": 0,
    "nbrOfDevices": 0
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`fields`	Comma-separated list	query	string
`deviceTypes`	Comma-separated list of device type keys	query	string
`serviceIds`	Comma-separated list of service ids	query	string
`geo`	Geo object	query	string
`search`	Search a devices	query	string
`page`	Request page of devices with pagination start with 0	query	double
`pageSize`	Request via devices with pagination with max pageSize	query	double

Response Messages

HTTP Status Code	Reason	Response Model	Headers

`403`	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

POST /devices

Implementation Notes

Create device on the server side

Response Class (Status 200)

Successful response.

Model | Example Value

OutputDevice {
id (string): Unique device identifier,
deviceType (DeviceType): Object defining device type for a device and its category,
externalId (string, *optional*): Additional custom identifier for a device (doesn't have to be unique),
deviceIdentifier (string): Additional custom identifier for a device (has to be unique),
name (string, *optional*): User defined device name,
serviceId (string): Identifier of service which device belongs to.,
smartObjects (Array[OutputSmartObject], *optional*): List of smart objects belonging to a device,
metas (object, *optional*): Static properties for a device,
location (object,null, *optional*): Standard GeoJSON format object for defining location of a device,
alarms (Array[Alarm], *optional*): List of alarms related to a device,
notifications (Array[Notification], *optional*): List of notification related to a device,
actions (Array[string], *optional*): List of actions related to a device,
attachments (Array[Attachment], *optional*): List of atachments related to a device,
timestamp (string, *optional*): Latest measurement received,
status (object,null, *optional*),
state (any,null, *optional*): Output property for state of a device, values are extracted from state smart object,
healthScore (integer,null, *optional*): Output property for health of a device, values are extracted from health score smart object
}

DeviceType {
key (number, *optional*): Predefined key for a device type,
name (string, *optional*): User defined device type name,
category (DeviceCategoryObject, *optional*): Related device category for a device type
}

OutputSmartObject {
instanceId (number, *optional*): Identifier of an instance of a smart object unique per device and device type,
smartObjectTypeKey (number): Predefined smart object key of a smart object instance on a device,
name (string, *optional*): Predefine name of a smart object type,
resources (Array[OutputResource], *optional*): List of resources related on a specific device type
}

Alarm {}

Notification {
id (string),
type (string): The types of notification. i.e = ['broadcastNotification', 'toast', 'notification', 'actionNotification'],
userId (string): user id,
title (string): The notification(messages) of title.,
validUntil (string): The time of notification when it is expired .,
timestamp (string): The time of notification.,
persisted (boolean): The notification is persisted or not.,
severity (number): The severity of notification. = ['1', '2'],
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).,
dismissed (boolean): whether the notification is dismissed.
}

Attachment {}

DeviceCategoryObject {
- **key** (number, *optional*): Predefined key for a device category,
- **name** (string, *optional*): Name of a device category,
- **iconName** (string, *optional*): Image name used for marker icons on the map,
- **description** (string, *optional*): Short description of device category

}

OutputResource {
- **url** (string, *optional*): Unique identifier of a resource,
- **resourceTypeKey** (number): Predefined key defining resource type,
- **name** (string, *optional*): Predefined name of a resource,
- **timestamp** (string, *optional*): Timestamp realated with a resource value,
- **value** (any,null, *optional*): Value of a resource (string, boolean,or integer)

}

```
{
  "id": "string",
  "deviceType": {
    "key": 0,
    "name": "string",
    "category": {
      "key": 0,
      "name": "string",
      "iconName": "string",
      "description": "string"
    }
  },
  "externalId": "string",
  "deviceIdentifier": "string",
  "name": "string",
  "serviceId": "string",
  "smartObjects": [
    {
      "instanceId": 0,
      "smartObjectTypeKey": 0,
      "name": "string",
      "resources": [
        {
          "url": "<deviceIdentifier>/<smartObjectTypeKey>/<instanceId>/<resourceTypeKey>",
          "resourceTypeKey": 0,
          "name": "string",
          "timestamp": "2019-10-03T07:46:34.673Z",
          "value": "string"
        }
      ]
    }
  ],
  "metas": {},
  "alarms": [
    {}
  ],
  "notifications": [
    {
      "id": "string",
      "type": "broadcastNotification",
      "userId": "string",
      "title": "string",
      "validUntil": "2019-10-03T07:46:34.673Z",
      "timestamp": "2019-10-03T07:46:34.673Z",
      "persisted": true,
      "severity": 1,
      "meta": {},
      "dismissed": true
    }
  ],
  "actions": [
    "string"
  ],
  "attachments": [
    {}
  ],
  "timestamp": "2019-10-03T07:46:34.674Z",
  "state": "",
  "healthScore": "7"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Parameter	Description	Parameter Type	Data Type
`device`		body	Model Example Value

InputCreateDevice {

deviceTypeKey (number): Device type key which defines a type for a device,

externalId (string): Additional custom identifier for a device (doesn't have to be unique),

deviceIdentifier (string): Additional custom identifier for a device (has to be unique),

name (string): User defined device name,

serviceId (string): Identifier of service which device belongs to.,

smartObjects (Array[InputSmartObject], *optional*): List of smart objects belonging to a device,

metas (object, *optional*): Static properties for a device

}

InputSmartObject {

instanceId (number, *optional*): Identifier of an instance of a smart object unique per device and device type,

smartObjectTypeKey (number): Predefined smart object key for creation of a smart object instance per device,

resources (Array[InputResource]): List of resources connected with the specific smart object on a device

}

InputResource {

resourceTypeKey (number): Predefined resource type key which defines a type of resource to create on a smart object on a device,

timestamp (string, *optional*): Time which we want to connect with the resource value.,

value (any,null, *optional*): Measured value of a resource (integer, string or boolean)

}

```
{
  "deviceTypeKey": 0,
  "externalId": "string",
  "deviceIdentifier": "string",
  "name": "string",
  "serviceId": "string",
  "smartObjects": [
    {
      "instanceId": 0,
      "smartObjectTypeKey": 0,
      "resources": [
        {
          "resourceTypeKey": 0,
          "timestamp": "2019-10-03T07:46:34.683Z",
          "value": "string"
        }
      ]
    }
  ],
  "metas": {}
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

DELETE /devices/{deviceId}

Implementation Notes

Delete the device specified by the deviceId.

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. Empty body object.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

<table>
<thead>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
</thead>
<tbody>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</tbody>
</table>

GET /devices/{deviceId}

Implementation Notes

Get All the information of the device specified by the deviceId.

Response Class (Status 200)

Successful response. The device details for the specified deviceId.

Model | Example Value

OutputDevice {

id (string): Unique device identifier,

deviceType (DeviceType): Object defining device type for a device and its category,

externalId (string, *optional*): Additional custom identifier for a device (doesn't have to be unique),

deviceIdentifier (string): Additional custom identifier for a device (has to be unique),

name (string, *optional*): User defined device name,

serviceId (string): Identifier of service which device belongs to.,

smartObjects (Array[OutputSmartObject], *optional*): List of smart objects belonging to a device,

metas (object, *optional*): Static properties for a device,

location (object,null, *optional*): Standard GeoJSON format object for defining location of a device,

alarms (Array[Alarm], *optional*): List of alarms related to a device,

notifications (Array[Notification], *optional*): List of notification related to a device,

actions (Array[string], *optional*): List of actions related to a device,

attachments (Array[Attachment], *optional*): List of atachments related to a device,

timestamp (string, *optional*): Latest measurement received,

status (object,null, *optional*),

state (any,null, *optional*): Output property for state of a device, values are extracted from state smart object,

healthScore (integer,null, *optional*): Output property for health of a device, values are extracted from health score smart object

}

DeviceType {

key (number, *optional*): Predefined key for a device type,

name (string, *optional*): User defined device type name,

category (DeviceCategoryObject, *optional*): Related device category for a device type

}

OutputSmartObject {

instanceId (number, *optional*): Identifier of an instance of a smart object unique per device and device type,

smartObjectTypeKey (number): Predefined smart object key of a smart object instance on a device,

name (string, *optional*): Predefine name of a smart object type,

resources (Array[OutputResource], *optional*): List of resources related on a specific device type

}

Alarm {}

Notification {

id (string),

type (string): The types of notification. i.e = ['broadcastNotification', 'toast', 'notification', 'actionNotification'],

userId (string): user id,

title (string): The notification(messages) of title.,

validUntil (string): The time of notification when it is expired .,

timestamp (string): The time of notification.,

persisted (boolean): The notification is persisted or not.,

severity (number): The severity of notification. = ['1', '2'],

meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).,

dismissed (boolean): whether the notification is dismissed.

}

Attachment {}

DeviceCategoryObject {

key (number, *optional*): Predefined key for a device category,
name (string, *optional*): Name of a device category,
iconName (string, *optional*): Image name used for marker icons on the map,
description (string, *optional*): Short description of device category

}

OutputResource {

url (string, *optional*): Unique identifier of a resource,
resourceTypeKey (number): Predefined key defining resource type,
name (string, *optional*): Predefined name of a resource,
timestamp (string, *optional*): Timestamp realated with a resource value,
value (any,null, *optional*): Value of a resource (string, boolean,or integer)

}

```
{
  "id": "string",
  "deviceType": {
    "key": 0,
    "name": "string",
    "category": {
      "key": 0,
      "name": "string",
      "iconName": "string",
      "description": "string"
    }
  },
  "externalId": "string",
  "deviceIdentifier": "string",
  "name": "string",
  "serviceId": "string",
  "smartObjects": [
    {
      "instanceId": 0,
      "smartObjectTypeKey": 0,
      "name": "string",
      "resources": [
        {
          "url": "<deviceIdentifier>/<smartObjectTypeKey>/<instanceId>/<resourceTypeKey>",
          "resourceTypeKey": 0,
          "name": "string",
          "timestamp": "2019-10-03T07:46:34.699Z",
          "value": "string"
        }
      ]
    }
  ],
  "metas": {},
  "alarms": [
    {}
  ],
  "notifications": [
    {
      "id": "string",
      "type": "broadcastNotification",
      "userId": "string",
      "title": "string",
      "validUntil": "2019-10-03T07:46:34.700Z",
      "timestamp": "2019-10-03T07:46:34.700Z",
      "persisted": true,
      "severity": 1,
      "meta": {},
      "dismissed": true
    }
  ],
  "actions": [
    "string"
  ],
  "attachments": [
    {}
  ],
  "timestamp": "2019-10-03T07:46:34.700Z",
  "state": "",
  "healthScore": "7"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`deviceId`		path	string
`fields`	Comma-separated list	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /devices/{deviceId}

Implementation Notes

Update the specified Device

Response Class (Status 200)

Successful response.

Model | Example Value

OutputDevice {
id (string): Unique device identifier,
deviceType (DeviceType): Object defining device type for a device and its category,
externalId (string, *optional*): Additional custom identifier for a device (doesn't have to be unique),
deviceIdentifier (string): Additional custom identifier for a device (has to be unique),
name (string, *optional*): User defined device name,
serviceId (string): Identifier of service which device belongs to.,
smartObjects (Array[OutputSmartObject], *optional*): List of smart objects belonging to a device,
metas (object, *optional*): Static properties for a device,
location (object,null, *optional*): Standard GeoJSON format object for defining location of a device,
alarms (Array[Alarm], *optional*): List of alarms related to a device,
notifications (Array[Notification], *optional*): List of notification related to a device,
actions (Array[string], *optional*): List of actions related to a device,
attachments (Array[Attachment], *optional*): List of atachments related to a device,
timestamp (string, *optional*): Latest measurement received,
status (object,null, *optional*),
state (any,null, *optional*): Output property for state of a device, values are extracted from state smart object,
healthScore (integer,null, *optional*): Output property for health of a device, values are extracted from health score smart object
}
DeviceType {
key (number, *optional*): Predefined key for a device type,
name (string, *optional*): User defined device type name,
category (DeviceCategoryObject, *optional*): Related device category for a device type
}
OutputSmartObject {
instanceId (number, *optional*): Identifier of an instance of a smart object unique per device and device type,
smartObjectTypeKey (number): Predefined smart object key of a smart object instance on a device,
name (string, *optional*): Predefine name of a smart object type,
resources (Array[OutputResource], *optional*): List of resources related on a specific device type
}
Alarm {}
Notification {
id (string),
type (string): The types of notification. i.e = ['broadcastNotification', 'toast', 'notification', 'actionNotification'],
userId (string): user id,
title (string): The notification(messages) of title.,
validUntil (string): The time of notification when it is expired .,
timestamp (string): The time of notification.,
persisted (boolean): The notification is persisted or not.,
severity (number): The severity of notification. = ['1', '2'],
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).,
dismissed (boolean): whether the notification is dismissed.
}
Attachment {}

DeviceCategoryObject {
key (number, *optional*): Predefined key for a device category,
name (string, *optional*): Name of a device category,
iconName (string, *optional*): Image name used for marker icons on the map,
description (string, *optional*): Short description of device category
}

OutputResource {
url (string, *optional*): Unique identifier of a resource,
resourceTypeKey (number): Predefined key defining resource type,
name (string, *optional*): Predefined name of a resource,
timestamp (string, *optional*): Timestamp realated with a resource value,
value (any,null, *optional*): Value of a resource (string, boolean,or integer)
}

```
{
  "id": "string",
  "deviceType": {
    "key": 0,
    "name": "string",
    "category": {
      "key": 0,
      "name": "string",
      "iconName": "string",
      "description": "string"
    }
  },
  "externalId": "string",
  "deviceIdentifier": "string",
  "name": "string",
  "serviceId": "string",
  "smartObjects": [
    {
      "instanceId": 0,
      "smartObjectTypeKey": 0,
      "name": "string",
      "resources": [
        {
          "url": "<deviceIdentifier>/<smartObjectTypeKey>/<instanceId>/<resourceTypeKey>",
          "resourceTypeKey": 0,
          "name": "string",
          "timestamp": "2019-10-03T07:46:34.713Z",
          "value": "string"
        }
      ]
    }
  ],
  "metas": {},
  "alarms": [
    {}
  ],
  "notifications": [
    {
      "id": "string",
      "type": "broadcastNotification",
      "userId": "string",
      "title": "string",
      "validUntil": "2019-10-03T07:46:34.713Z",
      "timestamp": "2019-10-03T07:46:34.713Z",
      "persisted": true,
      "severity": 1,
      "meta": {},
      "dismissed": true
    }
  ],
  "actions": [
    "string"
  ],
  "attachments": [
    {}
  ],
  "timestamp": "2019-10-03T07:46:34.713Z",
  "state": "",
  "healthScore": "7"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
deviceId		path	string

Parameter	Description	Parameter Type	Data Type
device		body	Model Example Value **InputPutDevice {** **deviceTypeKey** (number): Device type key which defines a type for a device, **externalId** (string): Additional custom identifier for a device (doesn't have to be unique), **deviceIdentifier** (string): Additional custom identifier for a device (has to be unique), **name** (string, *optional*): User defined device name, **serviceId** (string): Identifier of service which device belongs to., **smartObjects** (Array[InputSmartObject], *optional*): List of smart objects belonging to a device, **metas** (object, *optional*): Static properties for a device **}** **InputSmartObject {** **instanceId** (number, *optional*): Identifier of an instance of a smart object unique per device and device type, **smartObjectTypeKey** (number): Predefined smart object key for creation of a smart object instance per device, **resources** (Array[InputResource]): List of resources connected with the specific smart object on a device **}** **InputResource {** **resourceTypeKey** (number): Predefined resource type key which defines a type of resource to create on a smart object on a device, **timestamp** (string, *optional*): Time which we want to connect with the resource value., **value** (any,null, *optional*): Measured value of a resource (integer, string or boolean) **}**

```
{
  "deviceTypeKey": 0,
  "externalId": "string",
  "deviceIdentifier": "string",
  "name": "string",
  "serviceId": "string",
  "smartObjects": [
    {
      "instanceId": 0,
      "smartObjectTypeKey": 0,
      "resources": [
        {
          "resourceTypeKey": 0,
          "timestamp": "2019-10-03T07:46:34.722Z",
          "value": "string"
        }
      ]
    }
  ],
  "metas": {}
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

Documents

GET /documents

Implementation Notes

Get product documents meta information. **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request

Response Class (Status 200)

Successful response. Return all documents meta data.

Model | Example Value

DocumentArrayWithObjects {
  documents (Array[DocumentWithObjects])
}
DocumentWithObjects {
  title (string): Display title of the document,
  documentNumber (string, *optional*): Ericsson Document Number,
  documentName (string, *optional*): Document name/type,
  description (string): Description of the document,
  revision (string): revision number of the document,
  organization (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec,
  id (string): Id of the document,
  url (string): Url of the document,
  created (string): Timestamp when the object was created
}

```
{
  "documents": [
    {
      "title": "User Guide CTC UI",
      "documentNumber": "ETM-14:002325 Uen",
      "documentName": "User Guide",
      "description": "User guide of CTC UI for Dummies",
      "revision": "A",
      "organization": "Ericsson"
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`search-by-content`	Query for search docs by content	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /documents

Implementation Notes

Create new document meta data, This call is only suppored for Ericsson admin users **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request

Response Class (Status 201)

Successful response. Return created document meta data.

Model | Example Value

DocumentWithObjects {
title (string): Display title of the document,
documentNumber (string, *optional*): Ericsson Document Number,
documentName (string, *optional*): Document name/type,
description (string): Description of the document,
revision (string): revision number of the document,
organization (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec,
id (string): Id of the document,
url (string): Url of the document,
created (string): Timestamp when the object was created
}

```
{
  "title": "User Guide CTC UI",
  "documentNumber": "ETM-14:002325 Uen",
  "documentName": "User Guide",
  "description": "User guide of CTC UI for Dummies",
  "revision": "A",
  "organization": "Ericsson"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
body		body	Model Example Value **DocumentBase {** **title** (string): Display title of the document, **documentNumber** (string, *optional*): Ericsson Document Number, **documentName** (string, *optional*): Document name/type, **description** (string): Description of the document, **revision** (string): revision number of the document, **organization** (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec **}**

```
{
  "title": "User Guide CTC UI",
  "documentNumber": "ETM-14:002325 Uen",
  "documentName": "User Guide",
  "description": "User guide of CTC UI for Dummies",
  "revision": "A",
  "organization": "Ericsson"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

401	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

DELETE /documents/{documentId}

Implementation Notes

Delete document meta data by documentId, This call is only suppored by Ericsson admin users. **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request 404 | 1504 | Requested document is not found

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`documentId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. No Body		

401	API key is missing or invalid	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

GET /documents/{documentId}

Implementation Notes

Get document meta data by documentId **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request 404 | 1504 | Requested document is not found

Response Class (Status 200)

Successful response. Return document meta data.

Model | Example Value

DocumentWithObjects {
- **title** (string): Display title of the document,
- **documentNumber** (string, *optional*): Ericsson Document Number,
- **documentName** (string, *optional*): Document name/type,
- **description** (string): Description of the document,
- **revision** (string): revision number of the document,
- **organization** (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec,
- **id** (string): Id of the document,
- **url** (string): Url of the document,
- **created** (string): Timestamp when the object was created

}

```
{
  "title": "User Guide CTC UI",
  "documentNumber": "ETM-14:002325 Uen",
  "documentName": "User Guide",
  "description": "User guide of CTC UI for Dummies",
  "revision": "A",
  "organization": "Ericsson"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
documentId		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

`403`	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

`404`	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

`500`	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

PUT /documents/{documentId}

Implementation Notes

Update document meta data by documentId, This call is only suppored by Ericsson admin users. **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request 404 | 1504 | Requested document is not found

Response Class (Status 201)

Successful response. Return updated document meta data.

Model | Example Value

DocumentWithObjects {

- **title** (string): Display title of the document,
- **documentNumber** (string, *optional*): Ericsson Document Number,
- **documentName** (string, *optional*): Document name/type,
- **description** (string): Description of the document,
- **revision** (string): revision number of the document,
- **organization** (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec,
- **id** (string): Id of the document,
- **url** (string): Url of the document,
- **created** (string): Timestamp when the object was created

}

```
{
  "title": "User Guide CTC UI",
  "documentNumber": "ETM-14:002325 Uen",
  "documentName": "User Guide",
  "description": "User guide of CTC UI for Dummies",
  "revision": "A",
  "organization": "Ericsson"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
documentId		path	string

Parameter	Description	Parameter Type	Data Type
body		body	Model Example Value

DocumentBase {

title (string): Display title of the document,
documentNumber (string, *optional*): Ericsson Document Number,
documentName (string, *optional*): Document name/type,
description (string): Description of the document,
revision (string): revision number of the document,
organization (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec

}

```
{
  "title": "User Guide CTC UI",
  "documentNumber": "ETM-14:002325 Uen",
  "documentName": "User Guide",
  "description": "User guide of CTC UI for Dummies",
  "revision": "A",
  "organization": "Ericsson"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	API key is missing or invalid	Model Example Value	

Errors {

errors (Array[Error])

}

Error {

code (integer): Four-digits error code,
message (string, *optional*): Error description

}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

GET /documents/{documentId}/download

Implementation Notes

Download document by documentId **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request 404 | 1504 | Requested document is not found

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`documentId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
302	Redirect to download location		Location string

401	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /documents/{documentId}/upload

Implementation Notes

Upload document file for documentId Call will overwrite current document if exist This call is only suppored by Ericsson admin. **Error Response Info:** Status Code| Error Code | Description -- | -- | -- 500 | 1500 | Unknown error with document request 404 | 1504 | Requested document is not found

Response Class (Status 201)

Successful response. Return document meta data.

Model | Example Value

DocumentWithObjects {
title (string): Display title of the document,
documentNumber (string, *optional*): Ericsson Document Number,
documentName (string, *optional*): Document name/type,
description (string): Description of the document,
revision (string): revision number of the document,
organization (string, *optional*): Name of the organization that is the owner of the document (Ericsson) not connected to a organization objec,
id (string): Id of the document,
url (string): Url of the document,
created (string): Timestamp when the object was created
}

```
{
  "title": "User Guide CTC UI",
  "documentNumber": "ETM-14:002325 Uen",
  "documentName": "User Guide",
  "description": "User guide of CTC UI for Dummies",
  "revision": "A",
  "organization": "Ericsson"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`documentId`		path	string
`file`	**The file to upload.**	formData	file

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

401	API key is missing or invalid	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

404	Item was not found	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

EventTemplates

GET /eventTemplates

Implementation Notes

Get Event Template objects. The devices that belong to a service can trigger pre-defined Events. The definition of these events is done through an "Event Template". This "event template" is connected to a service and applies to devices of a specific type.

Response Class (Status 200)

Successful response. The Event Templates are returned.

Model | Example Value

eventTemplateArray {
 eventTemplates (Array[eventTemplate], *optional*): Array of Event Templates
}
eventTemplate {
 name (string): Name of the Event Template for display,
 description (string, *optional*): Description of the Event Template for display,
 serviceId (string): Id of the service this Template belongs to,
 deviceTypeKey (number): Device Type where this Event Template is linked to,
 condition (object): Condition of this Event Template,
 id (string): Id of occured Event Template
}

```
{
  "eventTemplates": [
    {
      "id": "e1ae7679-e97c-41fe-8342-1ff7a86da773",
      "name": "Easy Mile Low Battery",
      "description": "Monitors the Battery power and triggers when the % is below 5%",
      "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
      "deviceTypeKey": 100006,
      "condition": {
        "check": {
          "resourceTypeKey": 5513,
          "smartObjectTypeKey": 3336,
          "value": "Ten",
          "operator": "StringEquals",
          "resolutionAggregationType": "Realtime"
        }
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`serviceId`	serviceId to get filtered Event Templates	query	string

Parameter	Description	Parameter Type	Data Type
`deviceTypeKey`	deviceTypeKey to get filtered Event Templates	query	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>500</td><td>Unknown server error</td><td>

Model | Example Value

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre>
</td><td></td></tr>
</table>

POST /eventTemplates

Implementation Notes

Add specific Event Template.

Response Class (Status 201)

Successful response. The created shall be Event Template returned.

Model | Example Value

eventTemplate {
name (string): Name of the Event Template for display,
description (string, *optional*): Description of the Event Template for display,
serviceId (string): Id of the service this Template belongs to,
deviceTypeKey (number): Device Type where this Event Template is linked to,
condition (object): Condition of this Event Template,
id (string): Id of occured Event Template
}

```
{
  "id": "e1ae7679-e97c-41fe-8342-1ff7a86da773",
  "name": "Easy Mile Low Battery",
  "description": "Monitors the Battery power and triggers when the % is below 5%",
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 100006,
  "condition": {
    "check": {
      "resourceTypeKey": 5513,
      "smartObjectTypeKey": 3336,
      "value": "Ten",
      "operator": "StringEquals",
      "resolutionAggregationType": "Realtime"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
body	**Event Template to be added to the database.**	body	Model Example Value **eventTemplateWithoutId {** **name** (string): Name of the Event Template for display, **description** (string, *optional*): Description of the Event Template for display, **serviceId** (string): Id of the service this Template belongs to, **deviceTypeKey** (number): Device Type where this Event Template is linked to, **condition** (object): Condition of this Event Template **}**
`x-access-token`	User token.	header	string

```
{
  "name": "Easy Mile Low Battery",
  "description": "Monitors the Battery power and triggers when the % is below 5%",
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 100006,
  "condition": {
    "check": {
      "resourceTypeKey": 5513,
      "smartObjectTypeKey": 3336,
      "value": 5,
      "triggerOperator": "NumberLowerThan",
      "resolutionAggregationType": "Realtime"
    }
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /eventTemplates/{eventTemplateId}

Implementation Notes

Delete the specific Event Template from the solution.

Parameters

Parameter	Description	Parameter Type	Data Type
`eventTemplateId`	**eventTemplateId to delete Event Template**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The Event Template is deleted.		

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /eventTemplates/{eventTemplateId}

Implementation Notes

Get the specific Event Template.

Response Class (Status 200)

Successful response.The selected Event Template is returned.

Model | Example Value

eventTemplate {
name (string): Name of the Event Template for display,
description (string, *optional*): Description of the Event Template for display,
serviceId (string): Id of the service this Template belongs to,
deviceTypeKey (number): Device Type where this Event Template is linked to,
condition (object): Condition of this Event Template,
id (string): Id of occured Event Template
}

```
{
  "id": "e1ae7679-e97c-41fe-8342-1ff7a86da773",
  "name": "Easy Mile Low Battery",
  "description": "Monitors the Battery power and triggers when the % is below 5%",
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 100006,
  "condition": {
    "check": {
      "resourceTypeKey": 5513,
      "smartObjectTypeKey": 3336,
      "value": "Ten",
      "operator": "StringEquals",
      "resolutionAggregationType": "Realtime"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`eventTemplateId`	**eventTemplateId to get Event Template**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

PUT /eventTemplates/{eventTemplateId}

Implementation Notes

Get a specific Event Template

Response Class (Status 201)

Successful response. The updated Event Template is returned.

Model | Example Value

eventTemplate {
name (string): Name of the Event Template for display,
description (string, *optional*): Description of the Event Template for display,
serviceId (string): Id of the service this Template belongs to,
deviceTypeKey (number): Device Type where this Event Template is linked to,
condition (object): Condition of this Event Template,
id (string): Id of occured Event Template
}

```
{
  "id": "e1ae7679-e97c-41fe-8342-1ff7a86da773",
  "name": "Easy Mile Low Battery",
  "description": "Monitors the Battery power and triggers when the % is below 5%",
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 100006,
  "condition": {
    "check": {
      "resourceTypeKey": 5513,
      "smartObjectTypeKey": 3336,
      "value": "Ten",
      "operator": "StringEquals",
      "resolutionAggregationType": "Realtime"
    }
  }
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`eventTemplateId`	**eventTemplateId to update Event Template**	path	string

<table>
<tr><th>Parameter</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr>
<tr><td><b>body</b></td><td><b>New Event Template object that will replace the old one.</b></td><td>body</td><td>Model | Example Value
<b>eventTemplateWithoutId {</b>
<b>name</b> (string): Name of the Event Template for display,
<b>description</b> (string, <i>optional</i>): Description of the Event Template for display,
<b>serviceId</b> (string): Id of the service this Template belongs to,
<b>deviceTypeKey</b> (number): Device Type where this Event Template is linked to,
<b>condition</b> (object): Condition of this Event Template
<b>}</b>
<pre>
{
  "name": "Easy Mile Low Battery",
  "description": "Monitors the Battery power and triggers when the % is l
  "serviceId": "db39fd3e-7ad5-44be-a817-01fcb608efbb",
  "deviceTypeKey": 100006,
  "condition": {
    "check": {
      "resourceTypeKey": 5513,
      "smartObjectTypeKey": 3336,
      "value": 5,
      "triggerOperator": "NumberLowerThan",
      "resolutionAggregationType": "Realtime"
    }
  }
}
</pre></td></tr>
<tr><td><code>x-access-token</code></td><td>User token.</td><td>header</td><td>string</td></tr>
</table>

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td><code>403</code></td><td>User does not have permission to do the operation</td><td>Model | Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Layer Groups

GET /layerGroups

Implementation Notes

Get all Layer Groups.

Response Class (Status 200)

Successful response. The layer groups list is returned.

Model | Example Value

LayerGroups {
  layerGroups (Array[layerGroup], *optional*): All layer groups
}
layerGroup {
  stackOrder (number): The order of the group in the stack of groups,
  name (string): The name of this layer group,
  organizationId (string): Organization id of owner organization of layer group,
  domain (string): The domain of layer group,
  type (Array[string]): Allowed types of layers,
  layerIds (Array[string]): Layers linked to this layer group,
  disabled (boolean, *optional*): Is the layer group disabled
}

```
{
  "layerGroups": [
    {
      "stackOrder": 0,
      "name": "string",
      "organizationId": "string",
      "domain": "string",
      "type": [
        "string"
      ],
      "layerIds": [
        "string"
      ],
      "disabled": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationId`	Filter layer groups by organization	query	string
`domains`	Comma-separated list of domains	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

POST /layerGroups

Implementation Notes

Create Layer group

Response Class (Status 201)

Successful response. Returns created Layer group.

Model | Example Value

layerGroup {
stackOrder (number): The order of the group in the stack of groups,
name (string): The name of this layer group,
organizationId (string): Organization id of owner organization of layer group,
domain (string): The domain of layer group,
type (Array[string]): Allowed types of layers,
layerIds (Array[string]): Layers linked to this layer group,
disabled (boolean, *optional*): Is the layer group disabled
}

```
{
  "stackOrder": 0,
  "name": "string",
  "organizationId": "string",
  "domain": "string",
  "type": [
    "string"
  ],
  "layerIds": [
    "string"
  ],
  "disabled": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Parameter	Description	Parameter Type	Data Type
body	**Layer group to be added to the database.**	body	Model Example Value **layerGroup {** **stackOrder** (number): The order of the group in the stack of groups, **name** (string): The name of this layer group, **organizationId** (string): Organization id of owner organization of layer group, **domain** (string): The domain of layer group, **type** (Array[string]): Allowed types of layers, **layerIds** (Array[string]): Layers linked to this layer group, **disabled** (boolean, *optional*): Is the layer group disabled **}**

```
{
  "stackOrder": 0,
  "name": "string",
  "organizationId": "string",
  "domain": "string",
  "type": [
    "string"
  ],
  "layerIds": [
    "string"
  ],
  "disabled": true
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /layerGroups/{id}

Implementation Notes

Delete Layer Group

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**id of the layer group to be deleted**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

PUT /layerGroups/{id}

Implementation Notes

Update the Layer group

Response Class (Status 201)

Successful response. Action is done, layer group is returned.

Model | Example Value

layerGroup {
stackOrder (number): The order of the group in the stack of groups,
name (string): The name of this layer group,
organizationId (string): Organization id of owner organization of layer group,
domain (string): The domain of layer group,
type (Array[string]): Allowed types of layers,
layerIds (Array[string]): Layers linked to this layer group,
disabled (boolean, *optional*): Is the layer group disabled
}

```
{
  "stackOrder": 0,
  "name": "string",
  "organizationId": "string",
  "domain": "string",
  "type": [
    "string"
  ],
  "layerIds": [
    "string"
  ],
  "disabled": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**id of the layer group to update**	path	string

Parameter	Description	Parameter Type	Data Type
`layer group`	**New layer group that will replace the old one.**	body	Model Example Value **layerGroup {** **stackOrder** (number): The order of the group in the stack of groups, **name** (string): The name of this layer group, **organizationId** (string): Organization id of owner organization of layer group, **domain** (string): The domain of layer group, **type** (Array[string]): Allowed types of layers, **layerIds** (Array[string]): Layers linked to this layer group, **disabled** (boolean, *optional*): Is the layer group disabled **}**

```
{
  "stackOrder": 0,
  "name": "string",
  "organizationId": "string",
  "domain": "string",
  "type": [
    "string"
  ],
  "layerIds": [
    "string"
  ],
  "disabled": true
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

`403`	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

Layers

GET /layers

Implementation Notes

Get all layers.

Response Class (Status 200)

Successful response. Layers are returned.

Model | Example Value

Layers {
  layers (Array[Layer], *optional*): All layers
}
Layer {
  name (string): Name of the layer,
  domain (string): Domain of current layer,
  type (string): Type of current layer,
  serviceId (string): Id of service which layer belongs to,
  meta (object): List of key-value pairs where keys are customizable Strings,
  id (string, *optional*): Id of the layer
}

```
{
  "layers": [
    {
      "name": "Layer name",
      "domain": "devices",
      "type": "devices",
      "serviceId": "00000000-0000-0000-0000-000000001",
      "meta": {
        "deviceType": 101001
      }
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

POST /layers

Implementation Notes

Create layer

Response Class (Status 201)

Successful response. The created layer is returned.

Model | Example Value

Layer {
name (string): Name of the layer,
domain (string): Domain of current layer,
type (string): Type of current layer,
serviceId (string): Id of service which layer belongs to,
meta (object): List of key-value pairs where keys are customizable Strings,
id (string, *optional*): Id of the layer
}

```
{
  "name": "Layer name",
  "domain": "devices",
  "type": "devices",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "meta": {
    "deviceType": 101001
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

body	**Layer to be added to the database.**	body	Model | Example Value
			LayerBody {
			name (string): Name of the layer,
			domain (string): Domain of current layer,
			type (string): Type of current layer,
			serviceId (string): Id of service which layer belongs to,
			meta (object): List of key-value pairs where keys are customizable Strings
			}
			`{`
			`"name": "Layer name",`
			`"domain": "devices",`
			`"type": "devices",`
			`"serviceId": "00000000-0000-0000-0000-000000001",`
			`"meta": {`
			`"deviceType": 101001`
			`}`
			`}`

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

DELETE /layer/{layerId}

Implementation Notes

Delete the specific layer

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`layerId`	**layerId of the layer to be deleted**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The layer is deleted.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /layer/{layerId}

Implementation Notes

Get the specific layer

Response Class (Status 200)

Successful response. The selected layer is returned.

Model | Example Value

Layer {
name (string): Name of the layer,
domain (string): Domain of current layer,
type (string): Type of current layer,
serviceId (string): Id of service which layer belongs to,
meta (object): List of key-value pairs where keys are customizable Strings,
id (string, *optional*): Id of the layer
}

```
{
  "name": "Layer name",
  "domain": "devices",
  "type": "devices",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "meta": {
    "deviceType": 101001
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`layerId`	**layerId of the layer to get**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

PUT /layer/{layerId}

Implementation Notes

Update a specific layer

Response Class (Status 201)

Successful response. The updated layer is returned.

Model | Example Value

Layer {
  name (string): Name of the layer,
  domain (string): Domain of current layer,
  type (string): Type of current layer,
  serviceId (string): Id of service which layer belongs to,
  meta (object): List of key-value pairs where keys are customizable Strings,
  id (string, *optional*): Id of the layer
}

```
{
  "name": "Layer name",
  "domain": "devices",
  "type": "devices",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "meta": {
    "deviceType": 101001
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`layerId`	**layerId of the layer to update**	path	string

Parameter	Description	Parameter Type	Data Type
`layer`	**New layer to replace the old one (only layer name will be updated)**	body	Model Example Value **Layer {** **name** (string): Name of the layer, **domain** (string): Domain of current layer, **type** (string): Type of current layer, **serviceId** (string): Id of service which layer belongs to, **meta** (object): List of key-value pairs where keys are customizable Strings, **id** (string, *optional*): Id of the layer **}**

```
{
  "name": "Layer name",
  "domain": "devices",
  "type": "devices",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "meta": {
    "deviceType": 101001
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

GET /layers/{domain}

Implementation Notes

Get layers by domain.

Response Class (Status 200)

Successful response. Layers are returned.

Model | Example Value

inline_model_4 {
  layers (Array[Layer], *optional*)
}
Layer {
  name (string): Name of the layer,
  domain (string): Domain of current layer,
  type (string): Type of current layer,
  serviceId (string): Id of service which layer belongs to,
  meta (object): List of key-value pairs where keys are customizable Strings,
  id (string, *optional*): Id of the layer
}

```
{
  "layers": [
    {
      "name": "Layer name",
      "domain": "devices",
      "type": "devices",
      "serviceId": "00000000-0000-0000-0000-000000001",
      "meta": {
        "deviceType": 101001
      }
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`domain`	**Domain of the layers**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Map data

GET /mapdata

Implementation Notes

Get set of map data

Response Class (Status 200)

Successful response. Return set of map data info

Model | Example Value

mapDatas [
mapData
]
mapData {
id (string, *optional*): id of map data layer,
name (string): name of map data layer,
description (string, *optional*): description of map data layer,
layerUrl (string, *optional*): layer url of map data layer,
licenseKey (string, *optional*): license key of map data layer,
serviceId (string): service id of map data layer,
type (string): type of map data layer,
metas (metasData, *optional*): meta id of map data layer
}
metasData {
style (metaStyle, *optional*): style of meta,
maxNativeZoom (number, *optional*): max native zoom of meta,
subdomains (Array[string], *optional*): subdomain of meta,
attribution (string, *optional*): attribution of meta
}
metaStyle {
color (string, *optional*): Color of meta style. It can be a color name or hex color value, i.e. 'red' or '#3388ff',
opacity (number, *optional*): Opacity of meta style. Opacity must be a number from 0 to 1,
layer (layer, *optional*): Layer style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#path-option for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity,
circleMarker (circleMarker, *optional*): CircleMarker style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#circlemarker for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity
}
layer {
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}
circleMarker {
radius (number, *optional*): Radius of the circle marker, in pixels,
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles.,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,

opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles.,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}

```
[
  {
    "id": "string",
    "name": "string",
    "description": "string",
    "layerUrl": "string",
    "licenseKey": "string",
    "serviceId": "string",
    "type": "string",
    "metas": {
      "style": {
        "color": "string",
        "opacity": 0,
        "layer": {
          "stroke": true,
          "color": "string",
          "weight": 0,
          "opacity": 0,
          "fill": true,
          "fillColor": "string",
          "fillOpacity": 0
        },
        "circleMarker": {
          "radius": 0,
          "stroke": true,
          "color": "string",
          "weight": 0,
          "opacity": 0,
          "fill": true,
          "fillColor": "string",
          "fillOpacity": 0
        }
      },
      "maxNativeZoom": 0,
      "subdomains": [
        "string"
      ],
      "attribution": "string"
    }
  }
]
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`mapDataType`	mapDataType key to get filtered map data	query	string

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value
<b>Errors {</b>
errors (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model | Example Value
<b>Errors {</b>
errors (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

POST /mapdata

Implementation Notes

Create a new map data

Response Class (Status 201)

Successful response. The created map data is returned

Model | Example Value

mapData {
id (string, *optional*): id of map data layer,
name (string): name of map data layer,
description (string, *optional*): description of map data layer,
layerUrl (string, *optional*): layer url of map data layer,
licenseKey (string, *optional*): license key of map data layer,
serviceId (string): service id of map data layer,
type (string): type of map data layer,
metas (metasData, *optional*): meta id of map data layer
}
metasData {
style (metaStyle, *optional*): style of meta,
maxNativeZoom (number, *optional*): max native zoom of meta,
subdomains (Array[string], *optional*): subdomain of meta,
attribution (string, *optional*): attribution of meta
}
metaStyle {
color (string, *optional*): Color of meta style. It can be a color name or hex color value, i.e. 'red' or '#3388ff',
opacity (number, *optional*): Opacity of meta style. Opacity must be a number from 0 to 1,
layer (layer, *optional*): Layer style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#path-option for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity,
circleMarker (circleMarker, *optional*): CircleMarker style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#circlemarker for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity
}
layer {
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}
circleMarker {
radius (number, *optional*): Radius of the circle marker, in pixels,
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles.,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles.,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "layerUrl": "string",
  "licenseKey": "string",
  "serviceId": "string",
  "type": "string",
  "metas": {
    "style": {
      "color": "string",
      "opacity": 0,
      "layer": {
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      },
      "circleMarker": {
        "radius": 0,
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      }
    },
    "maxNativeZoom": 0,
    "subdomains": [
      "string"
    ],
    "attribution": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
body	**Map data to be added to the database.**	body	Model | Example Value **mapData {** **id** (string, *optional*): id of map data layer, **name** (string): name of map data layer, **description** (string, *optional*): description of map data layer, **layerUrl** (string, *optional*): layer url of map data layer, **licenseKey** (string, *optional*): license key of map data layer, **serviceId** (string): service id of map data layer, **type** (string): type of map data layer, **metas** (metasData, *optional*): meta id of map data layer

Parameter	Description	Parameter Type	Data Type

			}
			metasData {
			style (metaStyle, *optional*): style of meta,
			maxNativeZoom (number, *optional*): max native zoom of meta,
			subdomains (Array[string], *optional*): subdomain of meta,
			attribution (string, *optional*): attribution of meta
			}
			metaStyle {
			color (string, *optional*): Color of meta style. It can be a color name or hex color value, i.e. 'red' or '#3388ff',
			opacity (number, *optional*): Opacity of meta style. Opacity must be a number from 0 to 1,
			layer (layer, *optional*): Layer style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#path-option for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity,
			circleMarker (circleMarker, *optional*): CircleMarker style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#circlemarker for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity
			}
			layer {
			stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles,
			color (string, *optional*): Stroke color,
			weight (number, *optional*): Stroke width in pixels,
			opacity (number, *optional*): Stroke opacity,
			fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles,
			fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
			fillOpacity (number, *optional*): Fill opacity
			}
			circleMarker {
			radius (number, *optional*): Radius of the circle marker, in pixels,
			stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles.,
			color (string, *optional*): Stroke color,
			weight (number, *optional*): Stroke width in pixels,
			opacity (number, *optional*): Stroke opacity,
			fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles.,
			fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
			fillOpacity (number, *optional*): Fill opacity
			}
			`{`
			`  "id": "string",`
			`  "name": "string",`

Parameter	Description	Parameter Type	Data Type

```
  "description": "string",
  "layerUrl": "string",
  "licenseKey": "string",
  "serviceId": "string",
  "type": "string",
  "metas": {
    "style": {
      "color": "string",
      "opacity": 0,
      "layer": {
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      },
      "circleMarker": {
        "radius": 0,
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      }
    },
    "maxNativeZoom": 0,
    "subdomains": [
      "string"
    ],
    "attribution": "string"
  }
}
```

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model | Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

DELETE /mapdata/{mapDataId}

Implementation Notes

Delete the map data by mapDataId

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`mapDataId`	**layer id of mapData**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The map data is deleted		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /mapdata/{mapDataId}

Implementation Notes

Get map data by mapDataId.

Response Class (Status 200)

Successful response. Return a map data's info

Model | Example Value

mapData {
id (string, *optional*): id of map data layer,
name (string): name of map data layer,
description (string, *optional*): description of map data layer,
layerUrl (string, *optional*): layer url of map data layer,
licenseKey (string, *optional*): license key of map data layer,
serviceId (string): service id of map data layer,
type (string): type of map data layer,
metas (metasData, *optional*): meta id of map data layer
}
metasData {
style (metaStyle, *optional*): style of meta,
maxNativeZoom (number, *optional*): max native zoom of meta,
subdomains (Array[string], *optional*): subdomain of meta,
attribution (string, *optional*): attribution of meta
}
metaStyle {
color (string, *optional*): Color of meta style. It can be a color name or hex color value, i.e. 'red' or '#3388ff',
opacity (number, *optional*): Opacity of meta style. Opacity must be a number from 0 to 1,
layer (layer, *optional*): Layer style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#path-option for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity,
circleMarker (circleMarker, *optional*): CircleMarker style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#circlemarker for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity
}
layer {
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}
circleMarker {
radius (number, *optional*): Radius of the circle marker, in pixels,
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles.,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles.,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "layerUrl": "string",
  "licenseKey": "string",
  "serviceId": "string",
  "type": "string",
  "metas": {
    "style": {
      "color": "string",
      "opacity": 0,
      "layer": {
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      },
      "circleMarker": {
        "radius": 0,
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      }
    },
    "maxNativeZoom": 0,
    "subdomains": [
      "string"
    ],
    "attribution": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`mapDataId`	**layer id of mapData**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

PUT /mapdata/{mapDataId}

Implementation Notes

Update the map data by mapDataId

Response Class (Status 201)

Successful response. The updated map data is returned

Model | Example Value

mapData {
id (string, *optional*): id of map data layer,
name (string): name of map data layer,
description (string, *optional*): description of map data layer,
layerUrl (string, *optional*): layer url of map data layer,
licenseKey (string, *optional*): license key of map data layer,
serviceId (string): service id of map data layer,
type (string): type of map data layer,
metas (metasData, *optional*): meta id of map data layer
}
metasData {
style (metaStyle, *optional*): style of meta,
maxNativeZoom (number, *optional*): max native zoom of meta,
subdomains (Array[string], *optional*): subdomain of meta,
attribution (string, *optional*): attribution of meta
}
metaStyle {
color (string, *optional*): Color of meta style. It can be a color name or hex color value, i.e. 'red' or '#3388ff',
opacity (number, *optional*): Opacity of meta style. Opacity must be a number from 0 to 1,
layer (layer, *optional*): Layer style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#path-option for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity,
circleMarker (circleMarker, *optional*): CircleMarker style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#circlemarker for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity
}
layer {
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}
circleMarker {
radius (number, *optional*): Radius of the circle marker, in pixels,
stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles.,
color (string, *optional*): Stroke color,
weight (number, *optional*): Stroke width in pixels,
opacity (number, *optional*): Stroke opacity,
fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles.,
fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
fillOpacity (number, *optional*): Fill opacity
}

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "layerUrl": "string",
  "licenseKey": "string",
  "serviceId": "string",
  "type": "string",
  "metas": {
    "style": {
      "color": "string",
      "opacity": 0,
      "layer": {
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      },
      "circleMarker": {
        "radius": 0,
        "stroke": true,
        "color": "string",
        "weight": 0,
        "opacity": 0,
        "fill": true,
        "fillColor": "string",
        "fillOpacity": 0
      }
    },
    "maxNativeZoom": 0,
    "subdomains": [
      "string"
    ],
    "attribution": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
body	**The new map data object that will replace the old one.**	body	Model Example Value **mapData {** **id** (string, *optional*): id of map data layer, **name** (string): name of map data layer, **description** (string, *optional*): description of map data layer, **layerUrl** (string, *optional*): layer url of map data layer, **licenseKey** (string, *optional*): license key of map data layer, **serviceId** (string): service id of map data layer, **type** (string): type of map data layer, **metas** (metasData, *optional*): meta id of map data layer

Parameter	Description	Parameter Type	Data Type

			}
			metasData {
			style (metaStyle, *optional*): style of meta,
			maxNativeZoom (number, *optional*): max native zoom of meta,
			subdomains (Array[string], *optional*): subdomain of meta,
			attribution (string, *optional*): attribution of meta
			}
			metaStyle {
			color (string, *optional*): Color of meta style. It can be a color name or hex color value, i.e. 'red' or '#3388ff',
			opacity (number, *optional*): Opacity of meta style. Opacity must be a number from 0 to 1,
			layer (layer, *optional*): Layer style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#path-option for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity,
			circleMarker (circleMarker, *optional*): CircleMarker style options. Only ERSI feature layer supports this setting. Refer to https://leafletjs.com/reference.html#circlemarker for more details about the supported options. The color and opacity in this setting will override the setting of style.color and style.opacity
			}
			layer {
			stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles,
			color (string, *optional*): Stroke color,
			weight (number, *optional*): Stroke width in pixels,
			opacity (number, *optional*): Stroke opacity,
			fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles,
			fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
			fillOpacity (number, *optional*): Fill opacity
			}
			circleMarker {
			radius (number, *optional*): Radius of the circle marker, in pixels,
			stroke (boolean, *optional*): Whether to draw stroke along the path. Set it to false to disable borders on polygons or circles.,
			color (string, *optional*): Stroke color,
			weight (number, *optional*): Stroke width in pixels,
			opacity (number, *optional*): Stroke opacity,
			fill (boolean, *optional*): Whether to fill the path with color. Set it to false to disable filling on polygons or circles.,
			fillColor (string, *optional*): Fill color. Defaults to the value of the color option,
			fillOpacity (number, *optional*): Fill opacity
			}
			`{`
			`"id": "string",`
			`"name": "string",`

<table>
<tr><th>Parameter</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr>
<tr><td></td><td></td><td></td><td><pre>
    "description": "string",
    "layerUrl": "string",
    "licenseKey": "string",
    "serviceId": "string",
    "type": "string",
    "metas": {
      "style": {
        "color": "string",
        "opacity": 0,
        "layer": {
          "stroke": true,
          "color": "string",
          "weight": 0,
          "opacity": 0,
          "fill": true,
          "fillColor": "string",
          "fillOpacity": 0
        },
        "circleMarker": {
          "radius": 0,
          "stroke": true,
          "color": "string",
          "weight": 0,
          "opacity": 0,
          "fill": true,
          "fillColor": "string",
          "fillOpacity": 0
        }
      },
      "maxNativeZoom": 0,
      "subdomains": [
        "string"
      ],
      "attribution": "string"
    }
  }
</pre></td></tr>
<tr><td><code>x-access-token</code></td><td>User token.</td><td>header</td><td>string</td></tr>
<tr><td><b><code>mapDataId</code></b></td><td><b>layer id of mapData</b></td><td>path</td><td>string</td></tr>
</table>

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /shapeFiles

Implementation Notes

Post a shape files

Response Class (Status 201)

Successful response. Return metadata of uploaded shape file.

Model | Example Value

shapeFile {
id (string): id of shape file,
name (string): name of shape file,
description (string): description of shape file,
metas (shapeFileMeta, *optional*): meta of shape file
}
shapeFileMeta {
uploadTimeStamp (string, *optional*): upload time of shape file
}

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "metas": {
    "uploadTimeStamp": "2019-10-03T07:46:35.024Z"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`shapeFile`	**The file to upload.**	formData	file
`name`	name	formData	string
`description`	description	formData	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Unexpected error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

PUT /shapeFiles/{shapeFileId}

Implementation Notes

Update shape file content and metadata.

Response Class (Status 201)

Successful response. Return shape file.

Model | Example Value

shapeFile {
id (string): id of shape file,
name (string): name of shape file,
description (string): description of shape file,
metas (shapeFileMeta, *optional*): meta of shape file
}
shapeFileMeta {
uploadTimeStamp (string, *optional*): upload time of shape file
}

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "metas": {
    "uploadTimeStamp": "2019-10-03T07:46:35.031Z"
  }
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`shapeFileId`	**id of the object to update**	path	string
`shapeFile`	The file to upload.	formData	file
`name`	name	formData	string
`description`	description	formData	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Unexpected error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Measurements

POST /measurements

Implementation Notes

Sending device measurements

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`measurements`		body	Model Example Value **Measurements {** **measurements** (Array[Measurement]) **}** **Measurement {** **url** (string): unique resource identifier: deviceIdentifier/smartObjectTypeKey/instanceId/resourceTypeKey , **timestamp** (string, *optional*): Timestamp of the value changed, **value** (any): The changed value **}**

```
{
  "measurements": [
    {
      "url": "Device test externalId/3336/0/5513",
      "timestamp": "2018-03-16T08:59:16.632Z",
      "value": "changed value"
    }
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. Empty response body		

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

POST /measurements/read

Implementation Notes

Getting devices measurements. Max available urls amount by default set to 1000

Response Class (Status 200)

Successful response. List of devices measurements

Model | Example Value

Measurements {
  measurements (Array[Measurement])
}
Measurement {
  url (string): unique resource identifier: deviceIdentifier/smartObjectTypeKey/instanceId/resourceTypeKey,
  timestamp (string, *optional*): Timestamp of the value changed,
  value (any): The changed value
}

```
{
  "measurements": [
    {
      "url": "Device test externalId/3336/0/5513",
      "timestamp": "2018-03-16T08:59:16.632Z",
      "value": "changed value"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

urls	**List of resources urls**	body	Model | Example Value
			MeasurementsUrls {
			  urls (Array[string])
			}
			`{`
			`  "urls": [`
			`    "string"`
			`  ]`
			`}`

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

413	Counts of requested urls more than 1000	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /measurements/history

Implementation Notes

Getting devices measurements history

Response Class (Status 200)

Successful response. List of devices measurements

Model | Example Value

Measurements {
  measurements (Array[Measurement])
}
Measurement {
  url (string): unique resource identifier: deviceIdentifier/smartObjectTypeKey/instanceId/resourceTypeKey,
  timestamp (string, *optional*): Timestamp of the value changed,
  value (any): The changed value
}

```
{
  "measurements": [
    {
      "url": "Device test externalId/3336/0/5513",
      "timestamp": "2018-03-16T08:59:16.632Z",
      "value": "changed value"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
timestampFrom	Get device measurements history after the timestamp.	query	date-time
timestampTo	Get device measurements history before the timestamp.	query	date-time
deviceId	Filter measurements by device id	query	string
serviceId	Filter measurements by service id	query	string
deviceTypes	Comma-separated list of device type keys	query	string
smartObjectTypeKeys	Comma-separated list of smart object type keys	query	string
resourceTypeKeys	Comma-separated list of resource type keys	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Notifications

GET /notifications

Implementation Notes

Get notifications. Notifications are user-specific messages that are typically created from an automated rule.

Response Class (Status 200)

Successful response. Notifications returned

Model | Example Value

Notifications {
  Notification (Array[Notification], *optional*): all of the Notifications
}
Notification {
  id (string),
  type (string): The types of notification. i.e = ['broadcastNotification', 'toast', 'notification', 'actionNotification'],
  userId (string): user id,
  title (string): The notification(messages) of title.,
  validUntil (string): The time of notification when it is expired .,
  timestamp (string): The time of notification.,
  persisted (boolean): The notification is persisted or not.,
  severity (number): The severity of notification. = ['1', '2'],
  meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).,
  dismissed (boolean): whether the notification is dismissed.
}

```
{
  "Notification": [
    {
      "id": "string",
      "type": "broadcastNotification",
      "userId": "string",
      "title": "string",
      "validUntil": "2019-10-03T07:46:35.072Z",
      "timestamp": "2019-10-03T07:46:35.072Z",
      "persisted": true,
      "severity": 1,
      "meta": {},
      "dismissed": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

422	Unprocessable Entity	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /notifications

Implementation Notes

Create notification. Notifications are user-specific messages that are typically created from an automated rule.

Response Class (Status 201)

Successful response. The created shall be action type returned.

Model | Example Value

Empty

```
{}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
notification		body	Model Example Value

NotificationPostData {
  notification (NotificationDetail, *optional*),
  recipients (Recipients, *optional*),
  actionIds (Array[string], *optional*): Action Ids used for actionNotification action type
}

NotificationDetail {
  type (string),
  title (string),
  validUntil (string),
  persisted (boolean),
  severity (number),
  meta (object, *optional*)
}

Recipients {
  users (Array[string], *optional*): Users' Ids,
  roles (Array[string], *optional*): Roles' Ids,
  groups (Array[string], *optional*): Groups' Ids,
  organizations (Array[string], *optional*): Organizations' Ids
}

```
{
  "notification": {
    "type": "string",
    "title": "string",
    "validUntil": "string",
    "persisted": true,
    "severity": 0,
    "meta": {}
  },
  "recipients": {
    "users": [
      "string"
    ],
    "roles": [
      "string"
    ],
    "groups": [
      "string"
    ],
    "organizations": [
      "string"
    ]
  },
  "actionIds": [
    "string"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /notifications/dismiss/{id}

Implementation Notes

Dismiss notification by id

Response Class (Status 200)

Successful response. Notification dismissed

Model | Example Value

Notification {
id (string),
type (string): The types of notification. i.e = ['broadcastNotification', 'toast', 'notification', 'actionNotification'],
userId (string): user id,
title (string): The notification(messages) of title.,
validUntil (string): The time of notification when it is expired .,
timestamp (string): The time of notification.,
persisted (boolean): The notification is persisted or not.,
severity (number): The severity of notification. = ['1', '2'],
meta (object): List of key-value pairs where keys are customizable Strings. The value can be anything (String, Array, Object, and so on).,
dismissed (boolean): whether the notification is dismissed.
}

```
{
  "id": "string",
  "type": "broadcastNotification",
  "userId": "string",
  "title": "string",
  "validUntil": "2019-10-03T07:46:35.089Z",
  "timestamp": "2019-10-03T07:46:35.089Z",
  "persisted": true,
  "severity": 1,
  "meta": {},
  "dismissed": true
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`	**Identifier of the notification to be dismissed**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

<table>
<thead>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
</thead>
<tbody>
<tr><td>422</td><td>Unprocessable Entity</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown Server Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</tbody>
</table>

Organizations

GET /organizations

Implementation Notes

Get all organizations

Response Class (Status 200)

Successful response. Return all organizations

Model | Example Value

OrganizationArray {
  organizations (Array[Organization], *optional*)
}
Organization {
  id (string, *optional*): Id of the organization,
  name (string): Name of the organization,
  description (string): Name of the organization,
  address (string): Address of the organization,
  email (string): Email of the organization,
  telephone (string): Telephone of the organization,
  url (string): URL of the website,
  authentication (authentication): Object defining authentication type and domains for users which can be created in this organization,
  license (string): organization license data,
  storageSize (number): storage size,
  maxFileSize (number): max file size,
  sessionInactivityTimeout (number, *optional*): session inactivity timeout in seconds,
  deviceMetasTemplates (deviceMetasTemplates, *optional*): Meta templates for device types
}
authentication {
  domains (Array[string]): Allowed authentication domains for users which can be created in this organization,
  type (string): Allowed authentication type for users which can be created in this organization = ['DB', 'AAD', 'AADEXT']
}
deviceMetasTemplates {}

```
{
  "organizations": [
    {
      "name": "Nitzsche LLC",
      "description": "Organization description",
      "address": "352 Nona Islands",
      "email": "Mark_Langosh@yahoo.com",
      "telephone": "(634) 356-8525",
      "url": "https://mckenzie.biz",
      "authentication": {
        "domains": [
          "yahoo.com"
        ],
        "type": "DB"
      },
      "license": "organization license",
      "storageSize": 1024,
      "maxFileSize": 1024,
      "sessionInactivityTimeout": 3600,
      "deviceMetasTemplates": {
        "23456": [
          {
            "name": "vehicleType",
            "type": "enum",
            "values": [
              "None",
              "Standard",
              "Heavy Duty"
            ],
            "editable": true,
            "own": false
          },
          {
            "name": "schedulable",
            "type": "boolean",
            "editable": true,
            "own": false
          },
          {
            "name": "trackable",
            "type": "boolean",
            "editable": true,
            "own": true
          }
        ]
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /organizations

Implementation Notes

create the organization details

Response Class (Status 201)

Successful response.

Model | Example Value

Organization {
id (string, *optional*): Id of the organization,
name (string): Name of the organization,
description (string): Name of the organization,
address (string): Address of the organization,
email (string): Email of the organization,
telephone (string): Telephone of the organization,
url (string): URL of the website,
authentication (authentication): Object defining authentication type and domains for users which can be created in this organization,
license (string): organization license data,
storageSize (number): storage size,
maxFileSize (number): max file size,
sessionInactivityTimeout (number, *optional*): session inactivity timeout in seconds,
deviceMetasTemplates (deviceMetasTemplates, *optional*): Meta templates for device types
}
authentication {
domains (Array[string]): Allowed authentication domains for users which can be created in this organization,
type (string): Allowed authentication type for users which can be created in this organization = ['DB', 'AAD', 'AADEXT']
}
deviceMetasTemplates {}

```
{
  "name": "Nitzsche LLC",
  "description": "Organization description",
  "address": "352 Nona Islands",
  "email": "Mark_Langosh@yahoo.com",
  "telephone": "(634) 356-8525",
  "url": "https://mckenzie.biz",
  "authentication": {
    "domains": [
      "yahoo.com"
    ],
    "type": "DB"
  },
  "license": "organization license",
  "storageSize": 1024,
  "maxFileSize": 1024,
  "sessionInactivityTimeout": 3600,
  "deviceMetasTemplates": {
    "23456": [
      {
        "name": "vehicleType",
        "type": "enum",
        "values": [
          "None",
          "Standard",
          "Heavy Duty"
        ],
        "editable": true,
        "own": false
      },
      {
        "name": "schedulable",
        "type": "boolean",
        "editable": true,
        "own": false
      },
      {
        "name": "trackable",
        "type": "boolean",
        "editable": true,
        "own": true
      }
    ]
  }
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
Organization		body	Model | Example Value **Organization {** **id** (string, *optional*): Id of the organization, **name** (string): Name of the organization, **description** (string): Name of the organization, **address** (string): Address of the organization, **email** (string): Email of the organization, **telephone** (string): Telephone of the organization, **url** (string): URL of the website, **authentication** (authentication): Object defining authentication type and domains for users which can be created in this organization, **license** (string): organization license data, **storageSize** (number): storage size, **maxFileSize** (number): max file size, **sessionInactivityTimeout** (number, *optional*): session inactivity timeout in seconds, **deviceMetasTemplates** (deviceMetasTemplates, *optional*): Meta templates for device types **}** **authentication {** **domains** (Array[string]): Allowed authentication domains for users which can be created in this organization, **type** (string): Allowed authentication type for users which can be created in this organization = ['DB', 'AAD', 'AADEXT'] **}** **deviceMetasTemplates {}**

Parameter	Description	Parameter Type	Data Type

```
{
  "name": "Nitzsche LLC",
  "description": "Organization description",
  "address": "352 Nona Islands",
  "email": "Mark_Langosh@yahoo.com",
  "telephone": "(634) 356-8525",
  "url": "https://mckenzie.biz",
  "authentication": {
    "domains": [
      "yahoo.com"
    ],
    "type": "DB"
  },
  "license": "organization license",
  "storageSize": 1024,
  "maxFileSize": 1024,
  "sessionInactivityTimeout": 3600,
  "deviceMetasTemplates": {
    "23456": [
      {
        "name": "vehicleType",
        "type": "enum",
        "values": [
          "None",
          "Standard",
          "Heavy Duty"
        ],
        "editable": true,
        "own": false
      },
      {
        "name": "schedulable",
        "type": "boolean",
        "editable": true,
        "own": false
      },
      {
        "name": "trackable",
        "type": "boolean",
        "editable": true,
        "own": true
      }
    ]
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

DELETE /organizations/{organizationId}

Implementation Notes

Delete organization with id.

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. Empty body object.		

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /organizations/{organizationId}

Implementation Notes

Get a organization by id.

Response Class (Status 200)

Successful response. Return a organization's detail info

Model | Example Value

Organization {
id (string, *optional*): Id of the organization,
name (string): Name of the organization,
description (string): Name of the organization,
address (string): Address of the organization,
email (string): Email of the organization,
telephone (string): Telephone of the organization,
url (string): URL of the website,
authentication (authentication): Object defining authentication type and domains for users which can be created in this organization,
license (string): organization license data,
storageSize (number): storage size,
maxFileSize (number): max file size,
sessionInactivityTimeout (number, *optional*): session inactivity timeout in seconds,
deviceMetasTemplates (deviceMetasTemplates, *optional*): Meta templates for device types
}
authentication {
domains (Array[string]): Allowed authentication domains for users which can be created in this organization,
type (string): Allowed authentication type for users which can be created in this organization = ['DB', 'AAD', 'AADEXT']
}
deviceMetasTemplates {}

```
{
  "name": "Nitzsche LLC",
  "description": "Organization description",
  "address": "352 Nona Islands",
  "email": "Mark_Langosh@yahoo.com",
  "telephone": "(634) 356-8525",
  "url": "https://mckenzie.biz",
  "authentication": {
    "domains": [
      "yahoo.com"
    ],
    "type": "DB"
  },
  "license": "organization license",
  "storageSize": 1024,
  "maxFileSize": 1024,
  "sessionInactivityTimeout": 3600,
  "deviceMetasTemplates": {
    "23456": [
      {
        "name": "vehicleType",
        "type": "enum",
        "values": [
          "None",
          "Standard",
          "Heavy Duty"
        ],
        "editable": true,
        "own": false
      },
      {
        "name": "schedulable",
        "type": "boolean",
        "editable": true,
        "own": false
      },
      {
        "name": "trackable",
        "type": "boolean",
        "editable": true,
        "own": true
      }
    ]
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

PUT /organizations/{organizationId}

Implementation Notes

update the organization details

Response Class (Status 201)

Successful response.

Model | Example Value

Organization {
id (string, *optional*): Id of the organization,
name (string): Name of the organization,
description (string): Name of the organization,
address (string): Address of the organization,
email (string): Email of the organization,
telephone (string): Telephone of the organization,
url (string): URL of the website,
authentication (authentication): Object defining authentication type and domains for users which can be created in this organization,
license (string): organization license data,
storageSize (number): storage size,
maxFileSize (number): max file size,
sessionInactivityTimeout (number, *optional*): session inactivity timeout in seconds,
deviceMetasTemplates (deviceMetasTemplates, *optional*): Meta templates for device types
}
authentication {
domains (Array[string]): Allowed authentication domains for users which can be created in this organization,
type (string): Allowed authentication type for users which can be created in this organization = ['DB', 'AAD', 'AADEXT']
}
deviceMetasTemplates {}

```
{
  "name": "Nitzsche LLC",
  "description": "Organization description",
  "address": "352 Nona Islands",
  "email": "Mark_Langosh@yahoo.com",
  "telephone": "(634) 356-8525",
  "url": "https://mckenzie.biz",
  "authentication": {
    "domains": [
      "yahoo.com"
    ],
    "type": "DB"
  },
  "license": "organization license",
  "storageSize": 1024,
  "maxFileSize": 1024,
  "sessionInactivityTimeout": 3600,
  "deviceMetasTemplates": {
    "23456": [
      {
        "name": "vehicleType",
        "type": "enum",
        "values": [
          "None",
          "Standard",
          "Heavy Duty"
        ],
        "editable": true,
        "own": false
      },
      {
        "name": "schedulable",
        "type": "boolean",
        "editable": true,
        "own": false
      },
      {
        "name": "trackable",
        "type": "boolean",
        "editable": true,
        "own": true
      }
    ]
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationId`	**id of the organization to update**	path	string
`Organization`		body	Model Example Value **Organization {** **id** (string, *optional*): Id of the organization, **name** (string): Name of the organization, **description** (string): Name of the organization, **address** (string): Address of the organization, **email** (string): Email of the organization, **telephone** (string): Telephone of the organization, **url** (string): URL of the website, **authentication** (authentication): Object defining authentication type and domains for users which can be created in this organization, **license** (string): organization license data, **storageSize** (number): storage size, **maxFileSize** (number): max file size, **sessionInactivityTimeout** (number, *optional*): session inactivity timeout in seconds, **deviceMetasTemplates** (deviceMetasTemplates, *optional*): Meta templates for device types **}** **authentication {** **domains** (Array[string]): Allowed authentication domains for users which can be created in this organization, **type** (string): Allowed authentication type for users which can be created in this organization = ['DB', 'AAD', 'AADEXT']

Parameter	Description	Parameter Type	Data Type

			}
			deviceMetasTemplates {}

```
{
  "name": "Nitzsche LLC",
  "description": "Organization description",
  "address": "352 Nona Islands",
  "email": "Mark_Langosh@yahoo.com",
  "telephone": "(634) 356-8525",
  "url": "https://mckenzie.biz",
  "authentication": {
    "domains": [
      "yahoo.com"
    ],
    "type": "DB"
  },
  "license": "organization license",
  "storageSize": 1024,
  "maxFileSize": 1024,
  "sessionInactivityTimeout": 3600,
  "deviceMetasTemplates": {
    "23456": [
      {
        "name": "vehicleType",
        "type": "enum",
        "values": [
          "None",
          "Standard",
          "Heavy Duty"
        ],
        "editable": true,
        "own": false
      },
      {
        "name": "schedulable",
        "type": "boolean",
        "editable": true,
        "own": false
      },
      {
        "name": "trackable",
        "type": "boolean",
        "editable": true,
        "own": true
      }
    ]
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Permissions

GET /permissions

Implementation Notes

Get the list of all permissions

Response Class (Status 200)

Successful response. Returns all permissions.

Model | Example Value

PermissionArray {
  permissions (Array[Permission], *optional*)
}
Permission {
  name (string, *optional*): Name of the permission,
  serviceId (string): The service id associated with the permssion,
  domain (string): The domain for the permission,
  subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
  domainActions (Array[string]): The list of actions associated with the domain,
  description (string, *optional*),
  id (string, *optional*)
}

```
{
  "permissions": [
    {
      "id": "00000000-0000-0000-0000-000000001",
      "name": "Permission for Easymile bus type 1",
      "serviceId": "00000000-0000-0000-0000-000000001",
      "domain": "device",
      "subDomains": [
        "easymilebusType1"
      ],
      "domainActions": [
        "view",
        "add"
      ],
      "description": "Permission to view and to add Easymile bus type 1"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /permissions

Implementation Notes

Create a new permission

Response Class (Status 201)

Successful response.

Model | Example Value

Permission {

name (string, *optional*): Name of the permission,
serviceId (string): The service id associated with the permssion,
domain (string): The domain for the permission,
subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
domainActions (Array[string]): The list of actions associated with the domain,
description (string, *optional*),
id (string, *optional*)

}

```
{
  "id": "00000000-0000-0000-0000-000000001",
  "name": "Permission for Easymile bus type 1",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "domain": "device",
  "subDomains": [
    "easymilebusType1"
  ],
  "domainActions": [
    "view",
    "add"
  ],
  "description": "Permission to view and to add Easymile bus type 1"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Parameter	Description	Parameter Type	Data Type
permission		body	Model Example Value **PermissionBase {** **name** (string, *optional*): Name of the permission, **serviceId** (string): The service id associated with the permssion, **domain** (string): The domain for the permission, **subDomains** (Array[string], *optional*): The subdomain associated/connected to the domain, **domainActions** (Array[string]): The list of actions associated with the domain, **description** (string, *optional*) **}**

```
{
  "name": "Permission for Easymile bus type 1",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "domain": "device",
  "subDomains": [
    "easymilebusType1"
  ],
  "domainActions": [
    "view",
    "add"
  ],
  "description": "Permission to view and to add Easymile bus type 1"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /permissions/{permissionId}

Implementation Notes

Delete permission by permission id

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`permissionId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response.		

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /permissions/{permissionId}

Implementation Notes

Get permission by permission id

Response Class (Status 200)

Successful response. Return a permission detail info

Model | Example Value

Permission {
name (string, *optional*): Name of the permission,
serviceId (string): The service id associated with the permssion,
domain (string): The domain for the permission,
subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
domainActions (Array[string]): The list of actions associated with the domain,
description (string, *optional*),
id (string, *optional*)
}

```
{
  "id": "00000000-0000-0000-0000-000000001",
  "name": "Permission for Easymile bus type 1",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "domain": "device",
  "subDomains": [
    "easymilebusType1"
  ],
  "domainActions": [
    "view",
    "add"
  ],
  "description": "Permission to view and to add Easymile bus type 1"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`permissionId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

PUT /permissions/{permissionId}

Implementation Notes

Update existing permission

Response Class (Status 201)

Successful response.

Model | Example Value

Permission {
name (string, *optional*): Name of the permission,
serviceId (string): The service id associated with the permssion,
domain (string): The domain for the permission,
subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
domainActions (Array[string]): The list of actions associated with the domain,
description (string, *optional*),
id (string, *optional*)
}

```
{
  "id": "00000000-0000-0000-0000-000000001",
  "name": "Permission for Easymile bus type 1",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "domain": "device",
  "subDomains": [
    "easymilebusType1"
  ],
  "domainActions": [
    "view",
    "add"
  ],
  "description": "Permission to view and to add Easymile bus type 1"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`permissionId`	**permissionId of the permission to update**	path	string

Parameter	Description	Parameter Type	Data Type

body	**New permission object that will replace the old one**	body	Model Example Value
			PermissionBase {
			name (string, *optional*): Name of the permission,
			serviceId (string): The service id associated with the permssion,
			domain (string): The domain for the permission,
			subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
			domainActions (Array[string]): The list of actions associated with the domain,
			description (string, *optional*)
			}
			`{`
			`"name": "Permission for Easymile bus type 1",`
			`"serviceId": "00000000-0000-0000-0000-000000001",`
			`"domain": "device",`
			`"subDomains": [`
			`"easymilebusType1"`
			`],`
			`"domainActions": [`
			`"view",`
			`"add"`
			`],`
			`"description": "Permission to view and to add Easymile bus type 1"`
			`}`

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

`403`	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

GET /permissions/organization/{organizationId}

Implementation Notes

Get the list of all permissions for the specified organization id

Response Class (Status 200)

Successful response. Returns all permissions for the specified organization id

Model | Example Value

PermissionArray {
  permissions (Array[Permission], *optional*)
}
Permission {
  name (string, *optional*): Name of the permission,
  serviceId (string): The service id associated with the permssion,
  domain (string): The domain for the permission,
  subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
  domainActions (Array[string]): The list of actions associated with the domain,
  description (string, *optional*),
  id (string, *optional*)
}

```
{
  "permissions": [
    {
      "id": "00000000-0000-0000-0000-000000001",
      "name": "Permission for Easymile bus type 1",
      "serviceId": "00000000-0000-0000-0000-000000001",
      "domain": "device",
      "subDomains": [
        "easymilebusType1"
      ],
      "domainActions": [
        "view",
        "add"
      ],
      "description": "Permission to view and to add Easymile bus type 1"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
organizationId		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

GET	/permissions/user/{userId}

Implementation Notes

Get the list of all permissions for the specified user id

Response Class (Status 200)

Successful response. Returns all permissions for the specified user id

Model | Example Value

PermissionArray {
  permissions (Array[Permission], *optional*)
}
Permission {
  name (string, *optional*): Name of the permission,
  serviceId (string): The service id associated with the permssion,
  domain (string): The domain for the permission,
  subDomains (Array[string], *optional*): The subdomain associated/connected to the domain,
  domainActions (Array[string]): The list of actions associated with the domain,
  description (string, *optional*),
  id (string, *optional*)
}

```
{
  "permissions": [
    {
      "id": "00000000-0000-0000-0000-000000001",
      "name": "Permission for Easymile bus type 1",
      "serviceId": "00000000-0000-0000-0000-000000001",
      "domain": "device",
      "subDomains": [
        "easymilebusType1"
      ],
      "domainActions": [
        "view",
        "add"
      ],
      "description": "Permission to view and to add Easymile bus type 1"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Preferences

DELETE /preferences

Implementation Notes

Delete several preferences

Response Class (Status 200)

Return the array of deletion status objects. This doesn't mean that deletions were performed just that deletion request was processed. Individual per preference deletion statuses are returned as a response body in the same order of the provided preference IDs.

Model | Example Value

Inline Model [
  Inline Model 1
]
Inline Model 1 {
  success (boolean): True if deletion of preference was successful. False otherwise.,
  data (object): If success is false returns the object describing the error.
}

```
[
  {
    "success": false,
    "data": {
      "name": "CTC Error",
      "message": "Preference with ID first_pref_id does not exist",
      "code": 404
    }
  }
]
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
guids	**comma-separated list of preference guids**	query	Array[string]

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model | Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /preferences

Implementation Notes

Get all preferences

Response Class (Status 200)

Successful response. Return all preferences

Model | Example Value

PreferenceModel {
  preferences (Array[Preference], *optional*)
}
Preference {
  id (string,null, *optional*): id of preference,
  name (string): name of preference,
  key (string, *optional*): key of preference,
  type (string, *optional*): type of preference = ['system', 'organization', 'role', 'user'],
  value (object, *optional*): value of preference,
  serviceId (string, *optional*): service id of preference,
  ownerId (string, *optional*): owner id of preference,
  ownerName (string, *optional*): owner name of preference
}

```
{
  "preferences": [
    {
      "id": "00000000-0000-0000-0000-000000001",
      "name": "User1",
      "key": "core_windowLayout",
      "type": "user",
      "serviceId": "00000000-0000-0000-0000-000000001",
      "ownerId": "00000000-0000-0000-0000-000000002",
      "ownerName": "eric",
      "value": {
        "key1": "value1",
        "key2": "value2"
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceIds`	Filter on all preferences with a specifc serviceId. Multiple fields can be specified separated by a comma (,)	query	Array[string]

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	User does not have permission to do the operation	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /preferences

Implementation Notes

Create new preference

Response Class (Status 201)

Successful response. Return the created preference

Model | Example Value

Preference {
id (string,null, *optional*): id of preference,
name (string): name of preference,
key (string, *optional*): key of preference,
type (string, *optional*): type of preference = ['system', 'organization', 'role', 'user'],
value (object, *optional*): value of preference,
serviceId (string, *optional*): service id of preference,
ownerId (string, *optional*): owner id of preference,
ownerName (string, *optional*): owner name of preference
}

```json
{
  "id": "00000000-0000-0000-0000-000000001",
  "name": "User1",
  "key": "core_windowLayout",
  "type": "user",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "ownerId": "00000000-0000-0000-0000-000000002",
  "ownerName": "eric",
  "value": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
`preference`		body	Model Example Value

Preference {
- **id** (string,null, *optional*): id of preference,
- **name** (string): name of preference,
- **key** (string, *optional*): key of preference,
- **type** (string, *optional*): type of preference = ['system', 'organization', 'role', 'user'],
- **value** (object, *optional*): value of preference,
- **serviceId** (string, *optional*): service id of preference,
- **ownerId** (string, *optional*): owner id of preference,
- **ownerName** (string, *optional*): owner name of preference

}

```
{
  "id": "00000000-0000-0000-0000-000000001",
  "name": "User1",
  "key": "core_windowLayout",
  "type": "user",
  "serviceId": "00000000-0000-0000-0000-000000001",
  "ownerId": "00000000-0000-0000-0000-000000002",
  "ownerName": "eric",
  "value": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Not authenticated.	string	
403	User does not have permission to do the operation	Model Example Value	

Errors {
- **errors** (Array[Error])

}

Error {
- **code** (integer): Four-digits error code,
- **message** (string, *optional*): Error description

}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

DELETE /preferences/{preferenceId}

Implementation Notes

Delete preference with id.

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`preferenceId`		path	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>204</td><td>Successful response. The preference is deleted.</td><td></td><td></td></tr>
<tr><td>401</td><td>Not authenticated.</td><td>string</td><td></td></tr>
<tr><td>403</td><td>User does not have permission to do the operation</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Not able to process the request</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

PubSub Server

POST /pubsub/consumer

Implementation Notes

Authentification endpoint for consumer of the PubSub Server

Response Class (Status 200)

Successful response. PubSub socket, ttl and fallback urls are returned.

Model | Example Value

pubsubConsumer {
  pubsub (pubsub, *optional*): wss or ws of pubsub,
  fallback (object, *optional*): channels of pubsub
}
pubsub {
  wss (string, *optional*): https pubsub authentification url,
  ws (string, *optional*): http pubsub authentification url,
  ttl (string, *optional*): current time
}

```
{
  "pubsub": {
    "wss": "string",
    "ws": "string",
    "ttl": "string"
  },
  "fallback": {}
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /pubsub/producer

Implementation Notes

Authentification endpoint for producer of the PubSub Server

Response Class (Status 200)

Successful response. PubSub socket and ttl are returned.

Model | Example Value

pubsubProducer {
  pubsub (pubsub, *optional*)
}
pubsub {
  wss (string, *optional*): https pubsub authentification url,
  ws (string, *optional*): http pubsub authentification url,
  ttl (string, *optional*): current time
}

```
{
  "pubsub": {
    "wss": "string",
    "ws": "string",
    "ttl": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
422	Not able to process the request	Model | Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

ResourceTypes

GET /resourceTypes

Implementation Notes

Get resourceTypes based on filter

Response Class (Status 200)

Successful response. All resourceTypes are returned

Model | Example Value

ResourceTypeArray {
smartObjectTypes (Array[ResourceType], *optional*): ResourceType array
}
ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "smartObjectTypes": [
    {
      "id": "id1",
      "name": "Example ResourceType",
      "resourceTypeKey": 49,
      "resourceDataType": "String"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /resourceTypes

Implementation Notes

Add specific ResourceType

Response Class (Status 201)

Successful response

Model | Example Value

ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "id": "id1",
  "name": "Example ResourceType",
  "resourceTypeKey": 49,
  "resourceDataType": "String"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

body	**ResourceType to be added to the database.**	body	Model | Example Value
			ResourceTypeBase {
			name (string): ResourceType name,
			resourceTypeKey (number): ResourceType identifier,
			resourceDataType (string): Type of the data that resourceType provides,
			units (string, *optional*): Units of the ResourceType
			}
			`{`
			`"name": "Example ResourceType",`
			`"resourceTypeKey": 49,`
			`"resourceDataType": "String"`
			`}`

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

DELETE /resourceTypes/{resourceTypeId}

Implementation Notes

Delete the specific ResourceType

Parameters

Parameter	Description	Parameter Type	Data Type
`resourceTypeId`	**resourceTypeId of the ResourceType to be deleted**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The ResourceType is deleted.		

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /resourceTypes/{resourceTypeId}

Implementation Notes

Get the specific ResourceType

Response Class (Status 200)

Successful response. The selected ResourceType is returned

Model | Example Value

ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "id": "id1",
  "name": "Example ResourceType",
  "resourceTypeKey": 49,
  "resourceDataType": "String"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`resourceTypeId`	**resourceTypeId of the ResourceType to get**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model | Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /resourceTypes/{resourceTypeId}

Implementation Notes

Update the specific ResourceType

Response Class (Status 201)

Successful response. The updated ResourceType is returned

Model | Example Value

ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "id": "id1",
  "name": "Example ResourceType",
  "resourceTypeKey": 49,
  "resourceDataType": "String"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`resourceTypeId`	**resourceTypeId of the ResourceType to update**	path	string

body	**New ResourceType object that will replace the old one. if password field is not there it will not change**	body	Model | Example Value
			ResourceTypeBase {
			name (string): ResourceType name,
			resourceTypeKey (number): ResourceType identifier,
			resourceDataType (string): Type of the data that resourceType provides,
			units (string, *optional*): Units of the ResourceType
			}
			`{`
			`"name": "Example ResourceType",`
			`"resourceTypeKey": 49,`
			`"resourceDataType": "String"`
			`}`

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Roles

GET /roles

Implementation Notes

Get Roles based on filter

Response Class (Status 200)

Successful response. The selected roles are returned

Model | Example Value

RoleArray [
  Role
]
Role {
  name (string): Name of the role,
  organizationId (string): OrganizationId this role belongs to,
  permissions (Array[string], *optional*): List of permission ids,
  isAdmin (boolean, *optional*): Temporarily flag until we have access right,
  id (string, *optional*): Id of the role
}

```
[
  {
    "id": "1234567890",
    "name": "Role",
    "organizationId": "0123456789",
    "permissions": [
      "permision id 1",
      "permission id 2"
    ],
    "isAdmin": false
  }
]
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown Server Error</td><td>Model | Example Value

<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

POST /roles

Implementation Notes

Add specific role

Response Class (Status 200)

Successful response

Model | Example Value

RoleBase {
  name (string): Name of the role,
  organizationId (string): OrganizationId this role belongs to,
  permissions (Array[string], *optional*): List of permission ids,
  isAdmin (boolean, *optional*): Temporarily flag until we have access right
}

```
{
  "name": "Role",
  "organizationId": "0123456789",
  "permissions": [
    "permision id 1",
    "permission id 2"
  ],
  "isAdmin": false
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

body	**Role to be added to the database.**	body	Model | Example Value
			RoleBase {
			name (string): Name of the role,
			organizationId (string): OrganizationId this role belongs to,
			permissions (Array[string], *optional*): List of permission ids,
			isAdmin (boolean, *optional*): Temporarily flag until we have access right
			}
			`{`
			`"name": "Role",`
			`"organizationId": "0123456789",`
			`"permissions": [`
			`"permision id 1",`
			`"permission id 2"`
			`],`
			`"isAdmin": false`
			`}`

`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

DELETE /roles/{roleId}

Implementation Notes

Delete the specific role

Parameters

Parameter	Description	Parameter Type	Data Type
roleId	**roleId of the role to be deleted**	path	string
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The role is deleted.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		  "errors": [
		    {
		      "code": 1000,
		      "message": "Unknown error with API"
		    }
		  ]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		  "errors": [
		    {
		      "code": 1000,
		      "message": "Unknown error with API"
		    }
		  ]
		}	

HTTP Status Code	Reason	Response Model	Headers
500	Unknown Server Error	Model | Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /roles/{roleId}

Implementation Notes

Get the specific role

Response Class (Status 200)

Successful response. The selected role is returned

Model | Example Value

Role {
name (string): Name of the role,
organizationId (string): OrganizationId this role belongs to,
permissions (Array[string], *optional*): List of permission ids,
isAdmin (boolean, *optional*): Temporarily flag until we have access right,
id (string, *optional*): Id of the role
}

```
{
  "id": "1234567890",
  "name": "Role",
  "organizationId": "0123456789",
  "permissions": [
    "permision id 1",
    "permission id 2"
  ],
  "isAdmin": false
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`roleId`	**roleId of the role to get**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /roles/{roleId}

Implementation Notes

Update the specific role

Response Class (Status 200)

Successful response. The updated role is returned

Model | Example Value

Role {
name (string): Name of the role,
organizationId (string): OrganizationId this role belongs to,
permissions (Array[string], *optional*): List of permission ids,
isAdmin (boolean, *optional*): Temporarily flag until we have access right,
id (string, *optional*): Id of the role
}

```
{
  "id": "1234567890",
  "name": "Role",
  "organizationId": "0123456789",
  "permissions": [
    "permision id 1",
    "permission id 2"
  ],
  "isAdmin": false
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`roleId`	**roleId of the role to update**	path	string

body	**New role object that will replace the old one. if password field is not there it will not change**	body	Model | Example Value
			RoleBase {
			name (string): Name of the role,
			organizationId (string): OrganizationId this role belongs to,
			permissions (Array[string], *optional*): List of permission ids,
			isAdmin (boolean, *optional*): Temporarily flag until we have access right
			}
			`{`
			`"name": "Role",`
			`"organizationId": "0123456789",`
			`"permissions": [`
			`"permision id 1",`
			`"permission id 2"`
			`],`
			`"isAdmin": false`
			`}`

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /roles/organization/{organizationId}

Implementation Notes

Get the list of roles for the specified organization id

Response Class (Status 200)

Successful response. The selected roles are returned

Model | Example Value

RoleArray [
  Role
]
Role {
  name (string): Name of the role,
  organizationId (string): OrganizationId this role belongs to,
  permissions (Array[string], *optional*): List of permission ids,
  isAdmin (boolean, *optional*): Temporarily flag until we have access right,
  id (string, *optional*): Id of the role
}

```
[
  {
    "id": "1234567890",
    "name": "Role",
    "organizationId": "0123456789",
    "permissions": [
      "permision id 1",
      "permission id 2"
    ],
    "isAdmin": false
  }
]
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

Rules

GET /rules

Implementation Notes

Get all rules

Response Class (Status 200)

Successful response. Rules are returned.

Model | Example Value

RuleObjectArray {
  rules (Array[RuleObject], *optional*)
}
RuleObject {
  id (string): Id of the rule,
  refId (string, *optional*): Reference id of the rule,
  type (string): Rule type = ['ENTITY', 'ENTITY_TYPE', 'ENTITY_GROUP'],
  name (string): Name of the rule,
  description (string),
  organizationId (string): Id of organization the rule belongs to,
  condition (Condition),
  action (ActionObject): action object,
  resourceCollection (ResourceCollection): Resource collection,
  scheduleId (string): Id of the schedule,
  active (boolean): Whether to active
}
Condition {
  or (Array[Inline Model 1], *optional*): Array of "or" conditions,
  and (Array[Inline Model 2], *optional*): Array of "and" conditions,
  not (object, *optional*): "not" condition,
  check (object, *optional*): condition body
}
ActionObject {
  actionMeta (object): META of the action,
  actionId (string): Id of the action
}
ResourceCollection {
  id (string, *optional*): Id of the resourceCollection,
  deviceTypeKeys (Array[number], *optional*): An array of device type keys,
  deviceIds (Array[string], *optional*): An array of device ids
}
Inline Model 1 {}
Inline Model 2 {}

```
{
  "rules": [
    {
      "id": "string",
      "refId": "string",
      "type": "ENTITY",
      "name": "string",
      "description": "string",
      "organizationId": "string",
      "condition": {
        "and": [
          {
            "or": [
              {
                "check": {
                  "resourceTypeKey": 5517,
                  "smartObjectTypeKey": 3336,
                  "instanceId": 0,
                  "operator": "stringEquals",
                  "value": "OK"
                }
              },
              {
                "not": {
                  "check": {
                    "resourceTypeKey": 30021,
                    "smartObjectTypeKey": 26312,
                    "operator": "numberEquals",
                    "value": 200
                  }
                }
              }
            ]
          }
        ]
      },
      "action": {
        "actionMeta": {},
        "actionId": "string"
      },
      "resourceCollection": {
        "id": "string",
        "deviceTypeKeys": [
          0
        ],
        "deviceIds": [
          "string"
        ]
      },
      "scheduleId": "string",
      "active": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationIds`	Comma-separated list of organization ids to filter rules	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /rules

Implementation Notes

Create a new rule

Response Class (Status 201)

Successful response. Rule is returned.

Model | Example Value

RuleObjectArray {
  rules (Array[RuleObject], *optional*)
}
RuleObject {
  id (string): Id of the rule,
  refId (string, *optional*): Reference id of the rule,
  type (string): Rule type = ['ENTITY', 'ENTITY_TYPE', 'ENTITY_GROUP'],
  name (string): Name of the rule,
  description (string),
  organizationId (string): Id of organization the rule belongs to,
  condition (Condition),
  action (ActionObject): action object,
  resourceCollection (ResourceCollection): Resource collection,
  scheduleId (string): Id of the schedule,
  active (boolean): Whether to active
}
Condition {
  or (Array[Inline Model 1], *optional*): Array of "or" conditions,
  and (Array[Inline Model 2], *optional*): Array of "and" conditions,
  not (object, *optional*): "not" condition,
  check (object, *optional*): condition body
}
ActionObject {
  actionMeta (object): META of the action,
  actionId (string): Id of the action
}
ResourceCollection {
  id (string, *optional*): Id of the resourceCollection,
  deviceTypeKeys (Array[number], *optional*): An array of device type keys,
  deviceIds (Array[string], *optional*): An array of device ids
}
Inline Model 1 {}
Inline Model 2 {}

```
{
  "rules": [
    {
      "id": "string",
      "refId": "string",
      "type": "ENTITY",
      "name": "string",
      "description": "string",
      "organizationId": "string",
      "condition": {
        "and": [
          {
            "or": [
              {
                "check": {
                  "resourceTypeKey": 5517,
                  "smartObjectTypeKey": 3336,
                  "instanceId": 0,
                  "operator": "stringEquals",
                  "value": "OK"
                }
              },
              {
                "not": {
                  "check": {
                    "resourceTypeKey": 30021,
                    "smartObjectTypeKey": 26312,
                    "operator": "numberEquals",
                    "value": 200
                  }
                }
              }
            ]
          }
        ]
      },
      "action": {
        "actionMeta": {},
        "actionId": "string"
      },
      "resourceCollection": {
        "id": "string",
        "deviceTypeKeys": [
          0
        ],
        "deviceIds": [
          "string"
        ]
      },
      "scheduleId": "string",
      "active": true
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`rule`		body	Model | Example Value **RuleObjectWithoutIds {** **name** (string): Name of the object, **description** (string), **organizationId** (string): Id of organization the rule belongs to,

Parameter	Description	Parameter Type	Data Type

type (string): Rule type = ['ENTITY', 'ENTITY_TYPE', 'ENTITY_GROUP'],
condition (Condition),
action (ActionObject): action object,
resourceCollection (ResourceCollectionWithoutId): Resource collection,
scheduleId (string): Id of the schedule,
active (boolean): Whether to active
}

Condition {
or (Array[Inline Model 1], *optional*): Array of "or" conditions,
and (Array[Inline Model 2], *optional*): Array of "and" conditions,
not (object, *optional*): "not" condition,
check (object, *optional*): condition body
}

ActionObject {
actionMeta (object): META of the action,
actionId (string): Id of the action
}

ResourceCollectionWithoutId {
deviceTypeKeys (Array[number], *optional*): An array of device type keys,
deviceIds (Array[string], *optional*): An array of device ids
}

Inline Model 1 {}

Inline Model 2 {}

```
{
  "name": "string",
  "description": "string",
  "organizationId": "string",
  "type": "ENTITY",
  "condition": {
    "and": [
      {
        "or": [
          {
            "check": {
              "resourceTypeKey": 5517,
              "smartObjectTypeKey": 3336,
              "instanceId": 0,
              "operator": "stringEquals",
              "value": "OK"
            }
          },
          {
            "not": {
              "check": {
                "resourceTypeKey": 30021,
                "smartObjectTypeKey": 26312,
                "operator": "numberEquals",
                "value": 200
              }
            }
          }
        ]
      }
    ]
  },
```

Parameter	Description	Parameter Type	Data Type

```
    "action": {
      "actionMeta": {},
      "actionId": "string"
    },
    "resourceCollection": {
      "deviceTypeKeys": [
        0
      ],
      "deviceIds": [
        "string"
      ]
    },
    "scheduleId": "string",
    "active": true
  }
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers
422	Not able to process the request	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /rules/trigger/action

Implementation Notes

Trigger action by rule

Response Class (Status 202)

string

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`Authorization`	Base authentication	header	string

`rule`		body	Model | Example Value
			RuleTrigger {
			string
			}

```
{
  "RuleId": "17fd7ba2-c320-4da5-95c9-640622f7806b",
  "DeviceId": "035643d8-3900-4760-a51f-408de9f5511b",
  "Measurement": {}
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

DELETE /rules/{ruleId}

Implementation Notes

Delete a specific rule

Parameters

Parameter	Description	Parameter Type	Data Type
`ruleId`		path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. Rule is removed.		

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /rules/{ruleId}

Implementation Notes

Get a specific rule

Response Class (Status 200)

Successful response. Rule is created.

Model | Example Value

RuleObject {
- **id** (string): Id of the rule,
- **refId** (string, *optional*): Reference id of the rule,
- **type** (string): Rule type = ['ENTITY', 'ENTITY_TYPE', 'ENTITY_GROUP'],
- **name** (string): Name of the rule,
- **description** (string),
- **organizationId** (string): Id of organization the rule belongs to,
- **condition** (Condition),
- **action** (ActionObject): action object,
- **resourceCollection** (ResourceCollection): Resource collection,
- **scheduleId** (string): Id of the schedule,
- **active** (boolean): Whether to active

}

Condition {
- **or** (Array[Inline Model 1], *optional*): Array of "or" conditions,
- **and** (Array[Inline Model 2], *optional*): Array of "and" conditions,
- **not** (object, *optional*): "not" condition,
- **check** (object, *optional*): condition body

}

ActionObject {
- **actionMeta** (object): META of the action,
- **actionId** (string): Id of the action

}

ResourceCollection {
- **id** (string, *optional*): Id of the resourceCollection,
- **deviceTypeKeys** (Array[number], *optional*): An array of device type keys,
- **deviceIds** (Array[string], *optional*): An array of device ids

}

Inline Model 1 {}

Inline Model 2 {}

```
{
  "id": "string",
  "refId": "string",
  "type": "ENTITY",
  "name": "string",
  "description": "string",
  "organizationId": "string",
  "condition": {
    "and": [
      {
        "or": [
          {
            "check": {
              "resourceTypeKey": 5517,
              "smartObjectTypeKey": 3336,
              "instanceId": 0,
              "operator": "stringEquals",
              "value": "OK"
            }
          },
          {
            "not": {
              "check": {
                "resourceTypeKey": 30021,
                "smartObjectTypeKey": 26312,
                "operator": "numberEquals",
                "value": 200
              }
            }
          }
        ]
      }
    ]
  },
  "action": {
    "actionMeta": {},
    "actionId": "string"
  },
  "resourceCollection": {
    "id": "string",
    "deviceTypeKeys": [
      0
    ],
    "deviceIds": [
      "string"
    ]
  },
  "scheduleId": "string",
  "active": true
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
ruleId		path	string
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Not able to process the request	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /rules/{ruleId}

Implementation Notes

Update the specific rule

Response Class (Status 200)

Successful response. Rule is updated.

Model | Example Value

RuleObject {
  id (string): Id of the rule,
  refId (string, *optional*): Reference id of the rule,
  type (string): Rule type = ['ENTITY', 'ENTITY_TYPE', 'ENTITY_GROUP'],
  name (string): Name of the rule,
  description (string),
  organizationId (string): Id of organization the rule belongs to,
  condition (Condition),
  action (ActionObject): action object,
  resourceCollection (ResourceCollection): Resource collection,
  scheduleId (string): Id of the schedule,
  active (boolean): Whether to active
}
Condition {
  or (Array[Inline Model 1], *optional*): Array of "or" conditions,
  and (Array[Inline Model 2], *optional*): Array of "and" conditions,
  not (object, *optional*): "not" condition,
  check (object, *optional*): condition body
}
ActionObject {
  actionMeta (object): META of the action,
  actionId (string): Id of the action
}
ResourceCollection {
  id (string, *optional*): Id of the resourceCollection,
  deviceTypeKeys (Array[number], *optional*): An array of device type keys,
  deviceIds (Array[string], *optional*): An array of device ids
}
Inline Model 1 {}
Inline Model 2 {}

```
{
  "id": "string",
  "refId": "string",
  "type": "ENTITY",
  "name": "string",
  "description": "string",
  "organizationId": "string",
  "condition": {
    "and": [
      {
        "or": [
          {
            "check": {
              "resourceTypeKey": 5517,
              "smartObjectTypeKey": 3336,
              "instanceId": 0,
              "operator": "stringEquals",
              "value": "OK"
            }
          },
          {
            "not": {
              "check": {
                "resourceTypeKey": 30021,
                "smartObjectTypeKey": 26312,
                "operator": "numberEquals",
                "value": 200
              }
            }
          }
        ]
      }
    ]
  },
  "action": {
    "actionMeta": {},
    "actionId": "string"
  },
  "resourceCollection": {
    "id": "string",
    "deviceTypeKeys": [
      0
    ],
    "deviceIds": [
      "string"
    ]
  },
  "scheduleId": "string",
  "active": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
ruleId		path	string
x-access-token	User token.	header	string
rule		body	Model | Example Value **RuleObject {** **id** (string): Id of the rule, **refId** (string, *optional*): Reference id of the rule, **type** (string): Rule type = ['ENTITY', 'ENTITY_TYPE', 'ENTITY_GROUP'], **name** (string): Name of the rule, **description** (string), **organizationId** (string): Id of organization the rule belongs to, **condition** (Condition), **action** (ActionObject): action object,

Parameter	Description	Parameter Type	Data Type

resourceCollection (ResourceCollection): Resource collection,
scheduleId (string): Id of the schedule,
active (boolean): Whether to active

}

Condition {

or (Array[Inline Model 1], *optional*): Array of "or" conditions,
and (Array[Inline Model 2], *optional*): Array of "and" conditions,
not (object, *optional*): "not" condition,
check (object, *optional*): condition body

}

ActionObject {

actionMeta (object): META of the action,
actionId (string): Id of the action

}

ResourceCollection {

id (string, *optional*): Id of the resourceCollection,
deviceTypeKeys (Array[number], *optional*): An array of device type keys,
deviceIds (Array[string], *optional*): An array of device ids

}

Inline Model 1 {}

Inline Model 2 {}

```
{
  "id": "string",
  "refId": "string",
  "type": "ENTITY",
  "name": "string",
  "description": "string",
  "organizationId": "string",
  "condition": {
    "and": [
      {
        "or": [
          {
            "check": {
              "resourceTypeKey": 5517,
              "smartObjectTypeKey": 3336,
              "instanceId": 0,
              "operator": "stringEquals",
              "value": "OK"
            }
          },
          {
            "not": {
              "check": {
                "resourceTypeKey": 30021,
                "smartObjectTypeKey": 26312,
                "operator": "numberEquals",
                "value": 200
              }
            }
          }
        ]
      }
    ]
  },
  "action": {
```

Parameter	Description	Parameter Type	Data Type

```
    "actionMeta": {},
    "actionId": "string"
  },
  "resourceCollection": {
    "id": "string",
    "deviceTypeKeys": [
      0
    ],
    "deviceIds": [
      "string"
    ]
  },
  "scheduleId": "string",
  "active": true
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers
422	Not able to process the request	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

Schedules

GET /schedules

Implementation Notes

Get all schedules

Response Class (Status 200)

Successful response. Schedules are returned.

Model | Example Value

ScheduleArray {
  schedules (Array[Schedule], *optional*)
}
Schedule {
  name (string): Name of the schedule for display,
  description (string): Description of the schedule for display,
  organizationId (string): Id of organization the schedule belongs to,
  recurrence (string): RRULE format from iCalander to specify recurrence,
  id (string): Id of occured Schedule
}

```
{
  "schedules": [
    {
      "id": "22875bb9-8787-454d-b76d-44355a4dce55",
      "name": "Example Daily",
      "description": "Example Description",
      "organizationId": "String",
      "recurrence": "RRULE:FREQ=DAILY;DTSTART=20180130T145620Z;COUNT=5;UNTIL=20180130T145621Z"
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationIds`	Comma-separated list of organization ids to filter schedules	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /schedules

Implementation Notes

Create a new schedule

Response Class (Status 201)

Successful response. Actions are returned.

Model | Example Value

Schedule {
name (string): Name of the schedule for display,
description (string): Description of the schedule for display,
organizationId (string): Id of organization the schedule belongs to,
recurrence (string): RRULE format from iCalander to specify recurrence,
id (string): Id of occured Schedule
}

```
{
  "id": "22875bb9-8787-454d-b76d-44355a4dce55",
  "name": "Example Daily",
  "description": "Example Description",
  "organizationId": "String",
  "recurrence": "RRULE:FREQ=DAILY;DTSTART=20180130T145620Z;COUNT=5;UNTIL=20180130T145621Z"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
scheduleData		body	Model | Example Value **ScheduleWithoutId {** **name** (string): Name of the schedule for display, **description** (string): Description of the schedule for display, **organizationId** (string): Id of organization the schedule belongs to, **recurrence** (string): RRULE format from iCalander to specify recurrence **}**

```
{
  "name": "Example Daily",
  "description": "Example Description",
  "organizationId": "String",
  "recurrence": "RRULE:FREQ=DAILY;DTSTART=20180130T145620Z;COUNT=5;UNTIL=20180130T
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

DELETE /schedules/{scheduleId}

Implementation Notes

Delete a specific schedule

Parameters

Parameter	Description	Parameter Type	Data Type
`scheduleId`		path	string
`x-access-token`	User token.	header	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>204</td><td>Successful response. Actions are returned.</td><td></td><td></td></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
    <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
    <b>code</b> (integer): Four-digits error code,
    <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
    <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
    <b>code</b> (integer): Four-digits error code,
    <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

GET /schedules/{scheduleId}

Implementation Notes

Get a specific schedule

Response Class (Status 200)

Successful response. Schedule is returned.

Model | Example Value

Schedule {
  name (string): Name of the schedule for display,
  description (string): Description of the schedule for display,
  organizationId (string): Id of organization the schedule belongs to,
  recurrence (string): RRULE format from iCalander to specify recurrence,
  id (string): Id of occured Schedule
}

```
{
  "id": "22875bb9-8787-454d-b76d-44355a4dce55",
  "name": "Example Daily",
  "description": "Example Description",
  "organizationId": "String",
  "recurrence": "RRULE:FREQ=DAILY;DTSTART=20180130T145620Z;COUNT=5;UNTIL=20180130T145621Z"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
scheduleId		path	string
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value
<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

PUT /schedules/{scheduleId}

Implementation Notes

Update the schedule

Response Class (Status 201)

Successful response. Schedule is returned.

Model | Example Value

Schedule {
name (string): Name of the schedule for display,
description (string): Description of the schedule for display,
organizationId (string): Id of organization the schedule belongs to,
recurrence (string): RRULE format from iCalander to specify recurrence,
id (string): Id of occured Schedule
}

```
{
  "id": "22875bb9-8787-454d-b76d-44355a4dce55",
  "name": "Example Daily",
  "description": "Example Description",
  "organizationId": "String",
  "recurrence": "RRULE:FREQ=DAILY;DTSTART=20180130T145620Z;COUNT=5;UNTIL=20180130T145621Z"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
scheduleId		path	string
x-access-token	User token.	header	string

scheduleData		body	Model | Example Value
			ScheduleWithoutId {
			name (string): Name of the schedule for display,
			description (string): Description of the schedule for display,
			organizationId (string): Id of organization the schedule belongs to,
			recurrence (string): RRULE format from iCalander to specify recurrence
			}
			{
			"name": "Example Daily",
			"description": "Example Description",
			"organizationId": "String",
			"recurrence": "RRULE:FREQ=DAILY;DTSTART=20180130T145620Z;COUNT=5;UNTIL=20180130T
			}

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

Search

GET /search

Implementation Notes

Search

Response Class (Status 200)

Successful response

Model | Example Value

Search {
  devices (DevicesResult, *optional*): Devices search result
}
DevicesResult {
  devices (Array[FoundedDevice], *optional*): List of compact devices which match search query,
  pagination (PaginationModel, *optional*): Pagination object
}
FoundedDevice {
  id (string, *optional*): Unique device identifier,
  name (string, *optional*): User defined device name,
  deviceType (CompactDeviceType, *optional*): Object defining device compact category
}
PaginationModel {
  nbrOfPages (number, *optional*): Number of pages,
  nbrOfItems (number, *optional*): Number of items in total,
  page (number, *optional*): Page Number,
  pageSize (number, *optional*): Size of a Page
}
CompactDeviceType {
  category (CompactDeviceCategoryObject, *optional*): Related device compact category for a compact device type
}
CompactDeviceCategoryObject {
  iconName (string, *optional*): Image name used for marker icons in the search list
}

```
{
  "devices": {
    "devices": [
      {
        "id": "string",
        "name": "string",
        "deviceType": {
          "category": {
            "iconName": "string"
          }
        }
      }
    ],
    "pagination": {
      "nbrOfPages": 0,
      "nbrOfItems": 0,
      "page": 0,
      "pageSize": 0
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
q	Search text	query	string
pageSize	Request via devices with pagination with max pageSize	query	double
page	Request page of devices with pagination start with 0	query	double
modules	Search in that fields Available options are: devices, mapdata, devicegroups, alarms, notifications, wiki Multiple fields can be specified separated by a comma (,)	query	Array[string]
autocomplete	Returns limited amount of items and their names only if flag is set to True	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

Services

GET /services

Implementation Notes

Get all Services. Services are a place-holder for information related to an integrated system or set of devices

Response Class (Status 200)

Successful response. Return all services

Model | Example Value

Services {

Service (Array[Service], *optional*): all of the Services

}

Service {

name (string, *optional*): The name of service.,

description (string, *optional*): The description of service.,

preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,

ownerOrganization (string, *optional*): Own organization id,

type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],

subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,

adapter (inline_model_0, *optional*)

}

inline_model_0 {

commands (Commands, *optional*)

}

Commands {

url (string): Adapter URL where the API can POST the commands to the adapter,

username (string): Username,

password (string): Password,

timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000

}

```
{
  "Service": [
    {
      "name": "Service1",
      "description": "core_wl",
      "preferenceValidation": {
        "preference_key": {
          "type": "object",
          "properties": {
            "value1": {
              "type": "string"
            }
          }
        }
      },
      "ownerOrganization": "ownOrganizationId",
      "type": "EXTERNAL",
      "subscribedOrganizations": [
        "subscribedOrganizatioId1",
        "subscribedOrganizationId2"
      ],
      "adapter": {
        "commands": {
          "url": "https://adapter.com/command",
          "username": "Aladdin",
          "password": "open sesame",
          "timeout": "5000"
        }
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model | Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

POST /services

Implementation Notes

create the Service details

Response Class (Status 201)

Successful response.

Model | Example Value

Service {
name (string, *optional*): The name of service.,
description (string, *optional*): The description of service.,
preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
ownerOrganization (string, *optional*): Own organization id,
type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,
adapter (inline_model_0, *optional*)
}
inline_model_0 {
commands (Commands, *optional*)
}
Commands {
url (string): Adapter URL where the API can POST the commands to the adapter,
username (string): Username,
password (string): Password,
timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000
}

```
{
  "name": "Service1",
  "description": "core_wl",
  "preferenceValidation": {
    "preference_key": {
      "type": "object",
      "properties": {
        "value1": {
          "type": "string"
        }
      }
    }
  },
  "ownerOrganization": "ownOrganizationId",
  "type": "EXTERNAL",
  "subscribedOrganizations": [
    "subscribedOrganizatioId1",
    "subscribedOrganizationId2"
  ],
  "adapter": {
    "commands": {
      "url": "https://adapter.com/command",
      "username": "Aladdin",
      "password": "open sesame",
      "timeout": "5000"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Parameter	Description	Parameter Type	Data Type
`Service`		body	Model Example Value

Service {

name (string, *optional*): The name of service.,
description (string, *optional*): The description of service.,
preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
ownerOrganization (string, *optional*): Own organization id,
type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,
adapter (inline_model_0, *optional*)

}

inline_model_0 {

commands (Commands, *optional*)

}

Commands {

url (string): Adapter URL where the API can POST the commands to the adapter,
username (string): Username,
password (string): Password,
timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000

}

```
{
  "name": "Service1",
  "description": "core_wl",
  "preferenceValidation": {
    "preference_key": {
      "type": "object",
      "properties": {
        "value1": {
          "type": "string"
        }
      }
    }
  },
  "ownerOrganization": "ownOrganizationId",
  "type": "EXTERNAL",
  "subscribedOrganizations": [
    "subscribedOrganizatioId1",
    "subscribedOrganizationId2"
  ],
  "adapter": {
    "commands": {
      "url": "https://adapter.com/command",
      "username": "Aladdin",
      "password": "open sesame",
      "timeout": "5000"
    }
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

DELETE /services/{serviceId}

Implementation Notes

Delete service with id.

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. Empty body object.		

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /services/{serviceId}

Implementation Notes

Get all service id.

Response Class (Status 200)

Successful response. Return a service's detail info

Model | Example Value

Service {
name (string, *optional*): The name of service.,
description (string, *optional*): The description of service.,
preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
ownerOrganization (string, *optional*): Own organization id,
type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,
adapter (inline_model_0, *optional*)
}
inline_model_0 {
commands (Commands, *optional*)
}
Commands {
url (string): Adapter URL where the API can POST the commands to the adapter,
username (string): Username,
password (string): Password,
timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000
}

```
{
  "name": "Service1",
  "description": "core_wl",
  "preferenceValidation": {
    "preference_key": {
      "type": "object",
      "properties": {
        "value1": {
          "type": "string"
        }
      }
    }
  },
  "ownerOrganization": "ownOrganizationId",
  "type": "EXTERNAL",
  "subscribedOrganizations": [
    "subscribedOrganizatioId1",
    "subscribedOrganizationId2"
  ],
  "adapter": {
    "commands": {
      "url": "https://adapter.com/command",
      "username": "Aladdin",
      "password": "open sesame",
      "timeout": "5000"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

PUT /services/{serviceId}

Implementation Notes

update the service details

Response Class (Status 201)

Successful response.

Model | Example Value

Service {
name (string, *optional*): The name of service.,
description (string, *optional*): The description of service.,
preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
ownerOrganization (string, *optional*): Own organization id,
type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,
adapter (inline_model_0, *optional*)
}
inline_model_0 {
commands (Commands, *optional*)
}
Commands {
url (string): Adapter URL where the API can POST the commands to the adapter,
username (string): Username,
password (string): Password,
timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000
}

```
{
  "name": "Service1",
  "description": "core_wl",
  "preferenceValidation": {
    "preference_key": {
      "type": "object",
      "properties": {
        "value1": {
          "type": "string"
        }
      }
    }
  },
  "ownerOrganization": "ownOrganizationId",
  "type": "EXTERNAL",
  "subscribedOrganizations": [
    "subscribedOrganizatioId1",
    "subscribedOrganizationId2"
  ],
  "adapter": {
    "commands": {
      "url": "https://adapter.com/command",
      "username": "Aladdin",
      "password": "open sesame",
      "timeout": "5000"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceId`	**id of the service to update**	path	string

Parameter	Description	Parameter Type	Data Type
service		body	Model Example Value

Service {
name (string, *optional*): The name of service.,
description (string, *optional*): The description of service.,
preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
ownerOrganization (string, *optional*): Own organization id,
type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,
adapter (inline_model_0, *optional*)
}

inline_model_0 {
commands (Commands, *optional*)
}

Commands {
url (string): Adapter URL where the API can POST the commands to the adapter,
username (string): Username,
password (string): Password,
timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000
}

```
{
  "name": "Service1",
  "description": "core_wl",
  "preferenceValidation": {
    "preference_key": {
      "type": "object",
      "properties": {
        "value1": {
          "type": "string"
        }
      }
    }
  },
  "ownerOrganization": "ownOrganizationId",
  "type": "EXTERNAL",
  "subscribedOrganizations": [
    "subscribedOrganizatioId1",
    "subscribedOrganizationId2"
  ],
  "adapter": {
    "commands": {
      "url": "https://adapter.com/command",
      "username": "Aladdin",
      "password": "open sesame",
      "timeout": "5000"
    }
  }
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

GET /services/organization/{organizationId}

Implementation Notes

Get all Services

Response Class (Status 200)

Successful response. Return all services

Model | Example Value

Services {
  Service (Array[Service], *optional*): all of the Services
}
Service {
  name (string, *optional*): The name of service.,
  description (string, *optional*): The description of service.,
  preferenceValidation (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
  ownerOrganization (string, *optional*): Own organization id,
  type (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
  subscribedOrganizations (Array[string], *optional*): List of organization id(s) the service is subscribed to,
  adapter (inline_model_0, *optional*)
}
inline_model_0 {
  commands (Commands, *optional*)
}
Commands {
  url (string): Adapter URL where the API can POST the commands to the adapter,
  username (string): Username,
  password (string): Password,
  timeout (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000
}

```
{
  "Service": [
    {
      "name": "Service1",
      "description": "core_wl",
      "preferenceValidation": {
        "preference_key": {
          "type": "object",
          "properties": {
            "value1": {
              "type": "string"
            }
          }
        }
      },
      "ownerOrganization": "ownOrganizationId",
      "type": "EXTERNAL",
      "subscribedOrganizations": [
        "subscribedOrganizatioId1",
        "subscribedOrganizationId2"
      ],
      "adapter": {
        "commands": {
          "url": "https://adapter.com/command",
          "username": "Aladdin",
          "password": "open sesame",
          "timeout": "5000"
        }
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
organizationId		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /services/organizationCore/{organizationId}

Implementation Notes

Get organization core services

Response Class (Status 200)

Successful response. Return all services

Model | Example Value

Service {
- **name** (string, *optional*): The name of service.,
- **description** (string, *optional*): The description of service.,
- **preferenceValidation** (object, *optional*): validation object - https://github.com/epoberezkin/ajv,
- **ownerOrganization** (string, *optional*): Own organization id,
- **type** (string, *optional*): The type of service. = ['COMMON', 'ORGANIZATION_CORE', 'ADMIN', 'EXTERNAL'],
- **subscribedOrganizations** (Array[string], *optional*): List of organization id(s) the service is subscribed to,
- **adapter** (inline_model_0, *optional*)

}

inline_model_0 {
- **commands** (Commands, *optional*)

}

Commands {
- **url** (string): Adapter URL where the API can POST the commands to the adapter,
- **username** (string): Username,
- **password** (string): Password,
- **timeout** (number, *optional*): The maximum wait time for response from adapter (in milliseconds). Minimum value: 1000, Default value: 5000

}

```
{
  "name": "Service1",
  "description": "core_wl",
  "preferenceValidation": {
    "preference_key": {
      "type": "object",
      "properties": {
        "value1": {
          "type": "string"
        }
      }
    }
  },
  "ownerOrganization": "ownOrganizationId",
  "type": "EXTERNAL",
  "subscribedOrganizations": [
    "subscribedOrganizatioId1",
    "subscribedOrganizationId2"
  ],
  "adapter": {
    "commands": {
      "url": "https://adapter.com/command",
      "username": "Aladdin",
      "password": "open sesame",
      "timeout": "5000"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`organizationId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers
422	Server Error	Model Example Value	

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /services/serviceDetails/{serviceId}

Implementation Notes

Get service details

Response Class (Status 200)

Successful response. Returns service details

<span class="strong">serviceDetails is not defined!</span>

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`serviceId`	**id of the service to get details**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

Smart Object Types

GET /smartObjectTypes

Implementation Notes

Get Smart Object Types

Response Class (Status 200)

Successful response. The selected Smart Object Types are returned

Model | Example Value

SmartObjectTypeArray {
  smartObjectTypes (Array[SmartObjectType], *optional*)
}
SmartObjectType {
  name (string): Name of the Smart Object Type,
  smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
  multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
  resources (Array[ResourceType]): List of resource types keys,
  id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
  id (string, *optional*): Id of the object,
  name (string): ResourceType name,
  resourceTypeKey (number): ResourceType identifier,
  resourceDataType (string): Type of the data that resourceType provides,
  units (string, *optional*): Units of the ResourceType
}

```
{
  "smartObjectTypes": [
    {
      "name": "string",
      "smartObjectTypeKey": 0,
      "multipleInstances": true,
      "resources": [
        {
          "id": "id1",
          "name": "Example ResourceType",
          "resourceTypeKey": 49,
          "resourceDataType": "String"
        }
      ],
      "id": "string"
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

POST /smartObjectTypes

Implementation Notes

Add specific smart Object Type

Response Class (Status 201)

Successful response

Model | Example Value

SmartObjectType {
name (string): Name of the Smart Object Type,
smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
resources (Array[ResourceType]): List of resource types keys,
id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "name": "string",
  "smartObjectTypeKey": 0,
  "multipleInstances": true,
  "resources": [
    {
      "id": "id1",
      "name": "Example ResourceType",
      "resourceTypeKey": 49,
      "resourceDataType": "String"
    }
  ],
  "id": "string"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
body	**Smart Object Type to be added to the database.**	body	Model | Example Value **SmartObjectTypeCreateUpdate {** **name** (string): Name of the Smart Object Type, **smartObjectTypeKey** (number): Specifies the ObjectID of the Smart Object Type, **multipleInstances** (boolean): Indicates whether the Smart Object Type can occur multiple times on a device, **resources** (Array[number]): List of resource types keys **}**

```
{
  "name": "Example SmartObjectType",
  "smartObjectTypeKey": 1,
  "multipleInstances": true,
  "resources": [
    1,
    2,
    3
  ]
}
```

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

DELETE /smartObjectTypes/{id}

Implementation Notes

Delete the specific smart object type

Parameters

Parameter	Description	Parameter Type	Data Type
`id`	**id of the smart object type to be deleted**	path	string
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
`204`	Successful response. The smart object type is deleted.		

`403`	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

`422`	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /smartObjectTypes/{id}

Implementation Notes

Get the specific smart object type

Response Class (Status 200)

Successful response. The selected smart object type is returned

Model | Example Value

SmartObjectType {
name (string): Name of the Smart Object Type,
smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
resources (Array[ResourceType]): List of resource types keys,
id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
id (string, *optional*): Id of the object,
name (string): ResourceType name,
resourceTypeKey (number): ResourceType identifier,
resourceDataType (string): Type of the data that resourceType provides,
units (string, *optional*): Units of the ResourceType
}

```
{
  "name": "string",
  "smartObjectTypeKey": 0,
  "multipleInstances": true,
  "resources": [
    {
      "id": "id1",
      "name": "Example ResourceType",
      "resourceTypeKey": 49,
      "resourceDataType": "String"
    }
  ],
  "id": "string"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
id	**id of the smart object type to get**	path	string
x-access-token	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Unknown Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

PUT /smartObjectTypes/{id}

Implementation Notes

Update the specific smart object type

Response Class (Status 201)

Successful response. The updated smart object type is returned

Model | Example Value

SmartObjectType {
  name (string): Name of the Smart Object Type,
  smartObjectTypeKey (number): Specifies the ObjectID of the Smart Object Type,
  multipleInstances (boolean): Indicates whether the Smart Object Type can occur multiple times on a device,
  resources (Array[ResourceType]): List of resource types keys,
  id (string, *optional*): Id of the Smart Object Type
}
ResourceType {
  id (string, *optional*): Id of the object,
  name (string): ResourceType name,
  resourceTypeKey (number): ResourceType identifier,
  resourceDataType (string): Type of the data that resourceType provides,
  units (string, *optional*): Units of the ResourceType
}

```
{
  "name": "string",
  "smartObjectTypeKey": 0,
  "multipleInstances": true,
  "resources": [
    {
      "id": "id1",
      "name": "Example ResourceType",
      "resourceTypeKey": 49,
      "resourceDataType": "String"
    }
  ],
  "id": "string"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
id	**id of the smart object type to update**	path	string

Parameter	Description	Parameter Type	Data Type
body	**New smart object type object that will replace the old one**	body	Model | Example Value **SmartObjectTypeCreateUpdate {** **name** (string): Name of the Smart Object Type, **smartObjectTypeKey** (number): Specifies the ObjectID of the Smart Object Type, **multipleInstances** (boolean): Indicates whether the Smart Object Type can occur multiple times on a device, **resources** (Array[number]): List of resource types keys **}**
x-access-token	User token.	header	string

```
{
  "name": "Example SmartObjectType",
  "smartObjectTypeKey": 1,
  "multipleInstances": true,
  "resources": [
    1,
    2,
    3
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
   <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
   <b>code</b> (integer): Four-digits error code,
   <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown Server Error</td><td>Model | Example Value

<b>Errors {</b>
   <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
   <b>code</b> (integer): Four-digits error code,
   <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

UserGroups

GET /userGroups

Implementation Notes

Get User Group objects.

Response Class (Status 200)

Successful response. Return all UserGroups.

Model | Example Value

userGroupArray {
  userGroups (Array[userGroupResponse], *optional*)
}
userGroupResponse {
  id (string, *optional*),
  name (string): Name of the userGroup,
  description (string, *optional*): Description of the userGroup,
  roleIds (Array[string], *optional*): The id's of the roles the user has,
  organizationId (string, *optional*): Id of the organization
}

```
{
  "userGroups": [
    {
      "userGroups": [
        {
          "name": "Farrell-Medhurst",
          "description": "test Conroy-Bins",
          "roleIds": [
            "ec4850df-c9e2-4ddc-9832-4af12a12ca5b"
          ],
          "organizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
        }
      ]
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`ids`	Filter on all user groups with a specifc id. Multiple fields can be specified separated by a comma (,)	query	Array[string]
`organization`	Filter on all user groups with a specifc organization.	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{ "errors": [ { "code": 1000, "message": "Unknown error with API" } ] }`	

POST /userGroups

Implementation Notes

Create the User Group details.

Response Class (Status 201)

Successful response.

Model | Example Value

userGroupResponse {
id (string, *optional*),
name (string): Name of the userGroup,
description (string, *optional*): Description of the userGroup,
roleIds (Array[string], *optional*): The id's of the roles the user has,
organizationId (string, *optional*): Id of the organization
}

```
{
  "userGroups": [
    {
      "name": "Farrell-Medhurst",
      "description": "test Conroy-Bins",
      "roleIds": [
        "ec4850df-c9e2-4ddc-9832-4af12a12ca5b"
      ],
      "organizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`UserGroup`		body	Model | Example Value **userGroupRequest {** **name** (string): Name of the userGroup, **description** (string, *optional*): Description of the userGroup, **roleIds** (Array[string], *optional*): The id's of the roles the user has, **organizationId** (string, *optional*): Id of the organization **}**

```
{
  "name": "Farrell-Medhurst",
  "description": "test Conroy-Bins",
  "roleIds": [
    "ec4850df-c9e2-4ddc-9832-4af12a12ca5b"
  ],
  "organizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

DELETE /userGroups/{userGroupId}

Implementation Notes

Delete UserGroup by id.

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userGroupId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response.		

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value	

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /userGroups/{userGroupId}

Implementation Notes

Get a UserGroup by id.

Response Class (Status 200)

Successful response. Return a UserGroup's detail info

Model | Example Value

userGroupResponse {
id (string, *optional*),
name (string): Name of the userGroup,
description (string, *optional*): Description of the userGroup,
roleIds (Array[string], *optional*): The id's of the roles the user has,
organizationId (string, *optional*): Id of the organization
}

```
{
  "userGroups": [
    {
      "name": "Farrell-Medhurst",
      "description": "test Conroy-Bins",
      "roleIds": [
        "ec4850df-c9e2-4ddc-9832-4af12a12ca5b"
      ],
      "organizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userGroupId`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

PUT /userGroups/{userGroupId}

Implementation Notes

Update the UserGroup details.

Response Class (Status 201)

Successful response.

Model | Example Value

userGroupResponse {
id (string, *optional*),
name (string): Name of the userGroup,
description (string, *optional*): Description of the userGroup,
roleIds (Array[string], *optional*): The id's of the roles the user has,
organizationId (string, *optional*): Id of the organization
}

```
{
  "userGroups": [
    {
      "name": "Farrell-Medhurst",
      "description": "test Conroy-Bins",
      "roleIds": [
        "ec4850df-c9e2-4ddc-9832-4af12a12ca5b"
      ],
      "organizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userGroupId`		path	string

Parameter	Description	Parameter Type	Data Type
`UserGroup`		body	Model | Example Value **userGroupRequest {** **name** (string): Name of the userGroup, **description** (string, *optional*): Description of the userGroup, **roleIds** (Array[string], *optional*): The id's of the roles the user has, **organizationId** (string, *optional*): Id of the organization **}**

```
{
  "name": "Farrell-Medhurst",
  "description": "test Conroy-Bins",
  "roleIds": [
    "ec4850df-c9e2-4ddc-9832-4af12a12ca5b"
  ],
  "organizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

Users

GET /users

Implementation Notes

Get all users based on filter

Response Class (Status 200)

Successful response. The selected users are returned.

Model | Example Value

inline_model_5 {
  users (Array[User], *optional*)
}
User {
  id (string): Unique identifier representing this user,
  created (string): The time the data is created in the system,The Timestamp type expressed according to ISO 8601. We take the following format - Complete date plus hours, minutes, seconds and a decimal fraction of a second. For example, "1997-07-16T19:20:30.45+01:00",
  isLoggedIn (boolean): The user login into the system,
  lastActiveTimestamp (string, *optional*): Last timestamp the user was active on the system,
  email (string, *optional*): email address of the user,
  firstName (string, *optional*): First name of the user,
  lastName (string, *optional*): Last name of the user,
  isEnabled (boolean, *optional*): Access UI state of the user,
  isTelemetryEnabled (boolean, *optional*): Enabled or disabled client-side monitoring,
  roleIds (Array[string], *optional*): The id's of the roles the user has,
  groupIds (Array[string], *optional*): The id's of groups that a user belongs to,
  rootOrganizationId (string, *optional*): Primary organization id,
  organizationIds (Array[string], *optional*): The id's of organizations that a user has
}

```
{
  "users": [
    {
      "id": "string",
      "created": "2019-10-03T07:46:35.555Z",
      "isLoggedIn": true,
      "lastActiveTimestamp": "2019-10-03T07:46:35.555Z",
      "email": "string",
      "firstName": "string",
      "lastName": "string",
      "isEnabled": true,
      "isTelemetryEnabled": true,
      "roleIds": [
        "string"
      ],
      "groupIds": [
        "string"
      ],
      "rootOrganizationId": "string",
      "organizationIds": [
        "string"
      ]
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`email`	Filter on all users with a specifc email. Multiple fields can be specified separated by a comma (,)	query	Array[string]
`role`	Filter on all user that have a specific role (based on roleId) Multiple fields can be specified separated by a comma (,)	query	Array[string]
`group`	Filter on all user that are part of a specific group (based on groupId) Multiple fields can be specified separated by a comma (,)	query	Array[string]
`organization`	Filter on all user that have a specific organization (based on organizationId) Multiple fields can be specified separated by a comma (,)	query	Array[string]
`sort`	Sort the response by specified fields - prefix imply a descending sort order. Multiple fields can be specified separated by a comma (,)	query	Array[string]

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model | Example Value

<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

POST /users

Implementation Notes

Create user

Response Class (Status 201)

Successful response. The created users are returned.

Model | Example Value

User {

id (string): Unique identifier representing this user,
created (string): The time the data is created in the system,The Timestamp type expressed according to ISO 8601. We take the following format - Complete date plus hours, minutes, seconds and a decimal fraction of a second. For example, "1997-07-16T19:20:30.45+01:00",
isLoggedIn (boolean): The user login into the system,
lastActiveTimestamp (string, *optional*): Last timestamp the user was active on the system,
email (string, *optional*): email address of the user,
firstName (string, *optional*): First name of the user,
lastName (string, *optional*): Last name of the user,
isEnabled (boolean, *optional*): Access UI state of the user,
isTelemetryEnabled (boolean, *optional*): Enabled or disabled client-side monitoring,
roleIds (Array[string], *optional*): The id's of the roles the user has,
groupIds (Array[string], *optional*): The id's of groups that a user belongs to,
rootOrganizationId (string, *optional*): Primary organization id,
organizationIds (Array[string], *optional*): The id's of organizations that a user has

}

```
{
  "id": "string",
  "created": "2019-10-03T07:46:35.563Z",
  "isLoggedIn": true,
  "lastActiveTimestamp": "2019-10-03T07:46:35.563Z",
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "isEnabled": true,
  "isTelemetryEnabled": true,
  "roleIds": [
    "string"
  ],
  "groupIds": [
    "string"
  ],
  "rootOrganizationId": "string",
  "organizationIds": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
body		body	Model Example Value **CreateUser {** **type** (string): User authentication type = ['DB', 'AAD', 'AADEXT'], **password** (string, *optional*): Password of the user, **email** (string): email address of the user, **firstName** (string, *optional*): First name of the user, **lastName** (string, *optional*): Last name of the user, **isEnabled** (boolean, *optional*): Access UI state of the user, **isTelemetryEnabled** (boolean, *optional*): Enabled or disabled client-side monitoring, **roleIds** (Array[string], *optional*): The id's of the roles the user has, **groupIds** (Array[string], *optional*): The id's of groups that a user belongs to, **rootOrganizationId** (string, *optional*): Primary organization id, **organizationIds** (Array[string], *optional*): The id's of organizations that a user has **}**

```
{
  "email": "admin@connectedurbantransport.net",
  "firstName": "Admin",
  "lastName": "CUT",
  "isEnabled": true,
  "rootOrganizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15",
  "type": "DB"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

DELETE /users/{userId}

Implementation Notes

Delete the specific user

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userId`	**userId of the user to delete**	path	string

Response Messages

<table>
<thead>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
</thead>
<tbody>
<tr><td>204</td><td>Successful response. The user is deleted.</td><td></td><td></td></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value
<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value
<b>Errors {</b>
  <b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
  <b>code</b> (integer): Four-digits error code,
  <b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</tbody>
</table>

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /users/{userId}

Implementation Notes

Get the specific user

Response Class (Status 200)

Successful response. The selected user is returned.

Model | Example Value

User {
id (string): Unique identifier representing this user,
created (string): The time the data is created in the system,The Timestamp type expressed according to ISO 8601. We take the following format - Complete date plus hours, minutes, seconds and a decimal fraction of a second. For example, "1997-07-16T19:20:30.45+01:00",
isLoggedIn (boolean): The user login into the system,
lastActiveTimestamp (string, *optional*): Last timestamp the user was active on the system,
email (string, *optional*): email address of the user,
firstName (string, *optional*): First name of the user,
lastName (string, *optional*): Last name of the user,
isEnabled (boolean, *optional*): Access UI state of the user,
isTelemetryEnabled (boolean, *optional*): Enabled or disabled client-side monitoring,
roleIds (Array[string], *optional*): The id's of the roles the user has,
groupIds (Array[string], *optional*): The id's of groups that a user belongs to,
rootOrganizationId (string, *optional*): Primary organization id,
organizationIds (Array[string], *optional*): The id's of organizations that a user has
}

```
{
  "id": "string",
  "created": "2019-10-03T07:46:35.577Z",
  "isLoggedIn": true,
  "lastActiveTimestamp": "2019-10-03T07:46:35.577Z",
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "isEnabled": true,
  "isTelemetryEnabled": true,
  "roleIds": [
    "string"
  ],
  "groupIds": [
    "string"
  ],
  "rootOrganizationId": "string",
  "organizationIds": [
    "string"
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userId`	**userId of the user to get**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

PUT /users/{userId}

Implementation Notes

Update a specific user

Response Class (Status 201)

Successful response. The updated user is returned.

Model | Example Value

User {
id (string): Unique identifier representing this user,
created (string): The time the data is created in the system,The Timestamp type expressed according to ISO 8601. We take the following format - Complete date plus hours, minutes, seconds and a decimal fraction of a second. For example, "1997-07-16T19:20:30.45+01:00",
isLoggedIn (boolean): The user login into the system,
lastActiveTimestamp (string, *optional*): Last timestamp the user was active on the system,
email (string, *optional*): email address of the user,
firstName (string, *optional*): First name of the user,
lastName (string, *optional*): Last name of the user,
isEnabled (boolean, *optional*): Access UI state of the user,
isTelemetryEnabled (boolean, *optional*): Enabled or disabled client-side monitoring,
roleIds (Array[string], *optional*): The id's of the roles the user has,
groupIds (Array[string], *optional*): The id's of groups that a user belongs to,
rootOrganizationId (string, *optional*): Primary organization id,
organizationIds (Array[string], *optional*): The id's of organizations that a user has
}

```
{
  "id": "string",
  "created": "2019-10-03T07:46:35.584Z",
  "isLoggedIn": true,
  "lastActiveTimestamp": "2019-10-03T07:46:35.584Z",
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "isEnabled": true,
  "isTelemetryEnabled": true,
  "roleIds": [
    "string"
  ],
  "groupIds": [
    "string"
  ],
  "rootOrganizationId": "string",
  "organizationIds": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`userId`	**userId of the user to update**	path	string
`body`		body	Model Example Value **UpdateUser {** **password** (string, *optional*): Password of the user, **email** (string, *optional*): email address of the user, **firstName** (string, *optional*): First name of the user, **lastName** (string, *optional*): Last name of the user, **isEnabled** (boolean, *optional*): Access UI state of the user, **isTelemetryEnabled** (boolean, *optional*): Enabled or disabled client-side monitoring, **roleIds** (Array[string], *optional*): The id's of the roles the user has, **groupIds** (Array[string], *optional*): The id's of groups that a user belongs to , **rootOrganizationId** (string, *optional*): Primary organization id, **organizationIds** (Array[string], *optional*): The id's of organizations that a user has **}**

```
{
  "email": "admin@connectedurbantransport.net",
  "firstName": "Admin",
  "lastName": "CUT",
  "isEnabled": true,
  "rootOrganizationId": "aade0fd4-2479-4938-8c28-6b778fdb9e15"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

`403`	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

GET /users/email/{email}

Implementation Notes

Check if email exists

Response Class (Status 200)

Successful response. The email exists.

Model | Example Value

inline_model_6 {
  code (number, *optional*),
  message (string, *optional*)
}

```
{
  "code": 0,
  "message": "string"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`email`	**email to check**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	Successful response. The email does not exist.		

422	Server Error	Model | Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

HTTP Status Code	Reason	Response Model	Headers

500	Unknown server error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

GET /users/userinfo/me

Implementation Notes

Get all the user information

Response Class (Status 200)

Successful response. All user information is returned.

Model | Example Value

UserInfo {
- **user** (BaseUser, *optional*),
- **organizations** (Array[Inline Model 1], *optional*): organizations,
- **userGroups** (Array[Inline Model 2], *optional*): userGroups,
- **roles** (Array[Inline Model 3], *optional*): roles,
- **permissions** (Array[Inline Model 4], *optional*): permissions

}

BaseUser {
- **email** (string, *optional*): email address of the user,
- **firstName** (string, *optional*): First name of the user,
- **lastName** (string, *optional*): Last name of the user,
- **isEnabled** (boolean, *optional*): Access UI state of the user,
- **isTelemetryEnabled** (boolean, *optional*): Enabled or disabled client-side monitoring,
- **roleIds** (Array[string], *optional*): The id's of the roles the user has,
- **groupIds** (Array[string], *optional*): The id's of groups that a user belongs to,
- **rootOrganizationId** (string, *optional*): Primary organization id,
- **organizationIds** (Array[string], *optional*): The id's of organizations that a user has

}

Inline Model 1 {}

Inline Model 2 {}

Inline Model 3 {}

Inline Model 4 {}

```
{
  "user": {
    "email": "string",
    "firstName": "string",
    "lastName": "string",
    "isEnabled": true,
    "isTelemetryEnabled": true,
    "roleIds": [
      "string"
    ],
    "groupIds": [
      "string"
    ],
    "rootOrganizationId": "string",
    "organizationIds": [
      "string"
    ]
  },
  "organizations": [
    {}
  ],
  "userGroups": [
    {}
  ],
  "roles": [
    {}
  ],
  "permissions": [
    {}
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

<table>
<thead>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
</thead>
<tbody>
<tr><td>422</td><td>Server Error</td><td>Model Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>500</td><td>Unknown server error</td><td>Model Example Value
<b>Errors {</b>
<b>errors</b> (Array[Error])
<b>}</b>
<b>Error {</b>
<b>code</b> (integer): Four-digits error code,
<b>message</b> (string, <i>optional</i>): Error description
<b>}</b>
<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</tbody>
</table>

Wiki

GET /wiki/topics

Implementation Notes

Get all topics

Response Class (Status 200)

Successful response. Topics is returned.

Model | Example Value

TopicsResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (Array[TopicServerSide], *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
TopicServerSide {
id (string, *optional*): Unique identifier of the topic,
parent (string, *optional*): The parent topic of this topic, if the parent value is '0', means the topic is on the top level,
type (string, *optional*): Topic types, 'custom' means this is an user-defined topic, 'device' means this is an device related topic = ['custom', 'device'],
created_at (string, *optional*): the time when the topic is created,
updated_at (string, *optional*): the time when the topic is updated,
description (string, *optional*): the description of topic,
name (string, *optional*): name of topic
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": [
    {
      "id": "string",
      "parent": "string",
      "type": "custom",
      "created_at": "2019-10-03T07:46:35.611Z",
      "updated_at": "2019-10-03T07:46:35.611Z",
      "description": "string",
      "name": "string"
    }
  ]
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`type`	type key to get filtered topics, if type key is not pass, return all topics	query	string
`parent`	parent key to get filtered child topics. Pass '0' to get the top level topic, if type key is not pass, return all topics	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
`500`	Unknown server error	Model Example Value	

WikiResponse {

code (string, *optional*): Code of the Wiki,
data (object, *optional*): Data that can be any type,
error (WikiError, *optional*),
warning (Warning, *optional*)

}

WikiError {

error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error

}

Warning {

warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning

}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

POST /wiki/topics

Implementation Notes

Create a topic. If file is being uploaded through topic, files with specific names and extensions are not allowed. Check at Attachments upload for more informations.

Response Class (Status 200)

Successful response. Topic is created.

Model | Example Value

TopicResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (TopicServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
TopicServerSide {
id (string, *optional*): Unique identifier of the topic,
parent (string, *optional*): The parent topic of this topic, if the parent value is '0', means the topic is on the top level,
type (string, *optional*): Topic types, 'custom' means this is an user-defined topic, 'device' means this is an device related topic = ['custom', 'device'],
created_at (string, *optional*): the time when the topic is created,
updated_at (string, *optional*): the time when the topic is updated,
description (string, *optional*): the description of topic,
name (string, *optional*): name of topic
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "parent": "string",
    "type": "custom",
    "created_at": "2019-10-03T07:46:35.618Z",
    "updated_at": "2019-10-03T07:46:35.618Z",
    "description": "string",
    "name": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
body		body	Model | Example Value **WikiTopicPostBody {** **name** (string), **description** (string) **}**

```
{
  "name": "string",
  "description": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model | Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

DELETE /wiki/topics/{id}

Implementation Notes

Delete a topic

Response Class (Status 200)

Successful response. Topic is removed.

Model | Example Value

TopicResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (TopicServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
TopicServerSide {
id (string, *optional*): Unique identifier of the topic,
parent (string, *optional*): The parent topic of this topic, if the parent value is '0', means the topic is on the top level,
type (string, *optional*): Topic types, 'custom' means this is an user-defined topic, 'device' means this is an device related topic = ['custom', 'device'],
created_at (string, *optional*): the time when the topic is created,
updated_at (string, *optional*): the time when the topic is updated,
description (string, *optional*): the description of topic,
name (string, *optional*): name of topic
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "parent": "string",
    "type": "custom",
    "created_at": "2019-10-03T07:46:35.624Z",
    "updated_at": "2019-10-03T07:46:35.624Z",
    "description": "string",
    "name": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
403	Permission Error	Model Example Value	

WikiResponse {
code (string, *optional*): Code of the Wiki,
data (object, *optional*): Data that can be any type,
error (WikiError, *optional*),
warning (Warning, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

GET /wiki/topics/{id}

Implementation Notes

Get a topic

Response Class (Status 200)

Successful response. Topic is created.

Model | Example Value

TopicResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (TopicServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
TopicServerSide {
id (string, *optional*): Unique identifier of the topic,
parent (string, *optional*): The parent topic of this topic, if the parent value is '0', means the topic is on the top level,
type (string, *optional*): Topic types, 'custom' means this is an user-defined topic, 'device' means this is an device related topic = ['custom', 'device'],
created_at (string, *optional*): the time when the topic is created,
updated_at (string, *optional*): the time when the topic is updated,
description (string, *optional*): the description of topic,
name (string, *optional*): name of topic
}

```json
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "parent": "string",
    "type": "custom",
    "created_at": "2019-10-03T07:46:35.632Z",
    "updated_at": "2019-10-03T07:46:35.632Z",
    "description": "string",
    "name": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model | Example Value	

WikiResponse {
 code (string, *optional*): Code of the Wiki,
 data (object, *optional*): Data that can be any type,
 error (WikiError, *optional*),
 warning (Warning, *optional*)
}
WikiError {
 error (boolean, *optional*): Whether the error,
 message (string, *optional*): Message of the error
}
Warning {
 warning (boolean, *optional*): Whether the warning,
 message (string, *optional*): Message of the warning
}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

PUT /wiki/topics/{id}

Implementation Notes

Update a topic

Response Class (Status 200)

Successful response. Topic is updated.

Model | Example Value

TopicResponse {
  code (string, *optional*): Code of the Wiki,
  error (WikiError, *optional*),
  warning (Warning, *optional*),
  data (TopicServerSide, *optional*)
}
WikiError {
  error (boolean, *optional*): Whether the error,
  message (string, *optional*): Message of the error
}
Warning {
  warning (boolean, *optional*): Whether the warning,
  message (string, *optional*): Message of the warning
}
TopicServerSide {
  id (string, *optional*): Unique identifier of the topic,
  parent (string, *optional*): The parent topic of this topic, if the parent value is '0', means the topic is on the top level,
  type (string, *optional*): Topic types, 'custom' means this is an user-defined topic, 'device' means this is an device related topic = ['custom', 'device'],
  created_at (string, *optional*): the time when the topic is created,
  updated_at (string, *optional*): the time when the topic is updated,
  description (string, *optional*): the description of topic,
  name (string, *optional*): name of topic
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "parent": "string",
    "type": "custom",
    "created_at": "2019-10-03T07:46:35.638Z",
    "updated_at": "2019-10-03T07:46:35.638Z",
    "description": "string",
    "name": "string"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
id		path	string
body		body	Model Example Value **WikiTopicPutBody {** **name** (string), **description** (string), **id** (string, *optional*): Unique identifier representing this topic **}**

```
{
  "name": "string",
  "description": "string",
  "id": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

GET /wiki/search

Implementation Notes

Search Wiki

Response Class (Status 200)

Successful response. Wiki is returned.

Model | Example Value

ArticleArrayResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (Array[ArticleServerSide], *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
ArticleServerSide {
id (string, *optional*): Unique identifier of the article,
body (string, *optional*),
title (string, *optional*),
topic_id (string, *optional*),
attachmentIds (Array[string], *optional*): An array of attachment ids,
created_at (string, *optional*): the time when the article is created,
updated_at (string, *optional*): the time when the article is updated,
organizationId (string, *optional*): The id of organization which this article belongs to,
what_changed (string, *optional*): The difference between current version and previous version of the article,
archivesSize (integer, *optional*): The amount of archives of this article,
size (integer, *optional*): The size of this article,
user (WikiUserInfo, *optional*)
}
WikiUserInfo {
id (string, *optional*): Unique identifier of user,
displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": [
    {
      "id": "string",
      "body": "string",
      "title": "string",
      "topic_id": "string",
      "attachmentIds": [
        "string"
      ],
      "created_at": "2019-10-03T07:46:35.644Z",
      "updated_at": "2019-10-03T07:46:35.644Z",
      "organizationId": "string",
      "what_changed": "string",
      "archivesSize": 0,
      "size": 0,
      "user": {
        "id": "string",
        "displayName": "string"
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`query`		query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

GET /wiki/topic/{id}/articles

Implementation Notes

Get all articles of a topic

Response Class (Status 200)

Successful response. All articles is returned.

Model | Example Value

ArticleArrayResponse {
  code (string, *optional*): Code of the Wiki,
  error (WikiError, *optional*),
  warning (Warning, *optional*),
  data (Array[ArticleServerSide], *optional*)
}
WikiError {
  error (boolean, *optional*): Whether the error,
  message (string, *optional*): Message of the error
}
Warning {
  warning (boolean, *optional*): Whether the warning,
  message (string, *optional*): Message of the warning
}
ArticleServerSide {
  id (string, *optional*): Unique identifier of the article,
  body (string, *optional*),
  title (string, *optional*),
  topic_id (string, *optional*),
  attachmentIds (Array[string], *optional*): An array of attachment ids,
  created_at (string, *optional*): the time when the article is created,
  updated_at (string, *optional*): the time when the article is updated,
  organizationId (string, *optional*): The id of organization which this article belongs to,
  what_changed (string, *optional*): The difference between current version and previous version of the article,
  archivesSize (integer, *optional*): The amount of archives of this article,
  size (integer, *optional*): The size of this article,
  user (WikiUserInfo, *optional*)
}
WikiUserInfo {
  id (string, *optional*): Unique identifier of user,
  displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": [
    {
      "id": "string",
      "body": "string",
      "title": "string",
      "topic_id": "string",
      "attachmentIds": [
        "string"
      ],
      "created_at": "2019-10-03T07:46:35.650Z",
      "updated_at": "2019-10-03T07:46:35.650Z",
      "organizationId": "string",
      "what_changed": "string",
      "archivesSize": 0,
      "size": 0,
      "user": {
        "id": "string",
        "displayName": "string"
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

GET /wiki/archives/{id}

Implementation Notes

Get an archive

Response Class (Status 200)

Successful response. Archive is returned.

Model | Example Value

ArchiveResponse {
  code (string, *optional*): Code of the Wiki,
  error (WikiError, *optional*),
  warning (Warning, *optional*),
  data (ArchiveModel, *optional*)
}
WikiError {
  error (boolean, *optional*): Whether the error,
  message (string, *optional*): Message of the error
}
Warning {
  warning (boolean, *optional*): Whether the warning,
  message (string, *optional*): Message of the warning
}
ArchiveModel {
  id (string, *optional*): Unique identifier of the archive,
  article_id (string, *optional*): Unique identifier of the article which this archive belongs to,
  body (string, *optional*): The body of the archive,
  title (string, *optional*): The title of the archive,
  what_changed (string, *optional*): The difference between current version and previous version of the article,
  user (WikiUserInfo, *optional*),
  updated_at (string, *optional*): the time when the article is updated,
  organizationId (string, *optional*): The id of organization which this archive belongs to
}
WikiUserInfo {
  id (string, *optional*): Unique identifier of user,
  displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "article_id": "string",
    "body": "string",
    "title": "string",
    "what_changed": "string",
    "user": {
      "id": "string",
      "displayName": "string"
    },
    "updated_at": "2019-10-03T07:46:35.657Z",
    "organizationId": "string"
  }
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model | Example Value	

WikiResponse {
- **code** (string, *optional*): Code of the Wiki,
- **data** (object, *optional*): Data that can be any type,
- **error** (WikiError, *optional*),
- **warning** (Warning, *optional*)

}

WikiError {
- **error** (boolean, *optional*): Whether the error,
- **message** (string, *optional*): Message of the error

}

Warning {
- **warning** (boolean, *optional*): Whether the warning,
- **message** (string, *optional*): Message of the warning

}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

GET /wiki/articles

Implementation Notes

Get all articles

Response Class (Status 200)

Successful response. Article is returned.

Model | Example Value

ArticleArrayResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (Array[ArticleServerSide], *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
ArticleServerSide {
id (string, *optional*): Unique identifier of the article,
body (string, *optional*),
title (string, *optional*),
topic_id (string, *optional*),
attachmentIds (Array[string], *optional*): An array of attachment ids,
created_at (string, *optional*): the time when the article is created,
updated_at (string, *optional*): the time when the article is updated,
organizationId (string, *optional*): The id of organization which this article belongs to,
what_changed (string, *optional*): The difference between current version and previous version of the article,
archivesSize (integer, *optional*): The amount of archives of this article,
size (integer, *optional*): The size of this article,
user (WikiUserInfo, *optional*)
}
WikiUserInfo {
id (string, *optional*): Unique identifier of user,
displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": [
    {
      "id": "string",
      "body": "string",
      "title": "string",
      "topic_id": "string",
      "attachmentIds": [
        "string"
      ],
      "created_at": "2019-10-03T07:46:35.665Z",
      "updated_at": "2019-10-03T07:46:35.665Z",
      "organizationId": "string",
      "what_changed": "string",
      "archivesSize": 0,
      "size": 0,
      "user": {
        "id": "string",
        "displayName": "string"
      }
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

POST /wiki/articles

Implementation Notes

Create an article

Response Class (Status 200)

Successful response. Article is created.

Model | Example Value

ArticleResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (ArticleServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
ArticleServerSide {
id (string, *optional*): Unique identifier of the article,
body (string, *optional*),
title (string, *optional*),
topic_id (string, *optional*),
attachmentIds (Array[string], *optional*): An array of attachment ids,
created_at (string, *optional*): the time when the article is created,
updated_at (string, *optional*): the time when the article is updated,
organizationId (string, *optional*): The id of organization which this article belongs to,
what_changed (string, *optional*): The difference between current version and previous version of the article,
archivesSize (integer, *optional*): The amount of archives of this article,
size (integer, *optional*): The size of this article,
user (WikiUserInfo, *optional*)
}
WikiUserInfo {
id (string, *optional*): Unique identifier of user,
displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "body": "string",
    "title": "string",
    "topic_id": "string",
    "attachmentIds": [
      "string"
    ],
    "created_at": "2019-10-03T07:46:35.671Z",
    "updated_at": "2019-10-03T07:46:35.671Z",
    "organizationId": "string",
    "what_changed": "string",
    "archivesSize": 0,
    "size": 0,
    "user": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
body		body	Model | Example Value **ArticlePostBody {** **body** (string, *optional*), **title** (string, *optional*), **topic_id** (string, *optional*), **attachmentIds** (Array[string], *optional*): An array of attachment ids **}**

```
{
  "body": "string",
  "title": "string",
  "topic_id": "string",
  "attachmentIds": [
    "string"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

GET /wiki/articles/{id}/history

Implementation Notes

Get the history of an article

Response Class (Status 200)

Successful response. history of an article is returned.

Model | Example Value

ArchiveArrayResponse {
  code (string, *optional*): Code of the Wiki,
  error (WikiError, *optional*),
  warning (Warning, *optional*),
  data (Array[ArchiveModel], *optional*)
}
WikiError {
  error (boolean, *optional*): Whether the error,
  message (string, *optional*): Message of the error
}
Warning {
  warning (boolean, *optional*): Whether the warning,
  message (string, *optional*): Message of the warning
}
ArchiveModel {
  id (string, *optional*): Unique identifier of the archive,
  article_id (string, *optional*): Unique identifier of the article which this archive belongs to,
  body (string, *optional*): The body of the archive,
  title (string, *optional*): The title of the archive,
  what_changed (string, *optional*): The difference between current version and previous version of the article,
  user (WikiUserInfo, *optional*),
  updated_at (string, *optional*): the time when the article is updated,
  organizationId (string, *optional*): The id of organization which this archive belongs to
}
WikiUserInfo {
  id (string, *optional*): Unique identifier of user,
  displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": [
    {
      "id": "string",
      "article_id": "string",
      "body": "string",
      "title": "string",
      "what_changed": "string",
      "user": {
        "id": "string",
        "displayName": "string"
      },
      "updated_at": "2019-10-03T07:46:35.677Z",
      "organizationId": "string"
    }
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value	

WikiResponse {

code (string, *optional*): Code of the Wiki,
data (object, *optional*): Data that can be any type,
error (WikiError, *optional*),
warning (Warning, *optional*)

}

WikiError {

error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error

}

Warning {

warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning

}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

DELETE /wiki/articles/{id}

Implementation Notes

Delete an article

Response Class (Status 200)

Successful response. Article is removed.

Model | Example Value

ArticleResponse {
  code (string, *optional*): Code of the Wiki,
  error (WikiError, *optional*),
  warning (Warning, *optional*),
  data (ArticleServerSide, *optional*)
}
WikiError {
  error (boolean, *optional*): Whether the error,
  message (string, *optional*): Message of the error
}
Warning {
  warning (boolean, *optional*): Whether the warning,
  message (string, *optional*): Message of the warning
}
ArticleServerSide {
  id (string, *optional*): Unique identifier of the article,
  body (string, *optional*),
  title (string, *optional*),
  topic_id (string, *optional*),
  attachmentIds (Array[string], *optional*): An array of attachment ids,
  created_at (string, *optional*): the time when the article is created,
  updated_at (string, *optional*): the time when the article is updated,
  organizationId (string, *optional*): The id of organization which this article belongs to,
  what_changed (string, *optional*): The difference between current version and previous version of the article,
  archivesSize (integer, *optional*): The amount of archives of this article,
  size (integer, *optional*): The size of this article,
  user (WikiUserInfo, *optional*)
}
WikiUserInfo {
  id (string, *optional*): Unique identifier of user,
  displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "body": "string",
    "title": "string",
    "topic_id": "string",
    "attachmentIds": [
      "string"
    ],
    "created_at": "2019-10-03T07:46:35.683Z",
    "updated_at": "2019-10-03T07:46:35.683Z",
    "organizationId": "string",
    "what_changed": "string",
    "archivesSize": 0,
    "size": 0,
    "user": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		WikiResponse {
		code (string, *optional*): Code of the Wiki,
		data (object, *optional*): Data that can be any type,
		error (WikiError, *optional*),
		warning (Warning, *optional*)
		}
		WikiError {
		error (boolean, *optional*): Whether the error,
		message (string, *optional*): Message of the error
		}
		Warning {
		warning (boolean, *optional*): Whether the warning,
		message (string, *optional*): Message of the warning
		}
		`{`
		`  "code": "string",`
		`  "data": {},`
		`  "error": {`
		`    "error": true,`
		`    "message": "string"`
		`  },`
		`  "warning": {`
		`    "warning": true,`
		`    "message": "string"`
		`  }`
		`}`	

500	Unknown server error	Model Example Value
		WikiResponse {
		code (string, *optional*): Code of the Wiki,
		data (object, *optional*): Data that can be any type,
		error (WikiError, *optional*),
		warning (Warning, *optional*)
		}
		WikiError {
		error (boolean, *optional*): Whether the error,
		message (string, *optional*): Message of the error
		}
		Warning {
		warning (boolean, *optional*): Whether the warning,
		message (string, *optional*): Message of the warning
		}
		`{`
		`  "code": "string",`
		`  "data": {},`
		`  "error": {`
		`    "error": true,`
		`    "message": "string"`
		`  },`
		`  "warning": {`
		`    "warning": true,`
		`    "message": "string"`
		`  }`
		`}`	

GET /wiki/articles/{id}

Implementation Notes

Get an article

Response Class (Status 200)

Successful response. Article is returned.

Model | Example Value

ArticleResponse {
  code (string, *optional*): Code of the Wiki,
  error (WikiError, *optional*),
  warning (Warning, *optional*),
  data (ArticleServerSide, *optional*)
}
WikiError {
  error (boolean, *optional*): Whether the error,
  message (string, *optional*): Message of the error
}
Warning {
  warning (boolean, *optional*): Whether the warning,
  message (string, *optional*): Message of the warning
}
ArticleServerSide {
  id (string, *optional*): Unique identifier of the article,
  body (string, *optional*),
  title (string, *optional*),
  topic_id (string, *optional*),
  attachmentIds (Array[string], *optional*): An array of attachment ids,
  created_at (string, *optional*): the time when the article is created,
  updated_at (string, *optional*): the time when the article is updated,
  organizationId (string, *optional*): The id of organization which this article belongs to,
  what_changed (string, *optional*): The difference between current version and previous version of the article,
  archivesSize (integer, *optional*): The amount of archives of this article,
  size (integer, *optional*): The size of this article,
  user (WikiUserInfo, *optional*)
}
WikiUserInfo {
  id (string, *optional*): Unique identifier of user,
  displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "body": "string",
    "title": "string",
    "topic_id": "string",
    "attachmentIds": [
      "string"
    ],
    "created_at": "2019-10-03T07:46:35.690Z",
    "updated_at": "2019-10-03T07:46:35.690Z",
    "organizationId": "string",
    "what_changed": "string",
    "archivesSize": 0,
    "size": 0,
    "user": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value	

WikiResponse {
- **code** (string, *optional*): Code of the Wiki,
- **data** (object, *optional*): Data that can be any type,
- **error** (WikiError, *optional*),
- **warning** (Warning, *optional*)

}

WikiError {
- **error** (boolean, *optional*): Whether the error,
- **message** (string, *optional*): Message of the error

}

Warning {
- **warning** (boolean, *optional*): Whether the warning,
- **message** (string, *optional*): Message of the warning

}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

PUT /wiki/articles/{id}

Implementation Notes

Update an article

Response Class (Status 200)

Successful response. Article is updated.

Model | Example Value

ArticleResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (ArticleServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
ArticleServerSide {
id (string, *optional*): Unique identifier of the article,
body (string, *optional*),
title (string, *optional*),
topic_id (string, *optional*),
attachmentIds (Array[string], *optional*): An array of attachment ids,
created_at (string, *optional*): the time when the article is created,
updated_at (string, *optional*): the time when the article is updated,
organizationId (string, *optional*): The id of organization which this article belongs to,
what_changed (string, *optional*): The difference between current version and previous version of the article,
archivesSize (integer, *optional*): The amount of archives of this article,
size (integer, *optional*): The size of this article,
user (WikiUserInfo, *optional*)
}
WikiUserInfo {
id (string, *optional*): Unique identifier of user,
displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "body": "string",
    "title": "string",
    "topic_id": "string",
    "attachmentIds": [
      "string"
    ],
    "created_at": "2019-10-03T07:46:35.698Z",
    "updated_at": "2019-10-03T07:46:35.698Z",
    "organizationId": "string",
    "what_changed": "string",
    "archivesSize": 0,
    "size": 0,
    "user": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string
body		body	Model Example Value **ArticlePutBody {** **body** (string, *optional*), **title** (string, *optional*), **topic_id** (string, *optional*), **attachmentIds** (Array[string], *optional*): An array of attachment ids **}**

```
{
  "body": "string",
  "title": "string",
  "topic_id": "string",
  "attachmentIds": [
    "string"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value	

WikiResponse {

- **code** (string, *optional*): Code of the Wiki,
- **data** (object, *optional*): Data that can be any type,
- **error** (WikiError, *optional*),
- **warning** (Warning, *optional*)

}

WikiError {

- **error** (boolean, *optional*): Whether the error,
- **message** (string, *optional*): Message of the error

}

Warning {

- **warning** (boolean, *optional*): Whether the warning,
- **message** (string, *optional*): Message of the warning

}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

POST /wiki/entity/note

Implementation Notes

If the entity note doesn't exist, will be created, otherwise will be updated.
The topic and subtopic which relate to the entity note will be created implicitly if they don't exist.

Response Class (Status 200)

Successful response. Entity note is created or saved successfully.

Model | Example Value

ArticleResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (ArticleServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
ArticleServerSide {
id (string, *optional*): Unique identifier of the article,
body (string, *optional*),
title (string, *optional*),
topic_id (string, *optional*),
attachmentIds (Array[string], *optional*): An array of attachment ids,
created_at (string, *optional*): the time when the article is created,
updated_at (string, *optional*): the time when the article is updated,
organizationId (string, *optional*): The id of organization which this article belongs to,
what_changed (string, *optional*): The difference between current version and previous version of the article,
archivesSize (integer, *optional*): The amount of archives of this article,
size (integer, *optional*): The size of this article,
user (WikiUserInfo, *optional*)
}
WikiUserInfo {
id (string, *optional*): Unique identifier of user,
displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "body": "string",
    "title": "string",
    "topic_id": "string",
    "attachmentIds": [
      "string"
    ],
    "created_at": "2019-10-03T07:46:35.708Z",
    "updated_at": "2019-10-03T07:46:35.708Z",
    "organizationId": "string",
    "what_changed": "string",
    "archivesSize": 0,
    "size": 0,
    "user": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
body		body	Model Example Value **EntityNotePostBody {** **topicId** (string): device type key is used as topic id, **topicName** (string): Entity Type name is used as Topic name. If the entity type found under Devices in CUT GUI, the topicName named as the corresponding device type; i.e. TRN ATC; If the entity type found under Map Pin in CUT GUI, then the topicName should be named Map Pin/{corresponding type}, **subtopicId** (string): Entity id is used as subtopic id, **subtopicName** (string): Entity name is used as subtopic name, **body** (string): the article body, **title** (string): the article title, **attachmentIds** (Array[string]): An array of attachment ids **}**

```
{
  "topicId": "string",
  "topicName": "string",
  "subtopicId": "string",
  "subtopicName": "string",
  "body": "string",
  "title": "string",
  "attachmentIds": [
    "string"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value **WikiResponse {** **code** (string, *optional*): Code of the Wiki, **data** (object, *optional*): Data that can be any type, **error** (WikiError, *optional*), **warning** (Warning, *optional*) **}** **WikiError {** **error** (boolean, *optional*): Whether the error, **message** (string, *optional*): Message of the error **}** **Warning {** **warning** (boolean, *optional*): Whether the warning, **message** (string, *optional*): Message of the warning **}**	

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

POST /wiki/entityType/note

Implementation Notes

If the entity note doesn't exist, will be created, otherwise will be updated.
The topic which relate to the entity type note will be created implicitly if it doesn't exist.

Response Class (Status 200)

Successful response. Entity type note is created or saved successfully.

Model | Example Value

ArticleResponse {
code (string, *optional*): Code of the Wiki,
error (WikiError, *optional*),
warning (Warning, *optional*),
data (ArticleServerSide, *optional*)
}
WikiError {
error (boolean, *optional*): Whether the error,
message (string, *optional*): Message of the error
}
Warning {
warning (boolean, *optional*): Whether the warning,
message (string, *optional*): Message of the warning
}
ArticleServerSide {
id (string, *optional*): Unique identifier of the article,
body (string, *optional*),
title (string, *optional*),
topic_id (string, *optional*),
attachmentIds (Array[string], *optional*): An array of attachment ids,
created_at (string, *optional*): the time when the article is created,
updated_at (string, *optional*): the time when the article is updated,
organizationId (string, *optional*): The id of organization which this article belongs to,
what_changed (string, *optional*): The difference between current version and previous version of the article,
archivesSize (integer, *optional*): The amount of archives of this article,
size (integer, *optional*): The size of this article,
user (WikiUserInfo, *optional*)
}
WikiUserInfo {
id (string, *optional*): Unique identifier of user,
displayName (string, *optional*): user name
}

```
{
  "code": "string",
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  },
  "data": {
    "id": "string",
    "body": "string",
    "title": "string",
    "topic_id": "string",
    "attachmentIds": [
      "string"
    ],
    "created_at": "2019-10-03T07:46:35.716Z",
    "updated_at": "2019-10-03T07:46:35.716Z",
    "organizationId": "string",
    "what_changed": "string",
    "archivesSize": 0,
    "size": 0,
    "user": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
body		body	Model Example Value

EntityTypeNotePostBody {

topicId (string): device type key is used as topic id,

topicName (string): Entity Type name is used as Topic name. If the entity type found under Devices in CUT GUI, the topicName named as the corresponding device type; i.e. TRN ATC; If the entity type found under Map Pin in CUT GUI, then the topicName should be named Map Pin/{corresponding type},

body (string): the article body,

title (string): the article title,

attachmentIds (Array[string]): An array of attachment ids

}

```
{
  "topicId": "string",
  "topicName": "string",
  "body": "string",
  "title": "string",
  "attachmentIds": [
    "string"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
500	Unknown server error	Model Example Value	

WikiResponse {
- **code** (string, *optional*): Code of the Wiki,
- **data** (object, *optional*): Data that can be any type,
- **error** (WikiError, *optional*),
- **warning** (Warning, *optional*)

}

WikiError {
- **error** (boolean, *optional*): Whether the error,
- **message** (string, *optional*): Message of the error

}

Warning {
- **warning** (boolean, *optional*): Whether the warning,
- **message** (string, *optional*): Message of the warning

}

```
{
  "code": "string",
  "data": {},
  "error": {
    "error": true,
    "message": "string"
  },
  "warning": {
    "warning": true,
    "message": "string"
  }
}
```

Wiki Admin

GET /wikiAdmin/settings

Implementation Notes

Get wiki settings of all organizations

Response Class (Status 200)

Success response

Model | Example Value

WikiSettings {
organizationId (string, *optional*): Id of the organization,
maxSize (integer, *optional*): Maz size,
fileSize (integer, *optional*): File size,
warningSize (integer, *optional*): Warning size,
_id (string, *optional*, read only),
size (integer, *optional*, read only): Size,
name (string, *optional*, read only): Name of the setting,
initialized (boolean, *optional*, read only): Whether to initialized,
enabled (boolean, *optional*): Whether to enabled
}

```
{
  "organizationId": "string",
  "maxSize": 0,
  "fileSize": 0,
  "warningSize": 0,
  "_id": "string",
  "size": 0,
  "name": "string",
  "initialized": true,
  "enabled": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`  "errors": [`
		`    {`
		`      "code": 1000,`
		`      "message": "Unknown error with API"`
		`    }`
		`  ]`
		`}`	

POST /wikiAdmin/settings

Implementation Notes

Initialize wiki for a organization

Response Class (Status 201)

Success response

Model | Example Value

WikiSettings {
organizationId (string, *optional*): Id of the organization,
maxSize (integer, *optional*): Maz size,
fileSize (integer, *optional*): File size,
warningSize (integer, *optional*): Warning size,
_id (string, *optional*, read only),
size (integer, *optional*, read only): Size,
name (string, *optional*, read only): Name of the setting,
initialized (boolean, *optional*, read only): Whether to initialized,
enabled (boolean, *optional*): Whether to enabled
}

```
{
  "organizationId": "string",
  "maxSize": 0,
  "fileSize": 0,
  "warningSize": 0,
  "_id": "string",
  "size": 0,
  "name": "string",
  "initialized": true,
  "enabled": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
body		body	Model | Example Value

WikiSettingsPostBody {
organizationId (string): Id of the organization,
maxSize (integer): Maz size,
fileSize (integer): File size,
warningSize (integer): Warning size
}

```
{
  "organizationId": "string",
  "maxSize": 0,
  "fileSize": 0,
  "warningSize": 0
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

DELETE /wikiAdmin/settings/{id}

Implementation Notes

id is the id of organization, not wiki setting id

Parameters

Parameter	Description	Parameter Type	Data Type
`x-access-token`	User token.	header	string
`id`		path	string

Response Messages

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>204</td><td>Wiki data destroyed.</td><td></td><td></td></tr>
<tr><td>403</td><td>Permission Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
<tr><td>422</td><td>Server Error</td><td>Model | Example Value

Errors {
errors (Array[Error])
}
Error {
code (integer): Four-digits error code,
message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre></td><td></td></tr>
</table>

<table>
<tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th><th>Headers</th></tr>
<tr><td>500</td><td>Internal Server Error</td><td>Model | Example Value

Errors {
  errors (Array[Error])
}
Error {
  code (integer): Four-digits error code,
  message (string, *optional*): Error description
}

<pre>
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
</pre>
</td><td></td></tr>
</table>

GET /wikiAdmin/settings/{id}

Implementation Notes

Get a setting by its id

Response Class (Status 200)

Success resonponse

Model | Example Value

WikiSettings {
organizationId (string, *optional*): Id of the organization,
maxSize (integer, *optional*): Maz size,
fileSize (integer, *optional*): File size,
warningSize (integer, *optional*): Warning size,
_id (string, *optional*, read only),
size (integer, *optional*, read only): Size,
name (string, *optional*, read only): Name of the setting,
initialized (boolean, *optional*, read only): Whether to initialized,
enabled (boolean, *optional*): Whether to enabled
}

```
{
  "organizationId": "string",
  "maxSize": 0,
  "fileSize": 0,
  "warningSize": 0,
  "_id": "string",
  "size": 0,
  "name": "string",
  "initialized": true,
  "enabled": true
}
```

Response Content Type application/json ▾

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		{
		"errors": [
		{
		"code": 1000,
		"message": "Unknown error with API"
		}
		]
		}	

PUT /wikiAdmin/settings/{id}

Implementation Notes

id is the id of wiki setting, not organization id

Response Class (Status 201)

Success resonponse

Model | Example Value

WikiSettings {
organizationId (string, *optional*): Id of the organization,
maxSize (integer, *optional*): Maz size,
fileSize (integer, *optional*): File size,
warningSize (integer, *optional*): Warning size,
_id (string, *optional*, read only),
size (integer, *optional*, read only): Size,
name (string, *optional*, read only): Name of the setting,
initialized (boolean, *optional*, read only): Whether to initialized,
enabled (boolean, *optional*): Whether to enabled
}

```
{
  "organizationId": "string",
  "maxSize": 0,
  "fileSize": 0,
  "warningSize": 0,
  "_id": "string",
  "size": 0,
  "name": "string",
  "initialized": true,
  "enabled": true
}
```

Response Content Type application/json ▼

Parameters

Parameter	Description	Parameter Type	Data Type
x-access-token	User token.	header	string
id		path	string

Parameter	Description	Parameter Type	Data Type
body		body	Model Example Value **WikiSettings {** **organizationId** (string, *optional*): Id of the organization, **maxSize** (integer, *optional*): Maz size, **fileSize** (integer, *optional*): File size, **warningSize** (integer, *optional*): Warning size, **_id** (string, *optional*, read only), **size** (integer, *optional*, read only): Size, **name** (string, *optional*, read only): Name of the setting, **initialized** (boolean, *optional*, read only): Whether to initialized, **enabled** (boolean, *optional*): Whether to enabled **}**

```
{
  "organizationId": "string",
  "maxSize": 0,
  "fileSize": 0,
  "warningSize": 0,
  "_id": "string",
  "size": 0,
  "name": "string",
  "initialized": true,
  "enabled": true
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers

403	Permission Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}	

```
{
  "errors": [
    {
      "code": 1000,
      "message": "Unknown error with API"
    }
  ]
}
```

HTTP Status Code	Reason	Response Model	Headers

422	Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

500	Internal Server Error	Model Example Value
		Errors {
		errors (Array[Error])
		}
		Error {
		code (integer): Four-digits error code,
		message (string, *optional*): Error description
		}
		`{`
		`"errors": [`
		`{`
		`"code": 1000,`
		`"message": "Unknown error with API"`
		`}`
		`]`
		`}`	

[: /rest/ctcapi/v3]