

Operational Insights 1.0.0

[Base URL: dnaspaces.io/api/am/v1]

Operational Insights (OI) is a comprehensive single resource for managing, monitoring, and optimizing your assets, Internet of Things (IOT), alerting system, and operational workflows. Using a technology-agnostic approach, the solution can use a wide range of tags, sensors, including Wi-Fi, RFID, and environmental monitors, to continually integrate, monitor, and manage connected operations.

This document describes the Operational Insights (OI) APIs for:

1. Adding a new asset
2. Fetching assets using filters
3. Fetching a list of assets
4. Fetching a single asset
5. Updating an asset
6. Deleting an asset
7. Adding a new category
8. Fetching a list of categories
9. Fetching a single category
10. Updating a category
11. Deleting a category
12. Adding a new department
13. Fetching a list of departments
14. Fetching a single department
15. Updating a department
16. Deleting a department
17. Adding a new custom field
18. Fetching a list of custom fields
19. Updating a custom field
20. Deleting a custom field
21. Adding a new tag
22. Fetching a list of tags
23. Fetching a list of unassigned tags

- 24. Fetching a single tag
- 25. Updating a tag
- 26. Deleting a tag
- 27. Fetching location history of an asset
- 28. Fetching telemetry history of an asset

Accessing these APIs will require an access token for authentication. To retrieve this token, make a GET request to <https://dnaspaces.io/api/am/v1/auth/license/accesstoken>. Attach the permanent API token provided to you along with the prefix "JWT " in the request header under key "Authorization" This API will return an object where you can find the API access token under key "token". Attach this API access token along with the prefix "JWT " in the request header under key "Authorization" to the APIs you wish to execute. The token has an expiry time of 30 minutes.

Schemes

HTTPS

Authorize



Assets



POST

/entities/access/assets



This API is used to create assets. Before creating an asset, be sure you have the id of the category, department, tag(s), and/or custom fields that you wish to assign to the asset. Each entity is assigned an id when added to the system.

Parameters

Try it out

Name

Description

Name	Description
------	-------------

body * required

Request body for New Asset creation - Refer to "NewAsset" schema for details

object

Example Value | Model

(body)

```
{
  "serial": "serial-123",
  "name": "name-123",
  "description": "Asset Description",
  "status": "Active",
  "tags": [
    1
  ],
  "category": 100,
  "subcategories": [
    "subcategory"
  ],
  "department": 200,
  "customData": {
    "1": "test"
  },
  "categoryCustomData": {
    "1": "test"
  },
  "site": {
    "id": "abc",
    "level": "FLOOR",
    "name": "campus -> building -> floor",
  }
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
------	-------------

201

Successfully created entity

Example Value | Model

```
{
  "id": 0
}
```

Code	Description
------	-------------

400	The request body has invalid request values
-----	---

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401	Unauthorized
-----	--------------

Example Value | Model

```
Unauthorized
```

409	The serial number is not unique
-----	---------------------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/assets



This API is used to fetch a list of assets. This API requires that a limit and page is provided in the query parameters. The maximum limit that this API currently supports is 100. Pages start at page 0. To fetch the next series of assets, simply increment your page value.

Parameters

Try it out

Name	Description
------	-------------

limit * required

How many is to be retrieved at once. 100 is the max.

integer

(query)

limit - How many is to be retrieved at once. 100 is the max.

page * required

Which page to start fetching from. Starts from page 0

integer

(query)

page - Which page to start fetching from. Starts from page 0

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully retrieved all assets

Example Value | Model

```
[
  {
    "id": 1,
    "serial": "serial-123",
    "name": "name-123",
    "location": {
      "x": 100,
      "y": 100,
      "floorId": "abc",
      "zones": [
        {
          "color": "#abc",
          "id": "abc",
          "name": "abc"
        }
      ]
    },
    "chokePoints": [
      {
        "id": "abc",
        "name": "abc"
      }
    ],
    "hierarchy": "abc",
```

401

Unauthorized

Example Value | Model

Unauthorized

Code	Description
------	-------------

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/assets/{id}



This API is used to fetch a single asset using its ID.

Parameters

Try it out

Name	Description
------	-------------

id * required

id of the asset

integer

(path)

id - id of the asset

Responses

Response content type

application/json



Code	Description
------	-------------

Code**Description**

200

Successfully retrieved single asset

Example Value | Model

```
{
  "id": 1,
  "serial": "serial-123",
  "name": "name-123",
  "location": {
    "x": 100,
    "y": 100,
    "floorId": "abc",
    "zones": [
      {
        "color": "#abc",
        "id": "abc",
        "name": "abc"
      }
    ],
    "chokePoints": [
      {
        "id": "abc",
        "name": "abc"
      }
    ],
    "hierarchy": "abc",
    "lat": -999,
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Code**Description**

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

PUT`/entities/access/assets/{id}`

This API is used to update an asset using its ID. Only provide the attributes that you wish to update in the request body.

Parameters[Try it out](#)**Name****Description**

Name	Description
------	-------------

body * required

object
(*body*)

Request body for Asset Update - Refer to "UpdatedAsset" schema for details. Ensure that the request body has all the same attributes as the response body when fetching for the asset you want to update. All missing attributes will be overwritten with a null value in the system, which can interfere with current workflows. Only modify attributes given from the "NewAsset" schema.

Example Value | Model

```
{
  "id": 11,
  "serial": "serial-123",
  "name": "name-123",
  "description": "Asset Description",
  "status": "Active",
  "tags": [
    1
  ],
  "category": 100,
  "subcategories": [
    "subcategory 1"
  ],
  "department": 200,
  "customData": {
    "1": "test"
  },
  "categoryCustomData": {
    "1": "test"
  },
  "site": {
    "id": "abc",
    "level": "FLOOR",
```

Parameter content type

application/json

id * required

integer
(*path*)

id of the asset

id - id of the asset

Responses

Response content type

application/json

Code

Description

200

Successfully updated entity

Code	Description
------	-------------

400	The request body has invalid request values
-----	---

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401	Unauthorized
-----	--------------

Example Value | Model

```
Unauthorized
```

409	The serial number is not unique
-----	---------------------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

DELETE /entities/access/assets/{id}



This API is used to delete an asset using its ID. Deleting an asset will mark any of its assigned tags as "Available".

Parameters

Try it out

Name	Description
------	-------------

id * required

id of the asset

integer

(path)

Responses

Response content type

application/json



Code	Description
------	-------------

200

Successfully deleted entity

401

Unauthorized

Example Value | Model

```
Unauthorized
```

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

POST

/entities/access/search/assets



This API is used to fetch assets using a filter. Filters using asset serial number or tag mac address is currently supported. This API requires that a limit and page is provided as part of the request body. The maximum limit that this API currently supports is 100. Pages start at page 0. To fetch the next series of assets, simply increment your page value.

Parameters

Try it out

Name	Description
------	-------------

body * required

Example Value | Model

object

(body)

```
{
  "pagination": {
    "limit": 10,
    "page": 0
  },
  "fields": [
    {
      "key": "serial",
      "type": "TEXT",
      "operator": "contains",
      "values": [
        "serial123"
      ]
    }
  ]
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
------	-------------

Code**Description**

201

Successfully retrieved all assets using filters

Example Value | Model

```
[
  {
    "count": 1,
    "data": [
      {
        "id": 1,
        "serial": "serial-123",
        "name": "name-123",
        "location": {
          "x": 100,
          "y": 100,
          "floorId": "abc",
          "zones": [
            {
              "color": "#abc",
              "id": "abc",
              "name": "abc"
            }
          ]
        },
        "chokePoints": [
          {
            "id": "abc",
            "name": "abc"
          }
        ]
      }
    ]
  }
]
```

400

The request body has invalid request values

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model**Unauthorized**

Code	Description
------	-------------

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Categories



POST

/entities/access/categories



This API is used to create a new category. Before creating a category, be sure you have the id of the department and/or custom fields that you wish to assign to the category. Each entity is assigned an id when added to the system.

Parameters

Try it out

Name	Description
------	-------------

Name	Description
------	-------------

body * required

Request body for New Category creation - Refer to "NewCategory" schema for details

object

Example Value | Model

(body)

```
{
  "name": "cat1",
  "description": "description for category",
  "subcategories": [
    "subcategory 1",
    "subcategory 2"
  ],
  "customData": {
    "1": "test"
  },
  "template": {
    "entityName": "CATEGORY_CUSTOM",
    "fields": [
      {
        "label": "Custom Field",
        "key": "customField",
        "entityName": "CATEGORY_CUSTOM",
        "description": "This is a custom field",
        "templateLabel": "Category Custom Field",
        "type": "SELECT",
        "options": [
          "option 1",
          "option 2"
        ]
      }
    ]
  }
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
------	-------------

201

Successfully created entity

Example Value | Model

```
{
  "id": 0
}
```

Code	Description
------	-------------

400	The request body has invalid request values
-----	---

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401	Unauthorized
-----	--------------

Example Value | Model

```
Unauthorized
```

409	The name is not unique
-----	------------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/categories



This API is used to fetch a list of categories. This API requires that a limit and page is provided in the query parameters. The maximum limit that this API currently supports is 100. Pages start at page 0. To fetch the next series of categories, simply increment your page value.

Parameters

Try it out

Name	Description
------	-------------

limit * required

How many is to be retrieved at once. 100 is the max.

integer

(query)

limit - How many is to be retrieved at once. 100 is the max.

page * required

Which page to start fetching from. Starts from page 0

integer

(query)

page - Which page to start fetching from. Starts from page 0

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully retrieved all categories

Example Value | Model

```
{}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/categories/{id}



This API is used to fetch a category using its ID.

Parameters

[Try it out](#)

Name	Description
------	-------------

id * required

id of the category

integer

(path)

Responses

Response content type

application/json



Code	Description
------	-------------

200

Successfully retrieved single asset

Example Value | Model

```
{
  "id": 1,
  "name": "cat1",
  "description": "description for category",
  "subcategories": [
    "subcategory 1",
    "subcategory 2"
  ],
  "template": {
    "id": 1,
    "entityName": "CATEGORY_CUSTOM",
    "fields": [
      {
        "id": 100,
        "key": 100,
        "label": "custom field",
        "entityName": "CATEGORY_CUSTOM",
        "description": "description of field",
        "templateLabel": "category custom field",
        "type": "SELECT",
        "options": [
          "option 1",
          "option 2"
        ]
      }
    ]
  }
}
```

401

Unauthorized

Example Value | Model

Unauthorized

Code**Description**

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

PUT`/entities/access/categories/{id}`

This API is used to update a category using its ID. Only provide the attributes that you wish to update in the request body.

Parameters[Try it out](#)**Name****Description**

Name	Description
------	-------------

body * required

object
(*body*)

Request body for Category Update - Refer to "UpdatedCategory" schema for details. Ensure that the request body has all the same attributes as the response body when fetching for the asset you want to update. All missing attributes will be overwritten with a null value in the system, which can interfere with current workflows. Only modify attributes given from the "NewCategory" schema.

Example Value | Model

```
{
  "id": 1,
  "name": "cat1",
  "description": "description for category",
  "subcategories": [
    "subcategory 1",
    "subcategory 2"
  ],
  "customData": {
    "1": "test"
  },
  "template": {
    "entityName": "CATEGORY_CUSTOM",
    "fields": [
      {
        "label": "Custom Field",
        "key": "customField",
        "entityName": "CATEGORY_CUSTOM",
        "description": "This is a custom field",
        "templateLabel": "Category Custom Field",
        "type": "SELECT",
        "options": [
          "option 1",
```

Parameter content type

application/json

id * required

integer
(*path*)

id of the category

id - id of the category

Responses

Response content type

application/json

Code

Description

200

Successfully updated entity

Code Description

400 The request body has invalid request values

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401 Unauthorized

Example Value | Model

```
Unauthorized
```

409 The name is not unique

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500 Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

DELETE /entities/access/categories/{id}



Deleting a category using its ID. Deletion of a category can only be performed when there are no assets associated to the category.

Parameters

Try it out

Name	Description
------	-------------

id * required

id of the category

integer

(path)

id - id of the category

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully deleted entity

400

Assets are still assigned to this entity. Deletion could not be performed

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Code	Description
------	-------------

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Departments

POST

/entities/access/departments



This API is used to create a new department. Before creating a department, be sure you have the id of custom fields that you wish to assign to the category. Each entity is assigned an id when added to the system.

Parameters

Try it out

Name	Description
------	-------------

body * required

Request body for New Department creation - Refer to "NewDepartment" schema for details

object

(body)

Example Value | Model

```
{
  "name": "dept1",
  "customData": {
    "1": "test"
  },
  "description": "description for department"
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
------	-------------

201

Successfully created entity

Example Value | Model

```
{
  "id": 0
}
```

400

The request body has invalid request values

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

409

The name is not unique

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/departments



This API is used to fetch a list of departments. This API requires that a limit and page is provided in the query parameters. The maximum limit that this API currently supports is 100. Pages start at page 0. To fetch the next series of departments, simply increment your page value.

Parameters

[Try it out](#)

Name	Description
------	-------------

limit * required

How many is to be retrieved at once. 100 is the max.

integer

(query)

limit - How many is to be retrieved at once. 100 is the max.

page * required

Which page to start fetching from. Starts from page 0

integer

(query)

page - Which page to start fetching from. Starts from page 0

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully retrieved all departments

Example Value | Model

```
[
  {
    "id": 1,
    "name": "dept1",
    "assetCount": 10,
    "categories": [
      {
        "id": 1,
        "name": "cat1",
        "description": "description for category",
        "subcategories": [
          "subcategory 1",
          "subcategory 2"
        ],
        "template": {
          "id": 1,
          "entityName": "CATEGORY_CUSTOM",
          "fields": [
            {
              "id": 100,
              "key": 100,
              "label": "custom field",
              "entityName": "CATEGORY_CUSTOM",
```

Code	Description
401	Unauthorized Example Value Model <pre>Unauthorized</pre>
500	Internal server error Example Value Model <pre>{ "code": "ERROR_CODE", "message": "error message", "title": "error title", "httpStatus": 0 }</pre>

GET
/entities/access/departments/{id}
🔒

This API is used to fetch a department using its ID.

Parameters
Try it out

Name	Description
id * required integer (path)	id of the department <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">id - id of the department</div>

Responses

Response content type
application/json ▼

Code	Description
------	-------------

Code**Description**

200

Successfully retrieved single department

Example Value | Model

```
{
  "id": 1,
  "name": "dept1",
  "assetCount": 10,
  "categories": [
    {
      "id": 1,
      "name": "cat1",
      "description": "description for category",
      "subcategories": [
        "subcategory 1",
        "subcategory 2"
      ],
      "template": {
        "id": 1,
        "entityName": "CATEGORY_CUSTOM",
        "fields": [
          {
            "id": 100,
            "key": 100,
            "label": "custom field",
            "entityName": "CATEGORY_CUSTOM",
            "description": "description of field",
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Code	Description
------	-------------

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

PUT

/entities/access/departments/{id}



This API is used to update a department using its ID. Only provide the attributes that you wish to update in the request body.

Parameters

Try it out

Name	Description
------	-------------

body * required

object

(*body*)

Request body for Department Update - Refer to "UpdatedDepartment" schema for details. Ensure that the request body has all the same attributes as the response body when fetching for the asset you want to update. All missing attributes will be overwritten with a null value in the system, which can interfere with current workflows. Only modify attributes given from the "NewDepartment" schema.

Example Value | Model

```
{
  "id": 1,
  "name": "dept1",
  "customData": {
    "1": "test"
  },
  "description": "description for department"
}
```

Parameter content type

application/json



id * required

id of the department

integer

(*path*)

id - id of the department



Code	Description
------	-------------

200	Successfully updated entity
-----	-----------------------------

400	The request body has invalid request values
-----	---

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401	Unauthorized
-----	--------------

Example Value | Model

```
Unauthorized
```

409	The name is not unique
-----	------------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

DELETE

/entities/access/departments/{id}



This API is used to delete a department using its ID. Deletion of a department can only be performed when there are no assets and categories associated to the department.

Parameters

Try it out

Name	Description
------	-------------

id * required

id of the department

integer

(path)

id - id of the department

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully deleted entity

400

Assets are still assigned to this entity. Deletion could not be performed

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model

Unauthorized

Code

Description

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Fields



POST

/entities/access/fields



This API is used to create a new field for an entity.

Parameters

[Try it out](#)

Name	Description
------	-------------

body * required

Request body for New Field creation - Refer to "NewField" schema for details

object

(body)

Example Value | Model

```
{
  "field": {
    "label": "Custom Field",
    "key": "customField",
    "entityName": "TAGS",
    "description": "This is a custom field",
    "templateLabel": "Tag Custom Field",
    "type": "SELECT",
    "options": [
      "option 1",
      "option 2"
    ],
    "multiple": false,
    "validation": "ANY",
    "editable": true,
    "general": true,
    "required": false
  }
}
```

Parameter content type

application/json ▾

Responses

Response content type

application/json ▾

Code	Description
------	-------------

201

Successfully created entity

Example Value | Model

```
{
  "id": 0
}
```


Code	Description
------	-------------

400	The request body has invalid request values
-----	---

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401	Unauthorized
-----	--------------

Example Value | Model

```
Unauthorized
```

409	The label is not unique
-----	-------------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/fields/templates/{entityName}



This API is used to fetch all custom fields for an entity. Current entities includes ASSETS, CATEGORIES, DEPARTMENTS, and TAGS.

Parameters

Try it out

Name	Description
------	-------------

entityName * required

string

(path)

which entityName to fetch all fields for (ASSETS, CATEGORIES, DEPARTMENTS, TAGS)

entityName - which entityName to fetch all fields for (ASSETS)

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully retrieved all fields

Example Value | Model

```
[
  {
    "id": 100,
    "entityName": "TAGS",
    "lastModified": "2017-03-17T04:43:34.156Z",
    "fields": [
      {
        "id": 100,
        "key": 100,
        "label": "custom field",
        "entityName": "TAGS",
        "description": "description of field",
        "templateLabel": "Tag custom field",
        "type": "SELECT",
        "options": [
          "option 1",
          "option 2"
        ],
        "collection": null,
        "validation": "ANY",
        "isDefault": true,
        "required": true,
        "templateUnique": true,
      }
    ]
  }
]
```

401

Unauthorized

Example Value | Model

Unauthorized

Code	Description
------	-------------

500

Successfully created entity

Example Value | Model

```
{
  "id": 0
}
```

PUT

/entities/access/fields/{id}



This API is used to update a custom field using its ID. Only provide the attributes that you wish to update in the request body.

Parameters

Try it out

Name	Description
------	-------------

body * required

object

(body)

Request body for Field Update - Refer to "UpdatedField" schema for details. Ensure that the request body has all the same attributes as the response body when fetching for the asset you want to update. All missing attributes will be overwritten with a null value in the system, which can interfere with current workflows. Only modify attributes given from the "NewField" schema.

Example Value | Model

```
{
  "id": 100,
  "label": "Custom Field",
  "description": "This is a custom field",
  "templateLabel": "Tag Custom Field",
  "type": "SELECT",
  "options": [
    "option 1",
    "option 2"
  ],
  "multiple": false,
  "required": false
}
```

Parameter content type

application/json

id * required

integer

(path)

id of the field

id - id of the field

Responses

Response content type

application/json



Code	Description
------	-------------

200

Successfully updated entity

400

The request body has invalid request values

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

409

The label is not unique

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

DELETE

/entities/access/fields/{id}



This API is used to delete a field using its ID.

Parameters

Try it out

Name	Description
------	-------------

id * required

id of the field

integer

(path)

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully deleted entity

401

Unauthorized

Example Value | Model

```
Unauthorized
```

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Tags

POST

/entities/access/tags



This API is used to create a new tag.

Parameters

Try it out

Name	Description
------	-------------

body * required

Request body for New Tag creation - Refer to "NewTag" schema for details

object

(body)

Example Value | Model

```
{
  "serial": "11:22:33:44:55:66",
  "name": "Tag 1",
  "customData": {
    "1": "test"
  },
  "type": "RFID"
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
------	-------------

201

Successfully created entity

Example Value | Model

```
{
  "id": 0
}
```

Code	Description
------	-------------

400	The request body has invalid request values
-----	---

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401	Unauthorized
-----	--------------

Example Value | Model

```
Unauthorized
```

409	The mac address is not unique
-----	-------------------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/tags

This API is used to fetch a list of tags. This API requires that a limit and page is provided in the query parameters. The maximum limit that this API currently supports is 100. Pages start at page 0. To fetch the next series of tags, simply increment your page value.

Parameters

Try it out

Name	Description
------	-------------

limit * required

How many is to be retrieved at once. 100 is the max.

integer

(query)

limit - How many is to be retrieved at once. 100 is the max.

page * required

Which page to start fetching from. Starts from page 0

integer

(query)

page - Which page to start fetching from. Starts from page 0

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully retrieved all tags

Example Value | Model

```
[
  {
    "id": 1,
    "serial": "11:22:33:44:55:66",
    "name": "Tag 1",
    "data": [
      {}
    ],
    "assetId": 1,
    "type": "RFID",
    "location": {},
    "lastHeard": "2018-01-16T22:50:25.368Z"
  }
]
```

401

Unauthorized

Example Value | Model

Unauthorized

Code	Description
------	-------------

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/tags/unassigned

This API is used to fetch a list of unassigned tags. This API requires that a limit and page is provided in the query parameters. The maximum limit that this API currently supports is 100. Pages start at page 0. To fetch the next series of unassigned tags, simply increment your page value.

Parameters

Try it out

Name	Description
------	-------------

limit * required

How many is to be retrieved at once. 100 is the max.

integer

(query)

limit - How many is to be retrieved at once. 100 is the max.

page * required

Which page to start fetching from. Starts from page 0

integer

(query)

page - Which page to start fetching from. Starts from page 0

Responses

Response content type

application/json



Code	Description
------	-------------

Code	Description
------	-------------

200

Successfully retrieved all unassigned tags

Example Value | Model

```
[
  {
    "id": 1,
    "serial": "11:22:33:44:55:66",
    "name": "Tag 1",
    "data": [
      {}
    ],
    "assetId": 1,
    "type": "RFID",
    "location": {},
    "lastHeard": "2018-01-16T22:50:25.368Z"
  }
]
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/entities/access/tags/{id}



This API is used to fetch a tag using its ID.

Parameters

Try it out

Name	Description
------	-------------

Name	Description
------	-------------

id * required

id of the tag

integer

(path)

Responses

Response content type

application/json



Code	Description
------	-------------

200

Successfully retrieved single tag

Example Value | Model

```
{
  "id": 1,
  "serial": "11:22:33:44:55:66",
  "name": "Tag 1",
  "data": [
    {}
  ],
  "assetId": 1,
  "type": "RFID",
  "location": {},
  "lastHeard": "2018-01-16T22:50:25.368Z"
}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

404

The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Code	Description
------	-------------

500	Internal server error
-----	-----------------------

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

PUT

/entities/access/tags/{id}



This API is used to update a tag using its ID. Only provide the attributes that you wish to update in the request body.

Parameters

Try it out

Name	Description
------	-------------

body * required

object

(body)

Request body for Tag Update - Refer to "UpdatedTag" schema for details. Ensure that the request body has all the same attributes as the response body when fetching for the asset you want to update. All missing attributes will be overwritten with a null value in the system, which can interfere with current workflows. Only modify attributes given from the "NewTag" schema.

Example Value | Model

```
{
  "id": 1,
  "serial": "11:22:33:44:55:66",
  "name": "Tag 1",
  "customData": {
    "1": "test"
  },
  "type": "RFID"
}
```

Parameter content type

application/json

id * required

integer

(path)

id of the tag

id - id of the tag

Responses

Response content type

application/json



Code	Description
------	-------------

200

Successfully updated entity

400

The request body has invalid request values

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

409

The mac address is not unique

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

DELETE

/entities/access/tags/{id}



This API is used to delete a tag using its ID. A tag can only be deleted when unassigned to an asset.

Parameters

Try it out

Name	Description
------	-------------

id * required

id of the tag

integer

(path)

Responses

Response content type

application/json

Code	Description
------	-------------

200

Successfully deleted entity

400

Assets are still assigned to this entity. Deletion could not be performed

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

401

Unauthorized

Example Value | Model

```
Unauthorized
```

Code **Description**

404 The entity could not be found

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

500 Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

History

GET

/history/access/location/{assetId}



This API is used to fetch the location history of an asset using its ID. This API requires a start and end time to be provided. The maximum start and end time interval supported is one week.

Parameters

Try it out

Name	Description
------	-------------

assetId * required integer (path)	id of the asset <input type="text" value="assetId - id of the asset"/>
--	---

startTime * required integer (query)	Unix time stamp of start time of interval. <input type="text" value="startTime - Unix time stamp of start time of interval."/>
---	---

endtime * required integer (query)	Unix time stamp of end time of interval. <input type="text" value="endtime - Unix time stamp of end time of interval."/>
---	---



Code	Description
------	-------------

200

Successfully retrieved location history for asset

Example Value | Model

```
{
  "assetId": 1,
  "tenantId": 1,
  "startTime": 1524009600,
  "endTime": 1524009600,
  "series": [
    {
      "data": [
        {
          "x": 100,
          "y": 100,
          "floorId": "abc",
          "zones": [
            {
              "color": "#abc",
              "id": "abc",
              "name": "abc"
            }
          ]
        }
      ],
      "chokePoints": [
        {
          "id": "abc",
          "name": "abc"
        }
      ]
    }
  ]
}
```

500

Internal server error

Example Value | Model

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

GET

/history/access/telemetry/{assetId}



This API is used to fetch the telemetry history of an asset using its ID. This API requires a start and end time to be provided. The maximum start and end time interval supported is one week.

Parameters

Try it out

Name	Description
assetId * required integer (path)	id of the asset <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">assetId - id of the asset</div>
startTime * required integer (query)	Unix time stamp of start time of interval. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">startTime - Unix time stamp of start time of interval.</div>
endTime * required integer (query)	Unix time stamp of end time of interval. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">endTime - Unix time stamp of end time of interval.</div>
tagMacAddress * required string (query)	Mac address of tag assigned to asset. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">tagMacAddress - Mac address of tag assigned to asset.</div>
type * required string (query)	Type of telemetry history to be retrieved (TEMPERATURE or HUMIDITY) <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">type - Type of telemetry history to be retrieved (TEMPERATUI</div>

Responses

Response content type

application/json



Code	Description
------	-------------

Code **Description**

200 Successfully retrieved telemetry history for asset

Example Value | **Model**

```
{
  "assetId": 1,
  "tenantId": 1,
  "startTime": 1524009600,
  "endTime": 1524009600,
  "type": "TEMPERATURE",
  "units": "Degree Celsius",
  "series": [
    {
      "data": [
        {
          "value": 10,
          "timestamp": "2018-04-19T07:01:04.411Z"
        }
      ]
    }
  ]
}
```

500 Internal server error

Example Value | **Model**

```
{
  "code": "ERROR_CODE",
  "message": "error message",
  "title": "error title",
  "httpStatus": 0
}
```

Models

Asset ▾ {

description:	The following properties define an Asset. Any properties that appear in the response that have not been defined are considered custom field keys and values.
id	integer <i>example: 1</i> The identification number of the asset
serial	string <i>example: serial-123</i> The unique serial number of the asset
name	string <i>example: name-123</i> The name of the asset
location	LocationData > {...}
description	string <i>example: description of asset</i>

	The description of the asset
status	<p>string <i>example: Active</i></p> <p>Will be either "Active" or "Inactive" - identifies if asset is currently set to receive events</p>
tags	<p>▼ [</p> <p>The array of tag objects associated to the asset</p> <p>Tag > {...}</p>
category	Category > {...}
subcategories	<p>▼ [</p> <p><i>example: List ["subcategory 1"]</i></p> <p>The subcategories selected to categorize the asset</p>
department	<p>integer]</p> <p>Department > {...}</p>
data	<p>▼ {</p> <p> description: Telemetry and other arbitrary data of the asset</p>
categoryCustomData	<p>}</p> <p>▼ {</p> <p> description: The values for any category custom attributes related to the asset. The keys will be the ids of the custom asset field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options.</p>
maps	<p>}</p> <p>▼ [</p> <p><i>example: List ["abc", "def", "ghi"]</i></p> <p>The hierarchy of the asset's location</p>
lastHeard	<p>string]</p> <p>string <i>example: 2018-01-16T22:50:25.368Z</i></p> <p>Timestamp of when the asset last received a telemetry update/event</p>
lastModified	<p>string <i>example: 2018-01-16T22:50:25.368Z</i></p> <p>Timestamp of when the asset was last updated</p>
site	

```

static      AssetSite > {...}
            boolean
            example: true

            Identifies whether the asset is a static asset

staticLocation
telemetry   AssetStaticLocation > {...}
            {
              description:           The set of attributes that define
                                      the asset's telemetry data
            }

x           number
            example: 100

            The X coordinate of the static asset

y           number
            example: 100

            The Y coordinate of the static asset

icon       Icon > {...}
image      Image > {...}
}

```

```

NewAsset {
  serial*   string
            example: serial-123

            The unique serial number of the asset

  name     string
            example: name-123

            The name of the Asset

  description string
            example: Asset Description

            The description of the Asset

  status   string
            example: Active

            The status of the Asset - choose from either "Active" or
            "Inactive"

  tags     {
            example: List [ 1 ]

            The Ids of tags assigned to asset. These tags should not
            be assigned to another asset.

            integer]
  category* integer
            example: 100

            The category ID of the Asset; ensure the category belongs

```

to the department assigned to the asset

subcategories

▼ [

example: List ["subcategory"]

The subcategories of the Asset - options can be found under category selected for the asset.

string]

department*

integer

example: 200

The department ID of the Asset

customData

▼ {

description:

The values for any custom attributes for the asset. The keys will be the ids of the custom asset field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options.

}

categoryCustomData

▼ {

description:

The values for any category custom attributes for the asset. The keys will be the ids of the custom asset field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options.

}

site

AssetSite > {...}

staticLocation

AssetStaticLocation > {...}

static

boolean

example: true

True if the asset has been given a static location

x

number

example: 100

The X coordinate of asset's static location

y

number

example: 100

The Y coordinate of asset's static location

icon

```
icon      Icon > {...}
image     Image > {...}
}
```

UpdatedAsset ▾ {

id* **integer**
example: 11
The id of the asset

serial **string**
example: serial-123
The unique serial number of the asset

name **string**
example: name-123
The name of the Asset

description **string**
example: Asset Description
The description of the Asset

status **string**
example: Active
The status of the Asset - choose from either "Active" or "Inactive"

tags ▾ [
example: List [1]
The array of tag ids associated to the asset

category **integer**
example: 100
The category ID of the Asset

subcategories ▾ [
example: List ["subcategory 1"]
The subcategories of the Asset

department **integer**
example: 200
The department ID of the Asset

customData ▾ {
 description: The values for any custom attributes for the asset. The keys will be the ids of the custom asset field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If

some fields, provide any content. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options. Any properties from a Asset response object that have not been defined should be placed in this object.

```
categoryCustomData {  
  description:
```

The values for any category custom attributes for the asset. The keys will be the ids of the custom asset field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options.

```
site }  
AssetSite > {...}
```

```
static boolean  
example: true
```

Identifies whether the asset is a static asset

```
staticLocation AssetStaticLocation > {...}
```

```
x number  
example: 100
```

The X coordinate of the static asset

```
y number  
example: 100
```

The Y coordinate of the static asset

```
icon Icon > {...}
```

```
image Image > {...}
```

```
}
```



AssetSite ▼ {

description:

If a site is assigned to an asset, then only users that have access to the site will have access. The properties of site can be populated using the data from the map base hierarchy.

id*

string

example: abc

identification number of the site level assigned to asset

level*

string

example: FLOOR

the site level assigned to asset (CAMPUS, BUILDING, or FLOOR)

name*

string

example: campus -> building -> floor

name of site hierarchy assigned to asset

campus*

▼ [

example: List ["abc"]

id of campus assigned to asset

string]

building*

▼ [

example: List ["def"]

id of building assigned to asset

string]

floor*

▼ [

example: List ["ghi"]

id of floor assigned to asset

string]

}

AssetStaticLocation ∨ {



description: If a static location is assigned to an asset, then the asset will not receive any new location updates. The properties of static location can be populated using the data from the map base hierarchy.

id* **string**
example: abc

identification number of the static location level assigned to asset

level* **string**
example: FLOOR

the static location level assigned to asset (always has to be FLOOR)

name* **string**
example: campus -> building -> floor

name of static location hierarchy assigned to asset

campus* ∨ [
example: List ["abc"]

id of static campus assigned to asset

string]

building* ∨ [
example: List ["def"]

id of static building assigned to asset

string]

floor* ∨ [
example: List ["ghi"]

id of static floor assigned to asset

string]

}

FilteredAssets ∨ {



description: The following properties define the response of fetching assets using filters. Assets that fit the filters can be found as an array under key "data".

count **integer**
example: 1

data ∨ [**Asset** > {...}]

filter **AssetFilter** > {...}

}



AssetFilter ∨ {

description:

The following properties define the structure of a filter. To search for assets by serial number, add "serial" as the key as one of the fields. To search for assets by tag mac address, add "tags" as the key as one of the fields.

pagination*

∨ {

limit*

integer

example: 10

How many is to be retrieved at once. 100 is the max.

page*

integer

example: 0

Which page to start fetching from. Starts from page 0

}

fields*

∨ [∨ {

key*

string

example: serial

What to filter asset on.

type*

string

example: TEXT

What type the filter is. Use "TEXT"

operator*

string

example: contains

Use "contains" or "equals"

values*

∨ [**string**

example: serial123

The value to search assets on

]

}]

}

AssetLocationHistory ▾ {

description: The following properties define the location history of an asset

assetId **integer**
example: 1

tenantId **integer**
example: 1

startTime **integer**
example: 1524009600

endTime **integer**
example: 1524009600

series ▾ [LocationSeriesData > {...}]

}

LocationSeriesData ▾ {

data ▾ [LocationData > {...}]

}

LocationData ▾ {

description: The set of attributes that define the asset's location

x **integer**
example: 100

y **integer**
example: 100

floorId **string**
example: abc

zones ▾ [Zone > {...]]

chokePoints ▾ [ChokePoint > {...]]

hierarchy **string**
example: abc

lat **integer**
example: -999

lng **integer**
example: -999

timestamp **string**

}

Zone ▾ {

```
  color          string
                 example: #abc
  id             string
                 example: abc
  name          string
                 example: abc
}
```



ChokePoint ▾ {

```
  id            string
                 example: abc
  name         string
                 example: abc
}
```



AssetTelemetryHistory ▾ {

```
  description:   The following properties define the telemetry history of
                 an asset
  assetId       integer
                 example: 1
  tenantId     integer
                 example: 1
  startTime     integer
                 example: 1524009600
  endTime      integer
                 example: 1524009600
  type         string
                 example: TEMPERATURE
  units        string
                 example: Degree Celsius
  series       ▾ [TelemetrySeriesData > {...}]
}
```



TelemetrySeriesData ▾ {

```
  data          ▾ [TelemetryData > {...}]
}
```



TelemetryData ▾ {

```
  value          integer
                 example: 10
  timestamp      string
                 example: 2018-04-19T07:01:04.411Z
}
```



Error ▾ {

```
  code*         string
                 example: ERROR_CODE
  message*      string
                 example: error message
  title         string
                 example: error title
  httpStatus    integer
}
```



Unauthorized string

example: Unauthorized



Icon ▾ {

```
  description:   The properties of the icon chosen for the asset
  name*         string
  type*         string
}
```



Image ▾ {

```
  description:   The details of the image uploaded for the entity. The
                 attributes should be the same as the response body when
                 adding a new image. If you wish to assign the same image
                 to an entity, please re-upload the same image and assign
                 the new image details to the entity. Images are assets and
                 categories are currently supported
  name*         string
  id*           string
  type*         string
}
```





Category ▼ {

description:

The following properties define a Category. Any properties that appear in the response that have not been defined are considered custom fields.

id

integer

example: 1

The identification number of the category

name

string

example: cat1

The name of the category

description

string

example: description for category

The description of the category

subcategories

▼ [

example: List ["subcategory 1", "subcategory 2"]

The list of more distinctive categories to choose from while creating asset

string]

template

GetCategoryTemplate > {...}

icon

Icon > {...}

assetCount

integer

example: 10

The number of assets assigned to the category

department

Department > {...}

lastModified

string

example: 2018-01-16T22:50:25.368Z

The last time category was updated

image

Image > {...}

}



NewCategory ▾ {

```
  name*          string
                  example: cat1

                  The name of the category

  description    string
                  example: description for category

                  The description of the category

  subcategories  ▾ [
                  example: List [ "subcategory 1", "subcategory 2" ]

                  The list of more distinctive categories to choose from
                  while creating asset

                  string]

  customData     ▾ {
                  description:
                                The values for any custom
                                attributes for category

                  }

  template       AddCategoryTemplate > {...}

  icon           Icon > {...}

  department*   integer
                  example: 1

                  The department id that the category is assigned to

  image         Image > {...}

}
```



UpdatedCategory ▾ {

id* **integer**
example: 1

The identification number of the category

name* **string**
example: cat1

The name of the category

description* **string**
example: description for category

The description of the category

subcategories* ▾ [
example: List ["subcategory 1", "subcategory 2"]

The list of more distinctive categories to choose from while creating asset

customData* **string]**
▾ {
 description: The values for any custom attributes for category. The keys will be the ids of the custom category field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide timestamp string. If the field is a select field, provide the text of one of the select options. Any properties from a Category response object that have not been defined should be placed in this object.

template* **AddCategoryTemplate > {...}**

icon* **Icon > {...}**

image* **Image > {...}**

department* **integer**
example: 1

The department id that the category is assigned to

}

AddCategoryTemplate ▼ {



description:

The category custom fields created for the category. Any field objects added here will be added/updated. For example, if the template originally had custom fields "cf1" and "cf2", but the update request body now only has custom field "cf1", and new custom field "cf3", then custom field "cf2" will be deleted from the category and "cf3" will be added. Custom field "cf1" will remain in the category.

entityName

string

example: CATEGORY_CUSTOM

Which entity the field belongs to - should always be "CATEGORY_CUSTOM"

fields

▼ [

The list of custom fields that are to be added for the category

CategoryCustomFieldProperties > {...}]

}

GetCategoryTemplate ▼ {



description:

The category custom fields created for the category

id

integer

example: 1

The id of the category template

entityName

string

example: CATEGORY_CUSTOM

Which entity the field belongs to - should always be "CATEGORY_CUSTOM"

fields

▼ [

The list of custom fields that are to be added to the category

CategoryCustomField > {...}]

lastModified

string

example: 2018-01-16T22:50:25.368Z

Timestamp of the last time category template was modified

}

CategoryCustomFieldProperties ▼ {



label*

string

example: Custom Field

The label of the field viewed through user interface

key*

string

example: customField

The key of the field - usually a camelCase version of label. Key will eventually get converted to the id of the field

entityName*

string

example: CATEGORY_CUSTOM

Which entity the field belongs to (ASSETS, CATEGORIES, DEPARTMENTS, TAGS, CATEGORY_CUSTOM) CATEGORY_CUSTOM is for category custom fields. To add a category custom field, follow the NewCategory details.

description*

string

example: This is a custom field

The description of the field; used as helper text when viewed on the user interface

templateLabel*

string

example: Category Custom Field

How the field appears in a CSV template; Usually done by concatenating the entityName (Tag, Category, Department, Asset) with the field label. If entity is CATEGORY_CUSTOM, concatenate the category name with the field label (Cat1 Custom Field). The CATEGORY_CUSTOM entity is for category custom fields only. To add a category custom field, follow the NewCategory details.

type*

string

example: SELECT

The type of the field (TEXT, SELECT, TEL, DATE, PARAGRAPH)

options

▼ [

example: List ["option 1", "option 2"]

The array of string selection values that the field supports; only required when type of field is "SELECT"

string]

multiple

boolean

example: false

If multiple values can be selected for the field; only required when type of field is "SELECT"

validation*

string

example: ANY

What type of validation is needed for field values (ANY, TEL)

editable*

boolean

example: true

If the value of the field can be edited; always true

general*

boolean

example: true

```

        If the field should appear in the general tab; always true
    required*      boolean
                   example: false

        If the field value is required
    }

```

CategoryCustomField ∨ { ←

```

    id*            integer
                   example: 100

        The identification number of the field

    key*           string
                   example: 100

        The key of the field. Key will eventually get converted to
        the id of the field

    label*        string
                   example: custom field

        The label of field viewed through user interface

    entityName*   string
                   example: CATEGORY_CUSTOM

        Which entity the field belongs to

    description*  string
                   example: description of field

        The description of the field

    templateLabel* string
                   example: category custom field

        How the field appears in a CSV template

    type*         string
                   example: SELECT

        The type of the field (TEXT, SELECT, TEL, DATE, PARAGRAPH)

    options*      ∨ [
                   example: List [ "option 1", "option 2" ]

        The selection values that the field supports

    collection*   string
                   example: null

        What collection the field uses to populate data

    validation*   string
                   example: ANY

        What type of validation is needed for field values (ANY,
        TEL)

    isDefault*    boolean

```

	<i>example: true</i>	
		If the field is a default field
required*	boolean <i>example: true</i>	
		If a value is required for the field
templateUnique*	boolean <i>example: true</i>	
		If the value for the field must be unique during CSV import
templateHidden*	boolean <i>example: false</i>	
		If the field appears in the CSV template
hidden*	boolean <i>example: false</i>	
		If the field appears in the user interface
editable*	boolean <i>example: true</i>	
		If the value of the field can be edited
general*	boolean <i>example: true</i>	
		If the field should appear in the general tab
multiple*	boolean <i>example: false</i>	
		If multiple values can be selected for the field
defaultSelectAll*	boolean <i>example: false</i>	
		If all values for the field should be selected by default
clickableLink*	boolean <i>example: false</i>	
		If the value of the field can be clicked
min*	integer <i>example: 1</i>	
		The minimum value that can be entered for the field
max*	integer <i>example: 100</i>	
		The maximum value that can be entered for the field
filterable*	boolean <i>example: true</i>	
		If the field can be used to filter entities
sortable*	boolean <i>example: true</i>	
		If the field can be used to sort entities

```
}
```

Department ▾ {



description:

The following properties define a Department. Any properties that appear in the response that have not been defined are considered custom fields.

id

integer

example: 1

The identification number of the department

name

string

example: dept1

The name of the department

assetCount

integer

example: 10

The number of assets assigned to the department

categories

▾ [

The list of categories assigned to the department

Category > {...}]

description

string

example: description for department

The description of the department

lastModified

string

example: 2018-01-16T22:50:25.368Z

Timestamp of the last time the department was modified

```
}
```



NewDepartment ▾ {

name* **string**
example: dept1

The name of the department

customData ▾ {
 description:

The values for any custom attributes for the department. The keys will be the ids of the custom department field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options.

description **string**
 example: description for department

The description of the department

}



UpdatedDepartment ▼ {

id* **integer**
example: 1

The identification number of the department

name* **string**
example: dept1

The name of the department

customData* ▼ {
 description:

The values for any custom attributes for the department. The keys will be the ids of the custom department field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options. Any properties from a Department response object that have not been defined should be placed in this object.

description **string**
example: description for department

The description of the department

}

FieldTemplate ▾ {



```
  id integer
      example: 100

      The field template identification number. Each custom
      field is assigned a template.

  entityName string
      example: TAGS

      Which entity the field belongs to (ASSETS, CATEGORIES,
      DEPARTMENTS, TAGS, CATEGORY_CUSTOM). CATEGORY_CUSTOM is
      for category custom fields. To add a category custom
      field, follow the NewCategory details.

  lastModified string
      example: 2017-03-17T04:43:34.156Z

      Timestamp of the last time the field was updated

  fields
      ▾ [

      All the fields belonging to the entity

      Field > {...}]

}
```

Field ▾ {



```
  id* integer
      example: 100

      The field identification number

  key* string
      example: 100

      The key of the field. Key will eventually get converted to
      the id of the field.

  label* string
      example: custom field

      The label of the field viewed through user interface

  entityName* string
      example: TAGS

      Which entity the field belongs to

  description* string
      example: description of field

      The description of the field

  templateLabel* string
      example: Tag custom field

      How the field appears in a CSV template

  type* string
      example: SELECT
```


The type of the field (TEXT, SELECT, TEL, DATE, PARAGRAPH)

options* > [...]

collection* **string**
example: null

What collection the field uses to populate data

validation* **string**
example: ANY

What type of validation is needed for field values (ANY, TEL)

isDefault* **boolean**
example: true

If the field is a default field

required* **boolean**
example: true

If a value is required for the field

templateUnique* **boolean**
example: true

If the value for the field must be unique during CSV import

templateHidden* **boolean**
example: false

If the field appears in the CSV template

hidden* **boolean**
example: false

If the field appears in the user interface

editable* **boolean**
example: true

If the value of the field can be edited

general* **boolean**
example: true

If the field should appear in the general tab

multiple* **boolean**
example: false

If multiple values can be selected for the field

defaultSelectAll* **boolean**
example: false

If all values for the field should be selected by default

clickableLink* **boolean**
example: false

If the value of the field can be clicked

min* **integer**
example: 1

```

    max*                integer
                        example: 100

                        The minimum value that can be entered for the field

    filterable*         boolean
                        example: true

                        The maximum value that can be entered for the field

                        If the field can be used to filter entities

    sortable*           boolean
                        example: true

                        If the field can be used to sort entities

}

```

```

NewField ▾ {
  field*           FieldProperties > {...}
}

```

```

FieldProperties ▾ {
  description:       The properties of the field

  label*            string
                    example: Custom Field

                    The label of field viewed through user interface

  key*              string
                    example: customField

                    The key of the field - usually a camelCase version of
                    label. Key will eventually get converted to the id of the
                    field.

  entityName*       string
                    example: TAGS

                    Which entity the field belongs to (ASSETS, CATEGORIES,
                    DEPARTMENTS, TAGS, CATEGORY_CUSTOM) CATEGORY_CUSTOM is for
                    category custom fields. To add a category custom field,
                    follow the NewCategory details.

  description*      string
                    example: This is a custom field

                    The description of the field; used as helper text when
                    viewed on the user interface

  templateLabel*    string
                    example: Tag Custom Field

                    How the field appears in a CSV template; labelled by
                    concatenating the entityName (Tag, Category, Department,
                    Asset) with the field label. If entity is CATEGORY_CUSTOM,
                    concatenated with the department name and not concatenated with the

```

concatenate the department name and category name with the field label (Dept1 Cat1 Custom Field). CATEGORY_CUSTOM is for category custom fields. To add a category custom field, follow the NewCategory details.

type*

string

example: SELECT

The type of the field (TEXT, SELECT, TEL, DATE, PARAGRAPH)

options

▼ [

example: List ["option 1", "option 2"]

An array of string selection values that the field supports; only required when type of field is "SELECT"

multiple

string]

boolean

example: false

If multiple values can be selected for the field; only required when type of field is "SELECT"

validation*

string

example: ANY

What type of validation is needed for field values (ANY, TEL)

editable*

boolean

example: true

If the value of the field can be edited; always true

general*

boolean

example: true

If the field should appear in the general tab; always true

required*

boolean

example: false

If the field value is required

}

UpdatedField ▼ {



```
id*           integer
              example: 100

              The field identification number

label         string
              example: Custom Field

              The label of field viewed through user interface

description   string
              example: This is a custom field

              The description of the field; used as helper text when
              viewed on the user interface

templateLabel string
              example: Tag Custom Field

              How the field appears in a CSV template; labelled by
              concatenating the entityName (Tag, Category, Department,
              Asset) with the field label. If entity is CATEGORY_CUSTOM,
              concatenate the department name and category name with the
              field label (Dept1 Cat1 Custom Field). CATEGORY_CUSTOM is
              for category custom fields. To add a category custom
              field, follow the NewCategory details.

type          string
              example: SELECT

              The type of the field (TEXT, SELECT, TEL, DATE, PARAGRAPH)

options       ▼ [
              example: List [ "option 1", "option 2" ]

              An array of string selection values that the field
              supports; only required when type of field is "SELECT"

multiple      string]
              boolean
              example: false

              If multiple values can be selected for the field; only
              required when type of field is "SELECT"

required      boolean
              example: false

              If the field value is required

}
```



Tag

```
description:
  The following properties define a Tag. Any properties that
  appear in the response that have not been defined are
  considered custom fields.

id
  integer
  example: 1
  The identification number of the tag

serial
  string
  example: 11:22:33:44:55:66
  The unique mac address of the tag

name
  string
  example: Tag 1
  The name of the tag

data
   [
  example: List [ OrderedMap {} ]
  The telemetry data of tag

   {...}]

assetId
  integer
  example: 1
  The asset id that the tag is associated to

type
  string
  example: RFID
  The ype of the tag (RFID, BLE, WIFI)

location
   {
    description:
      The location data of tag
  }

lastHeard
  string
  example: 2018-01-16T22:50:25.368Z
  Timestamp of the last time the tag was heard

}
```



NewTag ▾ {

serial* **string**
example: 11:22:33:44:55:66

The unique mac address of the tag

name **string**
example: Tag 1

The name of the tag

customData

▾ {
 description:

The values for any custom attributes for the tag. The keys will be the ids of the custom tag field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options.

type **string**
example: RFID

The Type of the tag (RFID, BLE, WIFI)

}



UpdatedTag ∨ {

id* **integer**
example: 1

The identification number of the tag

serial **string**
example: 11:22:33:44:55:66

The unique mac address of the tag

name **string**
example: Tag 1

The name of the tag

customData ∨ {
 description:

The values for any custom attributes for tag. The keys will be the ids of the custom tag field, while the values will be dependent on the type of the custom field. If the field is a text field, provide any text. If the field is a date field, provide a timestamp string. If the field is a select field, provide the text of one of the select options. If you wish to update any custom data, please provide all new, old, and update key-value pairs here. The data included here will overwrite all custom data.

type **string**
example: RFID

The type of the tag (RFID, BLE, WIFI)

}