Customer

EASA

The European Authority for Aviation Safety

Document

E2 API Guide

About

This document provides a guide of use of the external API of ECCAIRS 2.

Author	Bilbomática
Date	31/07/2022
Version	4.19
Description of the version	fourth complete version.

Edificio Miribilla Santiago Compostela, 12, 4ª A
48003 Bilbao – Spain
Tel: +34 944 276 308 Fax: +34 944 271 547
Bilbao | Madrid | Pamplona

bilbomática

Table of Contents

1 Introduction

The purpose of this document is to guide on the use of the external API provided by E2.

In the sections below you will find samples of how each of the methods available shall be used. In order to execute your tests for the first time you must follow the steps under section 3 in the order provided. To run your tests, you can use third party tools like Postman or use Swagger itself.

Please note that for readability purposes, the body of the API methods are Microsoft Word objects and are by default not fully shown. Please double-click on these objects to get the full list of parameters.

Also, since the release performed on the 9th November, MFA is enabled. Thus, in order to execute API methods your IP needs to be whitelisted beforehand, otherwise you won't be able to generate a security token. **Please communicate with EASA so that these IPs can be whitelisted centrally by an administrator**.

2 What's new in this version

- A new 'User Management' section has been included.
    - New /usermgmt/users method (See 5.24), which returns the list of all users configured for the logged-in user's authority.

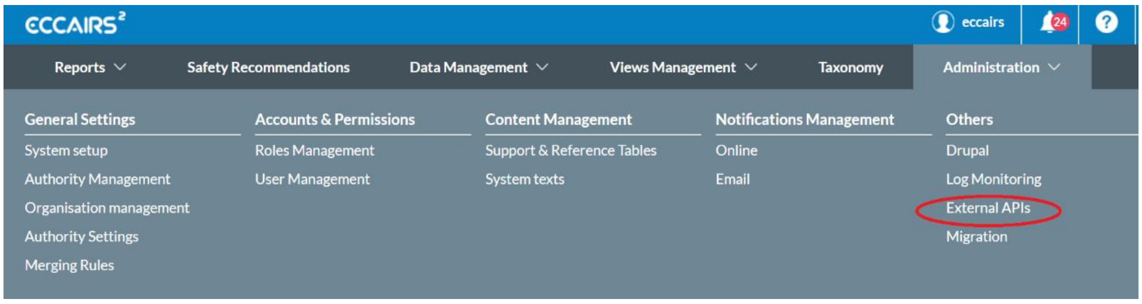
3 First time use

Before testing the API for the first time, you need to have users and grant them specific roles in E2 in order to query information and execute actions on saved data.

For that, please follow the steps under this section in the order provided.

3.1 Step 0: Entry point for ECCAIRS 2 API

The entry point to access the external API using Swagger is now also available through the E2 Web App, using the Administration menu entry as displayed in the image below:



Likewise, if you are already logged in E2 Web App, you can access the external API, typing the following URL:

- PROD: https://e2.aviationreporting.eu/guiding-page
- UAT: https://e2.uat-aviationreporting.eu/guiding-page
- Sandbox: https://e2.sandbox.aviationreporting.eu/guiding-page

You will see the following sections:

- **Login**: method to log into the system.
- **Taxonomy Management Section**: set of methods to interact with taxonomies and value lists.
- **Occurrence to Report Section**: group of methods focused on managing and consulting Reports and/or Occurrences.
- **Import from files**: group of methods to upload documents using files (e5x).

Note: in sections below, please, replace the {BASE_URL} by the value https://api.aviationreporting.eu

3.2 Step 1: Login as an elevated user

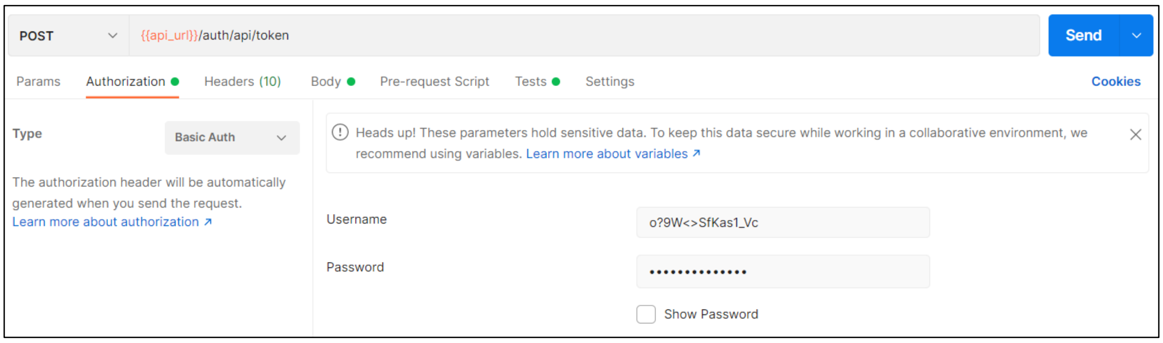
This step aims at creating user accounts and grant them privileges to execute actions on the API.

- Section: Login Section.
- Method URL: {BASE_URL}/auth/api/token
- Method Type: POST
- Authorization: Basic Auth
  - ✓ username: o?9W<>SfKas1_Vc
  - ✓ password: y]vfbuxrca#%sCksneQD/S:"MF^azq

- Body parameters:
  - ✓ Username: << national authority user with system administrator role >>
  - ✓ password: << password for the user with the username above>>
  - ✓ grant_type: “password”
- Response: the token to grant permissions to access and operate with ECCAIRS2 API

Example using **Postman**:

- Authorization:



POST {{api_url}}/auth/api/token

Params | Authorization | Headers (10) | Body | Pre-request Script | Tests | Settings | Cookies

Type: Basic Auth

The authorization header will be automatically generated when you send the request.
Learn more about authorization ↗

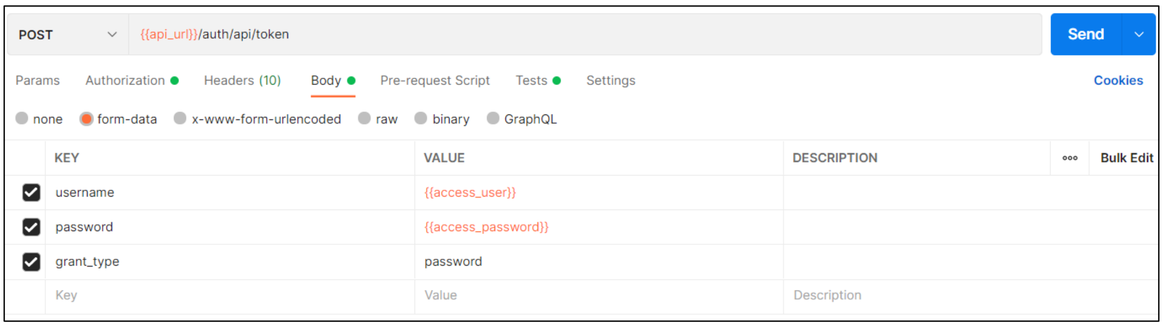
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Username: o?9W<>SfKas1_Vc

Password: •••••••••••••••

Show Password

- Body:



POST {{api_url}}/auth/api/token

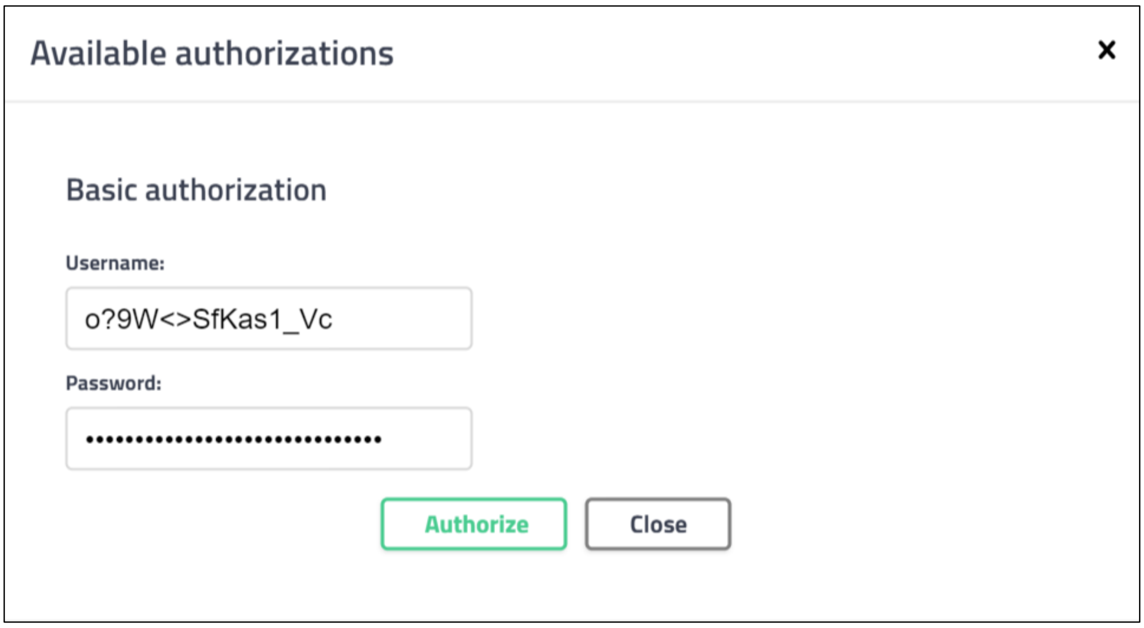
Params | Authorization | Headers (10) | Body | Pre-request Script | Tests | Settings | Cookies

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL

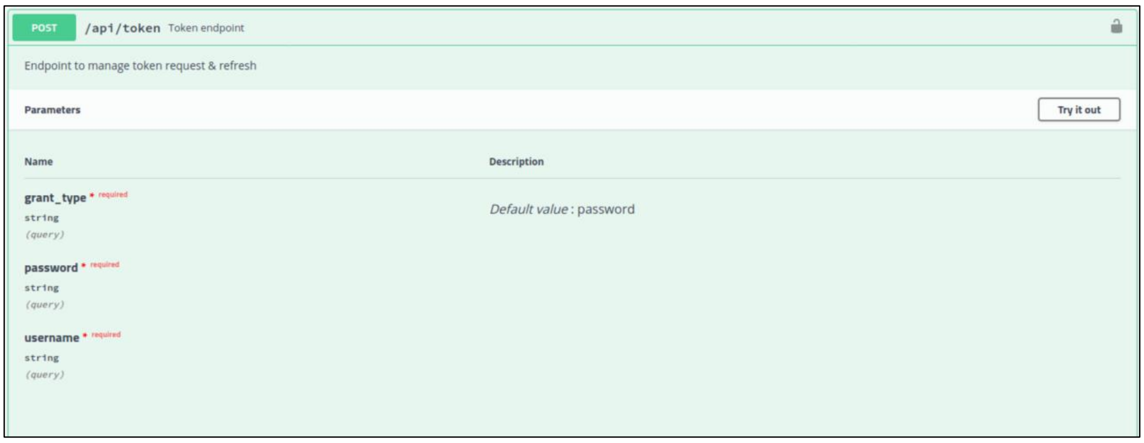
	KEY	VALUE	DESCRIPTION
✓	username	{{access_user}}	
✓	password	{{access_password}}	
✓	grant_type	password	
	Key	Value	Description

Example using **Swagger**

- Authorization:



- ✓ Parameters:



Endpoint to manage token request & refresh

Parameters

Name	Description
grant_type * required string (query)	Default value : password
password * required string (query)	
username * required string (query)	

- ✓ Response:

{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NzkyNjUzODE-sInVzZXIiOnsiaWQiOjEsInVzZXJuYW1lIjoiZWNjYWlycyIsImxhbmd1YWdlIjp7ImlkIjozLCJuYW1lI-joiRW5nbGlzaCIsImNvZGUiOiJlbiJ9LCJhdXRob3JpdHkiOnsiaWQiOjE-sIm5hbWUiOiJFQVNBIn0sInJvbGVzIjpbeyJuYW1lIjoiRUNDQUlSUyBTeXN0ZW0gQWR-taW5pc3RyYXRvciIsInBlcm1pc3Npb25zIjpbIkNyZWF0ZVF1ZXJ5IiwiUmVhZFF1ZXJ5Ii-wiRWRpdFF1ZXJ5IiwiRGVsZXRlUXVlcnkiLCJFeGVjdXRlUXVlcnkiLCJTaGFyZVF1ZXJ5Ii-wiQ3JlYXRlQmF0Y2hPcGVyYXRpb24iLCJSZWFkQmF0Y2hPcGVyYXRpb24iLCJFZGl0QmF0Y2hPcG
VyYXRpb24iLCJEZWxldGVCYXRjaE9wZXJhdGlvbiIsIkV4ZWN1dGVCYXR-jaE9wZXJhdGlvbiIsIkNyZWF0ZVZpZXciLCJSZWFkVmlldyIsIkVkaXRWaWV3Ii-wiRGVsZXRlVmlldyIsIkNyZWF0ZURhc2hib2FyZERhbGUiLCJSZWFkRGFzaGJvYXJkVGlsZSIsIkV-

3.3 Step 2: Token setting up

The token granted in the first step should be added to any subsequent call to the API as it contains information on the user executing each action. Thus, it helps to prevent any unauthorized access to information.

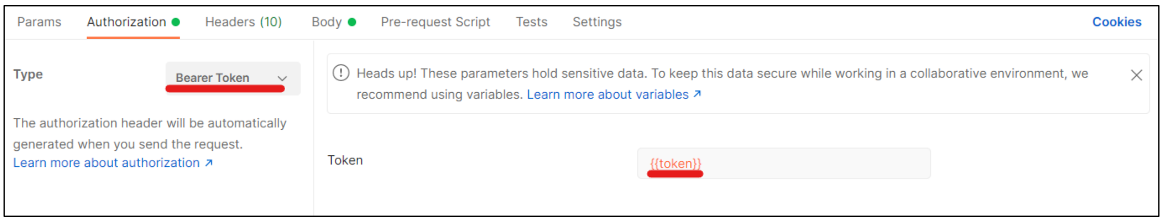
The token obtained in step 1 should be added as a Bearer Token type as follows:

- Authorization:
  - ✓ Type: Bearer Token
  - ✓ Token (example from step 1):

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NzkyNjUzODEsInVzZXIiOnsiaWQiOjEsInVzZXJuYW1lIjoiZWNjYWlycyIsI
mxhbmd1YWdlIjp7ImlkIjozLCJuYW1lIjoiRW5nbGlzaCIsImNvZGUiOiJlbiJ9LCJhdXRob3JpdHkiOnsiaWQiOjEsIm5hbWUiOiJFQV
NBIn0sInJvbGVzIjpbeyJuYW1lIjoiRUNDQUlSUyBTeXN0ZW0gQWRtaW5pc3RyYXRvciIsInBlcm1pc3Npb25zIjpbIkNyZWF0ZVF1
ZXJ5IiwiUmVhZFF1ZXJ5IiwiRWRpdFF1ZXJ5IiwiRGVsZXRlUXVlcnkiLCJFeGVjdXRlUXVlcnkiLCJTaGFyZVF1ZXJ5IiwiQ3JlYXRlQm
F0Y2hPcGVyYXRpb24iLCJSZWFkQmF0Y2hPcGVyYXRpb24iLCJFZGl0QmF0Y2hPcGVyYXRpb24iLCJEZWxldGVCYXRjaE9wZXJh
dGlvbiIsIkV4ZWN1dGVCYXRjaE9wZXJhdGlvbiIsIkNyZWF0ZVZpZXciLCJSZWFkVmlldyIsIkVkaXRWaWV3IiwiRGVsZXRlVmlldyIs
IkNyZWF0ZURhc2hib2FyZFRpbGUiLCJSZWFkRGFzaGJvYXJkVGlsZSIsIkVkaXREYXNoYm9hcmRUaWxlIiwiRGVsZXRlRGFzaGJvY
XJkVGlsZSIsIkNyZWF0ZVJvbGUiLCJSZWFkUm9sZSIsIkVkaXRSb2xlIiwiRGVsZXRlUm9sZSIsIk1hbmFnZUVDQ0FJUlNTZWN1cml
0eUFkbWluaXN0cmF0b3IiLCJDcmVhdGVVc2VyIiwiUmVhZFVzZXIiLCJFZGl0VXNlciIsIkRlbGV0ZVVzZXIiLCJEZWxldGVNZXJna
W5nUHJpbmNpcGxlIiwiQ3JlYXRlTm90aWZpY2F0aW9uIiwiUmVhZE5vdGlmaWNhdGlvbiIsIkVkaXROb3RpZmljYXRpb24iLCJEEZ
WxldGVOb3RpZmljYXRpb24iLCJBY2Nlc3NlZWxwIiwiQ3JlYXRlQXV0aG9yaXR5IiwiUmVhZEF1dGhvcml0eSIsIkVkaXRBdXRob3J
pdHkiLCJEZWxldGVBdXRob3JpdHkiLCJSZWFkRWNjYWlyczJTZXR0aW5ncyIsIkVkaXRFY2NhaXJzMlNldHRpbmdzIiwiQ3JlYXRlT
GFuZ3VhZ2UiLCJSZWFkTGFuZ3VhZ2UiLCJFZGl0TGFuZ3VhZ2UiLCJEZWxldGVMYW5ndWFnZSIsIkNyZWF0ZUNvdW50cnkiLC
JSZWFkQ291bnRyeSIsIkVkaXRDb3VudHJ5IiwiRGVsZXRlQ291bnRyeSJdfSx7Im5hbWUiOiJFQ0NBSVJTIFRheG9ub215IEFkbWlu
aXN0cmF0b3IiLCJwZXJtaXNzaW9ucyI6WyJDcmVhdGVRdWVyeSIsIlUlYWRRdWVyeSIsIkVkaXRRdWVyeSIsIkRlbGV0ZVF1ZXJ5Ii
wiRXhlY3V0ZVF1ZXJ5IiwiU2hhcmVRdWVyeSIsIkNyZWF0ZVRheG9ub215UHJvamVjdCIsIlUlYWRUYXhvbm9teVByb2plY3QiLCJ
FZGl0VGF4b25vbXlQcm9qZWN0IiwiRGVsZXRlVGF4b25vbXlQcm9qZWN0IiwiUmVsZWFzZVRheG9ub215UHJvamVjdCIsIkNvb
XBhcmVUYXhvbm9teVByb2plY3QiLCJFeHBvcnRUYXhvbm9teVByb2plY3QiLCJEZWlkZW50aWZ5VGF4b25vbXlQcm9qZWN0Ii
wiQ3JlYXRlTm90aWZpY2F0aW9uIiwiUmVhZE5vdGlmaWNhdGlvbiIsIkVkaXROb3RpZmljYXRpb24iLCJEZWxldGVOb3RpZmljY
XRpb24iLCJBY2Nlc3NlZWxwIiwiQ3JlYXRlTWFzdGVyVGFibGUiLCJSZWFkTWFzdGVyVGFibGUiLCJFZGl0TWFzdGVyVGFibGUiL
CJEZWxldGVNYXN0ZXJUYWJsZSJdfV19LCJ1c2VyX25hbWUiOiJlY2NhaXJzIiwiYXV0aG9yaXRpZXMiOlsiRUNDQUlSUyBUYXhv
bm9teSBBZG1pbmlzdHJhdG9yIiwiRUNDQUlSUyBTeXN0ZW0gQWRtaW5pc3RyYXRvciJdLCJqdGkiOiIyNzEwOWE0NC0yNDdiL
TQwOWUtOTlmYi1jN2ZiZDZiMTExODQiLCJjbGllbnRfaWQiOiJvPzlXPD5TZkthczFfVmMifQ.VQ5rFHjXGqm4KnAVd7DLGXIPzD
gHBHnr5HUtmaB0CyguGGf-gTpU-
S2MXiLWrGbXV8eeflBwGh_DDJkvSeA_ITk3r4Tjtfk_Y2xN4MdaC4WezhjXT58aLMgDzb1hCi6PJgefTQWbrlRbY-
vEkQ6UbDuUENy7zTDZc09WcjjGD0NtMge9MNJTGowBtUzhpUGF7U2Y_cxTzBSfohusEMrjdQRDYmYvuGenZUTfXVMdoUrytR
dwJPaaDYZUWij41ECWUWK3MVUzfOoWzmluPeKIv3j7JsFbsK4F0-9i9a_wGjG7FCkzuKnTyMVJ8x-
jtYfh7HHQFQUc4MSPbF2_gkvPuw
```

Example using Postman:

- Authorization: the picture below shows WHERE the type of authorization (bearer token) and the token must be entered.



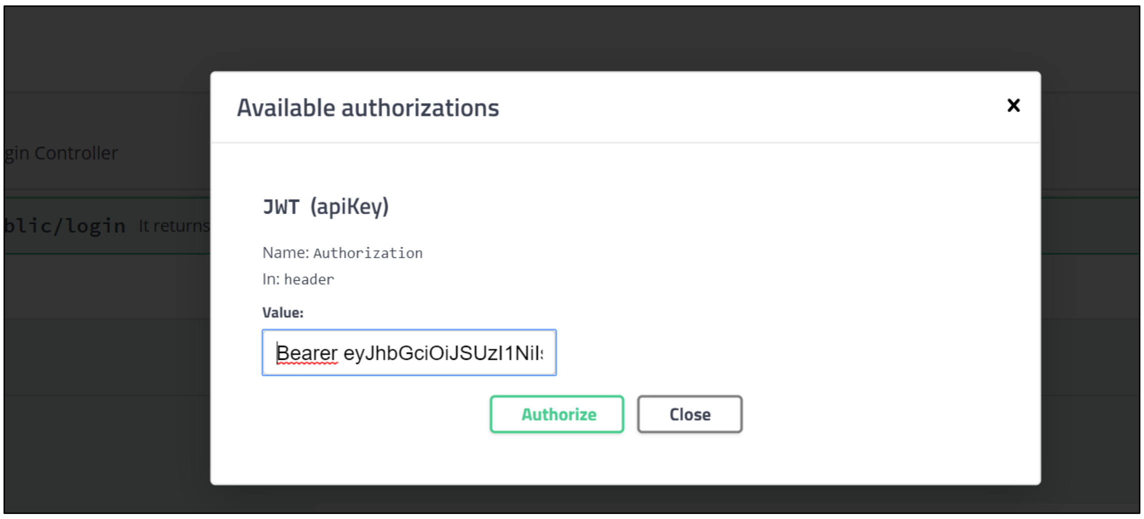
Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Token: {{token}}

Example using Swagger:



Available authorizations

JWT (apiKey)

Name: Authorization

In: header

Value: Bearer eyJhbGciOiJSUzI1Nil

3.4 Step 3: Logging in with an existing user

Since a new user has been created in ECCAIRS2, now this user can log in the API. Step 0 should be repeated with the new username and password (right now “ECCAIRS2” is used as default password until the final registration step is done). So, to log in with the user recently created:

- Section: Login Section
- Method URL: {BASE_URL}/auth/api/token
- Method Type: POST
- Body:

```
{
   "grant_type": "password",
   "password": "ECCAIRS2",
   "username": "sample_user"
}
```

- Response:

```
{⊟
   "access_token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1ODA0ODAzNDgsInVzZXIiOn-
siaWQiOjQsInVzZXJuYW1lIjoic2FtcGxlX3VzZXIi-
LCJsYW5ndWFnZSI6eyJpZCI6MywibmFtZSI6IkVuZ2xpc2giLCJjb2RlI-
joiZW4ifSwiYXV0aG9yaXR5Ijp7ImlkIjoxLCJuYW1lIjoiRUFTQSJ9LCJyb2xlcyI6W3si-
bmFtZSI6Ik9jdXJyZW5jZSBPZmZpY2VyIiwicGVybWlzc2lvbnMiOlsiQ3JlYXRlUmVwb3J0Ii-
wiRWRpdFJlcG9ydCIsIlJlYWRS-
```

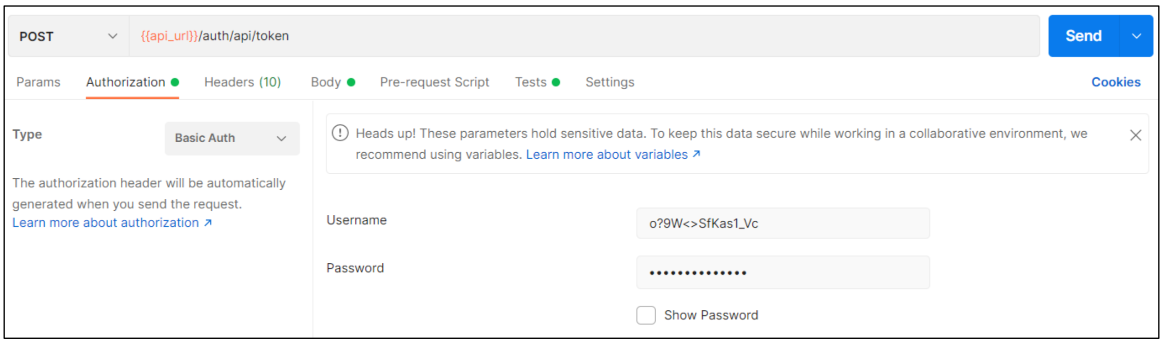
3.5 Token Refresh

To refresh an expired token, we have to use the refresh token

1. Section: Login Section.
2. Method URL: {BASE_URL}/auth/api/token
3. Method Type: POST
4. Authorization: Basic Auth
    a. username: o?9W<>SfKas1_Vc
    b. password: y]vfbuxrca#%sCksneQD/S:"MF^azq
5. Body parameters:
    a. grant_type: “refresh_token”
    b. refresh_token: << our refresh token >>
6. Response: the new token to grant permissions to access and operate with ECCAIRS2 API

Example using **Postman**:

7. Authorization:


POST {{api_url}}/auth/api/token

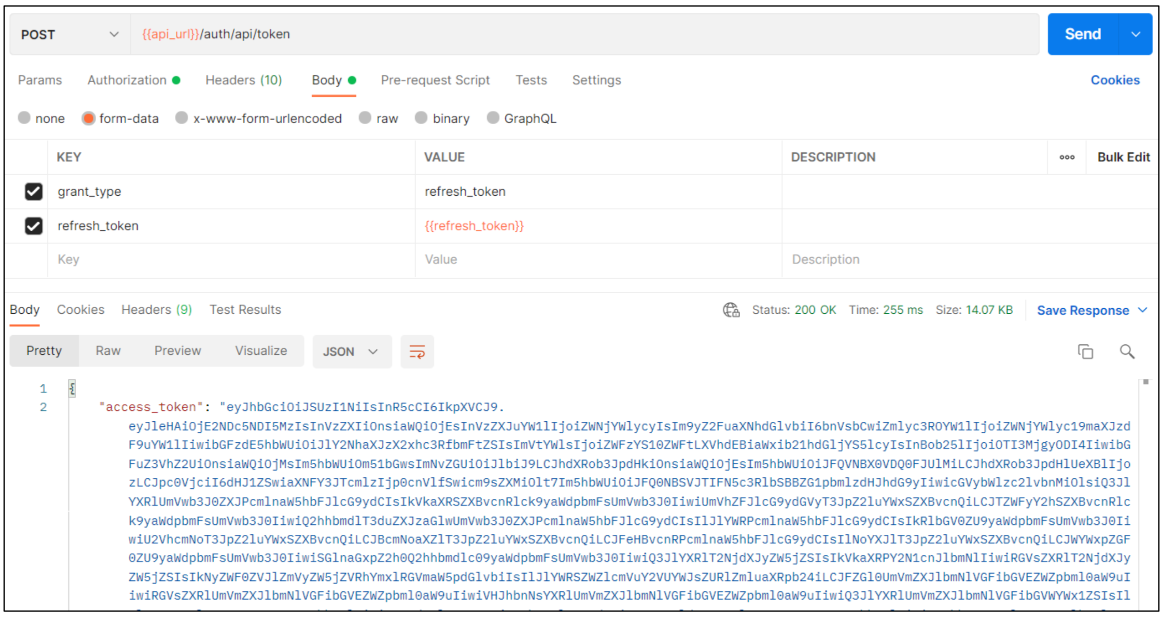
Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Username: o?9W<>SfKas1_Vc

Password: ••••••••••••••



{{api_url}}/auth/api/token

	KEY	VALUE	DESCRIPTION
✓	grant_type	refresh_token	
✓	refresh_token	{{refresh_token}}	
	Key	Value	Description

Status: 200 OK Time: 255 ms Size: 14.07 KB

```
{
    "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJleHAiOjE2NDc5NDI5MzIsInVzZXIiOnsiaWQiOjEsInVzZXJuYW1lIjoiZWNjYWlycyIsIm9yZ2FuaXNhdGlvbiI6bnVsbCwiZmlyc3ROYW1lIjoiZWNjYWlyc19maXJzd
        F9uYW1lIiwibGFzdE5hbWUiOiJlY2NhaXJzX2xhc3RfbmFtZSIsImVtYWlsIjoiZWFzYS10ZWFtLXVhdEBiaWxib21hdGljYS5lcyIsInBob25lIjoiOTI3MjgyODI4IiwibG
        FuZ3VhZ2UiOnsiaWQiOjMsIm5hbWUiOm51bGwsImNvZGUiOiJlbiJ9LCJhdXRob3JpdHkiOnsiaWQiOjEsIm5hbWUiOiJFQVNBX0VDQ0FJUlMiLCJhdXRob3JpdHlUeXBlIjo
        zLCJpc0VjciI6dHJ1ZSwiaXNFY3JTcmlzIjp0cnVlfSwicm9sZXMiOlt7Im5hbWUiOiJFQ0NBSVJTIFN5c3RlbSBBZG1pbmlzdHJhdG9yIiwicGVybWlzc2lvbnMiOlsiQ3Jl
        YXRlUmVwb3J0ZXJPcmlnaW5hbFJlcG9ydCIsIkVkaXRSZXBvcnRlck9yaWdpbmFsUmVwb3J0IiwiUmVhZFJlcG9ydGVyT3JpZ2luYWxSZXBvcnQiLCJTZWFyY2hSZXBvcnRlc
        k9yaWdpbmFsUmVwb3J0IiwiQ2hhbmdlT3duZXJzaGlwUmVwb3J0ZXJPcmlnaW5hbFJlcG9ydCIsIlJlYWRPcmlnaW5hbFJlcG9ydCIsIkRlbGV0ZU9yaWdpbmFsUmVwb3J0Ii
        wiU2VhcmNoT3JpZ2luYWxSZXBvcnQiLCJBcmNoaXZlT3JpZ2luYWxSZXBvcnQiLCJFeHBvcnRPcmlnaW5hbFJlcG9ydCIsIlNoYXJlT3JpZ2luYWxSZXBvcnQiLCJWYWxpZGF
        0ZU9yaWdpbmFsUmVwb3J0IiwiSGlnaGxpZ2h0Q2hhbmdlc09yaWdpbmFsUmVwb3J0IiwiQ3JlYXRlT2NjdXJyZW5jZSIsIkVkaXRPY2N1cnJlbmNlIiwiRGVsZXRlT2NjdXJy
        ZW5jZSIsIkNyZWF0ZVJlZmVyZW5jZVRhYmxlRGVmaW5pdGlvbiIsIlJlYWRSZWZlcmVuY2VUYWJsZURlZmluaXRpb24iLCJFZGl0UmVmZXJlbmNlVGFibGVEZWZpbml0aW9uI
        iwiRGVsZXRlUmVmZXJlbmNlVGFibGVEZWZpbml0aW9uIiwiVHJhbnNsYXRlUmVmZXJlbmNlVGFibGVEZWZpbml0aW9uIiwiQ3JlYXRlUmVmZXJlbmNlVGFibGVWYWx1ZSIsIl
```

4 Examples of use

Once you have followed steps provided in section 0 you can now test all the methods available in the API. In this section we provide an example for the testing of how occurrences can be managed.

4.1 Creating an occurrence

When creating a report, the type has to be specified. The input parameters are the following:

- reportType, with options:
  - ✓ REPORT
  - ✓ VALIDATED
  - ✓ OCCURRENCE
- taxonomyCodes - JSON with the report information. Please refer to section 9.3 for more information on the JSON structure
- reportingEntityId – who reports
- status
  - ✓ By default, it will be sent as Sent/Open.
  - ✓ Optionally, if specified, it can be saved as Draft.

Example

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/create
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Body:

```
{
  "type":"REPORT",
  "taxonomyCodes":{
    "24":{
      "ID":"#id_number#",
      "ENTITIES":{
        "1":[
          {
            "ID":"#id_number#",
```

- Response:

```
{
    "data": {
        "e2Id": "OR-0000000000044484",
        "status": "SENT",
        "version": "0.3"
    },
    "errorDetails": "",
```

4.2 Updating an occurrence

The input parameters are the following:

- e2id – ECCAIRS2 unique identifier
- taxonomyCodes - JSON with the report information. Please refer to section 9.3 for more information on the JSON structure
- versionType, which determines how to save the report, with options
  - ✓ DRAFT (for OR/VR/OC)
  - ✓ MINOR (for OR/VR/OC)
  - ✓ MAJOR (for OC)

Example

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/edit
- Method Type: PUT
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Body:

```
{
   "e2Id": "OR-0000000000044484",
   "versionType": "MINOR",
   "taxonomyCodes": {
      "24": {
            "ATTRIBUTES": {
               "440": ["new value"],
               "451": [98]
            },
```

- Response:

```
{
    "data": {
        "e2Id": "OR-0000000000044484",
        "status": "SENT",
        "version": "0.4"
    },
    "errorDetails": "",
    "returnCode": 1
```

4.3 Updating an occurrence - Deleting entities

To delete an entity of an Occurrence, the input parameters are the following:

- e2id – ECCAIRS2 unique identifier
- taxonomyCodes - JSON with the report information. Please refer to section 9.3 for more information on the JSON structure
- versionType, which determines how to save the report, with options
  - ✓ DRAFT (for OR/VR/OC)
  - ✓ MINOR (for OR/VR/OC)
  - ✓ MAJOR (for OC)

Example

- Section: Original Reports to Occurrences Section.

- Method URL: {BASE_URL}/occurrences/edit
- Method Type: PUT
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Body:

```
{
    "e2Id": "OC-0000000003292523",
    "versionType": "MINOR",
    "taxonomyCodes": {
       "24": {
             "ENTITIES": {
                "4": [
```

- Response:

```
{
    "data": {
        "e2Id": "OC-0000000003292523",
        "status": "OPEN",
```

4.4 Getting an occurrence URL

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/get-URL/{e2Id}
- Method Type: GET
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters: e2id (unique identifier of the occurrence)
- Body: Empty
- Response:

```
{
 "data": {
  "e2Id": "OR-0000000000044484",
  "version": "0.6",
```

4.5 Getting an occurrence (Taxonomy codes)

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/get/{e2id}
- Method Type: GET
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters: e2id (unique identifier of the occurrence)
- Body: Empty
- Response:

```
{
  "data": {
    "e2Id": "OR-0000000000044484",
    "version": "0.4",
    "type": "OR",
    "status": "SENT",
    "responsibleEntityId": 29,
    "reportingEntityId": 122,
    "pdfCode": null,
    "relatedReports": null,
    "rowsTaxCodes": {
      "24": {
        "ATTRIBUTES": {
          "1088": [
```

4.6 Getting an occurrence (xsd_tag)

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/{e2id}
- Method Type: GET
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters:
  - ✓ e2id (unique identifier of the occurrence)
  - ✓ singleValue (true/false): retrieve either full path to values (false) or just the value (true)
- Body: Empty
- Response:

```
{
  "data": {
    "e2Id": "OR-0000000000044484",
    "version": "0.4",
    "type": "OR",
    "status": "SENT",
    "responsibleEntityId": 29,
    "reportingEntityId": 122,
    "pdfCode": null,
    "relatedReports": null,
    "rowsTaxCodes": {
      "24": {
        "ATTRIBUTES": {
          "1088": [
            0
```

4.7 Getting an occurrence by fields (Taxonomy codes)

When getting an Occurrence by fields, the path of the attributes and/or entities to be retrieved are specified in the body.

- When including an Attribute in the request, only that specific one is included in the response.
- When including an Entity in the request, the response includes all the Attributes contained in it.
- If the requested Attribute/Entity is not included in the Occurrence in question, it will not appear in the response.

Example

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/fields/{e2id}
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters: e2id (unique identifier of the occurrence)
- Body:

```
[
   "24.ATTRIBUTES.473"
]
```

- Response:

```
{
    "data": {
        "e2Id": "OR-0000000000044484",
        "version": "0.4",
        "type": "OR",
        "status": "SENT",
        "responsibleEntityId": 29,
```

4.8 Getting an occurrence by fields (xsd_tag)

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/fields/{e2id}
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters:
  - ✓ e2id (unique identifier of the occurrence)
  - ✓ singleValue (true/false): retrieve either full path to values (false) or just the value (true)
- Body:

```
[
   "24.ATTRIBUTES.473"
]
```

- Response:

```
{
    "data": {
        "e2Id": "OR-0000000000000635",
        "version": "0.1",
        "type": "OR",
        "status": "PROCESSED",
        "responsibleEntityId": 3,
```

4.9 Changing an occurrence status

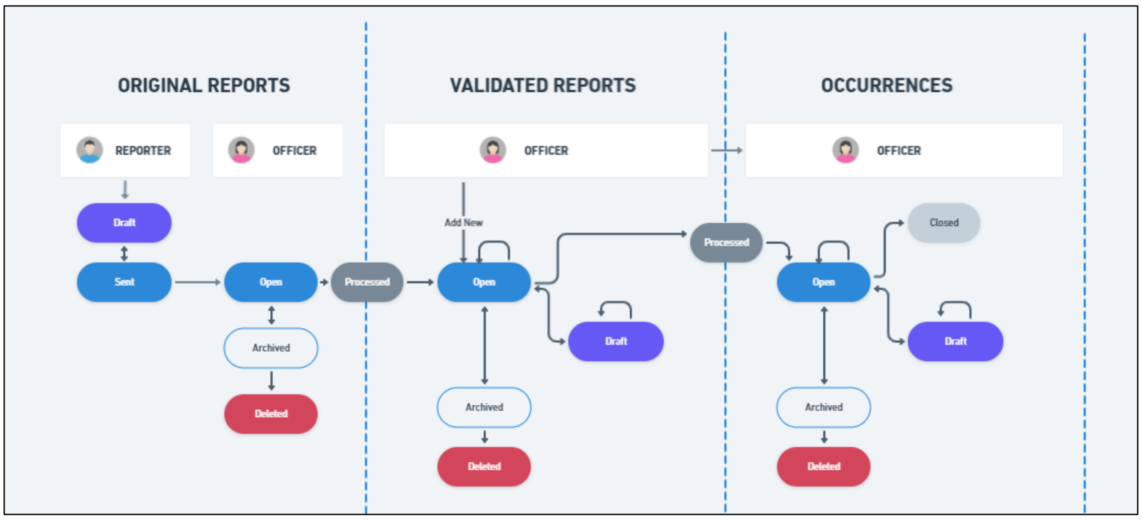
The input parameters are the following:

- e2id – ECCAIRS2 unique identifier
- Status to which the report will be changed, with options:

- ✓ DRAFT
- ✓ SENT
- ✓ OPEN
- ✓ ARCHIVED
- ✓ DELETED
- ✓ PROCESSED
- ✓ CLOSED

- VersionType. Mandatory for Open Occurrences, with options:
  - ✓ Minor
  - ✓ Major

Note that not all status changes are allowed, the previous status of the report has to be taken into account as the image above indicates.



Example

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/change-status
- Method Type: POST
- Authorization: Follow STEP 2 with the token obtained in STEP 7

- Body:

```
{
   "e2Id": "OR-0000000000000010",
   "status": "SENT"
}
```

- Response:

```
{
   "data": {
      "e2Id": "OR-0000000000000010",
      "status": "SENT",
      "version": "0.2"
   },
   "errorDetails": "",
   "returnCode": 1
}
```

5 Other methods

5.1 Get Query Result (Taxonomy codes)

This method executes the queries saved into the ECCAIRS 2 database.

It can be used to retrieve less than 10000 documents. The documents can be retrieved all in the same request or in multiple requests using the “pageSize” and “page” fields. The maximum pageSize supported is 10000. The first “page” is the “page” 0.

The message `"404 NOT_FOUND - The page requested does not exist"` will be received when a “page” does not exist.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/result
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "pageSize": 4,
  "page":0
}
```

- Response:

```
{
    "data": [
        {
            "e2Id": "OC-0000000000179425",
            "version": "0.1",
            "type": "OC",
            "status": "OPEN",
            "responsibleEntityId": 8,
            "reportingEntityId": null,
```

5.2 Get Query Result (xsd_tags)

This method executes the queries saved into the ECCAIRS 2 database.

It can be used to retrieve less than 10000 documents. The documents can be retrieved all in the same request or in multiple requests using the “pageSize” and “page” fields. The maximum pageSize supported is 10000. The first “page” is the “page” 0.

The message `"404 NOT_FOUND - The page requested does not exist"` will be received when a “page” does not exist.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/result
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters:
  - ✓ single-value (true/false): retrieve either full path to values (false) or just the value (true)
- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "pageSize": 4,
  "page":2
}
```

- Response:

```
{
    "data": [
        {
            "e2Id": "OC-0000000000183779",
            "version": "0.1",
            "type": "OC",
            "status": "OPEN",
            "responsibleEntityId": 8,
            "reportingEntityId": null,
```

5.3 Get Query Result (Taxonomy codes) with scroll-id

This method executes the queries saved into the ECCAIRS 2 database.

It can be used to retrieve more than 10000 documents. The documents can be retrieved in multiple requests using the “pageSize” in the first request to obtain the scrollId that will be used to retrieve the rest of documents in the amount indicated in the “pageSize” field. The maximum pageSize supported is 10000.

The scrollId expires after 30 minutes. Once expired it is required to execute the first request to obtain a new scrollId.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/result/scroll
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "pageSize": 2,
  "scrollId": null
}
```

- Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO7RgWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABO7RcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
```

Using the scrollId obtained in the response of this first request, execute the next requests as follows to obtain the number of documents indicated in the “pageSize” field until the end of the list. The total number of records that meets the query condition is indicated in the “total” field.

- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "pageSize": 2,
  "scrollId":"FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGd-
DYlJveWZSTHZuNWVuYU9RAAAAAABO7RgWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGd-
DYlJveWZSTHZuNWVuYU9RAAAAAABO7RcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ=="
}
```

- Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0
eWZSTHZuNWVuYU9RAAAAAABO7RgWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJve
HZuNWVuYU9RAAAAAABO7RcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000180000",
                "version": "0.1",
                "type": "OC",
                "status": "OPEN",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Nuuk/Godthåb - BGGH"
                            ],
                            "452": [
                                "85488"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        102
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2019-03-19"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID000000000000000000000000001
```

5.4 Get Query Result (xsd_tags) with scroll-id

This method executes the queries saved into the ECCAIRS 2 database.

This method can be used to retrieve more than 10000 documents. The documents can be retrieved in multiple requests using the “pageSize” in the first request to obtain the scrollId that will be used to retrieve the rest of documents in the amount indicated in the “pageSize” field. The maximum pageSize supported is 10000.

The scrollId expires after 30 minutes. Once expired it is required to execute the first request to obtain a new scrollId.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/result/scroll
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters:
  - ✓ singleValue (true/false): retrieve either full path to values (false) or just the value (true)
- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "pageSize": 2,
  "scrollId": null
}
```

- Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZF-
FoZTBTMTM4QmFSUTJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
```

Using the scrollId obtained in the response of this first request, execute the next requests as follows to obtain the number of documents indicated in the "pageSize" field until the end of the list. The total number of records that meets the query condition is indicated in the "total" field.

- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "scrollId":"FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGd-
DYlJveWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZF-
FoZTBTMTM4QmFSUTJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw=="
}
```

- Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZF-
FoZTBTMTM4QmFSUTJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000185518",
```

5.5 Get Query ResultByQuery (Taxonomy codes)

This method can be used to retrieve less than 10000 documents. The documents can be retrieved all in the same request or in multiple requests using the "pageSize" and "page" fields. The maximum pageSize supported is 10000. The first "page" is the "page" 0.

The message `"404 NOT_FOUND - The page requested does not exist"` will be received when a "page" does not exist.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/result/bool
- Method Type: POST
- Authorization: Follow STEP 2 with the token obtained in STEP 7
- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "pageSize": 3,
    "page": 0
}
```

- Response:

```
 {
   "data": [
        {
            "e2Id": "OC-0000000000177910",
            "version": "0.0",
            "type": "OC",
            "status": "DRAFT",
            "responsibleEntityId": 8,
            "reportingEntityId": null,
            "pdfCode": null,
```

5.6 Get Query ResultByQuery (xsd_tags)

This method can be used to retrieve less than 10000 documents. The documents can be retrieved all in the same request or in multiple requests using the “pageSize” and “page” fields. The maximum pageSize supported is 10000. The first “page” is the “page” 0.

The message `"404 NOT_FOUND - The page requested does not exist"` will be received when a “page” does not exist.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/result/bool
- Method Type: POST
- Authorization: Follow STEP 2 with the token obtained in STEP 7
- Parameters:
    - ✓ single-value (true/false): retrieve either full path to values (false) or just the value (true)
- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "pageSize": 3,
    "page":0
}
```

- Response:

```
{
 "data": [
        {
            "e2Id": "OC-0000000000177910",
            "version": "0.0",
            "type": "OC",
            "status": "DRAFT",
            "responsibleEntityId": 8,
            "reportingEntityId": null,
            "pdfCode": null,
```

5.7 Get Query ResultByQuery (Taxonomy codes) with scroll-id

This method can be used to retrieve more than 10000 documents. The documents can be retrieved in multiple requests using the “pageSize” in the first request to obtain the scrollId that will be used to retrieve the rest of documents in the amount indicated in the “pageSize” field. The maximum pageSize supported is 10000.

The scrollId expires after 30 minutes. Once expired it is required to execute the first request to obtain a new scrollId.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/result/bool/scroll
- Method Type: POST
- Authorization: Follow STEP 2 with the token obtained in STEP 7
- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "pageSize": 3,
    "scrollId": null
}
```

- Response:

```
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZLZHM3TTl6WVRu
R01OX2RFYy1CdmtnAAAAAABX1ZsWdks2TEE3YWlRemU5YjU3aHlPUmtKZxZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKc10Wa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
```

Using the scrollId obtained in the response of this first request, execute the next requests as follows to obtain the number of documents indicated in the “pageSize” field until the end of the list. The total number of records that meets the query condition is indicated in the ”total” field.

- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZLZHM3TTl6WVRuR01O
X2RFYy1CdmtnAAAAAABX1ZsWdks2TEE3YWlRemU5YjU3aHlPUmtKZxZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKc10Wa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ=="
}
```

- Response:

```
{    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZLZHM3TTl6WVRu
R01OX2RFYy1CdmtnAAAAAABX1ZsWdks2TEE3YWlRemU5YjU3aHlPUmtKZxZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKc10Wa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000178279",
                "version": "0.1",
                "type": "OC",
```

5.8 Get Query ResultByQuery (xsd_tags) with scroll-id

This method can be used to retrieve more than 10000 documents. The documents can be retrieved in multiple requests using the “pageSize” in the first request to obtain the scrollId that will be used to retrieve the rest of documents in the amount indicated in the “pageSize” field. The maximum pageSize supported is 10000.

The scrollId expires after 30 minutes. Once expired it is required to execute the first request to obtain a new scrollId.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/result/bool/scroll
- Method Type: POST
- Authorization: Follow STEP 2 with the token obtained in STEP 7
- Parameters:
  - ✓ single-value (true/false): retrieve either full path to values (false) or just the value (true)
- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "pageSize": 3,
    "scrollId": null
}
```

- Response:

```
{    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
```

Using the scrollId obtained in the response of this first request, execute the next requests as follows to obtain the number of documents indicated in the "pageSize" field until the end of the list. The total number of records that meets the query condition is indicated in the "total" field.

- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "scrollId":"FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJveWZST
HZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuYU9RA-
AAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ=="
}
```

- Response:

```
 {   "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000184145",
                "version": "0.1",
                "type": "OC",
```

5.9 Get Query Result (xsd_tags) shared with scroll-id

This method executes the queries saved into the ECCAIRS 2 database.

This method can be used to retrieve more than 10000 documents shared with the ECR. The documents can be retrieved in multiple requests using the "pageSize" in the first request to obtain the scrollId that will be used to retrieve the rest of documents in the amount indicated in the "pageSize" field. The maximum pageSize supported is 10000.

Only Eccairs authority or the authorities which belong to the ECR can use this service.

The scrollId expires after 30 minutes. Once expired it is required to execute the first request to obtain a new scrollId.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/shared/result/scroll
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters:
  - ✓ singleValue (true/false): retrieve either full path to values (false) or just the value (true)
  - ✓ auth_include (true/false): retrieve all Occurrences shared with ECR except those of your authority (false) or including those those of your authority (true). By default is true.
- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "pageSize": 2,
  "scrollId": null
}
```

- Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZF-
FoZTBTMTM4QmFSUTJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
```

Using the scrollId obtained in the response of this first request, execute the next requests as follows to obtain the number of documents indicated in the "pageSize" field until the end of the list. The total number of records that meets the query condition is indicated in the "total" field.

- Body:

```
{
  "type": "OCCURRENCE",
  "libraryName": "Test Library",
  "queryName": "Occurrence Class",
  "scrollId":"FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGd-
DYlJveWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZF-
FoZTBTMTM4QmFSUTJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw=="
}
```

- Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZF-
FoZTBTMTM4QmFSUTJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000185518",
```

5.10 Get Query ResultByQuery (xsd_tags) shared with scroll-id

This method can be used to retrieve less than 10000 documents shared with the ECR. The documents can be retrieved in multiple requests using the "pageSize" in the first request to obtain the scrollId that will be used to retrieve the rest of documents in the amount indicated in the "pageSize" field. The maximum pageSize supported is 10000.

Only Eccairs authority or the authorities which belong to the ECR can use this service.

The scrollId expires after 30 minutes. Once expired it is required to execute the first request to obtain a new scrollId.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/readable/shared/result/bool/scroll
- Method Type: POST
- Authorization: Follow STEP 2 with the token obtained in STEP 7
- Parameters:
  - ✓ Single-value (true/false): retrieve either full path to values (false) or just the value (true)
  - ✓ auth_include (true/false): retrieve all Occurrences shared with ECR except those of your authority (false) or including those those of your authority (true). By default is true.
- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "pageSize": 3,
    "scrollId": null
}
```

- Response:

```
{    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
```

Using the scrollId obtained in the response of this first request, execute the next requests as follows to obtain the number of documents indicated in the “pageSize” field until the end of the list. The total number of records that meets the query condition is indicated in the "total" field.

- Body:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modifica-
tionDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "scrollId":"FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJveWZST
HZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuYU9RA-
AAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ=="
}
```

- Response:

```
 {   "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZST-
HZuNWVuYU9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000184145",
                "version": "0.1",
                "type": "OC",
```

5.11 GetTaxonomy

It returns an object containing the properties of the specified entities and attributes in the released taxonomy of the user's authority. In the body of the request, the taxonomy codes of the attributes under a given entity taxonomy code must be provided as shown in the body of the example below.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy-service/external/taxonomy
- Method Type: POST
- Authorization: Follow STEP 2
- Body:

```
[
  {
    "entityTC": 24,
    "attributesList": [746, 19, 17]
  },
  {
```

- Response:

```
{
  "data": {
    "responseData": {
      "page": 0,
      "pageSize": 1,
      "totalRows": 1,
      "responseStatus": "OK",
      "responseMessage": "",
      "taxonomyName": "ECCAIRS Aviation",
      "language": "3",
      "responseCode": 200
    },
    "list": [
      {
        "id": 2,
        "taxonomyCode": 1,
        "description": "Aerodrome General",
        "detailed": "Aerodrome General",
        "explanation": "The identification of the aerodrome/helicopter landing
area by name, location and status.",
        "xsdTag": "Aerodrome_General",
        "isLink": null,
        "minInstance": null,
        "maxInstance": null,
        "attributesList": {
          "childrenAttributes": [
            {
              "id": 92,
              "taxonomyCode": 1,
              "description": "Aerodrome latitude",
```

5.12 Get TaxonomyVersion

This method returns the current taxonomy version of the user’s authority. If the Authority has a custom taxonomy in status released, it will return the version of the Custom Taxonomy. Otherwise, it will return the version of the General Taxonomy released.

- Section: Taxonomy Management Section.
- Method URL: {{API URL}}/taxonomy-service/released-taxonomy
- Type: GET
- Authorization: Follow STEP 2
- Body: empty
- Response:

```
Custom Taxonomy
{
    "data": [
        {
            "id": 36,
            "name": "ECCAIRS Aviation_EASA_5.1.0.0 (C0)",
            "version": "5.1.0.0 (C0)"
        }
    ],
    "errorDetails": "",
    "returnCode": 1
}
```

5.13 Get Taxonomy json schema

This method returns the json schema of a given taxonomy id.

- Section: Taxonomy Management Section.
- Method URL: {{API URL}} /taxonomy-service/public/getJsonSchema/{id}
- Type: GET
- Authorization: Follow STEP 2
- Parameters: id (identifier of the taxonomy)
- Body: empty
- Response:

```
{
  "data": "{\"type\": \"object\", \"title\": \"ECCAIRS Aviation 5.1.0.0\", \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"default\": \"\", \"examples\": \"\", \"required\": [\"taxonomy_codes\", \"status\", \"updated\", \"version\", \"responsible_entity_id\", \"au_creation_date\", \"au_active\", \"blocked\"], \"properties\": {\"_id\": {\"type\": \"string\", \"title\": \"Identifier - Primary Key\", \"description\": \"Report unique identifier (Primary Key)\"}, \"e2_id\": {\"$ref\": \"#/definitions/typeE2ID\", \"title\": \"Eccairs Number\", \"examples\": \"[OR-0000000000001234,VR-0000000000009876,OC-0000000000033556,SR-0000000000225789]\", \"description\": \"Eccairs Number. The unique identifier for the system user. Prefix: (OR-,VR-,OC-,SR-) | Format: PF-9(16) | Lenght: 19\"}, \"_class\": {\"type\": \"string\", \"title\": \"Class the report was inserted from. Optional\", \"examples\": \"\", \"description\": \"Class the report was inserted from. Optional\"}, \"status\": {\"enum\": [\"DRAFT\", \"SENT\", \"OPEN\", \"PROCESSED\", \"ARCHIVED\", \"DELETED\", \"CLOSED\"], \"type\": \"string\", \"title\": \"Status of the report\", \"examples\": \"[DRAFT,SENT,OPEN,PROCESSED,ARCHIVED,DELETED,CLOSED]\", \"description\": \"Report status. Only the states that have been defined in the listing, and which are indicated in the examples, will be allowed.\"}, \"blocked\": {\"type\": \"boolean\", \"title\": \"Automatic update\", \"examples\": \"[true, false]\", \"description\": \"It determines whether or not the report is blocked for editing. Used for BatchOperations and to avoid concurrent editions\"}, \"updated\": {\"$ref\": \"#/definitions/typeUpdated\", \"title\": \"It determines whether the report is locked for editing.\", \"description\": \"Determines if the report has been automatically updated, from a new Original Report.\"}, \"version\": {\"type\": \"string\", \"title\": \"Occurrence ver-
```

5.14 GetEntity

This method returns an object containing the properties for the specified entity and its attributes in the released taxonomy of the user's authority.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/ taxonomy-service/external/entity/{{entityID}}
- Method Type: GET
- Authorization: Follow STEP 2

Example

https://api.uat-aviationreporting.eu/taxonomy/external/entity/1

- Response:

```
{
   "data": {
      "id": 5131,
      "taxonomyCode": 1,
      "description": "Aerodrome General",
      "detailed": "Aerodrome General",
      "explanation": "The identification of the aerodrome/helicopter landing area by name, loca-
tion and status.",
      "xsdTag": "Aerodrome_General",
      "isLink": null,
      "minInstance": null,
      "maxInstance": null,
      "attributesList": {
         "childrenAttributes": [
            {
               "id": 92364,
               "taxonomyCode": 1,
               "description": "Aerodrome latitude",
               "detailed": "Aerodrome latitude",
```

5.15 Get ValueLists for Taxonomy Reference

This method retrieves all the Value Lists of a given Taxonomy when providing the Taxonomy Reference (ADREP/SRIS) and the Taxonomy status (Draft/Released).

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy-service/external/value-lists/taxonomy-reference/{{id}}
- Method Type: GET
- Authorization: Follow STEP 2
- Parameters:
  - ✓ Id: id of the Taxonomy Reference:
    - 1: ADREP (default)
    - 2: SRIS
  - ✓ fromDraft: status of the taxonomy from where the Value Lists will be retrieved
    - true: obtain the Value Lists of the Taxonomy in draft status
    - false: obtain the Value Lists of the Taxonomy in released status (by default)

- Body: empty

Example:

{{BASE_URL}}/taxonomy-service/external/valueLists/taxonomy-reference/1?fromDraft=true

- Response:

```
{
  "data": {
    "id": 3,
    "sourceName": "VL for AttrID  645 - Bird Species.xls",
    "identifier": 1041,
    "description": "VL for AttrID: 645 - Bird Species",
    "detailed": null,
    "explanation": "The list of bird species. Non-bird entries (mammal, rep-
tile) are also included.",
    "levels": 2,
    "lastIdentifier": 1133,
    "customId": null,
    "authority": {
      "creationDate": "2020-08-12T14:03:24.000+0000",
      "creationUser": "",
      "modificationDate": "2020-12-09T16:17:52.000+0000",
      "modificationUser": "eccairsroot@20.50.175.208",
      "active": true,
      "id": 1,
      "authorityName": "EASA_ECCAIRS",
```

5.16 Get ValueList

Use this method to retrieve a given Value List by its internal id.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy-service/external/value-lists/{{id}}
- Method Type: GET
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ Id: internal id of the value list to retrieve obtained in the previous step.
- Body: empty

Example:

{{BASE_URL}}/taxonomy-service/external/valueLists/22863

- Response:

```
{
    "data": {
        "id": 22863,
        "sourceName": "VL for AttrID  746 - Detection.xls",
        "identifier": 100000,
        "description": "VL for AttrID: 746 - Detection",
        "detailed": null,
        "explanation": null,
        "levels": 1,
        "lastIdentifier": 16,
        "customId": 1,
        "authority": {
            "id": 33,
            "authorityName": "EASA",
            "authorityCode": "EASA",
            "country": {
                "id": 5811636,
                "name": "Germany",
                "code": "DE"
            },
            "countryCode": "DE",
            "authorityType": {
```

5.17 Update ValueList

Used to update a Value List properties under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to update the Value List properties.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy-service/external/value-lists/{{id}}
- Method Type: PUT
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ Id: internal id of the Value List to be updated
- Body:

```
{
                "id": 27980,
                "sourceName":  "VL for AttrID  746 - Detection (1).xls",
                "identifier": 100000,
                "description": "VL for AttrID: 746 - Detection DEMO",
                "detailed": "Test API - Detailed",
                "explanation": "Test API - Explanation",
```

- Response:

```
{
    "data": {
        "id": 27980,
        "sourceName": "VL for AttrID  746 - Detection (1).xls",
        "identifier": 100000,
        "description": "VL for AttrID: 746 - Detection DEMO",
        "detailed": "Test API - Detailed",
        "explanation": "Test API - Explanation",
        "levels": 5,
        "lastIdentifier": null,
        "customId": 1,
        "isCustom": true,
        "locked": false,
        "releasedTaxonomyIdVersion": {
            "id": 311,
            "name": "Eccairs Aviation_DE_5.4.0.0 (C1)",
            "description": "EASA 2022",
            "version": null,
            "reference": null,
            "releasedDate": null,
            "tag": null,
            "taxonomyReference": {
                "id": 1
```

5.18 Create ValueList

Used to create a Value List under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to create a Value List.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/ taxonomy-service/external/value-lists/{{id}}
- Method Type: POST
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ Id: id of the Taxonomy Reference:
    - 1: ADREP (default)
    - 2: SRIS
- Body:

  Example: create a Value List without values (the values ca be created/updated or deleted separately)

```
{
   "id":null,
   "description":"NEW VL Description",
   "detailed":"NEW VL Detailed",
   "explanation":"NEW VL Explanation",
```

- Response:

```
{
    "data": {
        "id": 28014,
        "sourceName": null,
        "identifier": 100051,
        "description": "NEW VL Description",
        "detailed": "NEW VL Detailed",
        "explanation": "NEW VL Explanation",
        "levels": 3,
        "lastIdentifier": null,
        "customId": 1,
        "isCustom": true,
        "locked": false,
        "releasedTaxonomyIdVersion": {
            "id": 311,
            "name": "Eccairs Aviation_DE_5.4.0.0 (C1)",
            "description": "EASA 2022",
            "version": null,
            "reference": null,
            "releasedDate": null,
            "tag": null,
            "taxonomyReference": {
                "id": 1
```

- Body:

Example: create a Value List with values

```
{
   "id": null,
   "description":"NEW VL2 Description",
   "detailed":"NEW VL2 Detailed",
   "explanation":"NEW VL2 Explanation",
   "levels":3,
   "customId":1,
   "alias":[{
              "aliasId":3
    }],
   "listOfValues":[{
        "isNewRecord":0,
        "identifier":2,
        "parent":0,
        "level":1,
        "description":"NEW Value 2",
        "detailed":"NEW Value 2",
        "explanation": "NEW Value 2",
        "hasChild":0,
        "specialValue":{
            "id":1
        },
```

- Response:

```
{
    "data": {
        "id": 28020,
        "sourceName": null,
        "identifier": 100056,
        "description": "NEW VL2 Description",
        "detailed": "NEW VL2 Detailed",
        "explanation": "NEW VL2 Explanation",
        "levels": 3,
        "lastIdentifier": null,
        "customId": 1,
        "isCustom": true,
        "locked": false,
        "releasedTaxonomyIdVersion": {
            "id": 311,
            "name": "Eccairs Aviation_DE_5.4.0.0 (C1)",
            "description": "EASA 2022",
            "version": null,
            "reference": null,
            "releasedDate": null,
            "tag": null,
            "taxonomyReference": {
                "id": 1
```

5.19 Delete ValueList

Used to delete a Value List under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to delete the Value List.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy-service/external/value-lists/{{id}}
- Method Type: DELETE
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ Id: internal id of the Value List to be deleted from the draft Taxonomy
- Body: empty
- Response:

```
{
    "data": "CRUD_FBS_17",
    "errorDetails": "",
    "returnCode": 1
}
```

System messages:

`CRUD_FBS_17:` The Value List has been successfully deleted

5.20 Create Value

Used to create a Value in a Value List under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to create a Value in the Value List.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy-service/external/value/{valueListId}/{taxonomyId}
- Method Type: POST
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ valueListId: id of the Value List where the Value will be created
  - ✓ taxonomyId: id of the draft version of the Taxonomy where the value will be created
- Body:

```
{
    "valueList":{
       "id":28020
   },
   "identifier":4,
   "parent":0,
   "level":1,
   "description":"NEW Value 4",
   "detailed":"NEW Value 4",
   "hasChild":0,
   "specialValue":{
```

- Response:

```
{
    "data": {
        "id": 7027091,
        "valueList": {
            "id": 28020,
            "sourceName": null,
            "identifier": 100056,
            "description": "NEW VL2 Description",
            "detailed": "NEW VL2 Detailed",
            "explanation": "NEW VL2 Explanation",
            "levels": 3,
            "lastIdentifier": null,
            "customId": 1,
            "isCustom": null,
            "locked": null,
            "releasedTaxonomyIdVersion": {
                "id": 311,
                "name": "Eccairs Aviation_DE_5.4.0.0 (C1)",
                "description": "EASA 2022",
                "version": null,
                "reference": null,
                "releasedDate": null,
                "tag": null
```

5.21 Update Value

Used to update a Value in a Value List under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to update a Value in the Value List.

- Section: Taxonomy Management Section.
- Method URL:
- {BASE_URL}/taxonomy-service/external/value/{valueId}/{valueListId}/{taxonomyId}
- Method Type: PUT
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ valueId: id of the Value to be updated
  - ✓ valueListId: id of the Value List where the Value will be updated
  - ✓ taxonomyId: id of the draft version of the Taxonomy where the value will be updated
- Body:

```
{       "id": 7027085,
        "valueList": {
            "id": 28020
        },
        "identifier": 4,
        "parent": 0,
        "level": 1,
        "description": "Update Value 4",
        "detailed": "Update Value 4",
```

- Response:

```
{
    "data": {
        "id": 7027085,
        "valueList": {
            "id": 28020,
            "sourceName": null,
            "identifier": 100056,
            "description": "NEW VL2 Description",
            "detailed": "NEW VL2 Detailed",
            "explanation": "NEW VL2 Explanation",
            "levels": 3,
            "lastIdentifier": null,
            "customId": 1,
            "isCustom": null,
            "locked": null,
            "releasedTaxonomyIdVersion": {
                "id": 311,
                "name": "Eccairs Aviation_DE_5.4.0.0 (C1)",
                "description": "EASA 2022",
                "version": null,
                "reference": null,
                "releasedDate": null,
                "tag": null
```

5.22 Delete Value

Used to delete a Value in a Value List under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to delete a Value in the Value List.

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/ taxonomy-service/external/value/{valueId}
- Method Type: DELETE
- Authorization: Follow STEP 2 in section 3
- Parameter:
    - ✓ valueId: id of the Value to be deleted

- Body: empty
- Response:

```
{
    "data": "The List Of value was successfully deleted",
    "errorDetails": "",
    "returnCode": 1
```

5.23 Restore Value

Used to restore a Value previously deleted (logical delete) in a Value List under the Taxonomy project (draft) of the user's authority. A Taxonomy in draft status for the user's authority must previously exists to be able to restore a Value in the Value List.

Additionally, the restore of a Value is only possible when the Value was present is the released Taxonomy from where the draft version was created. If the Value only exists in the draft version of the Taxonomy, it is not possible to restore the deleted (physical delete) Value.

- Section: Taxonomy Management Section
- Method URL: {BASE_URL}/ taxonomy-service/external/value/restore/{valueId}
- Method Type: PUT
- Authorization: Follow STEP 2 in section 3
- Parameter:
  - ✓ valueId: id of the Value to be restored
- Body: empty
- Response:

```
{
    "data": "The List of value was successfully restored",
    "errorDetails": "",
    "returnCode": 1
```

5.24 GetUsers

It returns the list of all users configured for the logged-in user's authority.

- Section: Taxonomy Management Section
- Method URL: {BASE_URL}/usermgmt/users
- Method Type: GET

- Authorization: Follow STEP 2 in section 3
- Body: empty
- Parameters:
  - ✓ isOrganisational:
    - If TRUE, it returns the Organisational users;
    - If FALSE, it returns the non-Organisational users;
    - If NULL, it returns all.
  - ✓ returnAll:
    - If TRUE, it returns all the users (active and inactive),
    - if NULL or FALSE it returns only the active ones.
- Response:

```
{
    "data": [
        {
            "originalId": null,
            "id": 1,
            "firstName": "eccairs_first_name",
            "lastName": "eccairs_last_name",
            "username": "eccairs",
            "email": "easa-team-uat@bilbomatica.es",
```

6 Working with attachments

6.1 Uploading attachments to a folder

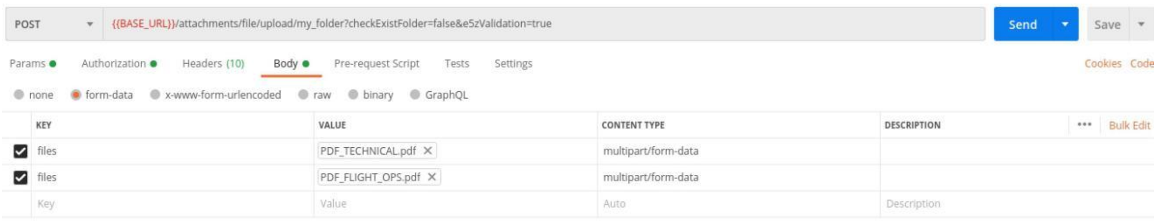
It allows to upload attachments to a folder in order to be referenced when creating an occurrence or updating an attribute (Datatype=ECCAIRS Resource Locator) with attachments.

- Section: Original Reports to Occurrences
- Method URL: {BASE_URL}/attachments/file/upload/{folder_name}
- Method Type: POST
- Authorization: Follow STEP 2 in section 3
- Parameters:

- ✓ folder_name: target folder name
- ✓ checkExistFolder (true/false): to check either if the folder already exists (true) or not (false)
- ✓ e5zValidation (true/false): both option check for valid attachment extensions but when equal to true only restrict from uploading executable files

- Body (form-data):
  - ✓ files: attachment filename. This parameter shall be included as many times as the attachments to be uploaded
    - Supported Media Types: multipart/form-data
    - Type: File

Example:



POST {{BASE_URL}}/attachments/file/upload/my_folder?checkExistFolder=false&e5zValidation=true

	KEY	VALUE	CONTENT TYPE	DESCRIPTION
✓	files	PDF_TECHNICAL.pdf	multipart/form-data	
✓	files	PDF_FLIGHT_OPS.pdf	multipart/form-data	
	Key	Value	Auto	Description

- Response:

```
{
    "data": "Test_BBM",
```

6.2 Creating an occurrence with attachments

We use the same method explained in section 4.1 but including, in the relevant attributes, the name of the folder where the attachments were previously upload (step 6.1) and their filenames.

- Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/create
- Method Type: POST
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Body:

```
{
    "type": "REPORT",
    "taxonomyCodes": {
        "24": {
            "ID": "#id_number#",
            "ATTRIBUTES": {
                "601":[ "OR with Attachment" ],
                "452":["2021_K1001"],
```

- Response:

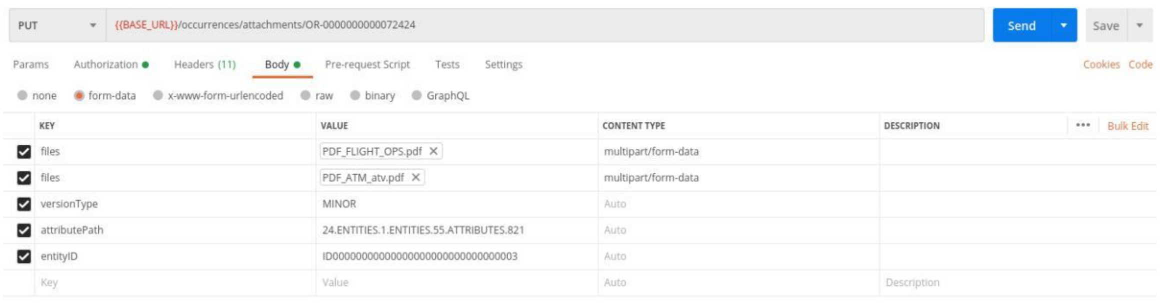
```
{
    "data": {
        "e2Id": "OR-0000000000072437",
        "status": "SENT",
```

6.3 Adding attachments to an existing occurrence

It allows to upload attachments of an attribute of an occurrence

- Section: Original Reports to Occurrences
- Method URL: {BASE_URL}/ occurrences/attachments/{e2id}
- Method Type: PUT
- Authorization: Follow STEP 2 in section 3
- Parameters: e2id (unique identifier of the occurrence)
- Body (form-data):
    - ✓ attributePath: attribute location where attachment is saved. If this attribute previously exists, then the attachment is added.
    - ✓ entityID: identification number of entity of attribute. Only is required when attribute belong to a sub-entity of Occurrence entity (taxonomy code 24).
    - ✓ versionType, which determines how to save the report, with options
        - DRAFT (for OR/VR/OC)
        - MINOR (for OR/VR/OC)
        - MAJOR (for OC)
    - ✓ files: attachment filename. This parameter shall be included as many times as the attachments to be uploaded. All files will be saved into the same attribute.
        - Supported Media Types: multipart/form-data
        - Type: File

Example:



{{BASE_URL}}/occurrences/attachments/OR-0000000000072424

none [✓ form-data] x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	CONTENT TYPE	DESCRIPTION
✓	files	PDF_FLIGHT_OPS.pdf	multipart/form-data	
✓	files	PDF_ATM_atv.pdf	multipart/form-data	
✓	versionType	MINOR	Auto	
✓	attributePath	24.ENTITIES.1.ENTITIES.55.ATTRIBUTES.821	Auto	
✓	entityID	ID00000000000000000000000000000003	Auto	
	Key	Value	Auto	Description

- Response:

```
{
    "data": {
        "e2Id": "OR-0000000000072437",
        "status": "SENT".
```

6.4 Downloading attachments

It allows to download all the attachments of an occurrence

- Section: Original Reports to Occurrences
- Method URL: {BASE_URL}/ occurrences/attachments/{e2id}
- Method Type: GET
- Authorization: Follow STEP 2 in section 3
- Parameters: e2id (unique identifier of the occurrence)
- Body: empty
- Response:

  The attached files will be included in a .zip file that will include all the occurrence attached files). The name of the zip file will be: timestamp + e2id

7 Getting all ORs as an organisational user

It allows to obtain the list of Original Reports in E2 from a given organisation. The list of Original Repots is pageable.

- Section: Working as an organizational user
- Method URL: {BASE_URL} /occurrences/reporter/original-report?&$skip={skip_length}&$top={top_length}
- Method Type: POST

- Authorization: Follow STEP 2 in section 3 and log in as an organisational user.
- Parameters:
  - ✓ Skip: the amount of records to skip.
  - ✓ Top: the amount of records to get in the response.
- Body:

```
{
    "status": [
        "DRAFT",
        "SENT",
```

- Response:

```
{
    "data": {
        "elements": [
            {
                "e2Id": "OR-0000000000044437",
                "version": "0.3",
                "reportStatus": "PROCESSED",
                "translatedStatus": null,
                "creationDate": "2022-03-15 08:01:03",
                "responsibleEntity": "United Kingdom > CAA",
                "responsibleEntityId": 29,
                "reportedByMe": true,
                "modificationUser": "MDCSpainOrgAdmin",
                "pdfCode": 6,
                "blocked": false
            },
            {
                "e2Id": "OR-0000000000044435",
                "version": "0.3",
```

8 Importing files

8.1 Create Original Reports from e5x files

It allows to create Original Reports in E2 from the xmls contained in one or more e5x files and save the xml from where the OR is created in the Original Report created in E2 under the attribute 802.

- Section: Import from Files
- Method URL: {BASE_URL}/ frontfile-api/files/migrate
- Method Type: POST

- Authorization: Follow STEP 2 in section 3
- Parameters:
  - ✓ Files (e5x files)
  - ✓ PDFCode: Code that identifies the Aviation Sector of the ORs imported
    - 2 (General Aviation)
    - 3 (Technical)
    - 4 (Aerodrome and Ground Handling)
    - 5 (ATM/ANS)
    - 6 (Flight Operations)

- Response:

```
{
    "data": "OCC_FBS_310",
    "errorDetails": "",
    "returnCode": 1
```

System messages:

`OCC_FBS_310:` Your file has been successfully uploaded and is being processed.
`OCC_FBE_388:` The PDFCode must be provided.

8.2 Getting the results of a migrated e5x file

It allows to obtain a CSV or JSON file with the results of a migration file. The file includes the results for every document included in the file that was sent to be migrated.

- Section: Import from Files
- Method URL: {BASE_URL}/frontfile-api/results/e5xresults
- Method Type: GET
- Authorization: Follow STEP 2 in section 3
- Parameters:
  - ✓ format: csv or json
  - ✓ idFile: the identifier of the file migrated

- Response: (CSV file)

```
E5z/E5x Id,File Name,User Id, Authority Name,Total Reports,Total
Attachments,Total Reports Loaded,Total Attachments Loaded,Initial Process
Date,Final Process Date,Migration Origin,Migration Status, Message
2024,DEMO.e5x,MDCSpainOrgAdmin,Spain (CAA),6,0,6,0,2022-03-15
08:01:02.0,2022-03-15 08:01:12.0,Load On-line,Processed OK,"success"

, XML Id,XML FileName,File Number,Responsible Entity,Initial Process Date,Final
Process Date, Total Attachments, Total Attachments loaded,Migration Status,
Message
 , 1486993, 0B1E734001484DBB99D23B4931D81B01, FROMOR1, 29, 2022-03-15
08:01:03.0, 2022-03-15 08:01:08.0, 0, 0, Successfully migrated, "OR-
0000000000044434"

, XML Id,XML FileName,File Number,Responsible Entity,Initial Process Date,Final
Process Date, Total Attachments, Total Attachments loaded,Migration Status,
Message
```

9 Annex

9.1 Permissions per method

The following table reflects the permissions (user rights) needed to execute each of the methods listed in this guide. The permissions needed to execute a specific method shall be a part of the role to which the execution user is enrolled.

METHOD	USER RIGHT	COMMENTS
3.4 Logging in with an existing user		(none)
4.1 Creating an occurrence	• OR: CreateReporterOriginalReport • VR: CreateValidatedReport • OC: CreateOccurrence • SR: CreateSafetyRecommendation	Depends on the type of record to be created (OR, VR, OC, SR)
4.2 Updating an occurrence	• OR: EditReporterOriginalReport • VR: EditValidatedReport • OC: EditOccurrence • SR: EditSafetyRecommendation	Depends on the type of record to be updated (OR, VR, OC, SR)

METHOD	USER RIGHT	COMMENTS
4.4 Getting an occurrence	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	Depends on the type of record to be read (OR, VR, OC, SR)
4.7 Getting an occurrence by fields	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	Depends on the type of record to be read (OR, VR, OC, SR)
4.9 Changing an occurrence status	• OR: EditReporterOriginalReport, DeleteOriginalReport, ArchiveOriginalReport, • VR: EditValidatedReport, DeleteValidatedReport, ArchiveValidatedReport, • OC: EditOccurrence, DeleteOccurrence, ArchiveOccurrence, CloseOccurrence, • SR: EditSafetyRecommendation, DeleteSafetyRecommendation	Depends on the type of record to be updated (OR, VR, OC, SR)
5.1 Get Query Result	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	Depends on the type of record to be read (OR, VR, OC, SR)
5.5 Get Query ResultByQuery	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	Depends on the type of record to be read (OR, VR, OC, SR)
5.11 GetTaxonomy		(none)

METHOD	USER RIGHT	COMMENTS
5.12 GetEntity		(none)
5.15 GetValueList	ReadTaxonomyValueList	
5.16 GetValueLists	ReadTaxonomyValueList	
Erreur ! Source du renvoi introuvable. CreateValueList	CreateTaxonomyValueList	
5.17 UpdateValueList	EditTaxonomyValueList	
Erreur ! Source du renvoi introuvable. DeleteValueList	DeleteTaxonomyValueList	

9.2 List of authorities

ID	AUTHORITY NAME
1	EASA
2	AUSTRIA
3	BELGIUM
4	BULGARIA
5	CROATIA
6	CYPRUS
7	"CZECH REPUBLIC"
8	DENMARK
9	ESTONIA
10	FINLAND
11	FRANCE

ID	AUTHORITY NAME
12	GERMANY
13	GREECE
14	HUNGARY
15	ICELAND
16	IRELAND
17	ITALY
18	LATVIA
19	LITHUANIA
20	LUXEMBOURG
21	MALTA
22	NETHERLANDS
23	NORWAY
24	POLAND
25	PORTUGAL
26	ROMANIA
27	SLOVAKIA
28	SLOVENIA
29	SPAIN
30	SWEDEN
31	SWITZERLAND
32	"UNITED KINGDOM"

9.3 Reports JSON

In this section, we provide the JSON structure of reports (OR, VR and Occurrence). It is used in:

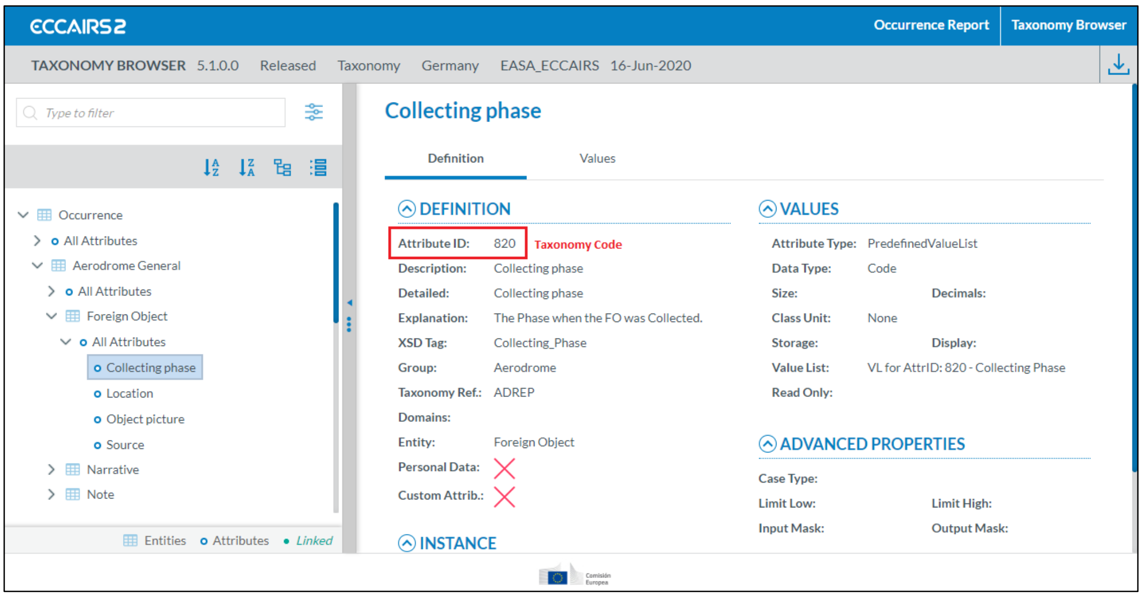
- 'taxonomyCodes' input parameter in:
  - ✓ Creating an occurrence
  - ✓ Updating an occurrence
- 'RowsTaxCodes' output parameter in:
  - ✓ Getting an occurrence URL
  - ✓ Section: Original Reports to Occurrences Section.
- Method URL: {BASE_URL}/occurrences/get-URL/{e2Id}
- Method Type: GET
- Authorization: Follow STEP 2 in section 3, with the token obtained in STEP 7
- Parameters: e2id (unique identifier of the occurrence)
- Body: Empty
- Response:

```
{
 "data": {
  "e2Id": "OR-0000000000044484",
  "version": "0.6",
```

  - ✓ Getting an occurrence
  - ✓ Getting an occurrence by fields
  - ✓ Get Query Result

9.3.1 Structure

Its structure is based in the Taxonomy, following the Entity/Attribute/List of Value/Value hierarchy, where each item is identified by its "Taxonomy Code" (also known as Entity ID, Attribute ID and Value ID).



Collecting phase

DEFINITION

Attribute ID:	820
Description:	Collecting phase
Detailed:	Collecting phase
Explanation:	The Phase when the FO was Collected.
XSD Tag:	Collecting_Phase
Group:	Aerodrome
Taxonomy Ref.:	ADREP
Domains:	
Entity:	Foreign Object
Personal Data:	✗
Custom Attrib.:	✗

INSTANCE

VALUES

Attribute Type:	PredefinedValueList		
Data Type:	Code		
Size:		Decimals:	
Class Unit:	None		
Storage:		Display:	
Value List:	VL for AttrID: 820 - Collecting Phase		
Read Only:			

ADVANCED PROPERTIES

Case Type:			
Limit Low:		Limit High:	
Input Mask:		Output Mask:	

For more information on the Taxonomy, you can access the Taxonomy Browser.

- In UAT here
- In PROD here
- And in the Sandbox environment here

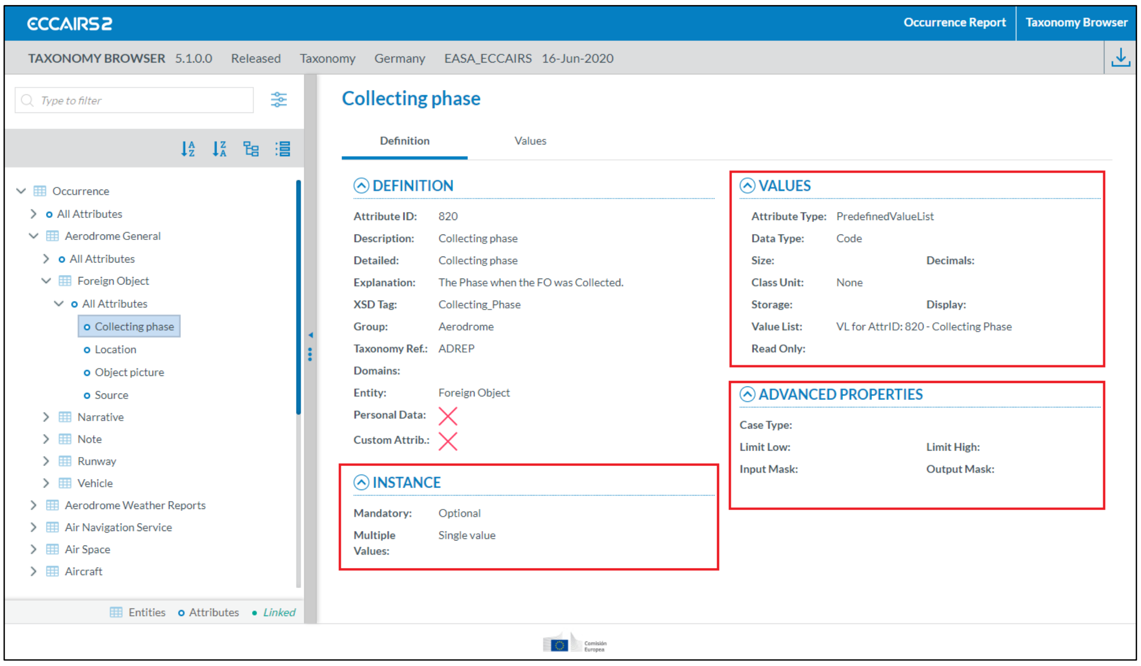
The JSON itself follows these basic guidelines:

- The first Entity is always 24 (Occurrence) and everything else lays below it.
- Only items (Entities or Attributes) with information are included in this JSON.
  - ✓ An attribute with no value should not be included.
  - ✓ An Entity with no attributes or child entities should not be included.
- All Entities (except for Entity 24) are included as an ARRAY OF OBJECTS. Each of these objects will consist of:
  - ✓ An ID, which is unique within the report.
    - When creating or editing a report with new entities, the ID will be included as **"ID": "#id_number#"**, the system will then assign the next available number.
  - ✓ ATTRIBUTES It includes all the Attributes that lay under the entity. They are always saved as arrays, to allow multivalued attributes.
  - ✓ CUSTOM_ATTRIBUTES It includes all the Custom Attributes that lay under the entity.
  - ✓ ENTITIES
    - Nested entities are possible in the Taxonomy.

  - This entity will in turn also have the same structure.
- ✓ LINKS
  - Linked entities will be referenced by their ID, with this nomenclature: **“REF": "ID25E475E37B394A05A39930C3D4735D20"**

9.3.2 Definition of attributes

For each attribute, the Taxonomy definition has to be met, regarding metadata like Data Type, Size, whether or not it is mandatory, maximum instances, etc. when applicable.



ECCAIRS2 | Occurrence Report | Taxonomy Browser

TAXONOMY BROWSER 5.1.0.0 Released Taxonomy Germany EASA_ECCAIRS 16-Jun-2020

Type to filter

- Occurrence
  - All Attributes
  - Aerodrome General
    - All Attributes
    - Foreign Object
      - All Attributes
        - Collecting phase
        - Location
        - Object picture
        - Source
    - Narrative
    - Note
    - Runway
    - Vehicle
  - Aerodrome Weather Reports
  - Air Navigation Service
  - Air Space
  - Aircraft

Entities · Attributes · Linked

Collecting phase

Definition | Values

DEFINITION

Attribute ID:	820
Description:	Collecting phase
Detailed:	Collecting phase
Explanation:	The Phase when the FO was Collected.
XSD Tag:	Collecting_Phase
Group:	Aerodrome
Taxonomy Ref.:	ADREP
Domains:	
Entity:	Foreign Object
Personal Data:	✗
Custom Attrib.:	✗

INSTANCE

Mandatory:	Optional
Multiple Values:	Single value

VALUES

Attribute Type:	PredefinedValueList		
Data Type:	Code		
Size:		Decimals:	
Class Unit:	None		
Storage:		Display:	
Value List:	VL for AttrID: 820 - Collecting Phase		
Read Only:			

ADVANCED PROPERTIES

Case Type:			
Limit Low:		Limit High:	
Input Mask:		Output Mask:	

When building the JSON, the **Data Type** is especially relevant, as the information is saved differently, depending on the case.

The attached PDF shows the detail for all the existing Data Types.

ECCAIRS-DataTypes.
pdf

Please note that it is an extract of the documentation used for the development, so please ignore the non-relevant information.

9.3.3 Example

The following code is the example of what a report JSON would look like, covering the scenarios explained above.

```
      "24": {
         "ATTRIBUTES": {
            "19": ["ECCAIRS Aviation"],
            "20": ["4.1.0.7"],
            "428": [4],
            "430": [106,5],
            "431": [300],
            "432": [99],
            "433": ["2013-10-13"],
            "434": ["2015-04-01T15:44:31"],
            "435": ["2019-09-02T09:30:01"],
            "436": [5],
            "440": ["approche"],
            "446": ["Gérard MASSINI"],
            "451": [99],
            "452": ["2015DNOR0957"],
            "453": [2060],
            "454": [{
               "content": [82],
               "AdditionalText": "string"
            }],
            "477": ["2013-10-12"],
            "601": ["Collision aviaire en approche"],
            "606": [99]
         },
         "CUSTOM_ATTRIBUTES": {“10101”: [6]},
         "ENTITIES": {
             “1”:   [
                    “ATTRIBUTES”:{
                    “8”: [3],
                    “24”: [4]
                    },
                    “LINKS":
                    {“14":
                           [
                           “REF": "ID25E475E37B394A05A39930C3D4735D20"
                           ]
                    },
                    ]
            "14": [
               {
                  "ATTRIBUTES": {
                     "390": [2050301]
                  },
                  "ID": ID25E475E37B394A05A39930C3D4735D20"
               }
            ],
            "53": [
               {
                  "ATTRIBUTES": {
                     "476": [2],
                     "495": [{
                           "content": [9808],
                           "AdditionalText": ""
```

```
                                }],
                        "800": [8],
                        "1070": [{
                            "EncodedText":"e1xydGYxXGFuc2lcYW5zaWNwZzEyNTJcZGVmZjBcZ
GVmbGFuZzEwNDB7XGZvbnR0Ymx7XGYwXGZuaWxcZmNoYXJzZXQwIEFyaW-
FsO319DQpcdmlld2tpbmQ0XHVjMVxwYXJkXGZzMjRccGFyDQpccGFyDQp9DQo="
                            }]
                        }
                    },
                    "ID": "#id_number#"
                }
            ]
        }
```