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 for M2M.

Author	Bilbomática
Date	31/03/2022
Version	4.16
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 4

2 First time use 4

2.1 Step 0: Entry point for ECCAIRS 2 API 4

2.2 Step 1: Login as an elevated user 5

2.3 Step 2: Token setting up 7

2.4 Step 3: Logging in with an existing user 9

3 Examples of use 10

3.1 Creating an occurrence 10

3.2 Updating an occurrence 11

3.3 Updating an occurrence - Deleting entities 12

3.4 Getting an occurrence (xsd_tag) 13

3.5 Getting an occurrence by fields (xsd_tag) 13

4 Other methods 14

4.1 Get TaxonomyVersion 14

4.2 Get ValueLists for Taxonomy Reference ..... **Error! Bookmark not defined.**

4.3 Get ValueList 15

4.4 GetValueLists 15

5 Working with attachments 17

5.1 Uploading attachments to a folder 17

5.2 Creating an occurrence with attachments 18

5.3 Adding attachments to an existing occurrence 19

6 Getting all ORs as an organisational user 20

7 Importing files 21

7.1 Create Original Reports from e5x files 21

7.2 Getting the results of a migrated e5x file 22

8 Annex 23

8.1 Reports JSON 23

8.1.1 Structure 23

8.1.2 Definition of attributes 24

8.1.3 Example 25

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 2 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**.

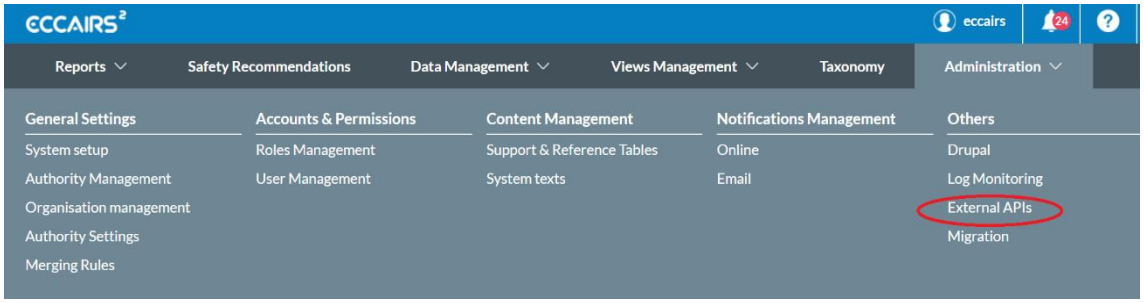
2 First time use

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.

2.1 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:

- 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.sandbox.aviationreporting.eu

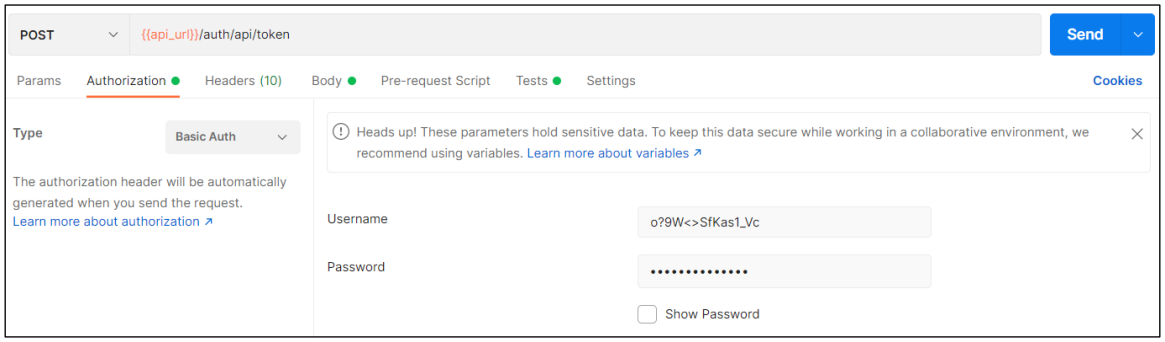
2.2 Step 1: Login as an elevated user

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

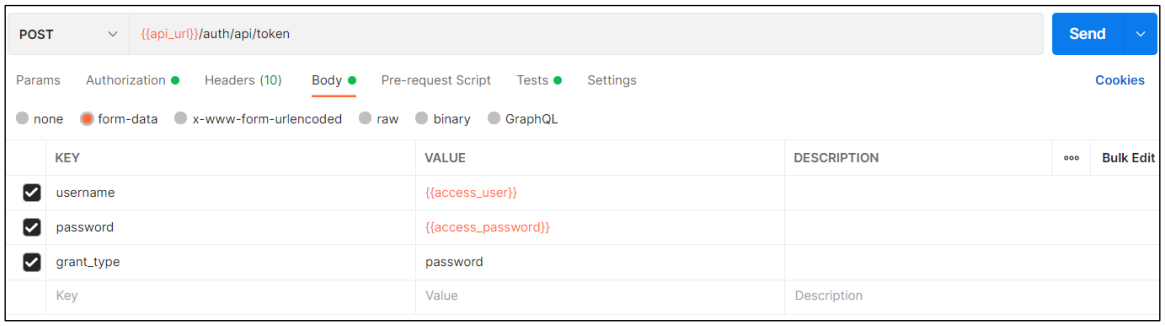
- Section: Login Section.
- Method URL: {BASE_URL}/auth/api/token
- Method Type: POST
- Authorization: Basic Auth
  - ✓ username: o?9W<>SfKas1_Vc
  - ✓ password: y]vfbuxrca#%sCksneQD/S:"MF^azq
- Body parameters:
  - ✓ Username: << national authority user with system administrator role >>
  - ✓ password: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

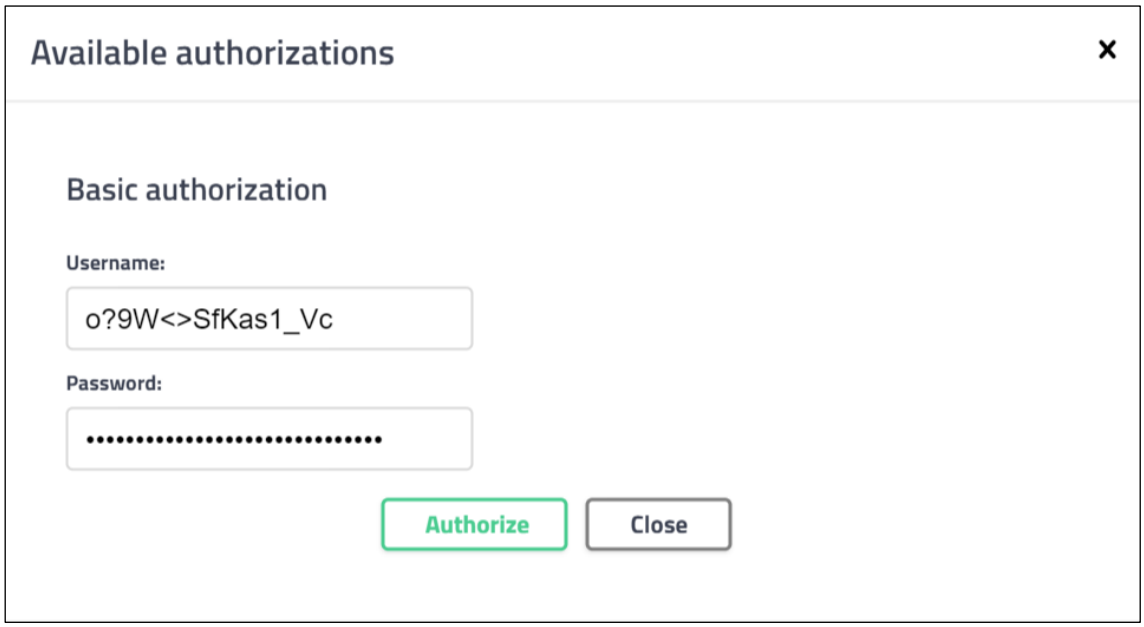


- Body:

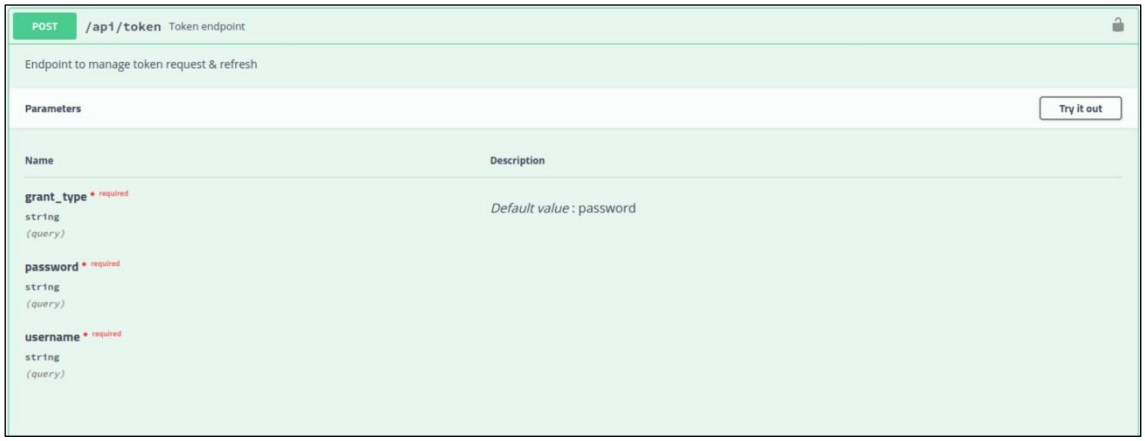


Example using **Swagger**

- Authorization:



✓ Parameters:



- ✓ Response:

{
"access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NzkyNjUzODE-
sInVzZXIiOnsiaWQiOjEsInVzZXJuYW1lIjoiZWNjYWlycyIsImxhbmd1YWdlIjp7ImlkIjozLCJuYW1lI-
joiRW5nbGlzaCIsImNvZGUiOiJlbiJ9LCJhdXRob3JpdHkiOnsiaWQiOjE-
sIm5hbWUiOiJFQVNBIn0sInJvbGVzIjpbeyJuYW1lIjoiRUNDQUlSUyBTeXN0ZW0gQWR-
taW5pc3RyYXRvciIsInBlcm1pc3Npb25zIjpbIkNyZWF0ZVF1ZXJ5IiwiUmVhZFF1ZXJ5Ii-
wiRWRpdFF1ZXJ5IiwiRGVsZXRlUXVlcnkiLCJFeGVjdXRlUXVlcnkiLCJTaGFyZVF1ZXJ5Ii-
wiQ3JlYXRlQmF0Y2hPcGVyYXRpb24iLCJSZWFkQmF0Y2hPcGVyYXRpb24iLCJFZGl0QmF0Y2hPcG
VyYXRpb24iLCJEZWxldGVCYXRjaE9wZXJhdGlvbiIsIkV4ZWN1dGVCYXR-
jaE9wZXJhdGlvbiIsIkNyZWF0ZVZpZXciLCJSZWFkVmlldyIsIkVkaXRWaWV3Ii-
wiRGVsZXRlVmlldyIsIkNyZWF0ZURhc2hib2FyZERpbGUiLCJSZWFkRGFzaGJvYXJkVGlsZSIsIkV-

2.3 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:

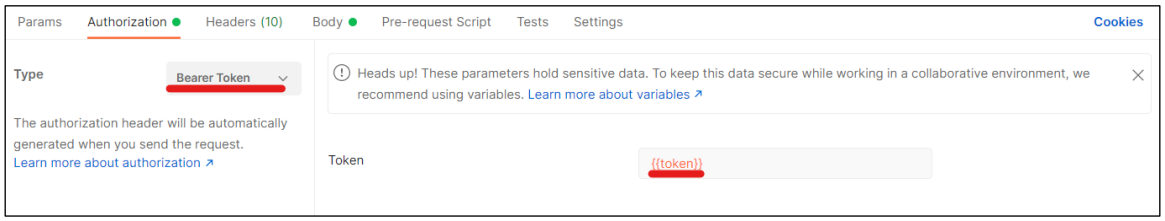
- Authorization:
  - ✓ Type: Bearer Token
  - ✓ Token (example from step 1):

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1NzkyNjUzODEsInVzZXIiOnsiaWQiOjEsInVzZXJuYW1lIjoiZWNjYWlycyIsI
mxhbmd1YWdlIjp7ImlkIjozLCJuYW1lIjoiRW5nbGlzaCIsImNvZGUiOiJlbiJ9LCJhdXRob3JpdHkiOnsiaWQiOjEsIm5hbWUiOiJFQV
NBIn0sInJvbGVzIjpbeyJuYW1lIjoiRUNDQUlSUyBTeXN0ZW0gQWRtaW5pc3RyYXRvciIsInBlcm1pc3Npb25zIjpbIkNyZWF0ZVF1
ZXJ5IiwiUmVhZFF1ZXJ5IiwiRWRpdFF1ZXJ5IiwiRGVsZXRlUXVlcnkiLCJFeGVjdXRlUXVlcnkiLCJTaGFyZVF1ZXJ5IiwiQ3JlYXRlQm
F0Y2hPcGVyYXRpb24iLCJSZWFkQmF0Y2hPcGVyYXRpb24iLCJFZGl0QmF0Y2hPcGVyYXRpb24iLCJEZWxldGVCYXRjaE9wZXJh

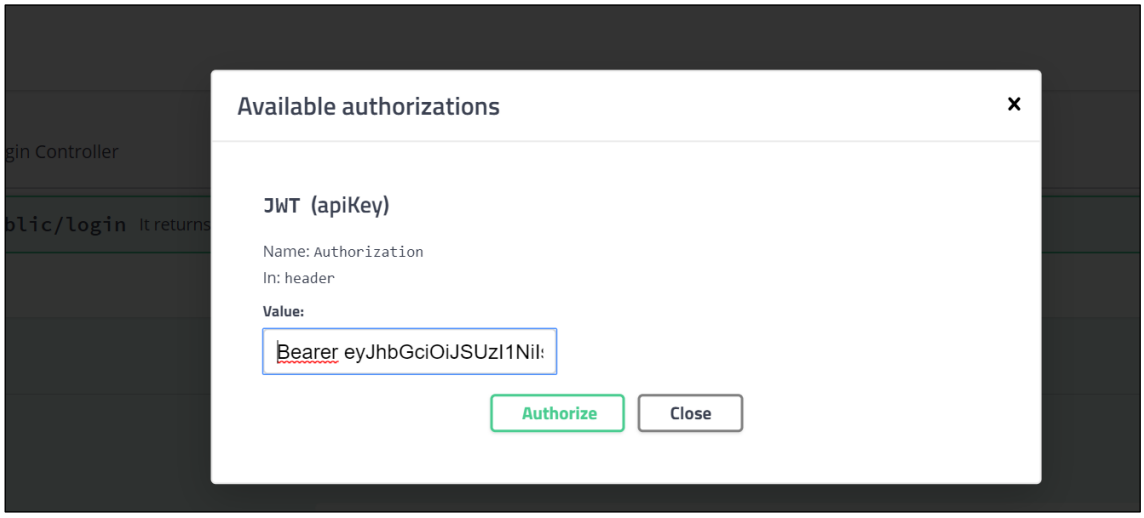
dGlvbiIsIkV4ZWN1dGVCYXRjaE9wZXJhdGlvbiIsIkNyZWF0ZVZpZXciLCJSZWFkVmlldyIsIkVkaXRWaWV3IiwiRGVsZXRlVmlldyIsIkNyZWF0ZURhc2hib2FyZFRpbGUiLCJSZWFkRGFzaGJvYXJkVGlsZSIsIkVkaXREYXNoYm9hcmRUaWxlIiwiRGVsZXRlRGFzaGJvYXJkVGlsZSIsIkNyZWF0ZVJvbGUiLCJSZWFkUm9sZSIsIkVkaXRSb2xlIiwiRGVsZXRlUm9sZSIsIk1hbmFnZUVDQ0FJUlNTZWN1cml0eUFkbWluaXN0cmF0b3IiLCJDcmVhdGVVc2VyIiwiUmVhZFVzZXIiLCJFZGl0VXNlciIsIkRlbGV0ZVVzZXIiLCJEZWxldGVVc2VyIiwiRGVsZXRlQ291bnRyeSIsIkVkaXRBdXRob3JpdHkiLCJEZWxldGVBdXRob3JpdHkiLCJSZWFkRWNjYWlyczJTZXR0aW5ncyIsIkVkaXRFY2NhaXJzMlNldHRpbmdzIiwiQ3JlYXRlTGFuZ3VhZ2UiLCJFZGl0TGFuZ3VhZ2UiLCJEZWxldGVMYW5ndWFnZSIsIkNyZWF0ZUNvdW50cnkiLCJSZWFkQ291bnRyeSIsIkVkaXRDb3VudHJ5IiwiRGVsZXRlQ291bnRyeSJdfSx7Im5hbWUiOiJFQ0NBSVJTIFRheG9ub215IEFkbWluaXN0cmF0b3IiLCJwZXJtaXNzaW9ucyI6WyJDcmVhdGVRdWVyeSIsIlUlYWRRdWVyeSIsIkVkaXRRdWVyeSIsIkRlbGV0ZVF1ZXJ5IiwiRXhlY3V0ZVF1ZXJ5IiwiU2hhcmVRdWVyeSIsIkNyZWF0ZVRheG9ub215IiwiRXhlY3V0ZVRheG9ub215IiwiU2hhcmVUYXhvbm9teVByb2plY3QiLCJFZGl0VGF4b25vbXlQcm9qZWN0IiwiRVgZWFzZVRheG9ub215UHJvamVjdCIsIlUlYWRUYXhvbm9teVByb2plY3QiLCJFeHBvcnRUYXhvbm9teVByb2plY3QiLCJEZWlkZW50aWZ5VGF4b25vbXlQcm9qZWN0IiwiUmVsZWFzZVRheG9ub215UHJvamVjdCIsIkNvbXBhcmVUYXhvbm9teVByb2plY3QiLCJFeHBvcnRUYXhvbm9teVByb2plY3QiLCJEZWlkZW50aWZ5VGF4b25vbXlQcm9qZWN0IiwiQ3JlYXRlTm90aWZpY2F0aW9uIiwiUmVhZE5vdGlmaWNhdGlvbiIsIkVkaXRPb3RpZmljYXRpb24iLCJEZWxldGVOb3RpZmljYXRpb24iLCJBY2NlcHRJbnZpdGUiLCJSZWFkTWFzdGVyVGFibGUiLCJSZWFkTWFzdGVyVGFibGUiLCJEZGl0TWFzdGVyVGFibGUiLCJEZWxldGVNYXN0ZXJUYWJsZSJdfV19LCJ1c2VyX25hbWUiOiJlY2NhaXJzIiwiYXV0aG9yaXRpZXMiOlsiRUNDQUlSUyBUYXhvbm9teSBBZG1pbmlzdHJhdG9yIiwiRUNDQUlSUyBTeXN0ZW0gQWRtaW5pc3RyYXRvciJdLCJjbGllbnRfaWQiOiJvPzlXPD5TZkthczFfVmMifQ.VQ5rFHjXGqm4KnAVd7DLGXIPzDgHBHnr5HUtmaB0CyguGGf-gTpU-S2MXiLWrGbXV8eeflBwGh_DDJkvSeA_ITk3r4Tjtfk_Y2xN4MdaC4WezhjXT58aLMgDzb1hCi6PJgefTQWbrlRbY-vEkQ6UbDuUENy7zTDZc09WcjjGD0NtMge9MNJTGowBtUzhpUGF7U2Y_cxTzBSfohusEMrjdQRDYmYvuGenZUTfXVMdoUrytRdwJPaaDYZUWij41ECWUWK3MVUzfOoWzmluPeKIv3j7JsFbsK4F0-9i9a_wGjG7FCkzuKnTyMVJ8x-jtYfh7HHQFQUc4MSPbF2_gkvPuw

Example using Postman:

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



Example using Swagger:



2.4 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 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.

3.1 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 8.1 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 2, 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": "",
```


3.2 Updating an occurrence

The input parameters are the following:

- e2id – ECCAIRS2 unique identifier
- taxonomyCodes - JSON with the report information. Please refer to section 8.1 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 2, 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
```


3.3 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 8.1 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 2, with the token obtained in STEP 7
- Body:

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

- Response:


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

3.4 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 2, 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
```


3.5 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 2, 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 Other methods

4.1 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
}
```

4.2 GetValueLists

Use this method to find all value lists given a taxonomy id

- Section: Taxonomy Management Section.
- Method URL: {BASE_URL}/taxonomy/valueLists/public/{{taxonomy_id}}
- Method Type: GET
- Authorization: Follow STEP 2 in section 2
- Parameter:
  - ✓ Taxonomy_id: id of the desired taxonomy (from 4.1)
- Body: empty
- Response:

```
{
  "data": {
    "responseData": {
      "page": 0,
      "pageSize": 448,
      "totalRows": 448,
      "responseStatus": "OK",
      "responseMessage": "",
      "taxonomyName": "ECCAIRS Aviation",
      "language": null,
      "responseCode": 200
    },
    "list": [
      {
        "id": 46381,
        "sourceName": null,
        "identifier": 100067,
        "description": "TestDiegoVL CustomSpain",
        "detailed": null,
        "explanation": "kkjl",
        "levels": 1,
        "lastIdentifier": null,
        "customId": 1,
        "isCustom": true,
        "locked": false,
```


4.3 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/listofvalue/valuelist/{{value_list_id}}/{{taxonomy_id}}
- Method Type: GET
- Authorization: Follow STEP 2 in section 2
- Parameter:
  - ✓ Value_list_id: internal id of the value list to retrieve obtained in the previous step.
  - ✓ Taxonomy_id: id of the desired taxonomy (from 4.1)
- Body: empty

Example:

{{BASE_URL}}/taxonomy-service/listofvalue/valuelist/46381/393

- Response:


```
{
    "data": [
        {
            "id": 11728665,
            "valueList": {
                "id": 46381,
                "description": "TestDiegoVL CustomSpain"
            },
            "sourceName": null,
            "isNewRecord": null,
            "identifier": 1,
            "parent": 0,
            "level": 1,
            "description": "TestDiegoVL CustomSpain2",
            "detailed": "TestDiegoVL CustomSpain2",
            "explanation": "TestDiegoVL CustomSpain2",
            "hasChild": false,
            "specialValue": {
                "id": 0,
                "name": "Normal"
            },
            "status": null,
            "version": null
```


5 Working with attachments

5.1 Uploading attachments to a folder

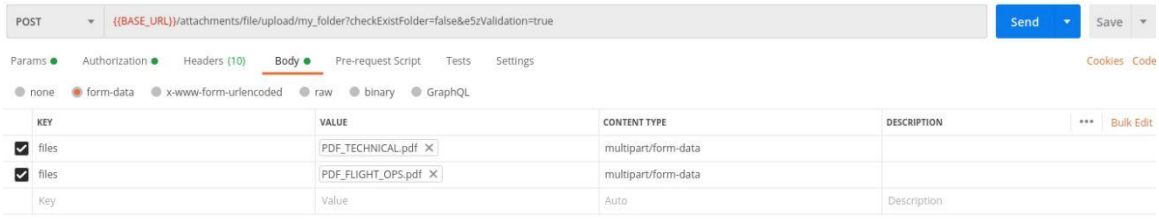
It allows to upload attachments to a folder in order to be referenced when creating an occurrence or updating an attribute (Datatype=ECCAIRS Resource Locator) with attachments.

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



- Response:

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

5.2 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 2, with the token obtained in STEP 7
- Body:

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

- Response:

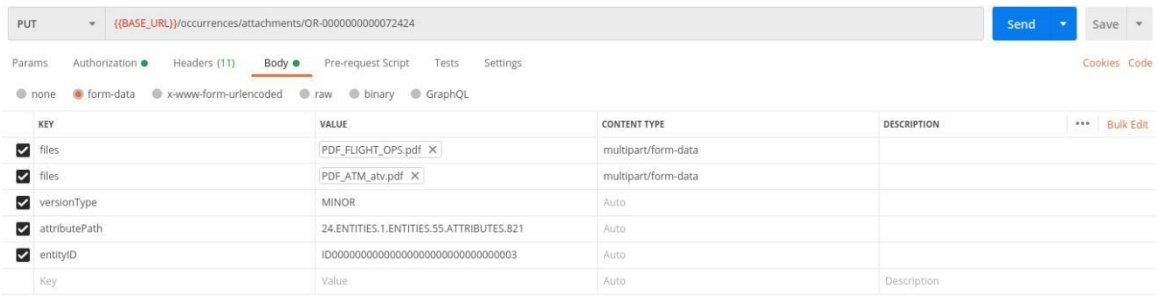
```
{
    "data": {
        "e2Id": "OR-0000000000072437",
        "status": "SENT",
```


5.3 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 2
- 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:



- Response:

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


6 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 2 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",
```


7 Importing files

7.1 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 2
- 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.

7.2 Getting the results of a migrated e5x file

It allows to obtain a CSV file with the results of a migration file. The CSV 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/getResultsCSV?idFile={file_id}
- Method Type: POST
- Authorization: Follow STEP 2 in section 2
- Parameters:
  - ✓ 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
```

8 Annex

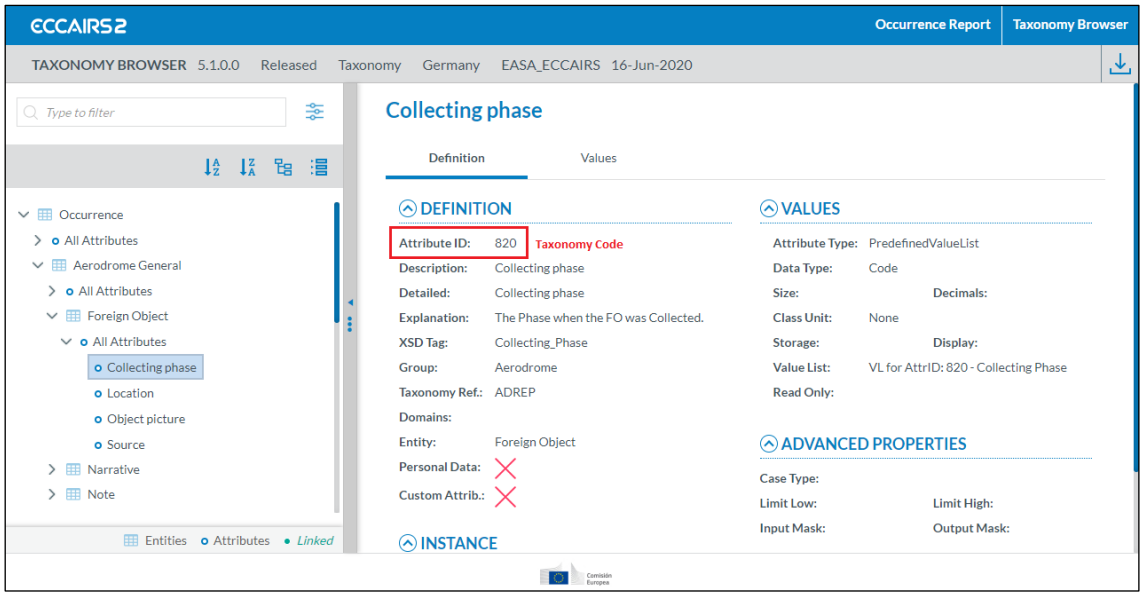
8.1 Reports JSON

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:
  - ✓ **Error! Reference source not found.**
  - ✓ **Error! Reference source not found.**
  - ✓ **Error! Reference source not found.**

8.1.1 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).



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

- In UAT here
- In PROD here

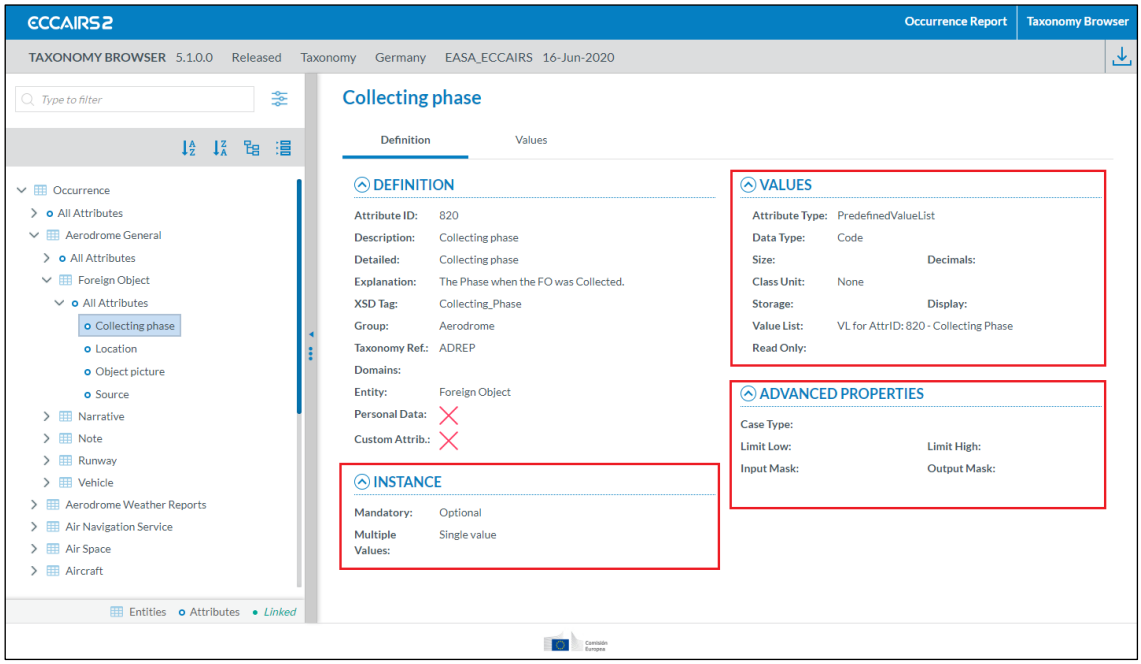
- And in the Sandbox environment here

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"**

8.1.2 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.



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.

8.1.3 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#"
            }
         ]
      }
```