ECCAIRS2 - SRIS2

ECCAIRS2

EASA - European Union Aviation Safety Agency

Title

E2 API Guide

This document provides a guide of use of the external API of ECCAIRS2

Date	30/09/2025
Version	*4.26*

EASA
European Union Aviation Safety Agency

1 Document Management

1.1 Terms and abbreviations

The following abbreviations have been used within this document:

Term/Abbreviation	Description

Table of Contents

2 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. 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.

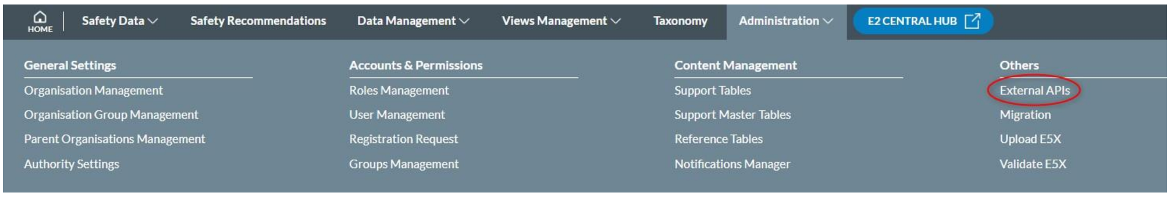
In order to execute API methods your IP needs to be whitelisted beforehand, otherwise you won't be able to generate a security token. **Please communicate with EASA so that these IPs can be whitelisted centrally by an administrator.**

3 What's new in this version

- API method to retrieve the e2ids of the Occurrences shared to the ECR resulting from the execution of a query stored in E2.
- API method to retrieve the list of queries and the type of library where they are saved:
  - 1: My Personal Libraries
  - 2: Local Common Libraries
  - 3: Shared with me Libraries

4 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. (Section 4)
- **User Management Section**: method to retrieve lists of users. (Section 7)
- **Taxonomy Management Section**: set of methods to interact with taxonomies and value lists. (Section 6)
- **Occurrence to Report Section**: group of methods focused on managing and consulting Reports and/or Occurrences. (Section 5)
- **Import from files**: group of methods to upload documents using files (e5x). (Section 9)

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

5 Login

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

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

5.1 Step 1: Logging in with an existing user

Method URL: {BASE_URL}**/idp/oauth2/token**

Method Type: POST

Content-Type:
application/x-www-form-urlencoded

This endpoint is used to log in with an existing M2M/API user. To access E2 API resources, first an access token must be obtained using the OAuth 2.1 Client Credentials Flow. This token acts as proof that your application/user is authorized.

The following parameters must be sent in the body **(application/x-www-form-urlencoded)**:

Parameter	Description	Example Value
grant_type	Specifies the OAuth 2.1 grant type. For client credentials, this must be **client_credentials**.	**client_credentials**
scope	**(Optional)** A space-separated list of scopes indicating the level of access your application is requesting. Only include scopes you need. If omitted, the default scopes configured for your client will be granted.	**openid**
client_id	Your unique application identifier, provided to you when you registered your client with our system. In other words, the **E2 username**.	**your-client-id-goes-here**

client_secret	The secret key associated with your client_id. In other words, the **E2 password**.	**your-client-secret-goes-here**

Response example:

```
{
     "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9
lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c",
     "scope": "openid",
     "token_type": "Bearer",
     "expires_in": 3599
}
```

Once the **access_token** expires e.g in 1 hour, which can be consulted in the token itself or in the **expires_in** response attribute, a new token will have to be requested.

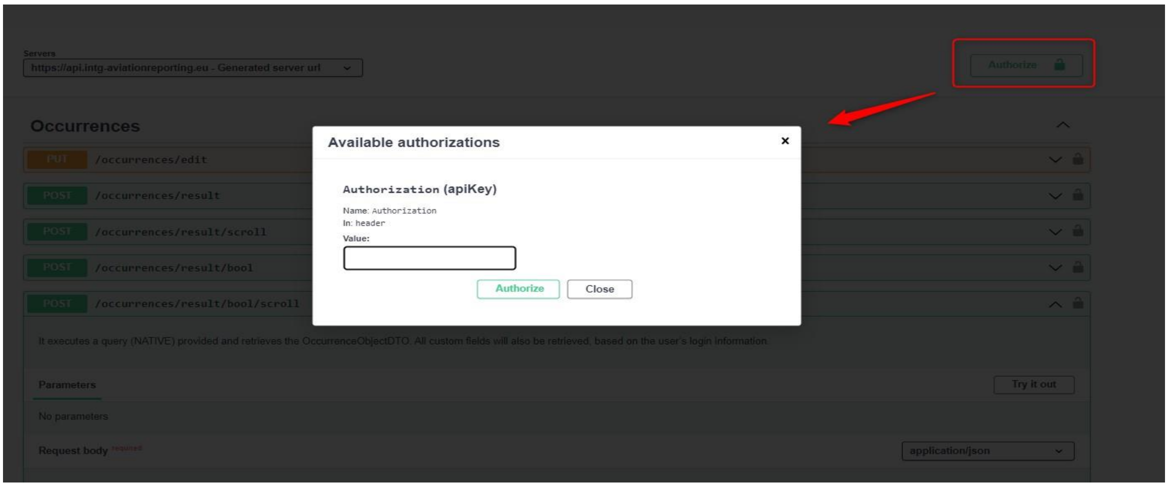
5.2 Step 2: Setting up the token

The **access_token** granted in the first step must be added to any subsequent calls to secure API endpoints. It must be sent in the **Authorization** request header as: Bearer + **access_token.**

Example header:

```
  Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

In Swagger:



Authorization (apiKey)

Name: Authorization

In: header

Value:

It executes a query (NATIVE) provided and retrieves the OccurrenceObjectDTO. All custom fields will also be retrieved, based on the user's login information.

No parameters

6 Original Reports to Occurrences

6.1 Occurrences

6.1.1 Creating an occurrence

Method URL: {BASE_URL}**/occurrences/create**

Method Type: POST

This method is used to create reports.

- The following information must be specified in the body:
  - ✓ type, with options:
    - ○ REPORT
    - ○ VALIDATED
    - ○ OCCURRENCE
  - ✓ taxonomyCodes – JSON with the report information. Please refer to section 11.4 for more information on the JSON structure
  - ✓ Optionally, you can include the reportingEntityId, that shows who reports the document
    - ○ It will be sent with the Reporting Entity value list code
  - ✓ status
    - ○ By default, it will be sent as Sent/Open.
    - ○ Optionally, if specified, it can be saved as Draft.
- Body example:

```
{
  "type":"VALIDATED",
  “reportingEntityId”:11,
  "taxonomyCodes":{
    "24":{
      "ID":"#id_number#",
      "ENTITIES":{
        "1":[
          {
```

- Response example:

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

6.1.2 Updating an occurrence

Method URL: {BASE_URL}**/occurrences/edit**

Method Type: PUT

- The following information must be specified in the body:
  - ✓ e2id – ECCAIRS2 unique identifier
  - ✓ taxonomyCodes – JSON with the report information.
    Please refer to section 11.4 for more information on the
    JSON structure
    - ○ Only the attributes that need to be changed should be included in the body.
    - ○ The attributes that are not included in the request body,
      will still be kept in the da- tabase.
    - ○ Further details on deleting in section 6.1.2.1.
  - ✓ versionType, which determines how to save the report, with options
    - ○ DRAFT (for OR/VR/OC)

  - MAJOR (for OC)
  - MINOR (for OR/VR/OC)

- Body example:

```
{
    "e2Id": "OR-0000000000044406",
    "versionType": "MINOR",
    "taxonomyCodes": {
        "24": {
            "ENTITIES": {
                "14": [
                    {
                        "ID": "#id_number#",
                        "ATTRIBUTES": {
                            "390": [
                                2050301
                            ]
                        }
                    },
                    {
                        "ID": "#id_number#",
                        "ATTRIBUTES": {
                            "390": [
                                2180301
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- Response example:

```
{
    "data": {
        "e2Id": "OR-0000000000044406",
        "status": "SENT",
        "version": "0.4"
    },
    "errorDetails": "",
    "returnCode": 1
```

6.1.2.1 Updating an occurrence - Deleting entities

To delete an item in a report, the method to user and input parameters are described in section 0.

As explained above, the attributes that are not included in the request body, will still be kept in the database. To delete an item, it must be stated with the **#unset#** parameter (in lowercase).

In can be done in for attributes individually and in bulk for attributes, entities and links. Please look at the body in the example for details.

- Body example:

```
{
    "e2Id": "OC-0000000003292523",
    "versionType": "MINOR",
    "taxonomyCodes": {
       "24": {
            "ATTRIBUTES": {
               "440": "#unset#",
               "451": "#unset#"

            },
             "ENTITIES": {
                "4": [
                    {
                        "ID":"ID00000000000000000000000000000002",
                        "ATTRIBUTES":"#unset#",
                        "ENTITIES":"#unset#",
                        "LINKS":"#unset#"
                    }
                ]
            }
       }
    }
 }
```

- Response example:

```
{
    "data": {
        "e2Id": "OC-0000000003292523",
        "status": "OPEN",
        "version": "0.2"
    },
    "errorDetails": "",
    "returnCode": 1
}
```

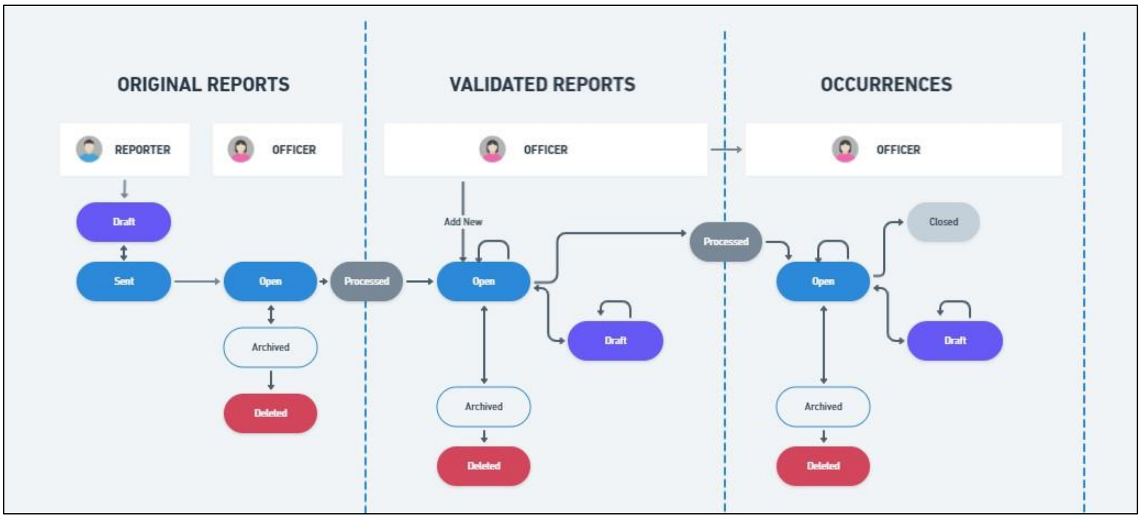
6.1.3 Changing an occurrence status

Method URL: {BASE_URL}/occurrences/change-status

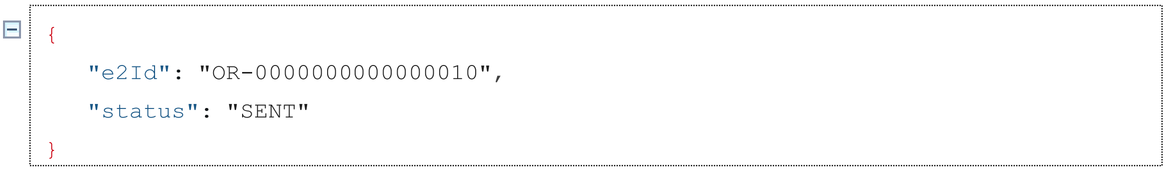
Method Type: POST

- The following information has to be specified in the body:
  - ✓ 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.

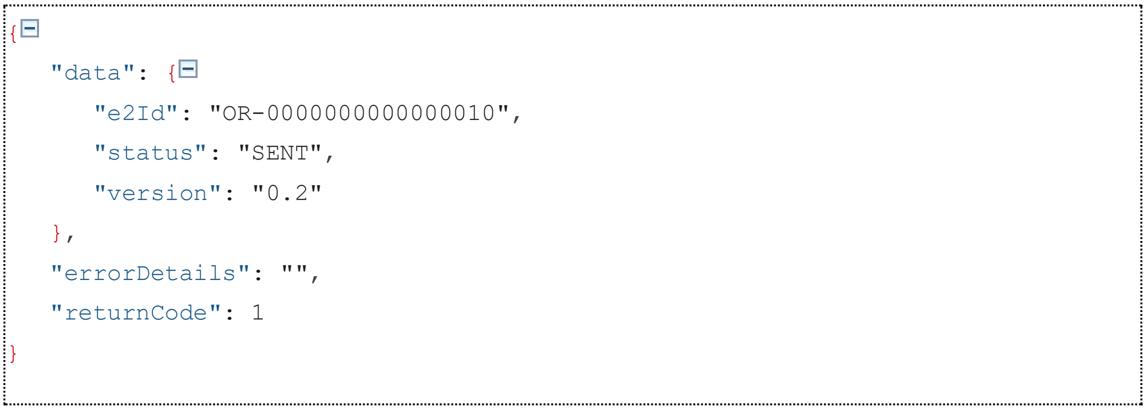


- Body example:



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

- Response example:



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

6.1.4 Delete the draft version of an Occurrence

Method URL: {BASE_URL}**/occurrences/OC/{e2Id}**

Method Type: DELETE

- The input parameter is the following:

  - ✓ e2id: unique identifier of the Occurrence.

- Body should always be empty.
- Response example:

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

6.1.5 Delete the draft version of an Original Report

Method URL: {BASE_URL}**/occurrences/OR/{e2Id}**

Method Type: DELETE

- The input parameter is the following:
  - ✓ e2id: unique identifier of the Original Report.
- Body should always be empty.
- Response example:

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

6.1.6 Delete the draft version of a Validated Report

Method URL: {BASE_URL}**/occurrences/VR/{e2Id}**

Method Type: DELETE

- The input parameter is the following:
  - ✓ e2id: unique identifier of the Validated Report.
- Body should always be empty.

- Response example:

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

6.1.7 Check if an Occurrence exists

Method URL: {BASE_URL}/occurrences/exists/{e2Id}

Method Type: GET

- The input parameter is the following:
  - ✓ e2id: unique identifier of the occurrence.
- Body should always be empty.
- Response example:

```
{
    "data": True,
    "errorDetails": "string",
    "returnCode": 0
```

If the response is TRUE, the report exists, if it is FALSE, it does not.

6.1.8 Getting an occurrence URL

Method URL: {BASE_URL}/occurrences/get-URL/{e2Id}

Method Type: GET

- The input parameter is the following:
  - ✓ e2id: unique identifier of the occurrence.
- Body should always be empty.
- Response example:

```
{
  "data": {
    "e2Id": "OR-0000000000044484",
    "version": "0.6",
    "url": "https://e2.aviationreporting.eu/report-
management/detail?encode=U2FsdGVkX1/nQas/oOCBMNfneAglNnFsOGMnR832gp88/1Y
X2tpP1PD7UcN1XfOdqu/OAbjL+/ygryFtYn03Ukc/7vPw4mZHqqgFXs1oWwq70WKhJai8XE5
uXV8lGB1ziFaMAYQ9HW+aVVb/phWdzRAsKku9w0rKi9Nz5EEXfFqVrUO+XVVj/2l4aPTJrrpyc
zwM9l23jnX99/uqF+AWCvD3MG3BLWVmwaWlkxCnQHA="
  },
  "errorDetails": "",
  "returnCode": 1
}
```

6.1.9 Getting an occurrence (Taxonomy codes)

Method URL: {BASE_URL}/occurrences/get/{e2id}

Method Type: GET

This method retrieves all the attributes saved in a report that the user has access to, depending on the views they have assigned to their role.

- The input parameters are the following:
  - ✓ e2id: unique identifier of the occurrence.
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.
- Body should always be empty.
- Response example:

```
{
   "data": {
      "e2Id": "OR-0000000000044484",
      "version": "0.4",
      "type": "OR",
      "status": "SENT",
      "responsibleEntityId": 29,
      "reportingEntityId": 122,
      "pdfCode": null,
      "relatedReports": null,
      "rowsTaxCodes": {
         "24": {
            "ATTRIBUTES": {
               "1088": [
                  9
               ],
               "1087": [
                  {
                     "text": "string"
                  }
               ],
               "470": [
                  9999
               ],
               "471": [
                  9999
               ],
               "472": [
                  9999
               ],
               "275": [
                  99
               ],
               "176": [
                  {
                     "unit": "kt",
                     "content": "1"
                  }
               ],
               "451": [
                  98
               ],
               "430": [
                  106,
                  5
               ],
               "310": [
                  {
                     "unit": "m",
                     "content": "1"
                  }
               ],
               "431": [
                  502
               ],
               "453": [
                  2165
```

6.1.10 Getting an occurrence by fields (Taxonomy codes)

Method URL: {BASE_URL}/occurrences/fields/{e2id}

Method Type: POST

This method retrieves the attributes specified in the request body. Only the attributes that the user has access to, depending on the views they have assigned to their role, will be retrieved.

- 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.
- The input parameter is the following:
  - ✓ e2id: unique identifier of the occurrence.
- Body example:

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

- Response example:

```
{
    "data": {
        "e2Id": "OR-0000000000044484",
        "version": "0.4",
        "type": "OR",
        "status": "SENT",
        "responsibleEntityId": 29,
        "reportingEntityId": 122,
        "pdfCode": null,
        "relatedReports": null,
        "rowsTaxCodes": {
            "24": {
                "ATTRIBUTES": {
                    "453": [
                        2165
                    ]
                },
                "ENTITIES": {
                    "53": [
                        {
                            "ATTRIBUTES": {
                                "447": [
                                    {
                                        "additionalText": "TEstring",
                                        "content": [
                                            122
                                        ]
                                    }
                                ],
                                "802": [
                                    {
                                        "path": "string",
                                        "fileName": "FileName-1",
                                        "description": "string"
                                    }
                                ]
                            }
                        }
                    ]
                }
            },
            "taxonomy_version_id": 320,
            "general_version_id": 327
        },
        "rowsLanguage": {
            "Occurrence": {
                "ATTRIBUTES": {
                    "Responsible_Entity": [
                        "Spain > CAA"
                    ]
                },
```

6.1.11 Get Query Result (Taxonomy codes)

Method URL: {BASE_URL}**/occurrences/result**

Method Type: POST

This method executes the queries saved into the ECCAIRS 2 database. It can be used to retrieve less than 10000 documents. They can be retrieved all in the same request or in multiple requests.

- The input parameter is the following:
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
  - ✓ type, with options:
    - ○ REPORT
    - ○ VALIDATED
    - ○ OCCURRENCE
  - ✓ queryName,
  - ✓ libraryName - the library where it is saved in E2 are specified in the body.
  - ✓ pageSize and page
    - ○ The maximum pageSize supported is 10000.
    - ○ The first “page” is the “page” 0.

- Body example:

```
{
  "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,
            "pdfCode": null,
            "relatedReports": null,
            "rowsTaxCodes": {
                "24": {
                    "ATTRIBUTES": {
                        "709": [
                            "2018-11-22"
                        ],
                        "451": [
                            98
                        ],
                        "440": [
                            "Kongsted (EKKS)"
                        ],
                        "452": [
                            "2018-471"
                        ],
                        "453": [
                            1047
                        ],
                        "432": [
                            2
                        ],
                        "454": [
                            {
                                "content": [
                                    67
                                ]
                            }
                        ],
                        "433": [
                            "2018-09-09"
                        ],
                        "455": [
                            3
                        ],
                        "477": [
                            "2018-09-09"
                        ],
                        "456": [
                            2
                        ],
```

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

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

Method URL: {BASE_URL}/occurrences/result/scroll

Method Type: POST

This method executes the queries saved into the ECCAIRS 2 database, with the scroll option. It can be used to retrieve more than 10000 documents. They can be retrieved all in the same request or in multiple requests (when over 10000 documents).

- The input parameter is the following:
    - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.
- The following information must be specified in the body
    - ✓ type, with options:
        - ○ REPORT
        - ○ VALIDATED
        - ○ OCCURRENCE
    - ✓ queryName,
    - ✓ libraryName - the library where it is saved in E2 are specified in the body.
    - ✓ pageSize
        - ○ The maximum pageSize supported is 10000.

    - 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.

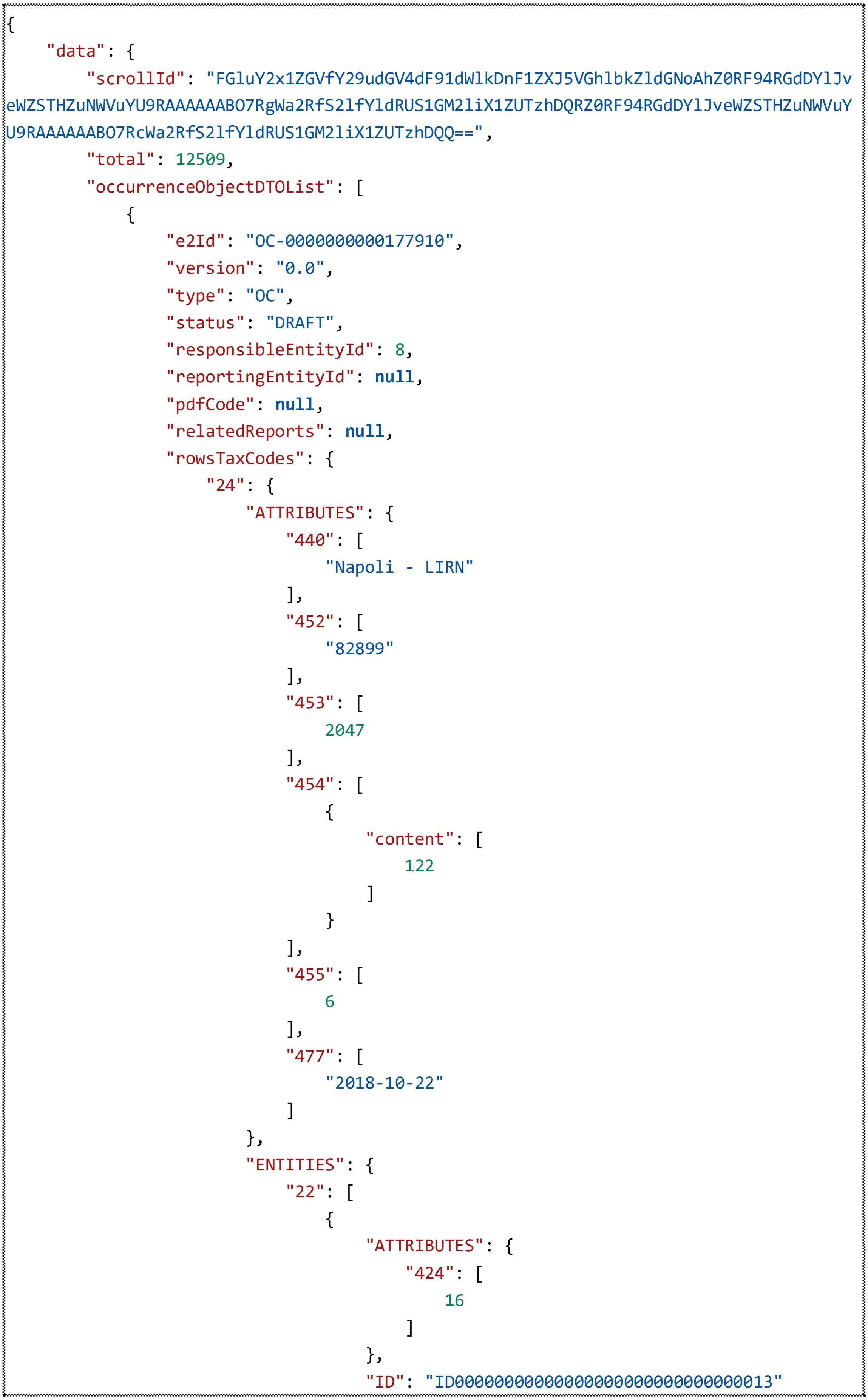
  - ✓ scrollId – It should always be null in the first request.

- First Request

  - ✓ Body example:

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

  - ✓ Response example:



```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO7RgWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABO7RcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
                "status": "DRAFT",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Napoli - LIRN"
                            ],
                            "452": [
                                "82899"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        122
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-22"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID00000000000000000000000000000013"
```

- Subsequent Requests

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

  The total number of records that meets the query condition is indicated in the “total” field.

  - ✓ Body example:

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

  - ✓ Response:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO7RgWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABO7RcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "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": "ID000000000000000000000000000015017"
```

6.1.13 Get Query ResultByQuery (Taxonomy codes)

Method URL: {BASE_URL}/occurrences/result/bool

Method Type: POST

This method executes the query specified in the body of the request. It can be used to retrieve less than 10000 documents. They can be retrieved all in the same request or in multiple requests.

- The input parameter is the following:
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
  - ✓ type, with options:

    - REPORT
    - VALIDATED
    - OCCURRENCE
  - ✓ queryJson
  - ✓ pageSize and page
    - The maximum pageSize supported is 10000.
    - The first “page” is the “page” 0.

- Body example:

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

- Response example:

```
{
  "data": [
       {
           "e2Id": "OC-0000000000177910",
           "version": "0.0",
           "type": "OC",
           "status": "DRAFT",
           "responsibleEntityId": 8,
           "reportingEntityId": null,
           "pdfCode": null,
           "relatedReports": null,
           "rowsTaxCodes": {
               "24": {
                   "ATTRIBUTES": {
                       "440": [
                           "Napoli - LIRN"
                       ],
                       "452": [
                           "82899"
                       ],
                       "453": [
                           2047
                       ],
                       "454": [
                           {
                               "content": [
                                   122
                               ]
                           }
                       ],
                       "455": [
                           6
                       ],
                       "477": [
                           "2018-10-22"
                       ]
                   },
                   "ENTITIES": {
                       "22": [
                           {
                               "ATTRIBUTES": {
                                   "424": [
                                       16
                                   ]
                               },
                               "ID": "ID0000000000000000000000000000013"
                           }
                       ],
                       "1": [
                           {
                               "ATTRIBUTES": {
```

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

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

Method URL: {BASE_URL}/occurrences/result/bool/scroll

Method Type: POST

This method executes the query specified in the body of the request, with the scroll option. It can be used to retrieve more than 10000 documents. They can be retrieved all in the same request or in multiple requests (when over 10000 documents).

- The input parameter is the following:
    - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
    - ✓ type, with options:
        - REPORT
        - VALIDATED
        - OCCURRENCE
    - ✓ queryJson
    - ✓ pageSize
        - The maximum pageSize supported is 10000.
        - 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.

- First Request
    - ✓ Body example:

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

    - ✓ Response example:

```
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZLZHM3TTl6WVRu
R01OX2RFYy1CdmtnAAAAAABX1ZsWdks2TEE3YWlRemU5YjU3aHlPUmtKZxZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABKc10Wa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
                "status": "DRAFT",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Napoli - LIRN"
                            ],
                            "452": [
                                "82899"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        122
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-22"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID000000000000000000000000000000013"
                                }
```

- Subsequent Requests

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

The total number of records that meets the query condition is indicated in the "total" field.

✓ Body example:

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

✓ Response example:

```
{    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZLZHM3TTl6WVRu
R01OX2RFYy1CdmtnAAAAAABX1ZsWdks2TEE3YWlRemU5YjU3aHlPUmtKZxZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABKc10Wa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000178279",
                "version": "0.1",
                "type": "OC",
                "status": "OPEN",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Unknown - ZZZZ"
                            ],
                            "452": [
                                "82390"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        999
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-08"
                            ],
                            "478": [
                                "09:39:00"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
```

6.1.15 Get Occurrences e2ids from ECR

Method URL: {BASE_URL}/occurrences/e2ids/query/{id}

Method Type: GET

This method retrieves the e2ids of the Occurrences shared to ECR resulting from the execution of the given query.

- The input parameter is the following:
  - ✓ id: unique identifier of the query.
  - ✓ Type: OCCURRENCE
- Body should always be empty.
- Response example:

```
{
    "data": [
        "OC-0000000003162950",
        "OC-0000000004692580",
        "OC-0000000004721293",
        "OC-0000000004607136",
        "OC-0000000004759065",
        "OC-0000000004883794",
        "OC-0000000004044004",
        "OC-0000000003672004",
        "OC-0000000004042085",
        "OC-0000000006160311",
        "OC-0000000006142564",
        "OC-0000000005337459",
        "OC-0000000005426255",
        "OC-0000000004862020",
        "OC-0000000006979769",
        "OC-0000000006068125",
        "OC-0000000004864386",
        "OC-0000000006260225",
        "OC-0000000005965762",
        "OC-0000000004861989",
        "OC-0000000006991645",
        "OC-0000000004768141",
        "OC-0000000005433450",
        "OC-0000000005792946",
        "OC-0000000005776934",
        "OC-0000000004569877",
        "OC-0000000004605710",
        "OC-0000000004613740",
        "OC-0000000004414634",
        "OC-0000000007885375",
        "OC-0000000007575281",
        "OC-0000000006959113",
        "OC-0000000007797889",
        "OC-0000000005763729",
        "OC-0000000005756334",
        "OC-0000000005690121",
        "OC-0000000005803124",
        "OC-0000000005921378",
        "OC-0000000007741905",
        "OC-0000000007888232",
        "OC-0000000004173673",
        "OC-0000000003995843",
        "OC-0000000003939874",
        "OC-0000000005752098",
        "OC-0000000005654925",
        "OC-0000000005678019",
        "OC-0000000004016078",
        "OC-0000000005411318",
        "OC-0000000004183108",
```

6.2 Attachments Occurrence Controller

6.2.1 Adding attachments to an existing occurrence

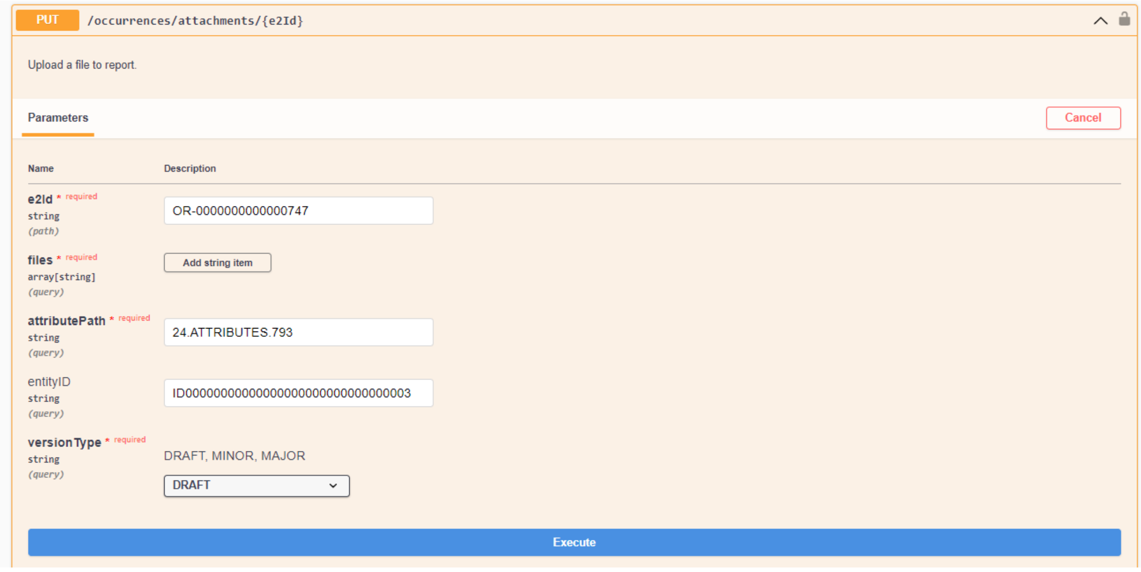
Method URL: {BASE_URL}/occurrences/attachments/{e2id}

Method Type: PUT

This method allows to upload attachments of an attribute of an occurrence.

- The input parameter is the following:
  - ✓ e2id: unique identifier of the occurrence.
- The following information must be specified in the body:
  - ✓ 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 at- tachments to be uploaded. All files will be saved into the same attribute.
    - Supported Media Types: multipart/form-data
    - Type: File
- Body example

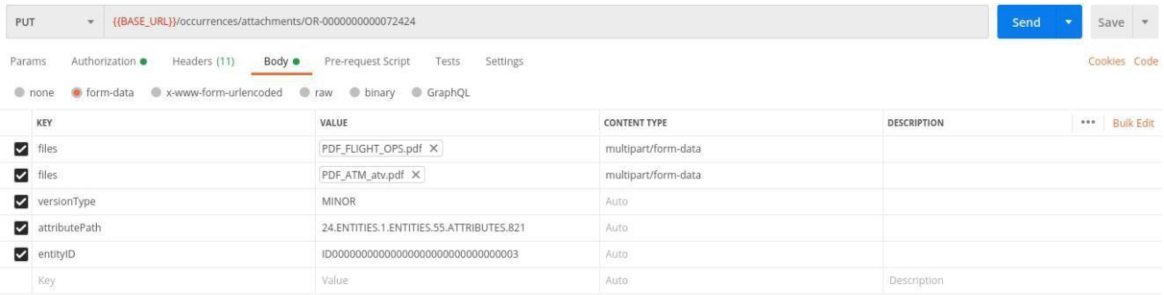
Swagger is configured in such a way so that the user can include the data as parameters and also update the file with a browser.



PUT /occurrences/attachments/{e2Id}

Upload a file to report.

Name	Description
e2Id * required string (path)	OR-0000000000000747
files * required array[string] (query)	Add string item
attributePath * required string (query)	24.ATTRIBUTES.793
entityID string (query)	ID00000000000000000000000000000003
versionType * required string (query)	DRAFT, MINOR, MAJOR DRAFT



PUT {{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",
```

6.2.2 Downloading attachments

Method URL: {BASE_URL}/occurrences/attachments/{e2id}

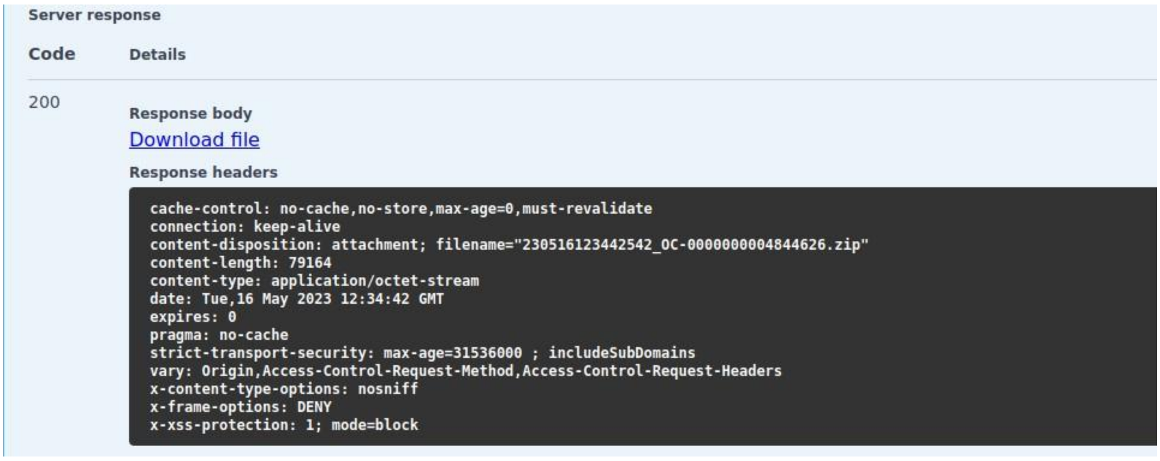
Method Type: GET

This method allows to download all the attachments of an occurrence.

- The input parameters are the following:

- ✓ e2id: unique identifier of the occurrence.

- Body should always be empty.
- Response:
  - ✓ The attached files will be included in a .zip file that will include all the occurrence attachments). The name of the zip file will be: timestamp + e2id



Server response

Code	Details
200	Response body Download file Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-disposition: attachment; filename="230516123442542_OC-0000000004844626.zip"
content-length: 79164
content-type: application/octet-stream
date: Tue,16 May 2023 12:34:42 GMT
expires: 0
pragma: no-cache
strict-transport-security: max-age=31536000 ; includeSubDomains
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

6.3 Occurrences Filtered Controller

6.3.1 Getting all ORs as an organisational user

Method URL: {BASE_URL}/occurrences/reporter/original-report ?&$skip={skip_length}&$top={top_length}

Method Type: POST

This method allows a reporter to obtain the list of Original Reports in E2 from a given organisation. The list of Original Reports can be retrieved in pages.

- The input parameters are the following:
  - ✓ Skip: the number of records to skip.
  - ✓ Top: the number of records to get in the response.
- The following information must be specified in the body
  - ✓ Status, with options

  - DRAFT,
  - SENT,
  - PROCESSED,
  - ARCHIVED,
  - DELETED.

- Body example:

```
{
    "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",
                "reportStatus": "PROCESSED",
                "translatedStatus": null,
                "creationDate": "2022-03-15 08:01:03",
                "responsibleEntity": "Netherlands > CAA",
                "responsibleEntityId": 29,
                "reportedByMe": true,
                "modificationUser": "MDCSpainOrgAdmin",
                "pdfCode": 6,
                "blocked": false
            },
            {
                "e2Id": "OR-0000000000044439",
                "version": "0.3",
                "reportStatus": "PROCESSED",
                "translatedStatus": null,
                "creationDate": "2022-03-15 08:01:04",
                "responsibleEntity": "Germany > BFU",
                "responsibleEntityId": 29,
                "reportedByMe": true,
                "modificationUser": "MDCSpainOrgAdmin",
                "pdfCode": 6,
                "blocked": false
            },
            {
                "e2Id": "OR-0000000000044434",
                "version": "0.3",
                "reportStatus": "PROCESSED",
                "translatedStatus": null,
                "creationDate": "2022-03-15 08:01:04",
                "responsibleEntity": "Spain > CAA",
                "responsibleEntityId": 29,
                "reportedByMe": true,
```

6.4 Readable Occurrences

These methods provide the same functionality as some of the aforementioned ones, with the difference that in the result it shows:

- rowsTaxCodes – where the information is shown with its codes
- rowsLanguage – where the description of the attributes is shown instead.

6.4.1 Getting an occurrence (xsd_tag)

*THIS IS THE READABLE VERSION OF THE METHOD SPECIFIED IN SECTION **6.1.9 GETTING AN OCCURRENCE (TAXONOMY CODES)***

Method URL: {BASE_URL}/occurrences/readable/{e2id}

Method Type: GET

- The input parameters are the following:
    - ✓ e2id: unique identifier of the occurrence.
    - ✓ singleValue (true/false):
        - ○ It retrieves either full path to values (false) or just the value (true), in case there are levels.
        - ○ False by default
    - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.
        - ○
- Body should always be empty.
- Response example:

```
{
   "data": {
      "e2Id": "OR-0000000000044484",
      "version": "0.4",
      "type": "OR",
      "status": "SENT",
      "responsibleEntityId": 29,
      "reportingEntityId": 122,
      "pdfCode": null,
      "relatedReports": null,
      "rowsTaxCodes": {
         "24": {
            "ATTRIBUTES": {
               "1088": [
                  9
               ],
               "1087": [
                  {
                     "text": "string"
                  }
               ],
               "470": [
                  9999
               ],
               "471": [
                  9999
               ],
               "472": [
                  9999
               ],
               "275": [
                  99
               ],
               "176": [
                  {
                     "unit": "kt",
                     "content": "1"
                  }
               ],
               "451": [
                  98
               ],
               "430": [
                  106,
                  5
               ],
               "310": [
                  {
                     "unit": "m",
                     "content": "1"
                  }
               ],
               "431": [
                  502
               ],
               "453": [
                  2165
```

6.4.2 Getting an occurrence by fields (xsd_tag)

THIS IS THE READABLE VERSION OF THE METHOD SPECIFIED IN SECTION ***206.1.10 GETTING AN OCCURRENCE BY FIELDS (TAXONOMY CODES)***

Method URL: {BASE_URL}/occurrences/readable/fields/{e2id}

Method Type: POST

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.

- The input parameter is the following:
  - e2id (unique identifier of the occurrence)
  - singleValue (true/false):
    - It retrieves either full path to values (false) or just the value (true), in case there are levels.
    - False by default

- Body example:

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

- Response example:

```
{
    "data": {
        "e2Id": "OR-0000000000000635",
        "version": "0.1",
        "type": "OR",
        "status": "PROCESSED",
        "responsibleEntityId": 3,
        "reportingEntityId": 122,
        "pdfCode": null,
        "relatedReports": [
            "VR-0000000000000313",
            "OC-0000000000081593"
        ],
        "rowsTaxCodes": {
            "24": {
                "ATTRIBUTES": {
                    "452": [
                        "236"
                    ],
                    "453": [
                        2018
                    ]
                }
            },
            "taxonomy_version_id": 57,
            "general_version_id": 1
        },
        "rowsLanguage": {
            "Occurrence": {
                "ATTRIBUTES": {
                    "File_Number": [
                        "236"
                    ],
                    "Responsible_Entity": [
                        "CAA"
                    ]
                }
            }
        },
        "updated": {
            "None": []
        },
        "blocked": false,
        "creationUser": "Belgium_Katalin",
        "creationDate": "11/02/2021 10:13:05",
        "lastModificationUser": "Belgium_Katalin",
        "lastModificationDate": "01/03/2021 12:47:46",
        "sharedEntities": {
            "sharedEntitiesId": [],
            "sharedAttachmentsEntitiesId": [],
            "deidentifiedTaxonomyCodes": []
```

6.4.3 Get Query Result (xsd_tags)

*THIS IS THE READABLE VERSION OF THE METHOD SPECIFIED IN SECTION **6.1.11 GET QUERY RESULT (TAXONOMY CODES)***

Method URL: {BASE_URL}/occurrences/readable/result

Method Type: POST

This method executes the queries saved into the ECCAIRS 2 database. It can be used to retrieve less than 10000 documents. They can be retrieved all in the same request or in multiple requests.

- The input parameters are the following:
    - single-value (true/false): retrieve either full path to values (false) or just the value (true)
    - excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
    - type, with options:
        - REPORT
        - VALIDATED
        - OCCURRENCE
    - queryName,
    - libraryName - the library where it is saved in E2 are specified in the body.
    - pageSize and page
        - The maximum pageSize supported is 10000.
        - The first “page” is the “page” 0.
- Body example:

```
{
  "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,
            "pdfCode": null,
            "relatedReports": null,
            "rowsTaxCodes": {
                "24": {
                    "ATTRIBUTES": {
                        "440": [
                            "Kristiansund - ENKB"
                        ],
                        "452": [
                            "88183"
                        ],
                        "453": [
                            2047
                        ],
                        "454": [
                            {
                                "content": [
                                    179
                                ]
                            }
                        ],
                        "455": [
                            6
                        ],
                        "477": [
                            "2019-07-31"
                        ]
                    },
                    "ENTITIES": {
                        "22": [
                            {
                                "ATTRIBUTES": {
                                    "424": [
                                        16
                                    ]
                                },
                                "ID": "ID000000000000000000000000000042165"
                            }
                        ],
                        "1": [
                            {
                                "ATTRIBUTES": {
```

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

6.4.4 Get Query Result (xsd_tags) with scroll-id

*THIS IS THE READABLE VERSION OF THE METHOD SPECIFIED IN SECTION **6.1.12 GET QUERY RESULT (TAXONOMY CODES) WITH SCROLL-ID***

Method URL: {BASE_URL}/occurrences/readable/result/scroll

Method Type: POST

This method executes the queries saved into the ECCAIRS 2 database, with the scroll option. It can be used to retrieve less than 10000 documents. They can be retrieved all in the same request or in multiple requests.

- The input parameters are the following:
  - ✓ single-value (true/false): retrieve either full path to values (false) or just the value (true)
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
  - ✓ type, with options:
    - o REPORT
    - o VALIDATED
    - o OCCURRENCE
  - ✓ queryName,
  - ✓ libraryName - the library where it is saved in E2 are specified in the body.
  - ✓ pageSize
    - o The maximum pageSize supported is 10000.
    - o 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.

- First Request

  ✓ Body example:

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

  ✓ Response example:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZFFoZTBTMTM4QmFSU
TJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
                "status": "DRAFT",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Napoli - LIRN"
                            ],
                            "452": [
                                "82899"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        122
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-22"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID000000000000000000000000000000013"
```

- Subsequent Requests

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

The total number of records that meets the query condition is indicated in the “total” field.

- ✓ Body example:

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

- ✓ Response example:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZFFoZTBTMTM4QmFSU
TJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000185518",
                "version": "0.1",
                "type": "OC",
                "status": "OPEN",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Unknown - ZZZZ"
                            ],
                            "452": [
                                "84389"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        999
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2019-01-15"
                            ],
                            "478": [
                                "03:00:00"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
```

6.4.5 Get Query ResultByQuery (xsd_tags)

THIS IS THE READABLE VERSION OF THE METHOD SPECIFIED IN SECTION ***6.1.13 GET QUERY RESULTBYQUERY (TAXONOMY CODES)***

Method URL: {BASE_URL}/occurrences/readable/result/bool

Method Type: POST

This method executes the query specified in the body of the request. It can be used to retrieve less than 10000 documents. They can be retrieved all in the same request or in multiple requests.

- The input parameters are the following:
  - ✓ single-value (true/false): retrieve either full path to values (false) or just the value (true)
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
  - ✓ type, with options:
    - ○ REPORT
    - ○ VALIDATED
    - ○ OCCURRENCE
  - ✓ queryJson
  - ✓ pageSize and page
    - ○ The maximum pageSize supported is 10000.

  - The first “page” is the “page” 0.

- Body example:

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

- Response example:

```
{
 "data": [
        {
            "e2Id": "OC-0000000000177910",
            "version": "0.0",
            "type": "OC",
            "status": "DRAFT",
            "responsibleEntityId": 8,
            "reportingEntityId": null,
            "pdfCode": null,
            "relatedReports": null,
            "rowsTaxCodes": {
                "24": {
                    "ATTRIBUTES": {
                        "440": [
                            "Napoli - LIRN"
                        ],
                        "452": [
                            "82899"
                        ],
                        "453": [
                            2047
                        ],
                        "454": [
                            {
                                "content": [
                                    122
                                ]
                            }
                        ],
                        "455": [
                            6
                        ],
                        "477": [
                            "2018-10-22"
                        ]
                    },
                    "ENTITIES": {
                        "22": [
                            {
                                "ATTRIBUTES": {
                                    "424": [
                                        16
                                    ]
                                },
                                "ID": "ID00000000000000000000000000000013"
                            }
                        ],
                        "1": [
                            {
                                "ATTRIBUTES": {
```

6.4.6 Get Query ResultByQuery (xsd_tags) with scroll-id

*THIS IS THE READABLE VERSION OF THE METHOD SPECIFIED IN SECTION **6.1.14 GET QUERY RESULTBYQUERY (TAXONOMY CODES) WITH SCROLL-ID***

Method URL: {BASE_URL}/occurrences/readable/result/bool/scroll

Method Type: POST

This method executes the query specified in the body of the request, with the scroll option. It can be used to retrieve more than 10000 documents. They can be retrieved all in the same request or in multiple requests (when over 10000 documents).

- The input parameters are the following:
  - ✓ single-value (true/false): retrieve either full path to values (false) or just the value (true)
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
  - ✓ type, with options:
    - ○ REPORT
    - ○ VALIDATED
    - ○ OCCURRENCE
  - ✓ queryJson
  - ✓ pageSize
    - ○ The maximum pageSize supported is 10000.
    - ○ 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.
- First Request
  - ✓ Body example:

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

- ✓ Response example:

```
{    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
                "status": "DRAFT",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Napoli - LIRN"
                            ],
                            "452": [
                                "82899"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        122
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-22"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID000000000000000000000000000000013"
                                }
```

- Subsequent Requests

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

The total number of records that meets the query condition is indicated in the “total” field.

✓ Body example:

```
{
    "type": "OCCURRENCE",
    "queryJson": "{\"bool\": {\"filter\": [{\"range\": {\"modificationDate\": {\"gte\": \"2000-09-23\"}}}]}}",
    "scrollId":"FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJveWZSTHZuNWVuYU9RAAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuYU9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ=="
}
```

- Response:

```
{   "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000184145",
                "version": "0.1",
                "type": "OC",
                "status": "OPEN",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Århus/Tirstrup - EKAH"
                            ],
                            "452": [
                                "80022"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        67
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-07-01"
                            ],
                            "478": [
                                "10:02:00"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            13
                                        ]
```

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

THERE IS NO “NON-READABLE” VERSION OF THIS METHOD.

Method URL: {BASE_URL}/occurrences/readable/shared/result/scroll

Method Type: POST

This method executes the queries saved into the ECCAIRS 2 database, with the scroll option. Its particularity is that it only retrieves the ones **shared with the ECR**. It can be used to retrieve more than 10000 documents. They can be retrieved all in the same request or in multiple requests (when over 10000 documents).

- The input parameters are the following:
    - 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 of your authority (true). By default, is true.
    - excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
    - type, with options:
        - REPORT
        - VALIDATED
        - OCCURRENCE
    - queryName,
    - libraryName - the library where it is saved in E2 are specified in the body.
    - pageSize
        - The maximum pageSize supported is 10000.
        - 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.
    - scrollId – It should always be null in the first request.
- First Request
    - Body example:

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

- Response example:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZFFoZTBTMTM4QmFSU
TJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
                "status": "DRAFT",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Napoli - LIRN"
                            ],
                            "452": [
                                "82899"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        122
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-22"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID000000000000000000000000000000013"
```

- Subsequent Requests

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

  The total number of records that meets the query condition is indicated in the "total" field.

  - ✓ Body example:

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

  - ✓ Response example:

```
{
    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAAABO9PcWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZrU3RCVXdmZFFoZTBTMTM4QmFSU
TJBAAAAAAAdgecWQzNIMmF5NlBRSXFPaEZ3RWd3NHBkZw==",
        "total": 12509,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000185518",
                "version": "0.1",
                "type": "OC",
                "status": "OPEN",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Unknown - ZZZZ"
                            ],
                            "452": [
                                "84389"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        999
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2019-01-15"
                            ],
                            "478": [
                                "03:00:00"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
```

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

THERE IS NO "NON-READABLE" VERSION OF THIS METHOD.

Method URL: {BASE_URL}/occurrences/readable/shared/result/bool/scroll

Method Type: POST

This method executes the query specified in the body of the request, with the scroll option. Its particularity is that it only retrieves the ones **shared with the ECR**. It can be used to retrieve more than 10000 documents. They can be retrieved all in the same request or in multiple requests (when over 10000 documents).

- The input parameters are the following:
  - ✓ 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 of your authority (true). By default, is true.
  - ✓ excludeMetadata: can be used optionally to exclude metadata fields from the method response.

- The following information must be specified in the body
  - ✓ type, with options:
    - ○ REPORT
    - ○ VALIDATED
    - ○ OCCURRENCE
  - ✓ queryJson
  - ✓ pageSize
    - ○ The maximum pageSize supported is 10000.
    - ○ 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.

- First Request
  - ✓ Body example:

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

  - ✓ Response example:

```
{    "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000177910",
                "version": "0.0",
                "type": "OC",
                "status": "DRAFT",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Napoli - LIRN"
                            ],
                            "452": [
                                "82899"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        122
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-10-22"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            16
                                        ]
                                    },
                                    "ID": "ID00000000000000000000000000000013"
                                }
```

- Subsequent Requests

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

The total number of records that meets the query condition is indicated in the “total” field.

✓ Body example:

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

✓ Response example:

```
{   "data": {
        "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDnF1ZXJ5VGhlbkZldGNoAhZ0RF94RGdDYlJv
eWZSTHZuNWVuYU9RAAAAAABKiioWa2RfS2lfYldRUS1GM2liX1ZUTzhDQRZ0RF94RGdDYlJveWZSTHZuNWVuY
U9RAAAAAABKiisWa2RfS2lfYldRUS1GM2liX1ZUTzhDQQ==",
        "total": 12511,
        "occurrenceObjectDTOList": [
            {
                "e2Id": "OC-0000000000184145",
                "version": "0.1",
                "type": "OC",
                "status": "OPEN",
                "responsibleEntityId": 8,
                "reportingEntityId": null,
                "pdfCode": null,
                "relatedReports": null,
                "rowsTaxCodes": {
                    "24": {
                        "ATTRIBUTES": {
                            "440": [
                                "Århus/Tirstrup - EKAH"
                            ],
                            "452": [
                                "80022"
                            ],
                            "453": [
                                2047
                            ],
                            "454": [
                                {
                                    "content": [
                                        67
                                    ]
                                }
                            ],
                            "455": [
                                6
                            ],
                            "477": [
                                "2018-07-01"
                            ],
                            "478": [
                                "10:02:00"
                            ]
                        },
                        "ENTITIES": {
                            "22": [
                                {
                                    "ATTRIBUTES": {
                                        "424": [
                                            13
                                        ]
```

7 Taxonomy Management

7.1 External Value List Controller

As a pre-requisite for most of these methods (except for GetValueList), a Taxonomy in draft status must previously exist in the user's authority to be able to execute them.

7.1.1 Create ValueList

Method URL: {BASE_URL}}/taxonomy-service/external/value-lists/{taxonomyReferenceId}

Method Type: POST

This method is used to create a Value List under the Taxonomy project (draft) of the user's authority.

- The input parameter is the following:
  - ✓ taxonomyReferenceId: id of the Taxonomy Reference, with options:
    - ○ 1: ADREP (default)
    - ○ 2: SRIS
- Example1: create a Value List without values (the values ca be created/updated or deleted separately)
  - ✓ Body:

```
{
   "id":null,
   "identifier":113340,
   "sourceName": "NewVL.xls",
   "description":"NEW VL Description",
   "detailed":"NEW VL Detailed",
   "explanation":"NEW VL Explanation",
   "levels":3,
   "lastIdentifier":31,
   "customId":1,
   "isCustom":0,
   "locked":0,
   "alias":[{
       }]
}
```

- ✓ 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,
                "description": "ADREP",
                "detailed": "ADREP 2000 Revision 2008 (Accident/Incident Data Reporti
ng)",
                "explanation": "ADREP"
            },
            "taxonomyStatus": {
                "id": 2,
                "name": "Draft",
                "description": "If any, there can only be one"
            },
            "taxonomyType": {
                "id": 2,
                "name": "Custom"
            },
            "authority": {
                "id": 33,
                "authorityName": "EASA",
                "authorityCode": "EASA",
                "countryId": null
            }
        },
        "listOfValues": null,
        "alias": [
            {
                "aliasId": 3,
                "valueListId": 28014,
                "hasListOfValueAlias": false,
                "action": {
```

- Example 2: create a Value List with values
- Body:

```
{
   "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
        },
        "aliasList":[{
                "alias": {
                     "id":3
                } ,
            "value":"New Alias 2"
            }]
   },
   {
        "isNewRecord":0,
        "identifier":3,
        "parent":0,
        "level":1,
        "description":"NEW Value 3",
        "detailed":"NEW Value 3",
        "explanation": "NEW Value 3",
        "hasChild":0,
        "specialValue":{
            "id":1
        },
        "aliasList":[{
                "alias": {
                     "id":3
                } ,
            "value":"New Alias 3"
            }]
   }]
}
```

- ✓ 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,
                "description": "ADREP",
                "detailed": "ADREP 2000 Revision 2008 (Accident/Incident Data Reporti
ng)",
                "explanation": "ADREP"
            },
            "taxonomyStatus": {
                "id": 2,
                "name": "Draft",
                "description": "If any, there can only be one"
            },
            "taxonomyType": {
                "id": 2,
                "name": "Custom"
            },
            "authority": {
                "id": 33,
                "authorityName": "EASA",
                "authorityCode": "EASA",
                "countryId": null
            }
        },
        "listOfValues": [
            {
                "id": 7027085,
                "valueList": {
                    "id": 28020,
                    "sourceName": null,
                    "identifier": 100056,
```

7.1.2 Update ValueList

Method URL: {BASE_URL}}/taxonomy-service/external/value-lists/{valueListId}

Method Type: PUT

This method is used to update a Value List properties under the Taxonomy project (draft) of the user's authority.

- The input parameter is the following:
  - ✓ valueListId: internal id of the Value List to be updated

- Body example:

```
{
                "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,
                "releasedTaxonomyIdVersion": {
                    "id": 311
                },
                "alias":[{
                           "aliasId":3
                }]
}
```

- Response example:

```
{
    "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,
                "description": "ADREP",
                "detailed": "ADREP 2000 Revision 2008 (Accident/Incident Data Reporti
ng)",
                "explanation": "ADREP"
            },
            "taxonomyStatus": {
                "id": 2,
                "name": "Draft",
                "description": "If any, there can only be one"
            },
            "taxonomyType": {
                "id": 2,
                "name": "Custom"
            },
            "authority": {
                "id": 33,
                "authorityName": "EASA",
                "authorityCode": "EASA",
                "countryId": null
            }
        },
        "listOfValues": null,
        "alias": [
            {
                "aliasId": 3,
                "valueListId": 27980,
                "hasListOfValueAlias": false,
                "action": {
```

7.1.3 Delete ValueList

Method URL: {BASE_URL}}/taxonomy-service/ external/value-lists/{valueListId}

Method Type: DELETE

This method is used to delete a Value List under the Taxonomy project (draft) of the user's authority.

- The input parameter is the following:
  - ✓ valueListId: internal id of the Value List to be deleted from the draft Taxonomy.
- Body should always be empty.
- Response example:

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

Note that CRUD_FBS_17 means that the Value List has been successfully deleted.

7.1.4 Get ValueList

Method URL: {BASE_URL}}/taxonomy-service/external/value-lists/{valueListsId}

Method Type: GET

This method is used to retrieve a given Value List by its internal id.

- The input parameter is the following:
  - ✓ valueListsId: internal id of the value list to retrieve obtained in the step 6.1.16.
- Body should always be empty.
- Response example:

```
{
    "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": {
                "id": 1,
                "name": "NAA"
            }
        },
        "action": null,
        "releasedTaxonomyIdVersionId": 233,
        "isCustom": true,
        "valueListModels": [
            {
                "id": 5651940,
                "valueListId": 22863,
                "identifier": 1,
                "parent": 0,
                "level": 1,
                "description": "Manufacturing",
                "detailed": "Manufacturing",
                "explanation": " ",
                "hasChild": false,
                "specialValue": {
                    "id": 0,
                    "name": "Normal"
                },
                "status": 1,
                "xsdTag": "1",
                "aliases": [],
                "active": true
            },
            {
                "id": 5651941,
```

7.1.5 Get ValueList by identifier

Method URL: {BASE_URL}}/taxonomy-service/external/value-lists/by-identifier{identifier}

Method Type: GET

This method is used to retrieve a given Value List data by its identifier.

- The input parameter is the following:
  - ✓ identifier: identifier of the value list to retrieve obtained in the step 6.1.16.
- Body should always be empty.
- Response example:

```
{
    "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": {
                "id": 1,
                "name": "NAA"
            }
        },
        "action": null,
        "releasedTaxonomyIdVersionId": 233,
        "isCustom": true,
        "valueListModels": [
            {
                "id": 5651940,
                "valueListId": 22863,
                "identifier": 1,
                "parent": 0,
                "level": 1,
                "description": "Manufacturing",
                "detailed": "Manufacturing",
                "explanation": " ",
                "hasChild": false,
                "specialValue": {
                    "id": 0,
                    "name": "Normal"
                },
                "status": 1,
                "xsdTag": "1",
                "aliases": [],
                "active": true
            },
            {
                "id": 5651941,
```

7.1.6 Get ValueLists for Taxonomy Reference

Method URL: {BASE_URL}}/taxonomy-service/external/value-lists/taxonomy-reference/{taxonomyReferenceId}

Method Type: GET

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

- The input parameters are the following:
  - ✓ taxonomyReferenceId: 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 should always be empty.
- Response example:

```
{
  "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,
reptile) 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",
      "authorityCode": "EASA_ECCAIRS",
      "country": {
        "id": 21023,
        "name": "International Organisations",
        "code": null
      },
      "countryCode": null,
      "authorityType": {
        "id": 3,
        "name": "NAA/SIA"
      }
    },
    "originalItemId": null,
    "action": null,
    "projectTaxonomyIdVersionId": null,
    "releasedTaxonomyIdVersionId": 1,
    "isCustom": false,
    "valueListModels": [
      {
        "id": 375,
        "valueListId": 3,
        "identifier": 1,
        "parent": 0,
        "level": 1,
        "description": "MAMMALS",
        "detailed": "MAMMALS",
```

7.2 External List of Value Controller

As a pre-requisite for most of these methods (except for GetValue), a Taxonomy in draft status must previously exist in the user's authority to be able to execute them.

Also, the Value List Id might be needed. It can be retrieved as described in **7.1.4 Get ValueList** and **7.1.5 Get ValueList by identifier**

Method URL: {BASE_URL}}/taxonomy-service/external/value-lists/by-identifier{identifier}

Method Type: GET

This method is used to retrieve a given Value List data by its identifier.

- The input parameter is the following:
  - ✓ identifier: identifier of the value list to retrieve obtained in the step 6.1.16.
- Body should always be empty.
- Response example:

```
{
    "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,
```

Get ValueLists for Taxonomy Reference

7.2.1 Create Value

Method URL: {BASE_URL}}/taxonomy-service/external/value/{valueListId}/ {taxonomyId}

Method Type: POST

This method is used to create a Value in a Value List under the Taxonomy project (draft) of the user's authority.

- The input parameters are the following:
  - ✓ 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 example:

```
{
    "valueList":{
       "id":141108
   },
   "identifier":4,
   "parent":0,
   "level":1,
   "description":"NEW Value 4",
   "detailed":"NEW Value 4",
   "hasChild":0,
   "specialValue":{
       "id": 1
   },
   "releasedTaxonomyIdVersion": {
            "id": 1108
        }
}
```

- Response example:

```
{
    "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,
                "taxonomyReference": {
                    "id": 1,
                    "description": "ADREP",
                    "detailed": "ADREP 2000 Revision 2008 (Accident/Incident Data Rep
orting)",
                    "explanation": "ADREP"
                },
                "taxonomyStatus": {
                    "id": 2,
                    "name": "Draft",
                    "description": "If any, there can only be one"
                },
                "taxonomyType": {
                    "id": 2,
                    "name": "Custom"
                },
                "authority": {
                    "id": 33,
                    "authorityName": "EASA",
                    "authorityCode": "EASA",
                    "countryId": null
                }
            },
            "listOfValues": null,
            "alias": null,
            "attributes": null,
            "action": {
                "id": 2,
```

7.2.2 Update Value

Method URL: {BASE_URL}}/taxonomy-service/external/value/{listOfvalueId}/ {valueListId}/{taxonomyId}

Method Type: PUT

This method is used to update a Value in a Value List under the Taxonomy project (draft) of the user's authority.

- The input parameters are the following:
  - ✓ 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 example:

```
{       "id": 7027085,
        "valueList": {
            "id": 28020
        },
        "identifier": 4,
        "parent": 0,
        "level": 1,
        "description": "Update Value 4",
        "detailed": "Update Value 4",
        "explanation": "Update Value 4",
        "hasChild": false,
        "specialValue": {
            "id": 2,
            "name": "None"
        },
        "xsdTag": "4",
        "releasedTaxonomyIdVersion": {
            "id": 311
        }
    }
```

- Response example:

```
{
    "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,
```

7.2.3 Delete Value

Method URL: {BASE_URL}}/taxonomy-service/external/value/{valueId}

Method Type: DELETE

This method is used to delete a Value in a Value List under the Taxonomy project (draft) of the user's authority.

- The input parameter is the following:
  - ✓ ValueId: unique identifier of the value to be deleted.
- Body should always be empty.
- Response example:

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

7.2.4 Restore Value

Method URL: {BASE_URL}}/taxonomy-service/external/value/restore/{valueId}

Method Type: PUT

This method is used to restore a Value previously deleted (logically deleted) in a Value List under the Taxonomy project (draft) of the user's authority.

Note that the restore of a Value is only possible when the Value was present in the released Taxonomy from where the draft version was created.

- The input parameter is the following:
  - ✓ ValueId: unique identifier of the value to be restored.
- Body should always be empty.
- Response example:

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

7.2.5 Get Value

Method URL: {BASE_URL}}/taxonomy-service/external/value/{valueId}

Method Type: GET

This method is used to get information about a Value in a Value List under the Taxonomy project.

- The input parameters are the following:
  - ✓ valueId: id of the Value wanted to get information
  - ✓ taxonomyId: id of the version of the Taxonomy
- Body should always be empty.
- Response example:

```
{
    "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,
                "taxonomyReference": {
                    "id": 1,
                    "description": "ADREP",
                    "detailed": "ADREP 2000 Revision 2008 (Accident/Incident Data Rep
orting)",
                    "explanation": "ADREP"
                },
                "taxonomyStatus": {
                    "id": 2,
                    "name": "Draft",
                    "description": "If any, there can only be one"
                },
                "taxonomyType": {
                    "id": 2,
                    "name": "Custom"
                },
                "authority": {
                    "id": 33,
                    "authorityName": "EASA",
                    "authorityCode": "EASA",
                    "countryId": null
                }
            },
            "listOfValues": null,
            "alias": null,
            "attributes": null,
            "action": {
                "id": 2,
```

7.3 External Taxonomy Controller

7.3.1 GetTaxonomy

Method URL: {BASE_URL}}/taxonomy-service/external/taxonomy

Method Type: POST

This method is used to retrieve the properties of the specified entities and attributes in the released taxonomy of the user’s authority

- The following information must be specified in the body
  - ✓ entityTC – Entity taxonomy code.
  - ✓ attributList – Including the taxonomy codes of the attributes under a given entity. The taxonomy codes must be provided as shown in the body of the example below.
- Body example:

```json
[
  {
    "entityTC": 24,
    "attributesList": [746, 19, 17]
  },
  {
    "entityTC": 2,
    "attributesList": [92, 93, 94]
  }
]
```

- Response example:

```
{
  "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",
              "detailed": "Aerodrome latitude",
              "explanation": "The latitude of the aerodrome reference point.
Aerodrome reference point: The designated geographical location of an
aerodrome.",
              "domains": null,
              "minInstance": null,
              "maxInstance": null,
              "releasedTaxonomyIdVersion": {
                "creationDate": "2020-08-12T14:03:45.000+0000",
                "creationUser": "root@localhost",
                "modificationDate": "2020-10-27T07:42:22.000+0000",
                "modificationUser": "root@localhost",
                "active": true,
                "id": 1,
                "name": "ECCAIRS Aviation 5.1.0.0",
                "description": "ECCAIRS Aviation..",
                "version": "5.1.0.0",
                "reference": null,
                "releasedDate": "2020-06-16T13:10:00.000+0000",
                "versionChange": null,
                "tag": null,
                "taxonomyReference": {
                  "creationDate": "2020-10-27T07:42:22.000+0000",
                  "creationUser": "eccairsroot@51.137.19.44",
                  "modificationDate": null,
                  "modificationUser": null,
                  "active": true,
                  "id": 1,
                  "description": "ADREP",
                  "detailed": "ADREP 2000 Revision 2008 (Accident/Incident
Data Reporting)",
                  "explanation": "ADREP"
                },
                "taxonomyStatus": {
                  "creationDate": "2020-08-12T14:03:46.000+0000",
                  "creationUser": "easa@172.17.0.1",
                  "modificationDate": null,
                  "modificationUser": null,
```

7.4 Taxonomy Controller

7.4.1 Get Taxonomy json schema

Method URL: {BASE_URL}}/taxonomy-service public/getJsonSchema/{taxonomyId}

Method Type: GET

This method is used to retrieve the json schema of a given taxonomy id.

- The input parameter is the following:
  - ✓ taxonomyId: identifier of the taxonomy
- Body should always be empty.
- Response example:

{
    "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\": {\"$ref\": \"#/definitions/typeUpdated\", \"title\": \"It determines whether the report is locked for editing.\", \"description\": \"Determines if the report has been automatically updated, from a new Original Report.\"}, \"version\": {\"type\": \"string\", \"title\": \"Occurrence version\", \"pattern\": \"^[0-9]{1,4}.[0-9]{1,4}$\", \"examples\": \"[0.0, 1.0, 7.5, 10.1,10.2,9999.9999]\", \"description\": \"Occurrence version. Two digits, separated by a dot. Minor version, determined by the digit on after the dot. Major version, determined by the digit on before the dot.   ORs and VRs will always be minor versions. Warning: 0.0 in Draft version. It is new, so in the migration, we should decide which version to include.\"}, \"pdf_code\": {\"enum\": [1, 2, 3, 4, 5, 6], \"type\": \"integer\", \"title\": \"Code to identify how the OR was first informed\", \"examples\": [\"1 - PDF_GA_INDIVIDUAL\", \"2 - PDF_GA_ORGANISATION\", \"3 - PDF_TECHNICAL\", \"4 - PDF_AERODROME\", \"5 - PDF_ATM\", \"6 - PDF_FLIGHT_OPS\"], \"description\": \"Code to identify how the OR was first informed. The codes refer to the options in the ‘pdf’ table in MySQL.\"}, \"au_active\": {\"type\": \"boolean\", \"title\": \"It determines whether or not the report is logically deleted.\", \"examples\": \"[true, false]\", \"description\": \"It determines whether or not the report is logically deleted. If the property (au_active) has a true value the report is active, if it has a false value the report is not active, it is logically deleted.\"}, \"migration_file\": {\"type\": \"object\", \"title\": \"\", \"required\": [\"e5z_name\", \"xml_name\"], \"properties\": {\"e5z_name\": {\"type\": \"string\", \"title\": \"\", \"description\": \"\"}, \"xml_name\": {\"type\": \"string\", \"title\": \"\", \"description\": \"\"}}, \"description\": \"\", \"additionalProperties\": false}, \"taxonomy_codes\": {\"$ref\": \"#/definitions/typeTaxonomyCodes\", \"title\": \"Report in JSON, translated to Taxonomy Code\", \"description\": \"Report in JSON, translated to Taxonomy Code. Property that stores the taxonomy codes, their properties. It does not include the translations.\"}, \"related_reports\": {\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/typeE2ID\", \"title\": \"Authority Identifier\", \"examples\": \"[OR-0000000000001234,VR-0000000000009876,OC-0000000000033556,SR-0000000000225789]\", \"description\": \"They consist of a list with the identifiers of the authority table that are active.\"}, \"title\": \"Related reports (Authority Identifier Array)\", \"description\": \"OR 1-1 VR n-1 Occ n-1 SR. It will be empty for ORs, as it does not come from any other records. Mandatory for all the others.\", \"uniqueItems\": true}, \"shared_entities\": {\"$ref\": \"#/definitions/typeSharedEntities\", \"title\":

7.4.2 Get TaxonomyVersion

Method URL: {BASE_URL}}/taxonomy-service/released-taxonomy

Method Type: GET

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.

- Body should always be empty.
- Response example:

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

7.5 External Entity Controller

7.5.1 GetEntity

Method URL: {BASE_URL}}/taxonomy-service/external/entity/{taxonomyCode}

Method Type: GET

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

- The input parameters are the following:
  - ✓ attributeList: attributes wanted to be retrieved
  - ✓ taxonomyCode: identifier of the taxonomy
- Body should always be empty.
- Response example:

```
{
  "data": {
    "id": 5131,
    "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": 92364,
          "taxonomyCode": 1,
          "description": "Aerodrome latitude",
          "detailed": "Aerodrome latitude",
          "explanation": "The latitude of the aerodrome reference point. Aerodrome reference point: The designated geographical location of an aerodrome.",
          "domains": null,
          "minInstance": "Optional",
          "maxInstance": "Single value",
          "releasedTaxonomyIdVersion": {
            "id": 425,
            "name": "Eccairs Aviation 5.10.2.0",
            "description": "ECCAIRS Aviation",
            "version": "5.10.2.0",
            "releasedDate": "2022-03-30T12:50:41.000+00:00",
            "versionChange": null,
            "tag": null,
            "taxonomyReference": {
              "id": 1,
              "description": "ADREP",
              "detailed": "ADREP 2000 Revision 2008 (Accident/Incident Data Reporting)",
              "explanation": "ADREP"
            },
            "taxonomyStatus": {
              "id": 1,
              "name": "Released",
              "description": "If any, there can only be one"
            },
            "taxonomyType": {
              "id": 1,
```

8 User Management

8.1 Get Users

Method URL: {BASE_URL}}**/usermgmt/users**

Method Type: GET

This method is used to get a list of all users configured for the logged-in user's authority.

- The input parameters are the following:
  - ✓ 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.
- Body should always be empty.
- Response example:

```
{
    "data": [
        {
            "originalId": null,
            "id": 1,
            "firstName": "eccairs_first_name",
            "lastName": "eccairs_last_name",
            "username": "eccairs",
            "email": "easa-team-uat@bilbomatica.es",
            "phone": "927282828",
            "description": "System Administration purposes",
            "disabled": false,
            "builtIn": true,
            "userAdministrator": true,
            "canChangePassword": true,
            "blocked": false,
            "superuser": true,
            "organisationAccount": false,
            "password": "",
            "jobTitle": "ECCAIRS System administrator",
            "country": {
                "id": 11,
                "countryName": "Germany"
            },
            "authority": {
                "id": 1,
                "authorityName": "EASA_ECCAIRS",
                "displayName": "EASA_ECCAIRS",
                "type": {
                    "id": 3,
                    "name": "NAA/SIA"
                },
                "isEcr": false,
                "isEcrSris": true,
                "countryCode": null
            },
            "organisation": null,
            "language": {
                "id": 3,
                "name": "English"
            },
            "roles": [
                {
                    "id": 1,
                    "roleName": "ECCAIRS System Administrator",
                    "hierarchy": 1
                },
                {
                    "id": 14,
                    "roleName": "E2 Central Hub Administrator",
                    "hierarchy": 1
```

8.2 Get Autofilled Attributes

Method URL: {BASE_URL}}**/usermgmt/Autofilled-attributes**

Method Type: GET

This method is used to get a list of the Autofilled Attributes defined for the logged-in user's organisation.

- The input parameters are the following:
  - ✓ taxonomyReferenceId: 1
- Body should always be empty.
- Response example:

```
{
   "data": [
      {
         "id": 9384,
         "taxonomyCode": 1088,
         "attributeDescription": "Aviation sector",
         "listOfValueDescription": "General Aviation",
         "autofilledValue": "6",
         "textValue": null
      }
   ],
   "errorDetails": "",
   "returnCode": 1
```

9 Reference Tables

9.1 Get Reference Table Values

Method URL: {BASE_URL}}/usermgmt/reference-table-values/{triggeringAttributetaxonomyCode}

Method Type: GET

This method is used to get the released reference table values with given trigger attribute taxonomy code.

- Body should always be empty.
- Response example:

```
{
    "data": [
        {
            "id": 177396,
            "triggerValue": "TSTFIN",
            "translatedTriggerValue": "TSTFIN",
            "fromDate": 1633824000000,
            "toDate": 1633824000000,
            "triggerAttribute": {
                "id": 346368,
                "description": "Aircraft registration",
                "taxonomyCode": 244,
                "taxonomyPath": "taxonomyCodes.24.ENTITIES.4.ATTRIBUTES.244",
                "attributeValueListId": null,
                "active": null,
                "invalid": null,
                "attributeDataTypeId": null
            },
            "valueList": [
                {
                    "id": 2437545,
                    "value": "21320",
                    "attribute": {
                        "id": 346162,
                        "description": "Manufacturer/model",
                        "taxonomyCode": 21,
                        "taxonomyPath":
"taxonomyCodes.24.ENTITIES.4.ATTRIBUTES.21",
                        "attributeValueListId": 177711,
                        "active": null,
                        "invalid": null,
                        "attributeDataTypeId": 3
                    },
                    "description": "AAI",
                    "translatedDescription": "AAI",
                    "fullDescription": "AAI",
                    "translatedFullDescription": "AAI",
                    "detailed": "AAI",
                    "explanation": "AAI"
                },
                {
                    "id": 2437546,
                    "value": null,
                    "attribute": {
                        "id": 346172,
                        "description": "Aircraft category",
                        "taxonomyCode": 32,
                        "taxonomyPath":
"taxonomyCodes.24.ENTITIES.4.ATTRIBUTES.32",
                        "attributeValueListId": 177516,
                        "active": null,
```

9.2 Create Reference Table Values

Method URL: {BASE_URL}}/usermgmt/reference-table/{id}/import

Method Type: POST

This method is used to update a reference table entry.

- The input parameter is the following:
  - ✓ Incremental
    - ○ If TRUE, it adds what is included in the file to the existing values
    - ○ If FALSE, it deletes the existing values and creates the ones in the file
- The following information should be included in the body:
  - ✓ files: attachment filename. CSV file with the values to be uploaded.
    - ○ Supported Media Types: multipart/form-data
  - ✓ Type: CSV
- Response example:

```
{
    "data": {
        "id": 177810,
        "triggerValue": "123456",
        "fromDate": 1694476800000,
        "toDate": 1694476800000,
        "triggerAttribute": null,
        "valueList": [
            {
                "id": 2437980,
                "value": "1075",
                "translatedValue": "MAXAIR > DRIFTER",
                "attribute": {
                    "id": 289844,
                    "description": "Manufacturer/model",
                    "taxonomyCode": 21,
                    "taxonomyPath": "taxonomyCodes.24.ENTITIES.4.ATTRIBUTES
.21",
                    "attributeValueListId": 145551,
                    "active": null,
                    "invalid": null,
                    "attributeDataTypeId": 3
                }
            },
            {
                "id": 2437981,
                "value": "REG123456",
                "attribute": {
                    "id": 290050,
                    "description": "Aircraft registration",
                    "taxonomyCode": 244,
                    "taxonomyPath": "taxonomyCodes.24.ENTITIES.4.ATTRIBUTES
.244",
                    "attributeValueListId": 0,
                    "active": null,
                    "invalid": null,
                    "attributeDataTypeId": 1
                }
            },
            {
                "id": 2437982,
                "value": "1993",
                "attribute": {
                    "id": 290121,
                    "description": "Year built",
                    "taxonomyCode": 327,
                    "taxonomyPath": "taxonomyCodes.24.ENTITIES.4.ATTRIBUTES
.327",
                    "attributeValueListId": 0,
                    "active": null,
                    "invalid": null,
```

9.3 Delete Reference Table Values

Method URL: {BASE_URL}}/usermgmt/reference-table/values/{id}

Method Type: DELETE

This method is used to delete a reference table entry.

- Body should always be empty.
- Response example:

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

10 Import from files

10.1 Create Original Reports from e5x files

Method URL: {BASE_URL}}/frontfile-api/files/migrate

Method Type: POST

This method is used to create an Original Reports in E2. One or more e5x files can be attached, and the xml file(s) contained in it, will be attached in attribute 802 of each corresponding OR.

- The input parameter is the following:
  - ✓ Files (e5x files)
- Body should always be empty.
- Response example 1:

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

Where:

```
OCC_FBS_310: Your file has been successfully uploaded and is being processed.
```

System messages:

`OCC_FBS_310:` Your file has been successfully uploaded and is being processed.

- Response example 2:

```
{
    "data": {
        "value": [
            "13059",
            "09c1bcc1-cfec-4bdb-aec2-76bbd4ca3576",
            "14222"
        ],
        "key": "OCC_FBW_309"
    },
    "errorDetails": "Your request has been queued. It may take a while to be
completed. It will be automatically processed once the queue gets released.",
    "returnCode": 1
}
```

Where:

```
OCC_FBW_309: Your request has been queued. It may take a while to be
completed. It will be automatically processed once the queue gets released.
```

`OCC_FBW_309:` Your request has been queued. It may take a while to be completed. It will be automatically processed once the queue gets released.

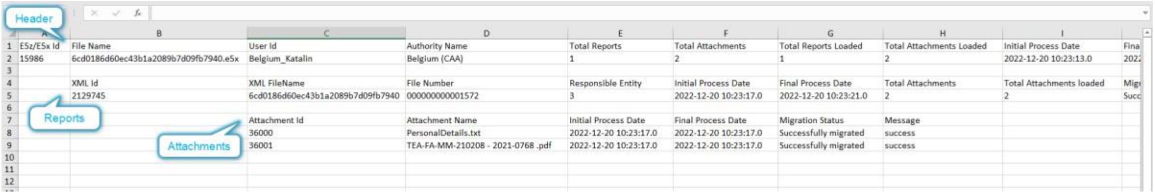
10.2 Get the results of a migrated e5x file

Method URL: {BASE_URL}}/frontfile-api/results/e5xresults

Method Type: GET

This method is used 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.

- The input parameters are the following:
  - ✓ format: csv or json
  - ✓ idFile: the identifier of the file migrated
- Body should always be empty.
- Response:
  - ✓ CSV file, or



	A	B	C	D	E	F	G	H	I	
1	E5z/E5x Id	File Name	User Id	Authority Name	Total Reports	Total Attachments	Total Reports Loaded	Total Attachments Loaded	Initial Process Date	Fina
2	15986	6cd0186d60ec43b1a2089b7d09fb7940.e5x	Belgium_Katalin	Belgium (CAA)	1	2	1	2	2022-12-20 10:23:13.0	2022
3										
4		XML Id	XML FileName	File Number	Responsible Entity	Initial Process Date	Final Process Date	Total Attachments	Total Attachments loaded	Migr
5		2129745	6cd0186d60ec43b1a2089b7d09fb7940	000000000001572	3	2022-12-20 10:23:17.0	2022-12-20 10:23:21.0	2	2	Succ
6										
7			Attachment Id	Attachment Name	Initial Process Date	Final Process Date	Migration Status	Message		
8			36000	PersonalDetails.txt	2022-12-20 10:23:17.0	2022-12-20 10:23:17.0	Successfully migrated	success		
9			36001	TEA-FA-MM-210208 - 2021-0768 .pdf	2022-12-20 10:23:17.0	2022-12-20 10:23:17.0	Successfully migrated	success		
10										
11										
12										

- ○ In the attached Excel you can see the explanation of the fields

CSV Migration result structure.xlsx

- ○ Example

Migration results - example.xlsx

- ✓ JSON file
  - ○ In the attached Excel you can see the explanation of the fields

JSON Migration result structure.xlsx

- Response example 1:

```
{
        "e5zE5xId": 1426,
        "fileName": "Teste5z.e5z",
        "userId": "Holanda_User",
        "authorityName": "Netherlands (CAA)",
        "totalReports": 0,
        "totalAttachments": 0,
        "totalReportsLoaded": 55,
        "totalAttachmentsLoaded": 36,
        "initialProcessDate": 1621939619000,
        "finalProcessDate": 1621939744000,
        "migrationOrigin": "Migration Report",
        "migrationStatus": "Processed with Errors",
        "message": "\"Error Unzipping File\"",
        "reportInfoList": [
                {
                        "xmlId": 1020575,
                        "xmlFileName": "4EB149BEDA6F4BA8B330350467343C7C",
                        "fileNumber": "AR.EU-SI-2020-000020",
                        "responsibleEntity": 22,
                        "initialProcessDate": 1621939650000,
                        "finalProcessDate": 1621939654000,
                        "totalAttachments": 0,
                        "totalAttachmentsLoaded": 0,
                        "migrationStatus": "Not migrated",
                        "message": "\"I/O error on POST request for \\\"http://e2-
occurrences-api-svc.easa:8081/private-occurrences/createFromBatch\\\": Connect
to e2-occurrences-api-svc.easa:8081 [e2-occurrences-api-svc.easa/10.0.193.81]
failed: Connection refused (Connection refused); nested exception is
org.apache.http.conn.HttpHostConnectException: Connect to e2-occurrences-api-
svc.easa:8081 [e2-occurrences-api-svc.easa/10.0.193.81] failed: Connection
refused (Connection refused)\"",
                        "csvMigrationAttachmentInfos": null
                },
                {
                        "xmlId": 1020576,
                        "xmlFileName": "52022A68406E4E5A9D69C27D7D8CAD82",
                        "fileNumber": "Test 211/2020",
                        "responsibleEntity": 22,
                        "initialProcessDate": 1621939651000,
                        "finalProcessDate": 1621939654000,
                        "totalAttachments": 2,
                        "totalAttachmentsLoaded": 2,
                        "migrationStatus": "Not migrated",
                        "message": "\"I/O error on POST request for \\\"http://e2-
occurrences-api-svc.easa:8081/private-occurrences/createFromBatch\\\": Connect
to e2-occurrences-api-svc.easa:8081 [e2-occurrences-api-svc.easa/10.0.193.81]
failed: Connection refused (Connection refused); nested exception is
org.apache.http.conn.HttpHostConnectException: Connect to e2-occurrences-api-
svc.easa:8081 [e2-occurrences-api-svc.easa/10.0.193.81] failed: Connection
refused (Connection refused)\"",
```

- Response example 2:

```
{
    "data": "OCC_FBW_486",
    "errorDetails": "Migration is not yet finished",
    "returnCode": 1
}
```

Where:

```
OCC_FBW_486: Migration is not yet finished.
```

System messages:

`OCC_FBW_486:` Migration is not yet finished.

11 Queries Management

11.1 Get queries

Method URL: {BASE_URL}}/queries/bi

Method Type: GET

This method retrieves the queries available for the logged user together with the type of library in which the query is stored.

- Body should always be empty.
- Response example:

E2 API Guide

```
{
    "data": [
        {
            "id": 1497,
            "name": "Class",
            "type": 1
        },
        {
            "id": 4309,
            "name": "Drag & Drop",
            "type": 1
        },
        {
            "id": 8756,
            "name": "Prueba",
            "type": 1
        },
        {
            "id": 11265,
            "name": "Additional Text",
            "type": 1
        },
        {
            "id": 13075,
            "name": "By dates ECR",
            "type": 1
        },
        {
            "id": 15595,
            "name": "test2",
            "type": 1
        },
        {
            "id": 15606,
            "name": "test rtf",
            "type": 1
        },
        {
            "id": 15726,
            "name": "au_modification_date",
            "type": 1
        },
        {
            "id": 16697,
            "name": "has automatic changes",
            "type": 1
        },
        {
            "id": 17139,
            "name": "Occurrence Class",
            "type": 1
```

12 Annex

12.1 Response format

All the methods responses have the same format

```
{
    "data": "<response data>",
    "errorDetails": "",
    "returnCode": 1
}
```

Where:

- Data
  - ✓ It includes a code, when applicable, if a message needs to be shown to the user
  - ✓ In some cases, it also returns parameters to be shown embedded in the message.
- Error Details is the explanation of the message
- returnCode can have two values:
  - ✓ 1: OK
  - ✓ 2: Error

12.1.1 Response with no errors

Simple case of returning a message ‘key’ as:

```
{
    "data": " CRUD_XXX_NNN",
    "errorDetails": "Message description",
    "returnCode": 1
}
```

12.1.2 Response with errors

Simple case of returning just the error ‘key’ as:

```
{
    "data": " CRUD_XXX_NNN",
    "errorDetails": "Error description",
    "returnCode": 2
}
```

12.1.3 Data embedded into the error message

```
{
    "data":
       {
        "key": "CRUD_XXX_NNN",
        "value": ["attribute1, attibute2,attribute3","query1,query2","bo1,bo2"]
    },
    "errorDetails": "Error description",
    "returnCode": 2
}
```

12.2 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
0 Logging in with an existing user		(none)
Error! Reference source not found. Creating an occurrence	• OR: CreateReporterOriginalReport • VR: CreateValidatedReport • OC: CreateOccurrence • SR: CreateSafetyRecommendation	It depends on the type of record to be created (OR, VR, OC, SR)
0 Updating an occurrence	• OR: EditReporterOriginalReport • VR: EditValidatedReport • OC: EditOccurrence • SR: EditSafetyRecommendation	It depends on the type of record to be updated (OR, VR, OC, SR)
6.1.9 Getting an occurrence	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence,	It depends on the type of record to be read (OR, VR, OC, SR)

METHOD	USER RIGHT	COMMENTS
	• SR: ReadSafetyRecommendationResponse	
6.1.10 Getting an occurrence by fields	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	It depends on the type of record to be read (OR, VR, OC, SR)
6.1.11 Changing an occurrence status	• OR: EditReporterOriginalReport, DeleteOriginalReport, ArchiveOriginalReport, • VR: EditValidatedReport, DeleteValidatedReport, ArchiveValidatedReport, • OC: EditOccurrence, DeleteOccurrence, ArchiveOccurrence, CloseOccurrence, • SR: EditSafetyRecommendation, DeleteSafetyRecommendation	It depends on the type of record to be updated (OR, VR, OC, SR)
6.1.11 Get Query Result	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	It depends on the type of record to be read (OR, VR, OC, SR)
6.1.13 Get Query ResultByQuery	• OR: ReadOriginalReport, • VR: ReadValidatedReport, • OC: ReadOccurrence, • SR: ReadSafetyRecommendationResponse	It depends on the type of record to be read (OR, VR, OC, SR)
7.3.1 GetTaxonomy		(none)
7.4.2 GetEntity		(none)
7.1.5 GetValueList	ReadTaxonomyValueList	
7.1.4 GetValueLists	ReadTaxonomyValueList	

METHOD	USER RIGHT	COMMENTS
Error! Reference source not found. CreateValueList	CreateTaxonomyValueList	
7.1.1 UpdateValueList	EditTaxonomyValueList	
Error! Reference source not found. DeleteValueList	DeleteTaxonomyValueList	

12.3 List of authorities

ID	AUTHORITY NAME	DISPLAY NAME	AUTHORITY CODE	AUTHORITY TYPE
1	EASA_ECCAIRS	EASA_ECCAIRS	EASA_ECCAIRS	3 (NAA/SIA)
2	Austria (CAA)	Austria (CAA)	CAA	1 (NAA)
3	Belgium (CAA)	Belgium (CAA)	CAA	1 (NAA)
4	Bulgaria (CAA)	Bulgaria (CAA)	CAA	1 (NAA)
5	Croatia (CAA)	Croatia (CAA)	CAA	1 (NAA)
6	Cyprus (CAA)	Cyprus (DCA)	CAA	1 (NAA)
7	Czech Republic (CAA)	Czech Republic (CAA)	CAA	1 (NAA)
8	Denmark (CAA)	Denmark (CAA)	CAA	1 (NAA)
9	Estonia (CAA)	Estonia (CAA)	CAA	1 (NAA)
10	Finland (CAA)	Finland (CAA)	CAA	1 (NAA)
11	France (DGAC)	France (DGAC)	DGAC	1 (NAA)

ID	AUTHORITY NAME	DISPLAY NAME	AUTHORITY CODE	AUTHORITY TYPE
12	Germany (LBA)	Germany (CAA)	LBA	1 (NAA)
13	Greece (HCAA/SORC)	Greece (HCAA)	HCAA/SORC	1 (NAA)
14	Hungary (CAA)	Hungary (CAA)	CAA	1 (NAA)
15	Iceland (FlugmÃ¡lastjÃ³rn Ã•slands)	Iceland (CAA)	CAA	1 (NAA)
16	Ireland (CAA)	Ireland (IAA)	CAA	1 (NAA)
17	Italy (ENAC)	Italy (ENAC)	ENAC	1 (NAA)
18	Latvia (CAA)	Latvia (CAA)	CAA	1 (NAA)
19	Lithuania (CAA)	Lithuania (CAA)	CAA	1 (NAA)
20	Luxembourg (CAA)	Luxembourg (CAA)	CAA	1 (NAA)
21	Malta (CAA)	Malta (CAA)	CAA	1 (NAA)
22	Netherlands (CAA)	Netherlands (ILT)	CAA	1 (NAA)
23	Norway (CAA)	Norway (CAA)	CAA	1 (NAA)
24	Poland (CAA)	Poland (CAA)	CAA	1 (NAA)
25	Portugal (CAA)	Portugal (ANAC)	CAA	1 (NAA)
26	Romania (CAA)	Romania (CAA)	CAA	1 (NAA)
27	Slovakia (CAA)	Slovakia (CAA)	CAA	1 (NAA)
28	Slovenia (CAA)	Slovenia (CAA)	CAA	1 (NAA)
29	Spain (CAA)	Spain (AESA)	CAA	1 (NAA)

ID	AUTHORITY NAME	DISPLAY NAME	AUTHORITY CODE	AUTHORITY TYPE
30	Sweden (CAA)	Sweden (CAA)	CAA	1 (NAA)
31	Switzerland (FOCA)	Switzerland (FOCA)	FOCA	1 (NAA)
32	United Kingdom (CAA)	UK (as ICAO State, NOT reporting under EU R376/2014)	CAA	1 (NAA)
33	EASA	EASA (in its role as Competent Authority)	EASA	3 (NAA/SIA)
35	Sweden (AIB)	Sweden (SIA)	AIB	3 (NAA/SIA)
36	Hungary (TSB)	Hungary (TSB)	TSB	3 (NAA/SIA)
37	Austria (AAIB)	Austria (SIA)	AAIB	2 (SIA)
38	Belgium (AIB)	Belgium (SIA)	AIB	2 (SIA)
39	Bulgaria (AAIU)	Bulgaria (SIA)	AAIU	3 (NAA/SIA)
40	Croatia (AIB)	Croatia (AIN)	AIB	3 (NAA/SIA)
41	Cyprus (AIIB)	Cyprus (SIA)	AIIB	2 (SIA)
42	Czech Republic (AAII)	Czech Republic (AAII)	AAII	3 (NAA/SIA)
43	Denmark (AAIB)	Denmark (AAIB)	AAIB	3 (NAA/SIA)
44	Germany (BFU)	Germany (BFU)	BFU	3 (NAA/SIA)
45	Greece (AAIASB)	Greece (SIA)	AAIASB	3 (NAA/SIA)
46	Iceland (AIB)	Iceland (SIA)	AIB	3 (NAA/SIA)
47	Ireland (AAIU)	Ireland (SIA)	AAIU	3 (NAA/SIA)
48	Italy (ANSV)	Italy (ANSV)	ANSV	3 (NAA/SIA)

ID	AUTHORITY NAME	DISPLAY NAME	AUTHORITY CODE	AUTHORITY TYPE
49	Latvia (TAIIB)	Latvia (SIA)	TAIIB	2 (SIA)
50	Luxembourg (AIB)	Luxembourg (SIA)	AIB	2 (SIA)
51	Netherlands (DSB)	Netherlands (SIA)	DSB	3 (NAA/SIA)
52	Norway (AAIB)	Norway (NSIA)	AAIB	2 (SIA)
53	Poland (AIB)	Poland (SIA)	AIB	2 (SIA)
54	Portugal (GPIAA)	Portugal (SIA)	GPIAA	2 (SIA)
55	Romania (AAIB)	Romania (AAIB)	AAIB	3 (NAA/SIA)
56	Slovenia (AAIIO)	Slovenia (SIA)	AAIIO	2 (SIA)
57	Switzerland (AAIB)	Switzerland (SIA)	AAIB	3 (NAA/SIA)
58	EUROPEAN COMMISSION - JRC	EUROPEAN COMMISSION	EUROPEAN COMMISSION	3 (NAA/SIA)
59	Finland (AIB)	Finland (SIA)	AIB	2 (SIA)
60	Lithuania (AIB)	Lithuania (SIA)	AIB	2 (SIA)
61	Estonia (AIB)	Estonia (SIA)	AIB	2 (SIA)
62	Slovakia (AAIIB)	Slovakia (AMIA)	AAIIB	3 (NAA/SIA)
63	United Kingdom (AAIB)	United Kingdom (AAIB)	AAIB	2 (SIA)
64	Malta (AIB)	Malta (AIB)	AIB	2 (SIA)
65	France (BEA)	France (BEA)	BEA	3 (NAA/SIA)
66	Spain (CIAIAC)	Spain (CIAIAC)	CIAIAC	3 (NAA/SIA)
67	Other entities	EASA_ASR	Other	3 (NAA/SIA)
68	Other entities	IT support only	Other	3 (NAA/SIA)

ID	AUTHORITY NAME	DISPLAY NAME	AUTHORITY CODE	AUTHORITY TYPE
69	Other	Non - ENCASIA	Other	2 (SIA)
70	Other entities	UKAB (non-EU State)	Other	1 (NAA)

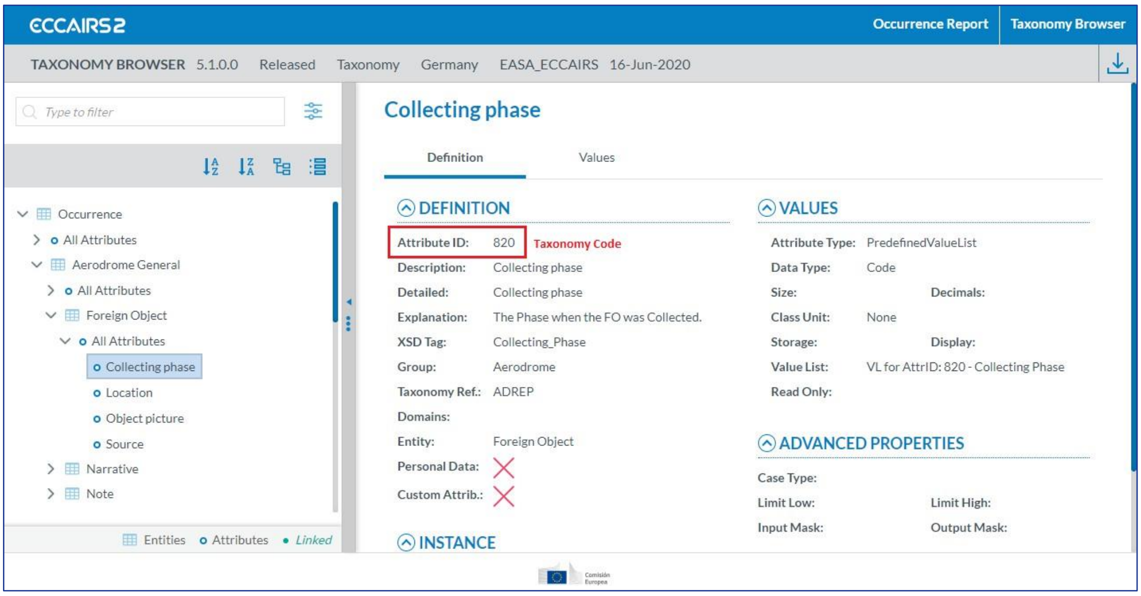
12.4 Reports JSON

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

12.4.1 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

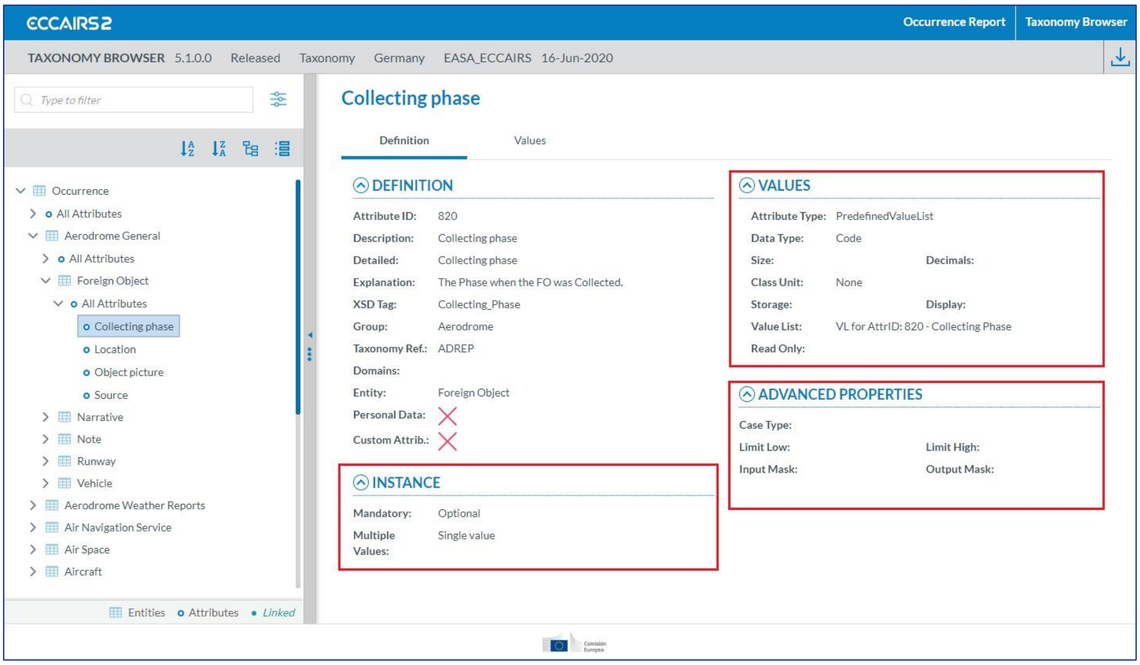
The JSON itself follows these basic guidelines:

- 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"**

12.4.2 Definition of attributes

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



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

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 details for all the existing Data Types.

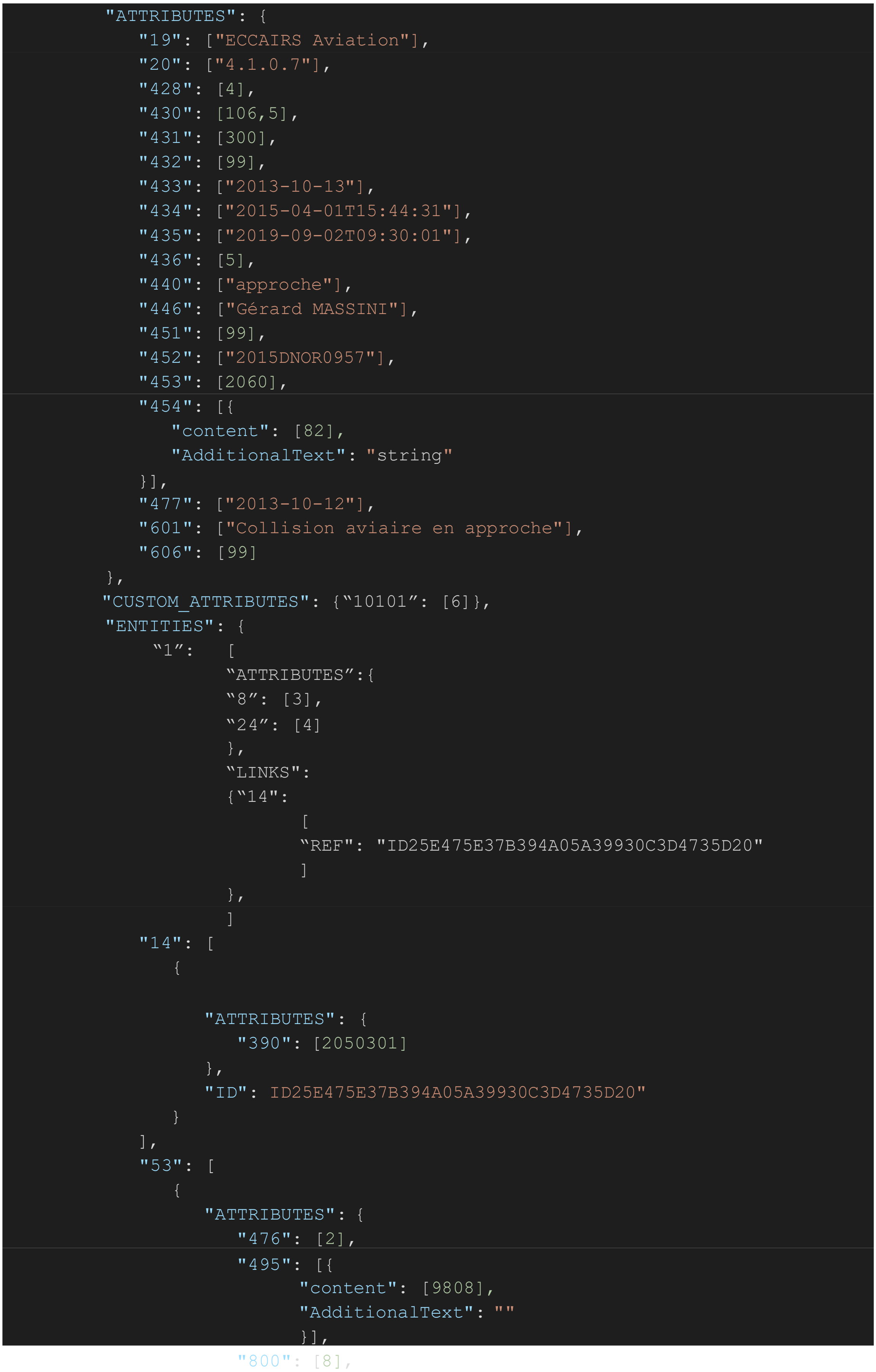
ECCAIRS-DataTypes.
pdf

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

12.4.3 Example

The following code is an 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":"e1xydGYxXG-
Fuc2lcYW5zaWNwZzEyNTJcZGVmZjBcZGVmbGFuZzEwNDB7XGZvbnR0Ymx7XGYwXGZuaWxcZ-
mNoYXJzZXQwIEFyaW-FsO319DQpcdmlld2tpbmQ0XHVjMVxwYXJkXGZz-
MjRccGFyDQpccGFyDQp9DQo="
                        }]
                     }
                  },
                  "ID": "#id_number#"
               }
            ]
         }
```