EUROPEAN MEDICINES AGENCY
SCIENCE MEDICINES HEALTH

Information Management Division

# ESMP API specifications

# Table of contents

# 1. Scope

The design of the API must address the current functional scope, with possibility for extending the scope to further datasets and operations on those datasets.

The current scope covers the submission of:

- data about national demand by NCA during MSSG-led preparedness period;
- data about routine shortage by MAH during crisis preparedness period.

## 1.1. Conform to the definitions of the rules in place

The specifications for the datasets and their validation rules were defined based on a coarse definition of the information to be submitted. As an example, the *availability* and *routine* MAH datasets each contain information of different natures as shortage information, volume of sales, market share, alternative therapies, free text information and other attributes. The business validation rules define that those data elements, all of different natures, are consolidated into a single record (row in the spreadsheet), are validated as a whole, and as a consequence cannot be submitted independently.

*The current specification and implementation of the API will however provide an additional format (JSON) to already specified format (spreadsheet) and with the same capability.*

## 1.2. Versioning

The initial version of the API is assigned the value `v1` which is the API version that mirrors the spreadsheet upload, as per the above section. The corollary is that the API must remain in-line over time with the validation rules and conformance.

## 1.3. Asynchronous write operations

For write operations, the server handles the request **asynchronously** and returns the information needed by the client to query for the completion status of the processing of the data.

The identifier of the status to query is returned in response header `Location`.

## 1.4. Authentication and authorisation

The target for the authentication and authorisation mechanism holds for a machine-to-machine approach with the concept of **client id** and **client secret** leading to OAuth 2.0 as in place for other systems such as UPD.

The registration to obtain access to ESMP API for your organisation must be done through our EMA service desk, for which detailed instructions will follow. There is one registration for the non-production environment and one registration for the production environment.

# 2. Specification

In the workshops on interoperability, while we got away from having to deliver an XML-format API, there was a request by NCA to provide a clear technical specification of the data elements in a way that is similar to XSD (XML schema definition), which is widely used by many organisations. For JSON format, the best equivalent to XSD is JSON schema (https://json-schema.org/).

The technical specifications are supported by OpenAPI 3.0.1 that supports a dialect of JSON schema.

## *2.1. National demand - NCA*

National demand is a dataset that is in the scope of **NCAs** during the **MSSG-led preparedness**.

### 2.1.1. Spreadsheet format representation

The structure of the data as spreadsheet can be represented as in the table below; it is a representation of a **subset of the data elements**, as the definition of all the columns can be found under https://www.ema.europa.eu/en/documents/other/european-shortages-monitoring-platform-esmp-implementation-guide-national-competent-authorities_en.pdf, but it represents all the attributes that are relevant for the submission of the data.

| | PMS ID | Month 1 | Month 2 | Month 3 | Month 4 | Month 5 | Month 6 |
|---|---|---|---|---|---|---|---|
| 1 | 600000750315 | 19750 | 150000 | 32000 | 300000 | 160000 | 100000 |
| 2 | 600000451015 | 10000 | 160000 | 42000 | 310000 | 170000 | 110000 |

Where "*Month n*" represents the *current month of the year (= month of submission)* + *n*. So for the medicinal product with PMS id 600000750315, if the data is submitted any day in *September 2024*, the value for Month 1 (19750) applies for the month of *October 2024*, Month 2 (*150000*) applies for the month of *November 2024* …

### 2.1.2. JSON format representation

The version 1 of the representation of the data follows closely the representation above, with the notable difference of the information on the months, in which they are represented explicitly as a *period*.

Note that the full specification in OpenAPI format is part of attached documents.

```
{
  "demands": {
    "items": [
      {
        "quantity": 19750,
        "period": {
          "start": "20241001",
          "end": "20241031"
        },
        "subject": "600000750315"
      },
      {
        "quantity": 150000,
        "period": {
          "start": "20241101",
          "end": "20241130"
        },
        "subject": "600000750315"
      },
      {
        "quantity": 32000,
        "period": {
          "start": "20241201",
          "end": "20241231"
        },
        "subject": "600000750315"
      },
      {
        "quantity": 300000,
        "period": {
          "start": "20250101",
          "end": "20250131"
        },
        "subject": "600000750315"
      },
      {
        "quantity": 160000,
        "period": {
          "start": "20250201",
          "end": "20250228"
        },
        "subject": "600000750315"
      },
      {
        "quantity": 100000,
        "period": {
          "start": "20250301",
          "end": "20250331"
        },
        "subject": "600000750315"
      },
      {
        "quantity": 10000,
        "period": {
          "start": "20241001",
          "end": "20241031"
        },
        "subject": "600000451015"
      },
      {
        "quantity": 160000,
        "period": {
          "start": "20241101",
          "end": "20241130"
        },
        "subject": "600000451015"
      },
      {
        "quantity": 42000,
        "period": {
          "start": "20241201",
          "end": "20241231"
        },
        "subject": "600000451015"
      },
      {
        "quantity": 310000,
        "period": {
          "start": "20250101",
          "end": "20250131"
        },
        "subject": "600000451015"
      },
      {
        "quantity": 170000,
        "period": {
          "start": "20250201",
          "end": "20250228"
        },
        "subject": "600000451015"
      },
      {
        "quantity": 110000,
        "period": {
          "start": "20250301",
          "end": "20250331"
        },
        "subject": "600000451015"
      }
    ]
  }
}
```

The schema behind the structure comes in dialects of JSON schemas and implemented under OpenAPI 3.0.1.

## *2.2. Routine shortage - MAH*

Routine shortage is a dataset that is in the scope of **MAHs** during **preparedness period** and is only applicable for products under the centralised marketing authorisation procedure.

### 2.2.1. Spreadsheet format representation

The structure of the data as spreadsheet can be represented as in the table below; it is a representation of a **subset of the data elements**, as the definition of all the columns can be found under https://www.ema.europa.eu/en/documents/other/european-shortages-monitoring-platform-esmp-implementation-guide-marketing-authorisation-holders_en.pdf.

|  | Package PMS ID | Country of authorisation | Shortage status | Shortage start date | Shortage end date | Root cause of shortage |
|---|---|---|---|---|---|---|
| 1 | 326523 | BE | No shortage |  |  |  |
| 2 | 326523 | ES | Potential | 15/03/2025 | 31/12/2025 | 200000028683;200000028684 |
| 3 | 523255 | NL | Actual | 01/04/2024 | 30/04/2025 | 200000028683 |

### 2.2.2. JSON format representation

The version 1 of the representation of the data follows closely the representation above.

Note that the full specification in OpenAPI format is part of attached documents.

```
{
  "routineShortages": {
    "items": [
      {
        "subject": "326523",
        "shortageStatus": "No shortage",
        "authorisationCountry": "BE"
      },
      {
        "subject": "326523",
        "shortageStatus": "Potential",
        "shortagePeriod": {
          "start": "2025-03-01",
          "end": "2025-03-31"
        },
        "authorisationCountry": "ES",
        "shortageRootCauses": [
          "200000028683",
          "200000028684"
        ]
      },
      {
        "subject": "523255",
        "shortageStatus": "Actual",
        "shortagePeriod": {
          "start": "2024-04-01",
          "end": "2025-04-30"
        },
        "authorisationCountry": "NL",
        "shortageRootCauses": [
          "200000028683"
        ]
      }
    ]
  }
}
```

### 2.3. *Submission status*

The resource submission status represents the status of the processing of a dataset submitted. The processing of the datasets is asynchronous and must be retrieved through a separate request.

An instance of submission status has the following attributes:

- `submissionId`: the unique identifier assigned by the ESMP API to the submission if the submission was accepted.
- `status`: the status of the processing of the submission (e.g. `Success`, `InProgress`, `Failed`).
- `diagnostics`: an array of elements (diagnostics) that contains information about validation issues, if any.

Example for the **successful processing** of the dataset:

```
{
  "submissionId": "8c9da296-b936-43c1-8e91-caa77e841014",
  "status": "Success",
  "diagnostics": null
}
```

Example for the **unsuccessful processing** of the dataset:

```
{
  "submissionId": "8ce17a45-7093-452d-aeb7-25bacf9bea7e",
  "status": "Failed",
  "diagnostics": [
    "Value for 'shortageRootCauses' must be in RMS list '200000028648', item[12]"
  ]
}
```

# 3. OpenAPI documents

The OpenAPI documents are available for MAH and NCA in both JSON and YAML, they are provided outside of this document.

NCA:

- openapi-esmp-nca-v1.json
- openapi-esmp-nca-v1.yaml

MAH:

- openapi-esmp-mah-v1.json
- openapi-esmp-mah-v1.yaml

# 4. Sequence diagrams

The diagrams represent the sequence of interactions between an **API client**, **ESMP API** and **Microsoft Entra ID**, for the submission of a **dataset X**, which is a placeholder name as the sequences apply to any of the datasets in scope (**national demand** and **routine shortage**).

## *4.1. Successful sequence*

In step 3, the JSON payload posted to the endpoint is well-formed and valid as per the JSON schema validation, and is **accepted** for further processing by the ESMP API. The processing includes:

- execution of the **business data validation** according to the rules in the implementation guide for the corresponding dataset;
- persistence of the data in ESMP.

For the API client to know whether the steps above completed, it must use the value of `submissionId` returned in step #4 and poll the endpoint for the submission status.

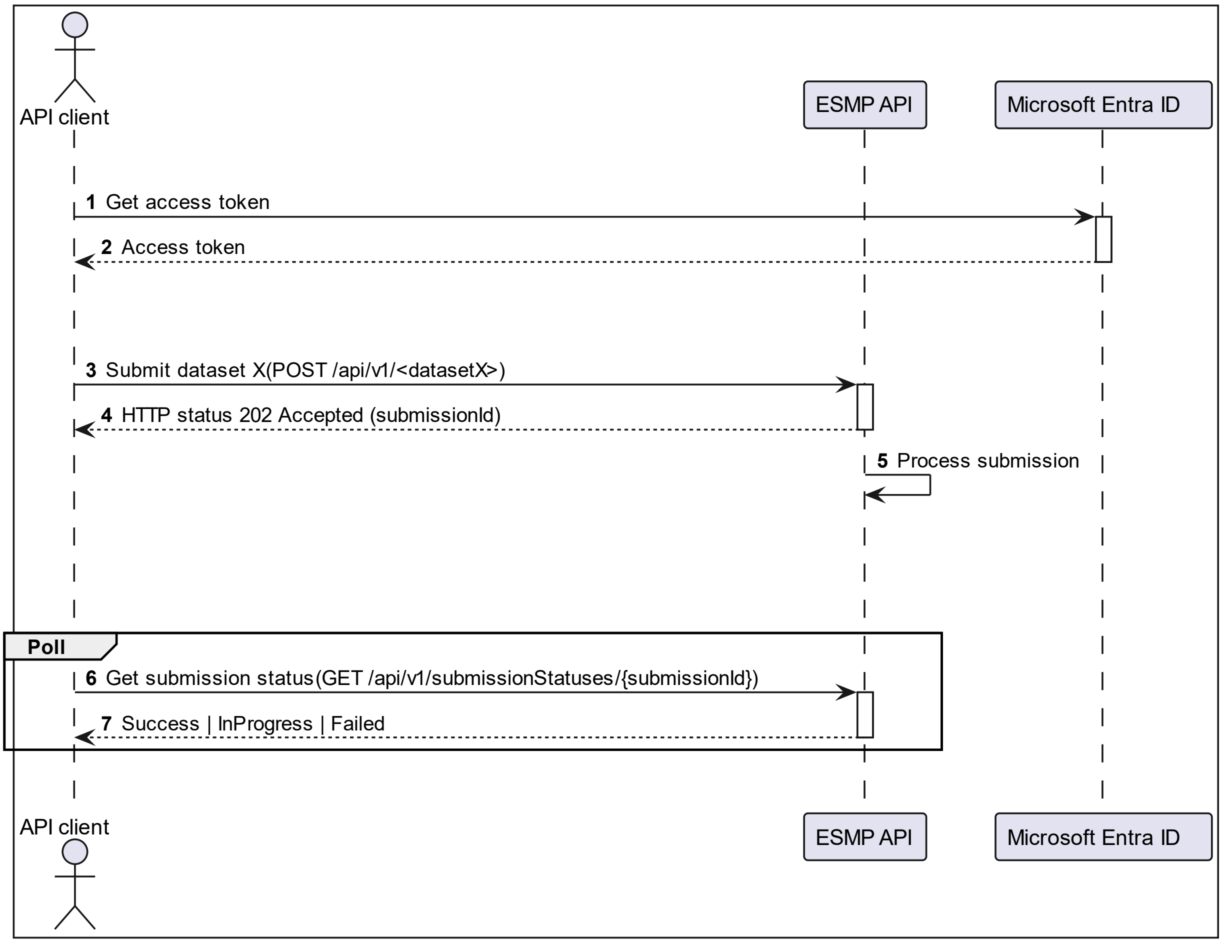
API client
ESMP API
Microsoft Entra ID
1 Get access token
2 Access token
3 Submit dataset X(POST /api/v1/<datasetX>)
4 HTTP status 202 Accepted (submissionId)
5 Process submission
Poll
6 Get submission status(GET /api/v1/submissionStatuses/{submissionId})
7 Success | InProgress | Failed
API client
ESMP API
Microsoft Entra ID

*Figure 1 Sequence diagram for the submission of a dataset, with success.*

## *4.2. Unsuccessful sequence*

In step 3, the JSON payload posted to the endpoint is not well-formed or is invalid as per the JSON schema validation. The response will have the **HTTP status 400 (Bad Request)** and contain details about the reason of the error. Note that in this case, there is no `submissionId` provided in the response.

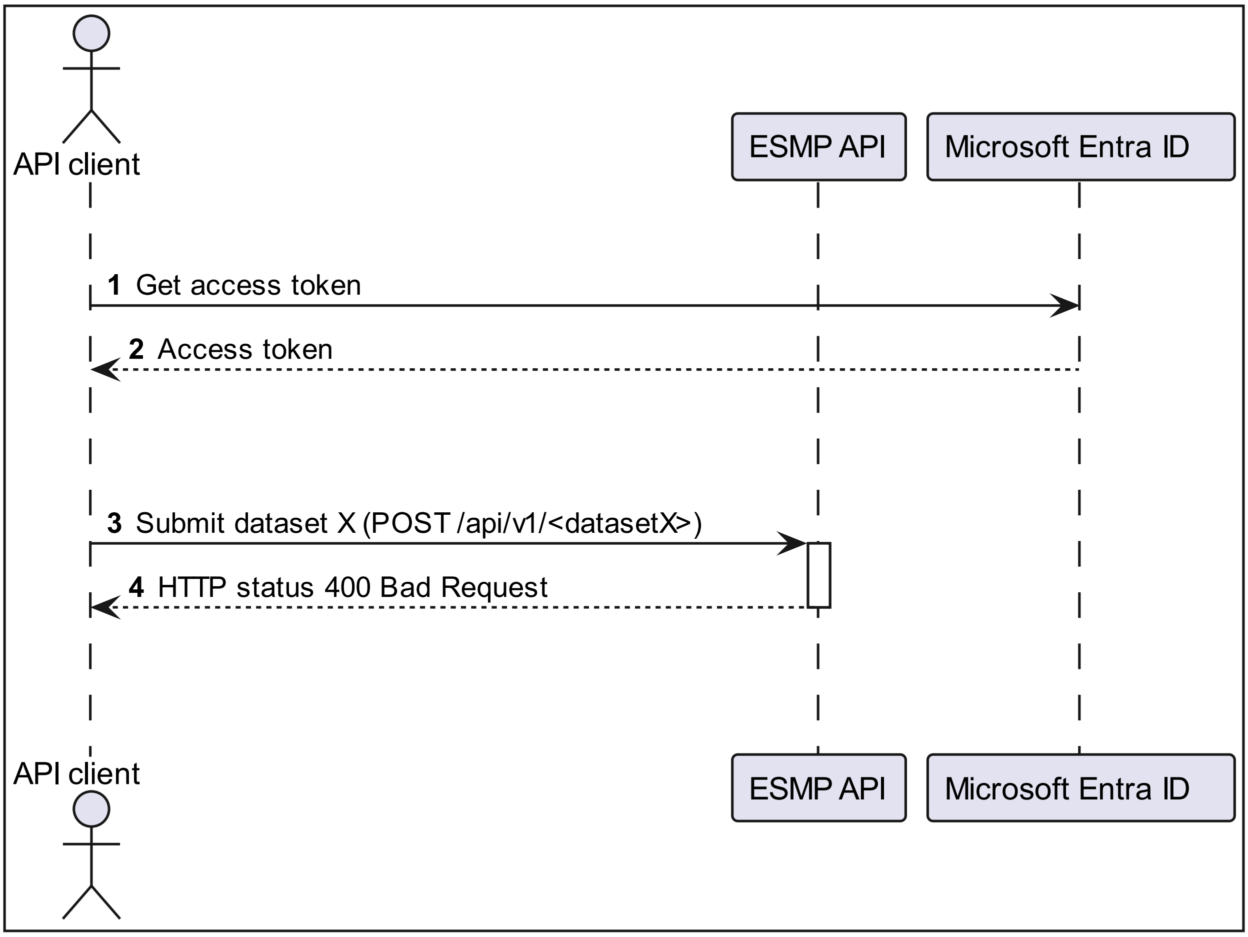

*Figure 2 Sequence diagram for the submission of a dataset, with non-valid JSON.*

## 5. Non-production environment

EMA will open a non-production environment during Q1/2025 to both NCA and MAH on a voluntary basis and benefit from EMA support for questions or issues.