

API - Getting started -

- [CATALOGUE endpoints](#)
- [STATISTICS endpoints](#)
- [SDMX 2.1 endpoints](#)
- [ASYNC endpoints for asynchronous response delivery](#)
 - [Purpose of ASYNC endpoints](#)
 - [ASYNC endpoints](#)
- [How to implement asynchronous requests?](#)
 - [Example](#)
 - [Step 1: Initial request](#)
 - [Step 2: Get the current status of the request](#)
 - [Step 3: Get the data](#)
 - [Errors returned](#)
 - [Data not yet ready](#)
 - [Invalid key](#)

The API exposes different endpoints for different purposes:

- [CATALOGUE](#), which is used for data discovery;
- [STATISTICS](#), which replaces Eurostat JSON web service;
- [SDMX 2.1](#), used for [synchronous](#) metadata and data queries;
- [ASYNC](#), used for [asynchronous](#) data delivery.

CATALOGUE endpoints

The CATALOGUE endpoint serves data discovery purposes and is a programmatic access to catalogues of data. The CATALOGUE endpoint is **only available through REST**.

Agency	Catalogue endpoint URL
DG COMP	https://webgate.ec.europa.eu/comp/redisstat/api/dissemination/catalogue/catalogue.wadl
DG EMPL	https://webgate.ec.europa.eu/empl/redisstat/api/dissemination/catalogue/catalogue.wadl
DG GROW	https://webgate.ec.europa.eu/grow/redisstat/api/dissemination/catalogue/catalogue.wadl

STATISTICS endpoints

API Statistics is a web service for data query where parameters don't have to respect a specific order opposed to API SDMX 2.1.

This Web Service:

- provides data in JSON-stat 2.0 format;
- supports only REpresentation State Transfer (REST) protocol;
- delivers responses in English, French or German.

SDMX 2.1 endpoints

The API SDMX endpoints are a programmatic access to [metadata](#) and [data](#), with the possibility to:

- Get a complete list of publicly available datasets
- Get the complete structure definition for a specific dataset
- Download a dataset in full, or a subset of a specific dataset. This supports customized requests for data.

Processing time and the data response depend on the query, i.e.

- A request for a small subset of a dataset could result in a response with few kilobytes, and can be returned within seconds
- A request for full dataset could result in a response of up to several megabytes, and can be returned after several minutes. When submitting this kind of request, asynchronous requests (described hereafter), may be useful.



This page mentions the SDMX 2.1 endpoint with SOAP protocol [only for reference](#). Please refer to the WSDL of each SOAP endpoint for more information.

The endpoints for SDMX artefacts are in line with [SDMX technical specifications for version 2.1](#).

These endpoints are also extended for returning artefacts in formats other than SDMX 2.1:

- JSON-stat
- TSV
- SDMX-CSV



Use the SDMX 2.1 endpoint to get content in JSON-stat, TSV or SDMX-CSV data format.

Agency	SDMX 2.1. Endpoints URL's
Eurostat	REST: https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/sdmx-rest.wadl SOAP: https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/sdmx-ws?wsdl REST: https://ec.europa.eu/eurostat/api/comext/dissemination/sdmx/2.1/sdmx-rest.wadl (Comext and Prodcum datasets) SOAP: https://ec.europa.eu/eurostat/api/comext/dissemination/sdmx/2.1/sdmx-ws?wsdl (Comext and Prodcum datasets)
DG COMP	REST: https://webgate.ec.europa.eu/comp/redisstat/api/dissemination/sdmx/2.1/sdmx-rest.wadl SOAP: https://webgate.ec.europa.eu/comp/redisstat/api/dissemination/sdmx/2.1/sdmx-ws?wsdl
DG EMPL	REST: https://webgate.ec.europa.eu/empl/redisstat/api/dissemination/sdmx/2.1/sdmx-rest.wadl SOAP: https://webgate.ec.europa.eu/empl/redisstat/api/dissemination/sdmx/2.1/sdmx-ws?wsdl
DG GROW	REST: https://webgate.ec.europa.eu/grow/redisstat/api/dissemination/sdmx/2.1/sdmx-rest.wadl SOAP: https://webgate.ec.europa.eu/grow/redisstat/api/dissemination/sdmx/2.1/sdmx-ws?wsdl

ASYNCR endpoints for asynchronous response delivery

Purpose of ASYNCR endpoints

The ASYNCR endpoint is a programmatic access for asynchronous responses to large data requests.

For the different endpoints listed above, data can be returned either synchronously or asynchronously:

- Synchronously: the data is returned directly in the response to the request. This is the default operation
- Asynchronously: the data is not returned directly in the response. Instead a key is returned in the response which allows to access the data through the async endpoint once it is available.

The decision whether to deliver the data synchronously or asynchronously is related to factors such as the complexity of the query and the volume of the data (number of rows) to be returned.

ASYNCR endpoints

Agency	Asynchronous Endpoints URL's
Eurostat	REST: https://ec.europa.eu/eurostat/api/dissemination/1.0/async/async.wadl REST: https://ec.europa.eu/eurostat/api/comext/dissemination/1.0/async/async.wadl
DG COMP	REST: https://webgate.ec.europa.eu/comp/redisstat/api/dissemination/1.0/async/async.wadl
DG EMPL	REST: https://webgate.ec.europa.eu/empl/redisstat/api/dissemination/1.0/async/async.wadl
DG GROW	REST: https://webgate.ec.europa.eu/grow/redisstat/api/dissemination/1.0/async/async.wadl

How to implement asynchronous requests?

The asynchronous delivery process can be summarized as follows:

1. **Step 1.** A client issues a request to one of these asynchronous endpoints: CATALOGUE, SDMX 2.1. The API returns a response indicating asynchronous delivery pattern, with a unique key
2. **Step 2.** The client issues to the asynchronous endpoint at regular interval a request with the unique key, to enquire about the availability readiness of the requested data
3. **Step 3.** Once the data is available, the client receives the data attached in the response to the request for the unique key

Example

Step 1: Initial request

The initial request must be sent to one of the asynchronous endpoints [mentioned above](#).

The response to such a request is structured as follows:

```

<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <S:Body>
    <ns0:syncResponse xmlns:ns0="http://estat.ec.europa.eu/disschain/soap/extraction">
      <dataResponse>
        <queued>
          <id>98de05ea-540a-43d3-903b-7c9e14faf808</id>
          <status>SUBMITTED</status>
        </queued>
      </dataResponse>
    </ns0:syncResponse>
  </S:Body>
</S:Envelope>

```



The <id> value ("98de05ea-540a-43d3-903b-7c9e14faf808" in this example), is the key to use for checking data availability against the asynchronous endpoint.

Step 2: Get the current status of the request

The status of a request that is processed asynchronously can be one of the following values:

Value	Meaning
SUBMITTED	The request is submitted for processing
PROCESSING	The request is currently being processed
AVAILABLE	The data is available for download
EXPIRED	The request was processed but the data is no longer available
UNKNOWN_REQUEST	In case the key provided cannot be matched to a request

The current status of a given request can be obtained:

- Via a REST request:
 - **Request URL:** `https://<api_base_uri>/1.0/async/status/<id>`
 - **Example of URL request:** <https://ec.europa.eu/eurostat/api/dissemination/1.0/async/status/98de05ea-540a-43d3-903b-7c9e14faf808>
- Via a SOAP request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:asy="http://estat.ec.europa.eu/disschain/soap/asynchronous" xmlns:asyl="http://estat.ec.europa.eu/disschain/asynchronous">
  <soapenv:Header/>
  <soapenv:Body>
    <asy:async>
      <asyl:status>
        <asyl:key>98de05ea-540a-43d3-903b-7c9e14faf808</asyl:key>
      </asyl:status>
    </asy:async>
  </soapenv:Body>
</soapenv:Envelope>

```

This request may provide different results, depending on the current status of the request:

- **DATA_NOT_YET_AVAILABLE:** As long as the request is not processed/finished, the following result will be returned:

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <S:Body>
    <S:Fault>
      <faultcode>100</faultcode>
      <faultstring>DATA_NOT_YET_AVAILABLE: Requested data is not yet available for download. Check the status of your request.</faultstring>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

- **AVAILABLE:** The request is processed/finished. When the query is fully executed, the returned status will be AVAILABLE and the following result will be returned:

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <S:Body>
    <ns0:asyncResponse xmlns:ns1="http://estat.ec.europa.eu/disschain/asynchronous" xmlns:ns0="http://estat.ec.europa.eu/disschain/soap/asynchronous">
      <ns1:status>
        <ns1:key>98de05ea-540a-43d3-903b-7c9e14faf808</ns1:key>
        <ns1:status>AVAILABLE</ns1:status>
      </ns1:status>
    </ns0:asyncResponse>
  </S:Body>
</S:Envelope>
```

Step 3: Get the data

When the results are AVAILABLE, it is possible to download the data. Data can be obtained:

- Via a REST request:
 - **Request URL:** `https://<api_base_uri>/1.0/async/data/<id>`
 - **Example of URL request:** <https://ec.europa.eu/eurostat/api/dissemination/1.0/async/data/98de05ea-540a-43d3-903b-7c9e14faf808>
- Via a SOAP request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:asy="http://estat.ec.europa.eu/disschain/soap/asynchronous" xmlns:asyl="http://estat.ec.europa.eu/disschain/asynchronous">
  <soapenv:Header/>
  <soapenv:Body>
    <asy:async>
      <asyl:data>
        <asyl:key>98de05ea-540a-43d3-903b-7c9e14faf808</asyl:key>
      </asyl:data>
    </asy:async>
  </soapenv:Body>
</soapenv:Envelope>
```

When the results are returned, the data is attached to the SOAP response:

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <S:Body>
    <ns0:asyncResponse xmlns:ns1="http://estat.ec.europa.eu/disschain/asynchronous" xmlns:ns0="http://estat.ec.europa.eu/disschain/soap/asynchronous">
      <ns1:data>
        <ns1:key>98de05ea-540a-43d3-903b-7c9e14faf808</ns1:key>
        <ns1:include href="cid:1568a7ca-792b-4730-bd81-a0a152eafd96"/>
      </ns1:data>
    </ns0:asyncResponse>
  </S:Body>
</S:Envelope>
```

Errors returned

Data not yet ready

As long as the data is not ready, the returned SOAP result will be:

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <S:Body>
    <S:Fault>
      <faultcode>100</faultcode>
      <faultstring>DATA_NOT_YET_AVAILABLE: Requested data is not yet available for download. Check the status of your request.</faultstring>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

Invalid key

If the key provided is not valid, the returned SOAP result will be:

```
<S:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <S:Body>
    <S:Fault>
      <faultcode>100</faultcode>
      <faultstring>UNKNOWN_REQUEST: Unknown request.</faultstring>
    </S:Fault>
  </S:Body>
</S:Envelope>
```