

# Web-API FAQ

- Starting
  - How can I start?
  - How can I download Public Datasets from ECMWF?
  - What is the difference between ECMWF Web-API and the Web User Interface?
  - Is the Web-API procedure easy?
  - How to download the Web-API library?
- Checking the availability
  - How can I see the contents of your archive?
  - Why some attributes (eg "parameters" or "time steps" etc) are grayed out and are not "checkable"?
  - Can you pass me an example on how the Datasets Web Interface could help me to understand the availability?
- Writing a Web-API request
  - What is a Web-API request ?
  - How to write my first Web-API script?
  - What is a MARS request?
- Monitoring your request
  - How long will it take for my request to complete?
  - How can I trace back an old request ?
  - Which is the best way to cancel a request?
  - Can I kill my request on my local environment (e.g. by CTRL+C)
  - My request has been queued (or active) for a long time. Do I have to kill it?
- Going further
  - Can I ask for data in "netcdf" format?
  - Can I request a limited area region?
  - How to write more complex scripts?
  - What are the differences between the full MARS client and API MARS service?
  - Can I use the Python requests to retrieve data using API MARS service ?
  - Can I use the Web-API to retrieve data directly from Metview?
  - How can I improve the performance of my request?
  - Why MARS activity is important?
- Limitations and restrictions
  - Is there any limitation on the volume or number of fields per request?
  - Is there any limitation on the selection of the period for a request on the Web User Interface?
  - Is there any limitation on the number of requests that a user can submit?
  - Is there any limitation in a single Web-API request?
  - Are there any data restrictions?
- Advanced topics
  - The Web-API flow chart
  - The Web-API flow in more details
  - Organise the output in separate files
  - MARS interpolation
- Problems

## Starting

### How can I start?



- Read the information on [ECMWF Web API Home](#)
- If you are a **public** user visit the [Access ECMWF Public Datasets](#) and follow the step-by-step guide
- If you are an **authorised** user you may wish to visit the [Access MARS](#) pages

### How can I download Public Datasets from ECMWF?



There are **2 ways** to access ECMWF public datasets:

1. The [Datasets Web Interface](#). In brief:
  - This interface will help you to navigate through the public datasets in a dynamic and user-friendly way.
  - After each selection, the web page is updated automatically in order to reflect the availability of the dataset.
  - The purpose of this interface is to help users navigate the content of the datasets, get familiar with it, extract sample data interactively and/or obtain the script to download data in batch. (see below)
2. The [Web-API](#) ie the **programmatic** way :
  - This is the **preferred** way for data download via scripting languages such as Python.

### What is the difference between ECMWF Web-API and the Web User Interface?

 Web-API is just the **programmatic** way to download data.

## Is the Web-API procedure easy?

- 
- The procedure is **simple**
  - You just need to write a request, and to submit it.
  - The service will keep you posted about your request status.
  - Once your request has been completed the requested data will be transferred from ECMWF server to your local server.

## How to download the Web-API library?

 Please see the [Web-API Downloads](#)

## Checking the availability

### How can I see the contents of your archive?

- 
- Our archive consists of several Public Datasets and each of them has its **own availability**.
  - Some of the datasets are **not homogeneous**. (eg different parameters are available for different periods etc)
  - The [Datasets Web Interface](#) will help you to navigate through the content of each dataset and to get familiar with the availability of the dataset..
  - For any kind of selections the system will update the attributes in a dynamic way to reflect the current availability. eg for some datasets if you change the "steps" you will spot that some parameters will be grayed out and can't be selected

### Why some attributes (eg "parameters" or "time steps" etc) are grayed out and are not "checkable"?

- 
- If an item is **not "checkable"** it means that it is **not available**.
  - The availability of an attribute is changing dynamically when you change the selections of another attribute (eg "steps") due to the **application dynamic features**.

### Can you pass me an example on how the [Datasets Web Interface](#) could help me to understand the availability?

-  Follow the steps below:
1. Click on the dataset [ERA-Interim \(Jan 1979 - present\)](#) to get a web page reflecting the availability of this specific dataset ( ie the period that this specific dataset is available, the forecast steps and the parameters that are available from this dataset etc.)
  2. From this page you have the option to select the attributes of your requests ie to define the "date", the "time" the "steps" and the "parameters" that you are interested in.
  3. In the case of the dataset above ("ERA Interim") if you select time "06:00:00" only, the system updates the availability of the "steps" dynamically. In that case only the "step" 00 is available (ie checkable).
  4. Additionally if you select "step" 00 you will see that only a subset of the parameters are now available and only these parameters can now be checked.

 Spend some time to understand how it works, make some selections and check how the page is updated dynamically

## Writing a Web-API request

### What is a Web-API request ?



Presumably before you start using Web-API you need to be able to write your own "request" which describes the data you wish to download. For that you need :

- to have an idea of what is available from our archive
- to decide what you wish to download
- to know how to write the request

## How to write my first Web-API script?



You may start with the examples available on [ERA-interim sample scripts](#) or by creating your request using our [Datasets Web Interface](#) . Please note the following:

- We strongly advice you to **start** with a **simple request**. ( 1-2 parameters 1 time step 1-2 steps etc)
- In case of **Python** the request will be a dictionary with "keys" and "values" that represent your selection. (eg "step": "00", "time": "00")
- The request is strongly connected to the availability of the data
- Use the example below as a basis to write your own script

```
#!/usr/bin/env python
from ecmwfapi import ECMWFDataServer

server = ECMWFDataServer()

server.retrieve({
    "stream": "oper",
    "levtype" : "sfc",
    "param": "165.128/41.128",
    "dataset" : "interim",
    "step": "0",
    "grid" : "0.75/0.75",
    "time" : "00",
    "date" : "2013-09-01/to/2013-09-30",
    "type" : "an",
    "class" : "ei",
    "target" : "data.grib"
})
```

## What is a MARS request?



- MARS stands for Meteorological Archive System. MARS is the ECMWF archival system which handles the requests that are submitted externally through the Web-API.
- A MARS request is basically a request for retrieving data from MARS using a specific language
- For more information about MARS you may visit the [MARS user documentation](#).
- An easy way to create your own MARS request is by using our [Datasets Web Interface](#) .
- Make some selections and click on "[View the MARS request](#)". The system will automatically translate your selections to a MARS request.
- This request can be used as a pilot to build your WebAPI script.

In the case of ERA Interim an example could be:

### era-interim.py

```
#!/usr/bin/env python
from ecmwfapi import ECMWFDataServer

server = ECMWFDataServer()

server.retrieve({
    'stream' : "oper",
    'levtype' : "sfc",
    'param' : "165.128/41.128",
    'dataset' : "interim",
    'step' : "0",
    'grid' : "0.75/0.75",
    'time' : "00",
    'date' : "2013-09-01/to/2013-09-30",
    'type' : "an",
    'class' : "ei",
    'target' : "interim_2013-09-01to2013-09-30_00.grib"
})
```

### era-interim.mars

```
retrieve,
  stream=oper,
  levtype=sfc,
  param=165.128/41.128,
  dataset=interim,
  step=0,
  grid=0.75/0.75,
  time=00,
  date=2013-09-01/to/2013-09-30,
  type=an,
  class="ei",
  target="interim_2013-09-01to2013-09-30_00.grib"
```

## Monitoring your request

### How long will it take for my request to complete?

 The request may take some time ( from some minutes, to many hours ) to complete, depending on some factors and limitations. (eg Nr of the requests you have submitted, Nr of the total active requests running currently, availability of the resources involved and most importantly [your request efficiency](#) )

### How can I trace back an old request ?

 [Your job list](#) can be used for **tracing back** tour requests.

You can also use the [Web-API activity](#) and the [MARS activity page](#). Please see [Why MARS activity is important](#)

### Which is the best way to cancel a request?

- 
- If you wish to **cancel** a request please visit [your job list](#) and click on the cancel option.
  - Once you have **cancelled** it, the request's status will become **aborted**
  - Cancelling is not a recommended approach because it may affect the performance of your other submitted requests.
  - See also next FAQ.

## Can I kill my request on my local environment (e.g. by CTRL+C)



- ⚠️ If you just kill a Web-API request on your **local environment** (e.g. by CTRL+C), the corresponding job, on the web-API service level, is **NOT cancelled but still running**.
- Your request will continue to be **active** in [your job list](#).
- See also the previous FAQ to check how you can cancel your request properly.

## My request has been queued (or active) for a long time. Do I have to kill it?



- A request may take some time to complete depending on many factors and limitations.
- Visit [your job list](#) to check the status of the request
- You may need to visit our [Web-API Troubleshooting](#) pages for more information.

## Going further

### Can I ask for data in "netcdf" format?



- Yes 😊 You only have to add in your request **"format" : "netcdf"**

### Can I request a limited area region?



- Yes 😊
- If you have already set the "grid" keyword in your request, you can add the **"area" : "coordinates"** keyword. You can set pre-defined area regions e.g. europe or set the area using coordinates North/West/South/East.
- You may also wish to visit the MARS area keyword for more information: [Post-processing keywords](#).
- See examples below.

```
"area": "europe",
```

```
#area: N/W/S/E  
  
#europe  
"area": "75/-20/10/60",  
  
#africa  
"area": "40/-20/-40/60",
```

## How to write more complex scripts?



Once you are happy with your 1st request we are suggesting you the following:

1. Try to write a new request for the **same dataset** by adding more items :
  - a. Add **more parameters** and submit
  - b. Add **more steps** and submit again
  - c. Add **more times** and submit again.
  - d. ⚠️ You always need to keep in mind that you have to check the **availability** of your new selections before you submit your request.
2. Try to check the availability of other datasets and write a simple script for another dataset.
3. Continue with other datasets

## What are the differences between the full MARS client and [API MARS service](#)?

**i** Some users might be used to the full MARS client to access MARS. There are some limitations if you use the [Access MARS](#) service compared to the full MARS client:

- You can only use **retrieve** or **list** verbs. compute, read, write... are not available
- You can only execute **one action** at the time
- You can **not** use **multi-target** using the keywords, only one final single file can be transferred at the end of the execution. See how to [Organise the output in separate files](#)
- You can only access to the **archived data** (no fdb access)

## Can I use the Python requests to retrieve data using [API MARS service](#) ?

**i** Yes! You can use the **Access MARS service** through ECMWF WebAPI using a Python script. This will allow any user with access to MARS to retrieve data. The syntax is:

```
server = ECMWFService("mars")
server.execute({ "python":"query" }, "target")
```

Example:

```
#!/usr/bin/env python
from ecmwfapi import *

server = ECMWFService("mars")
server.execute(
    {
        "class": "od",
        "date": "-1",
        "expver": "1",
        "levtype": "sfc",
        "param": "167.128",
        "step": "0/to/240/by/12",
        "stream": "oper",
        "time": "00",
        "type": "fc"
    },
    "target.grib")
```

## Can I use the Web-API to retrieve data directly from [Metview](#)?

**i** Metview versions from 4.4.3 onwards have support for the Web API service. More information: [Using the MARS Web API from Metview](#).

## How can I improve the performance of my request?

- i**
- This is **extremely useful** if you wish to download **big amounts** of data.
  - **Efficiency tips:**
    - **Smaller Nr of tapes** (a request accesses) means **better request efficiency**.
    - Requests for data which is **online** in general **go faster**
    - **Ideally** requests should not access more than 1 tape.
  - Please read the [retrieval efficiency pages](#)
  - Please also read [why MARS activity is important](#)

## Why MARS activity is important?

  In MARS activity you can see only the requests that are currently **"active" on the Web API level**. A **"queued"** request is **not visible** in MARS activity.

 **MARS activity** is important for many reasons:

- It can give you a good insight of your current request status and of your request details (like Nr of fields, Nr of tapes accessed etc)
- It can give you a very good indication of why your request is **slow**
- It can provide you a lot of information that you need to improve the efficiency of your request like
  - **how many tapes** your request is accessing (in the MARS log output)
  - the reason why a requests is queued.
  - other users requests.

#### Other users requests

 Many requests submitted by other users may be running in parallel consuming the services resources (MARS slots, Web-API slots, Tapes access slots etc). These requests may be very useful to understand the **efficiency factors**.

- **Compare** the activity of your requests to the **other users activity** using the [MARS activity](#) page.
- See the **reason** that some requests are **"queued"** and **how long they** have been "queued".
- Have a look on the MARS logs about **limitations** like:
  - *"In marsod-core, the total number of Requests from ECMWF Web API with 1 tape mount [source] is **limited to 6**"*
  - *etc*

## Limitations and restrictions

### Is there any limitation on the volume or number of fields per request?

 Yes. Check the current limits here: [News feed](#)

The limit of the target file size is also shown in the first lines of the output. e.g.

```
mars - INFO - 20180515.134138 - Maximum retrieval size is XX G
```

### Is there any limitation on the selection of the period for a request on the Web User Interface?

- 
- We have limited the time period selection for retrievals from the Web User Interface (UI), in order to **improve** the service performance but also to help the users to learn how to submit efficient MARS request.
  - The Web User Interface is for **sampling** and **browsing** only.
  - If you wish to download **long periods** of data we would strongly recommend you to use the **Web-API** and to follow the [retrieval efficiency pages](#)
  - The current limits can be found on our [News feed](#)

### Is there any limitation on the number of requests that a user can submit?

- 
- **20 queued** requests per user (see also [Troubleshooting#user\\_queued\\_limit\\_exceeded](#))
  - **3 active** requests per user

 You may also wish to check [how to trace back an old request](#)

### Is there any limitation in a single Web-API request?



There are **limitations** on

- the **total number of fields** and
- the **size** of the target file.

Check the current limits here: [News feed](#)

## Are there any data restrictions?

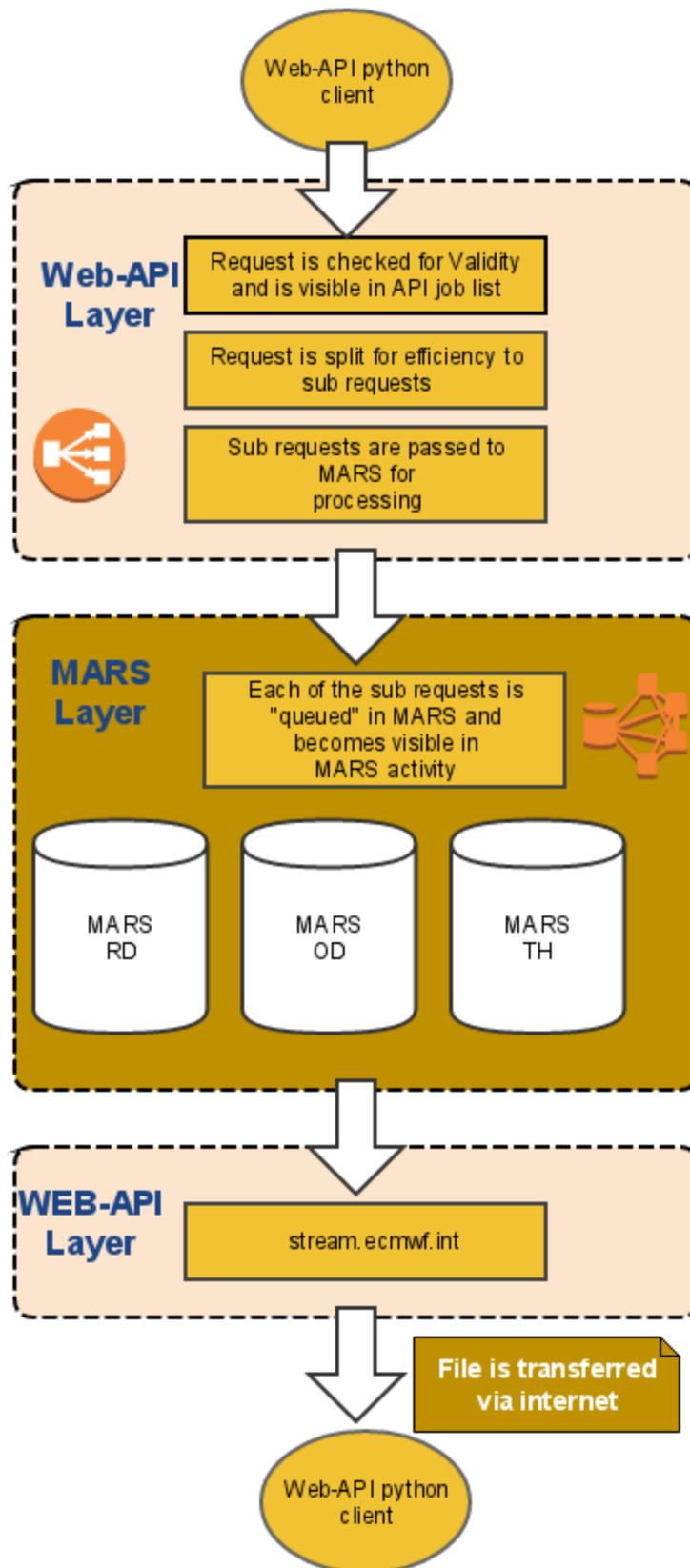


- Yes, you can only access datasets that you have been given access rights to. If you try to access datasets that you have no access rights for, your request will be **"aborted"**.
- For more information please have a look at [MARS access restrictions](#)

## Advanced topics

### The Web-API flow chart





## The Web-API flow in more details



### 1. Request submitted by the user and arrives in the the Web-API queue

- The request arrives to **the service back-end** and a **new record** is added in the service database. (**MongoDB**).
- At this phase the request gets the status "**queued**" and it becomes **visible** on [API jobs list](#) and on [Web-API activity](#)
- **Please note:**
  - The request is **not yet visible** on the [MARS activity](#)
  - Each **request** is stored in mongoDB. The corresponding record includes all the request's details (status, timers, host, user etc).
  - The request record is updated **periodically** to reflect its **current** status.
- The "**queuing time**" depends on several factors, ie priorities, limitations, current load etc
- Please note that to improve the service some **QoS** limitations and priorities are applied, like:
  - The total maximum number of active requests
  - The maximum number of active requests per user
  - The maximum number of queued requests per user
  - etc
- Once an API slot is available and if all the QoS rules are met, the request will become "**active**" on the Web-API level.

### 2. Request becomes active on Web-API level

- Split: On the Web-API level**, the request is **split** into a number of **smaller** requests to achieve a better **performance from the MARS point of view**.
  - For instance if a request is requesting one year of "ERA interim" daily data it is split per **month**. ie it is split into 12 requests.
- Validate:** Each of these **sub-requests** are validated and passed to a **MARS client** running on one of the available hosts. The hosts' names are visible in both [Web-API activity](#) and [MARS activity](#)
- Create file:** Once all the sub requests have **completed successfully** the corresponding sub-files are merged into a final one and the request status becomes **complete**
  - If something goes wrong during the procedure above the request status will get the value "**aborted**"

### 3. Request becomes active on MARS level

- At this phase the request becomes **visible** on the [MARS activity](#) (with status active or queued in MARS)
- Note that a request can be at status **active** on the level of API but **queued** on MARS.
- We always try to **tune** the Web-API and MARS systems in order to improve the performance but in most cases it is not so straightforward.
- On the level of **MARS server** there are several MARS QoS rules (**ie limitations, restrictions and priorities**) like the ones below.
  - *On marsod-core, the total number of Requests from ECMWF Web API with 1 tape mount [source] is limited to 6*
  - *On marsst-core, the total number of Requests from ECMWF Web API with 1 tape mount [source] is limited to 12*
  - *To avoid long queuing times it is strongly recommended to improve your [Retrieval efficiency](#)*

### 4. File transfer procedure

- The produced file is then transferred by the Web-API client from ECMWF proxy to the user's local disk.
- Once the file has been transferred successfully the file will be removed from our disks and it won't be any more available.

#### If transfer procedure fails



1. The file transfer procedure may fail due to several reasons (eg network connectivity problems between our end and the users) see more here: [Web-API Troubleshooting#asserttotal==sizeAssertionError](#)
2. If transfer procedure fails difficult to trace back what has happened ⚠️
3. The API client checks the volume of the transferred file and if the size is not the expected one the transfer procedure will start again. The MAX Nr of retries is 10 |

## Organise the output in separate files



Some applications require the input files separated each on them containing one single parameter or step. This is not the most efficient way to retrieve it from MARS but the solution is very easy.

For example, imagine that you want to retrieve 3 parameters for the same date 20181001,run 00 and type=pl but you need to have them in 3 separate files. The steps would be:

1. Retrieve the data following the [Retrieval efficiency](#) recommendations, in this case the 3 parameters together:

```
retrieve,  
class=od,  
date=2018-10-01,  
expver=1,  
levelist=250/500/850/900/1000,  
levtype=pl,  
param=129.128/130.128/157.128,  
step=0/3/6,  
stream=oper,  
grid=1/1,  
time=00:00:00,  
type=fc,  
target="pl_20181001_00_all.grib"
```

2. Install the [ecCodes](#) tools and use [grib\\_copy](#) to split the file using "multitarget" name (note that the quotes (") in the output name are required):

```
grib_copy pl_20181001_00_all.grib "pl_20181001_00_[shortName].grib"
```

This will create 3 different files, one per parameter:

```
pl_20181001_00_r.grib  
pl_20181001_00_t.grib  
pl_20181001_00_z.grib
```

## MARS interpolation

[MARS interpolation with MIR](#)

## Problems



For any kind of problems please visit our [Web-API Troubleshooting](#) pages