

Using MAST APIs

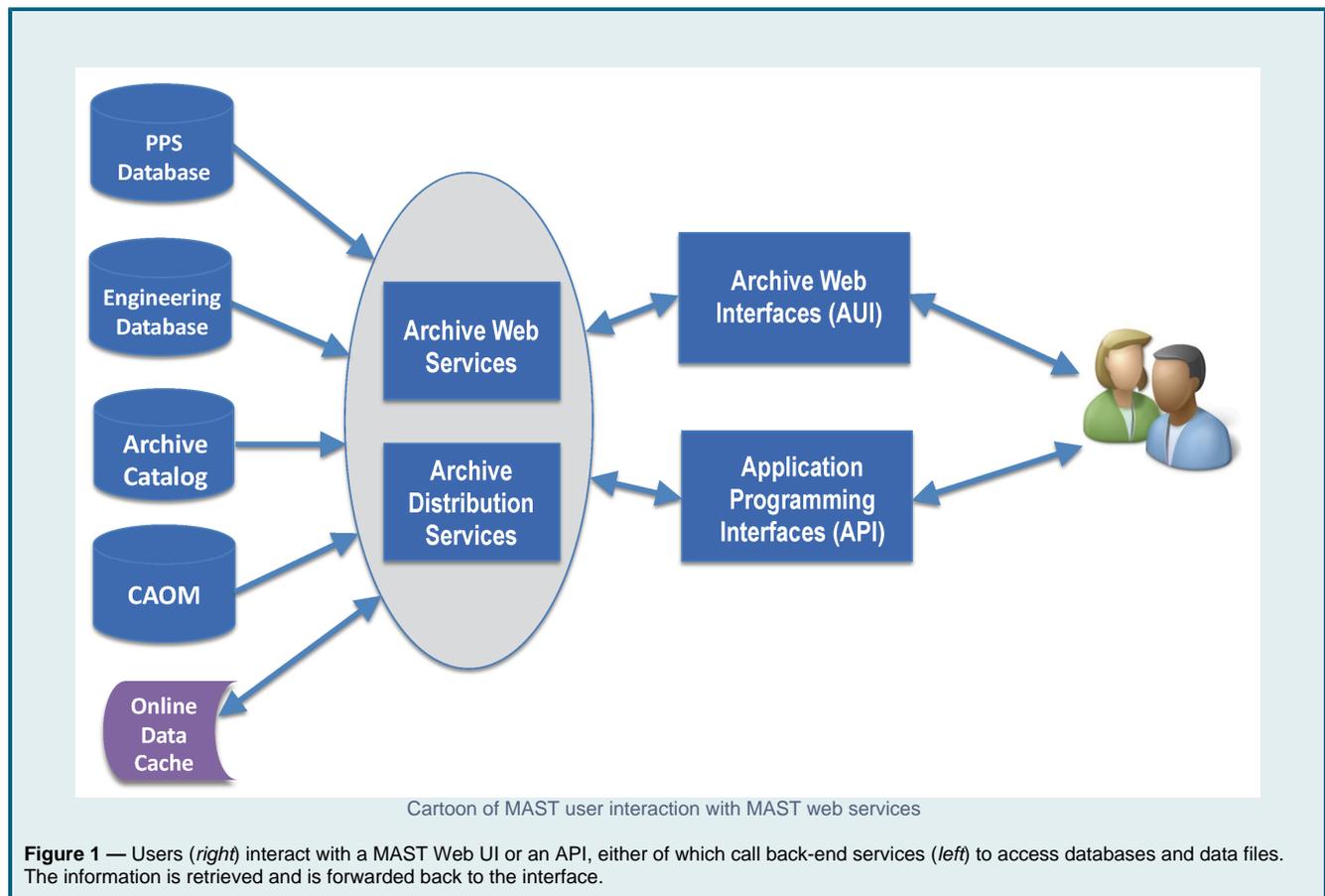
MAST application programming interfaces (APIs) provide a direct and efficient means for experienced users to search for and retrieve data products. The APIs that are most relevant for JWST are illustrated via a collection of tutorials which you can customize to your own scientific needs.

On this page...

- [MAST Web Services](#)
 - [Auth.MAST](#)
- [API Tutorials](#)
 - [Astroquery Search and Retrieval](#)
 - [Curl Download](#)
 - [MAST API](#)
 - [Duplication Check](#)
 - [JWST Engineering Data Retrieval](#)
- [For Further Reading...](#)

MAST Web Services

Much of the functionality of MAST is provided through a variety of [web services](#). The MAST Portal itself is a Web application that calls [MAST web services](#) to provide data search, selection, and retrieval functionality. Most of these same services are available to users via multiple MAST application programming interfaces (APIs). These APIs offer a very efficient means to create custom, scripted access to JWST data products. The figure below is a simplified view of the interaction between users and MAST back-end services.



Auth.MAST

Most data in MAST can be retrieved anonymously. But retrieving protected data with an API requires both **authentication** and **authorization**. Authorization is granted by MAST to program Principal Investigators, and may be granted by a PI to their collaborators. Authentication in the Portal is achieved by logging in with your MyST credentials; authentication for scripted retrieval is achieved with a [MAST API token](#). These tokens, which are long alpha-numeric strings, may be provided to a script:

- with a command-line argument
- by a user responding to a prompt
- by storing the token string in the shell environment variable `$MAST_API_TOKEN`
 - Users of bash can do this *in the shell they use to access an API* with the following command (or, optionally, store in their `.bashrc` file):

```
export MAST_API_TOKEN=<token string>
```

Generate the token with the [tokens](#) page.



Token Expiration Policy

For security reasons MAST tokens expire after 10 days of inactivity or 60 days after creation, whichever comes first. If you have a script which used to work but now fails with an authorization error, check to see if your token is still active. It is easy to generate a new MAST token.

API Tutorials

The following subsections provide a variety of examples of how to use the MAST APIs with Python or shell scripts. For a more extensive discussion of MAST APIs, with many more worked examples, see [MAST Web Services](#). Some of the tutorials below take the form of Jupyter Notebooks, which reside in the [MAST Notebook repository](#).

Astroquery Search and Retrieval

The Python package **astroquery** provides a general means of creating custom scripts to access astronomical data repositories; [astroquery.mast](#) provides specific functionality for the MAST archive, which is analogous to Portal searches.



Large JWST Queries

The most highly calibrated and combined JWST observations may contain a very large number (thousands) of data products. These observations may cause timeouts or other performance problems for the Portal application. The tutorials mentioned here provide a robust method for dealing with large numbers of observations and data products.

- [API Advanced Search Tutorial](#)
- [API Large Download Notebook](#)
- [API Search by Instrument Keywords Notebook](#)

Curl Download

One option for retrieving data via the Portal [Download Basket](#) is to retrieve a "Curl" script. This is in reality a **bash** script which includes **cURL** commands to actually retrieve the data. See a description of an [example download script](#) to learn how it works; from this you can create your own download script.



You must know the URI(s) for the data file(s) you wish to download. These may be obtained with a different API, e.g., with an astroquery search.

MAST API

The MAST web services may be called directly, through a [mashup](#) interface. This is a collection of interfaces that allows more functionality and flexibility for tailoring request parameters. It does, however, require significant knowledge of the supported web services, the formatting of service requests and response results, and careful programming to wrangle the arcane syntax of web service calls. See:

- [MAST API documentation](#)

Duplication Check

It is important for investigators who are preparing JWST observing proposals to know in advance if their intended targets might duplicate existing or planned JWST observations. This Jupyter notebook will help:

- [JWST Duplication Check Notebook](#)



While helpful, JWST planned observations in MAST do not provide sufficient information to determine if a genuine duplication exists. For instance, parallel observations are omitted, and the prime target orientation may not be known until the visits are scheduled. The results in the notebook will include a link to the program APT file, which should be consulted to evaluate any potential duplication identified in MAST.

JWST Engineering Data Retrieval

The JWST Engineering Database contents are described in the chapter on [Engineering Data](#). The EDB Retrieval tutorial shows how to retrieve multiple mnemonics over independent time ranges using Python. There is also an example Python script which you may customize for your own use. Both of these methods require a valid [MAST.auth token](#).

- [Engineering Data Jupyter Notebook](#)
- [Calibrated Engineering Data access via edb_script.py](#)

For Further Reading...

- [MAST Web Services](#)
- [MAST Authorization](#)
- [Portal Guide](#) chapter on [MAST web services](#)
- [CAOM Field Descriptions](#)