How to access Radiant MLHub data

The Radiant MLHub API gives access to open Earth observation (EO) training data for machine learning. You can learn more about the repository at the Radiant MLHub site and the organization behind it at the Radiant Earth Foundation site.

This is an introductory guide to accessing data. Full documentation for the API is available at docs.mlhub.earth. A Jupyter Notebook with sample codes to use the API is also available on Radiant Earth's GitHub.

You will begin by setting up your authorization, viewing the list of available training data collections and datasets, and retrieving the *items* (the pieces of data contained within them) from those collections. Basic familiarity with Python will be required to access and use the data.

Radiant MLHub uses STAC standard for cataloging training datasets. Each item in our collection is explained in json format compliant with STAC label extension definition.

Setting up and viewing Collections

Access to the Radiant MLHub API requires an access token. To get your access token, go to dashboard.mlhub.earth. If you have not used Radiant MLHub before, you will need to sign up and create a new account. Otherwise, sign in. Under **Usage**, you'll see your access token, which you will need. *Do not share* your access token with others: your usage may be limited and sharing your access token is a security risk.

To see what training data is available, you will need to access the collections available through the API. A collection represents the top-most data level. Typically, this means the data comes from the same source for the same geography. It might include different years or sub-geographies.

To find data with specific parameters, see the API documentation. Use the collections endpoint to retrieve a list of all collections available and their associated ids.

Access items

Once you have found the collection that you want to access, you can call the item from the API.

You can then limit what data you get from the item using the optional parameters:

- **Limit** limits how many items will be returned, with a minimum of 1 and a maximum of 10000.
- **Bounding box** limits the returned items to a specific geographic area.
- **Date time** limits the returned items to those that fall within a specific time-frame.

See the get features API documentation for more information.