

How To:

Using API to retrieve data from Macrostrat

Macrostrat is a database containing geospatial information for geological records (especially in lithology and stratigraphy) of the upper crust in North America and several other sites beneath the ocean. It has an interactive web platform to view those geospatial and geological data, which are also accessible for free. Macrostrat is maintained by Dr. Shanan Peters and <https://macrostrat.org> is the URL link to its main page.

To retrieve or download data from Macrostrat, you need to use its API. API is short for application programming interface. It is a group of backstage tools that help people use different devices, databases, programs and applications all together in an easier way. In this case, the API for Macrostrat serves as the nexus to connect the Macrostrat database to other software or programming languages that you use, including but not limited to: Python, R, JavaScript, MATLAB, Mapbox, ArcGIS, GeoDa and QGIS. Once you have the querying API ready, it will take your orders to the data archiver in the Macrostrat system and then return with the dataset that you just requested.

Macrostrat is an active project that may update its database aperiodically. Using an API to extract data directly from Macrostrat can insure that you have the most recent data in your analysis. Also, the API is convenient and user-friendly. Its functions operate behind the scene and all you need to do is to have the correct routes to get access to the data you want.

In order to use API for data retrieving in the Macrostrat, you need to get the access to the API Root of this platform. Open your web browser and type in <https://macrostrat.org/api> into the address bar and hit return. Now you will reach the API Root of Macrostrat. You can see the following page:

```

{
  "success": {
    "v": 2,
    "description": "This is the root of the Macrostrat API",
    "changelog": "/changes",
    "license": "CC-BY 4.0. More info at /meta",
    "routes": {
      "/columns": "Search and summarize columns based on unit properties or geographic location",
      "/sections": "Summarize units by gap-bound packages",
      "/units": "all Macrostrat units matching search criteria",
      "/fossils": "Paleobiology Database (http://paleobiodb.org) collections matched to Macrostrat units",
      "/stats": "statistics about the Macrostrat database",
      "/paleogeography": "Returns paleogeography geometry from \"Wright et al. (2013) Towards community-driven paleogeographic reconstructions: integrating open-access paleogeographic and paleobiology data with plate tectonics. Biogeosciences 10:1529-1541\" If you use this service and provide attribution, you should cite the associated paper via this service.",
      "/geologic_units/gmna": "Geologic map units. Continental-scale North American map data (gmna) adapted from the 2005 Geologic Map of North America (http://ngmdb.usgs.gov/gmna/)",
      "/geologic_units/gmus": "Geologic map units. State-level (gmus) data adapted from http://mrdata.usgs.gov/geology/state/.",
      "/geologic_units/map": "Geologic map units from various data sources",
      "/geologic_units/map/points": "Query point features from geologic maps",
      "/measurements": "measurements",
      "/defs": "Routes giving access to standard fields and dictionaries used in Macrostrat",
      "/carto": "Routes for creating geologic maps",
      "/grids": "Returns grids with the specified parameters",
      "/mobile": "Simplified data delivery, ideal for mobile applications"
    }
  }
}

```

Here, you can find information regarding different routes that you can use in searching for your data. Descriptions of each route are also attached. In the next step, you are going to use one of these routes to retrieve data.

If you want to retrieve data from the paleontology database, for example, you need to add “/fossils” to your route. In this case, you need to go to the address bar of your browser and type “/fossils” at the end of your existing route of <https://macrostrat.org/api> so that you will get a new route of <https://macrostrat.org/api/fossils>.

Once you reach there, you will see the following page:

```

{
  "success": {
    "v": 2,
    "license": "CC-BY 4.0",
    "description": "Paleobiology Database (http://paleobiodb.org) collections matched to Macrostrat units",
    "options": {
      "parameters": {
        "interval_name": "string, chronostratigraphic time interval name",
        "age": "numerical age in millions of years before present",
        "age_top": "numerical age (Ma) - must be used with age_bottom and be less than age_bottom",
        "age_bottom": "numerical age (Ma) - must be used with age_top and be greater than age_top",
        "lith_id": "integer, ID of a lithology from /defs/lithologies",
        "lith": "string, specific lithology name (e.g., shale, sandstone)",
        "lith_type": "string, groups of lithologies (e.g., carbonate, siliciclastic)",
        "lith_class": "string, general lithologies (sedimentary, igneous, metamorphic)",
        "environ_id": "integer, specific environment ID from /defs/environments",
        "environ": "string, specific environment",
        "environ_type": "string, groups of environments",
        "environ_class": "string, general environments",
        "econ_id": "integer, ID of an economic attribute from /defs/econs",
        "econ": "string, name of an economic attribute",
        "econ_type": "string, name of an economic attribute type",
        "econ_class": "string, name of an economic attribute class",
        "unit_id": "One or more comma-separated valid unit IDs",
        "col_id": "One or more comma-separated valid column IDs",
        "project_id": "One or more comma-separated valid project IDs",
        "strat_name_id": "One or more comma-separated valid strat_name_ids from /defs/strat_names",
        "strat_name_concept_id": "One or more stratigraphic name concepts from /defs/strat_name_concepts",
        "format": "Desired output format"
      },
      "output_formats": [
        "json",
        "csv",
        "geojson",
        "geojson_bare",
        "topojson",
        "topojson_bare"
      ],
      "examples": [
        "/api/fossils?interval_name=Permian",
        "/api/fossils?age=271",
        "/api/fossils?age_top=200&age_bottom=250",
        "/api/fossils?col_id=446"
      ],
      "fields": {
        "cltn_id": "integer, Paleobiology Database collection ID",
        "cltn_name": "text, Paleobiology Database collection name",
        "pbdb_occs": "integer, count of PBDB occurrences in units in column",
        "genus_no": "integer, corresponds to the genus_no in Paleobiology Database",
        "unit_id": "integer, unique identifier for unit",
        "col_id": "integer, unique identifier for column",
        "t_age": "number, continuous time age model estimated for truncation, in Myr before present",
        "b_age": "number, continuous time age model estimated for initiation, in Myr before present"
      }
    }
  }
}

```

At this point, you will find the “parameters” which indicates that, under the route you have chosen, what parameters are available for you to query the data. In this example, you have chosen the route of the paleontology database, so you can use parameters listed under “parameters” to query the database. For each parameter, a short description containing general information of this parameter including the type of input in the query is attached. It will tell you how to query each parameter. For instance, the parameter “interval_time” can be used to query the database based on the “chronostratigraphic time interval name”, and the input for this query should be a string.

There are several other things that you can find out. The “output_format” gives you options on the format of the data that you will get. The “examples” shows how you

can extract data by adding parameters to your route. The “fields” tells you what columns are included in the output dataset.

After checking out the “examples”, you may have a sense on how exactly you should do the data query by adding things to the route. The first step is to add a question mark (“?”) to your route which indicates the beginning of your query string.

Then, add the first parameter that you want to use in your data query right after the question mark. Following the parameter should be an equal sign connecting to your query inputs. The inputs that you type here should follow the instruction in the attached sentence to each parameter. For example, if you want to pull out data of the paleobiology collections from Jurassic period, the address bar of your browser should have https://macrostrat.org/api/fossils?interval_name=Jurassic in it. In this way, you can preview the data returned from your query.

If you have multiple parameters to use during your query, you may add “&” symbols between segments of the string for different parameters. For example, if you want to pull out data of the paleobiology collections from Jurassic period and with the lithology of shale, you may type the route https://macrostrat.org/api/fossils?interval_name=Jurassic&lith=shale into your address bar to preview the returned data.

Once you have previewed your data selection in the API Root, you will need to tell the system what format of data that you want to have. You will complete your querying route by adding “&format=” as well as your choice of format to the end. For example, if you want to have your Jurassic and shale paleobiology data in the csv format, the route https://macrostrat.org/api/fossils?interval_name=Jurassic&lith=shale&format=csv is what you need. If you put this querying route into the address bar of your browser and hit return, you will download a csv file containing your returning dataset. Alternatively, you can use the route to import the dataset directly to your programming environment or other software that you are working on.

Say that you are coding in R and you need to use the above-mentioned dataset, what you can do is to use the `read.csv` function to import the dataset directly. The command in R will be:

```
read.csv("https://macrostrat.org/api/fossils?interval_name=Jurassic&lith=shale&format=csv")
```

and by doing so, you will have returned data imported to your R environment directly.

Same steps can be applied to other routes including the “/units” which can return all available units matching your querying strings. Under this route, there is an additional option called “response_types” which will let you choose between key information and full information of inquired units to be included in the returned dataset. One example of this is that

https://macrostrat.org/api/units?strat_name=Waldron_shale&response=short will

bring you key information about the Waldron Shale while

https://macrostrat.org/api/units?strat_name=Waldron_shale&response=long will give

you additional information regarding this shale formation, including more detailed lithology, depositional environment, names of the geochronological intervals that contain its top and bottom boundaries, hex codes for its recommended coloring, and economic uses, etc.