DOL API User Guide

Last updated: 08/05/2024

Table of Contents

Overview

This guide offers an overview of the Department of Labor's Open Data Application Programming Interface (DOL API), along with guidance and instructions to assist you in accessing our metadata and data.

What is the DOL API?

The DOL API is a web service that provides on-demand access to machine readable metadata and data.

Using our API offers many benefits, including:

- **Customizable response**. You can request subsets of data you need instead of the entire dataset.
- **Automation.** You can reuse your customized data requests, which allows for more productive workflows.
- **Live and updated data**. You can be confident that every API request you make will always return the latest data.
- **Programming-language agnostic.** You can connect to the API through your preferred statistical software programs including Python, R, and SAS.

Getting Started

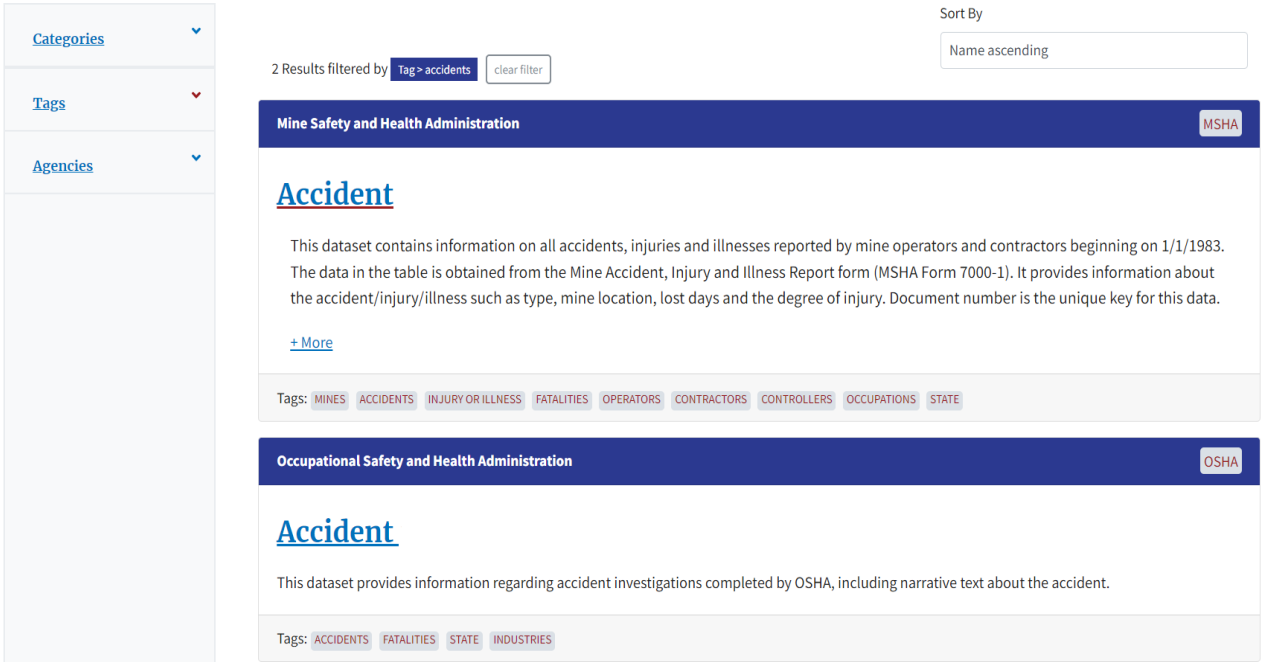
To use our APIs, you need to register for an account at https://dataportal.dol.gov/registration and complete our questionnaire to get an API key. Once you register, there may be a slight delay while the system incorporates your credentials, and you may need to wait for a few minutes and/or refresh your screen before an API key is generated in your account. Your API key will look like a mixture of numbers, letters, and special characters. You can create additional keys (up to 5) if you need to use them for different purposes, client applications, or automation.

Your API key is what grants you access to the DOL API and is a required parameter for all metadata and data requests. You can always find, manage, and create new keys by signing into the open data portal and going to https://dataportal.dol.gov/api-keys. The sharing of your API key(s) with others, including on public code repositories or other public forums is strictly prohibited and may result in the termination of your API service.

Finding Datasets

The U.S. Department of Labor publishes a large amount of data, including statistical, enforcement, legal, policy, administrative, and financial data. The Data Catalog is an inventory of datasets that are currently available for query through the API. You can view the full list of available datasets in the Data Catalog at https://dataportal.dol.gov/datasets. The catalog can be sorted by dataset name, or filtered by category, tag, or agency using the left sidebar.

Example: Filter on accidents tag



To programmatically browse our Data Catalog in JSON format, use our Datasets API: https://apiprod.dol.gov/v4/datasets. No api key is required to access this endpoint. Code examples for accessing this endpoint and using the resulting data are also available on our API Examples page.

Discovering Metadata

Every dataset in the API is comprehensively described by our metadata. Metadata can help you better understand the properties and characteristics of the variables within a dataset, inform whether the data will be potentially useful for specific purposes prior to access, and provide considerable benefit in automating the processing of the resulting data.

The Metadata API can be accessed using this template:

JSON Format: https://apiprod.dol.gov/v4/get/**<agency>**/**<endpoint>**/json/metadata?X-API-KEY=**<api_key>**
CSV Format: https://apiprod.dol.gov/v4/get/**<agency>**/**<endpoint>**/csv/metadata?X-API-KEY=**<api_key>**

To retrieve metadata, paste a link into a web browser (or use your preferred software) and replace <agency> with an *agency name abbreviation* and <endpoint> with an *api_url* from the Datasets API. Also, substitute <api_key> with your unique API key. You will need to register or sign-in to retrieve your API key.

Common use cases for accessing metadata and code examples can be found on the API Examples page.

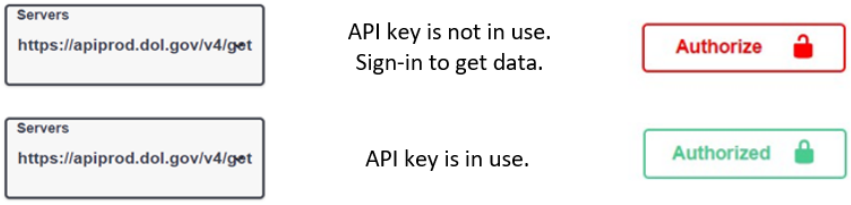
Building an API query

To start building API queries, we recommend using our API Query Builder Tool which is located towards the bottom of every dataset page you navigate to from the Data Catalog. The API Query Builder will provide you with basic templates for making API calls for each endpoint and allows you to interactively select which fields and records you want, choose how records are formatted and sorted, and generate a proper request URL and curl command that you can use to get your desired data.

Parameters

Parameters are options you can add to an API call that allow you to customize your data request: what subsets of records are returned; how many records are returned, which variables are included; and how the data are formatted and sorted. Listed below are the parameters that can be accepted in a request.

- **X-API-KEY** (string, required): User's API key for authentication.
  Your key is automatically populated by the API Query Builder if you are signed into your open data portal account. The color of the "Authorize" button indicates whether your API key is in use. Red indicates that you are not signed into your account and will not be able to request data. Green indicates that you are signed into your account and will be able to request data.



- **format** (string, required): The data format returned by the API.
  - Options: "json", "xml", "csv"
  - Default: json

- **limit** (integer): The maximum number of records to be returned.
  - Options: You can receive up to 5 MB of data or 10,000 records per request, whichever limit you hit first.
  - Default: 10

- **offset** (integer): The number of records to skip from the top of a dataset. This parameter can be used in conjunction with the **limit** parameter when you are requesting more than the maximum record limit for a dataset.

For example, if you need 20,000 records you would get the first chunk of 10,000 records by setting the limit to 10000 and offset to 0, and the second chunk of 10,000 records by setting the limit to 10000 and offset to 10000.
    - Options: Positive integer
    - Default: 0

- **fields** (string): The specific fields you want to select from a dataset. For example, you would use "case_identifier,naics_code,back_wages" to only return these 3 fields from a dataset.
    - Options: Comma separated string of dataset field names.
    - Default: All variables are included in the output.

- **sort** (string): The sort direction that you want applied to the records of a dataset.
    - Options: "asc" for ascending or "desc" for descending.

- **sort_by** (string): The field name you want to sort by.
    - Options: Field name

- **filter_object** (string): The conditional column filters to apply to a dataset.
    - Options: A JSON formatted string that specifies the values for *field*, *operator*, and *value*. The operators we support are eq (equals), neq (not equals), gt (greater than), lt (less than), in, not_in, and like.

Additional information on filtering datasets can be found in the Conditional filtering section.

Executing and Debugging Queries

Once parameters have been populated into the API Query Builder, you can proceed with clicking the "Execute" button to receive your response. The API Query Builder will display the request URL as well as the server response.

(a) Request URL

The request URL is what you will need to use external software and applications to pull data in from the API. You can also execute the URL within a browser to verify the call or view the results.

(b) Server response

If everything is successful, you will see data in the response body as well as the code “200” in the server response. You will also have the option to download the response through a blue button in the lower right of the Responses box.

If your API query fails and you get a server response code other than “200”, verify that your api key is in use by checking for a green “Authorize button” and check your output and parameters for any issues.

Conditional Filtering

Next, we are going to cover dataset filtering in more detail.

The **filter_object** parameter allows you to subset a dataset based on conditions for one or more fields.
To execute this in a query, it requires a JSON formatted string that has three key-value pairs.

- *field* – The field name the filter will be applied on.
- *operator* – The type of equality operation to perform.
- *value* – The conditional value to apply to the field.

Examples

For the following filtering examples, consider the dataset below:

Industry	Year	Value
A	2021	301
A	2022	456
B	2019	983
B	2020	475
B	2021	159
C	2021	979
C	2022	459
AB	2021	1200
AB	2022	1489
AC	2020	854
AC	2021	533
AC	2022	89

Single condition filter

Example (equals operator): Filter records where industry is equal to “B”.

```
{
   "field":"industry",
   "operator":"eq",
   "value":"B"
}
```

Example (not equals operator): Filter records where industry is not equal to “B”.

```
{
   "field":"industry",
   "operator":"neq",
   "value":"B"
}
```

Example (in operator): Filter records where industry is in “A”, “B”, or “C”.

```
{
   "field":"industry",
   "operator":"in",
   "value":[“A”,”B",”C”]
}
```

Example (not_in operator): Filter records where industry is not in “A”, “B”, or “C”.

```
{
   "field":"industry",
   "operator":"not_in",
   "value":[“A”,”B",”C”]
}
```

Example (like operator): Filter records where industry is like “A”.

```
{
   "field":"industry",
   "operator":"like",
   "value":"%A%"
}
```

Multiple condition filter

Example (and operator): Filter records where industry is equal to “B” and year is equal to “2020”.

```
{
   "and":[
      {
         "field":"industry",
         "operator":"eq",
         "value":"B"
      },
      {
         "field":"year",
         "operator":"eq",
         "value":"2020"
      }
   ]
}
```

Single variable, multiple condition filter

Example (or operator): Filter records where value is less than 500 or greater than 999.

```
{
   "or":[
      {
         "field":"value",
         "operator":"lt",
         "value":500
      },
      {
         "field":"value",
         "operator":"gt",
         "value":999
      }
   ]
}
```

Multiple variable, multiple condition filter

Example (and/or operators): Filter records where year is equal to “2021” and where industry is equal to either “A” or “C”.

```
{
   "and":[
      {
         "field":"year",
         "operator":"eq",
         "value":"2021"
      },
      {
         "or":[
            {
               "field":"industry",
               "operator":"eq",
               "value":"A"
            },
            {
               "field":"industry",
               "operator":"eq",
               "value":"C"
            }
         ]
      }
   ]
}
```

Final Query

The final API query, which is using filters to produce a specific subset of records from an API data source will look something like this:

https://apiprod.dol.gov/v4/get/agency/dataset/json?**limit**=10&**offset**=0&**sort**=asc&**sort_by**=industry &**filter_object**={"and":[{"or":[{"field":"industry","operator":"eq","value":"A"}, {"field":"industry","operator": "eq","value":"C"}]}, {"field":"year","operator":"eq","value":"2021"}]}&**X-API-KEY**= ijEJ5wN8dyLCmTgXMDeqmocprI-gcCy4PMQz0a7cFYo

All API requests will follow a similar structure:

```
<API server address>/get/<agency name>/<api endpoint url
>/<format>?<parameter>=<value>&<parameter>=<value>&...
```

Help and Contact Us

If you need help using the API or have additional questions, try out the resources that we make available:

- Frequently Asked Questions: https://dataportal.dol.gov/faq
- API Examples: https://dataportal.dol.gov/api-examples
- Videos Tutorials: https://dataportal.dol.gov/video-tutorials

For further assistance or to provide feedback on our Open Data Portal and API, Contact Us.