..

Working with Data APIs

Date published: 2020-10-30
Date modified: 2022-09-21

https://docs.cloudera.com/

Legal Notice

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

How to Enable Unauthenticated Access

Before invoking the CDP Data Visualization Data API, you must first enable it in the configuration file of the Web Server, and obtain an API key.

You might have to enable “Enable Unauthenticated Access” for the application if you are attempting to access the API outside of CML or CDSW jobs or sessions.

See the “Public Applications” section of https://docs.cloudera.com/machine-learning/1.4.1/applications/topics/ml-securing-applications.htmlSecuring Applications

Enabling Data APIs

To enable CDP Data Visualization Data API, update the configuration settings, and restart the CDP Data Visualization instance.

About this task

Note: This is only possible for CDP Data Visualization instance running in CML and CDSW

Procedure

1. Navigate to the Site Settings from the navigation menu and Advanced Site Settings section at the bottom of the page.
2. Add the ENABLE_API_KEYS and DATA_API_ENABLED settings in the configuration file of the Arcadia Web Server:

```
ENABLE_API_KEYS = True  # If you plan to access via API keys
              DATA_API_ENABLED = True
```

3. Restart the CDP Data Visualization service.

Using API keys

When you do not login directly into the visual layer of CDP Data Visualization, you must obtain an API key and implement it when using Data APIs.

Before you begin

Follow the instructions shown in Creating new API keys.

Procedure

1. After obtaining an API Key, you can use it to authenticate the user to access the dataset, and invoke the data layer of CDP Data Visualization at the following endpoint:

```
http(s)://server:port/arc/api/data
```

2. Proceed with using the Data API interface.

   For more information, see Example of Data API usage.

Related Information

Creating new API keys

Example of Data API usage

Accessing Data API request payload

To avoid creating Data API requests from scratch, CDP Data Visualization provides access to details of the data requests.

About this task

This task is optional.

Procedure

1. Add the following settings in Advanced Site Settings section of the Site Settings page:

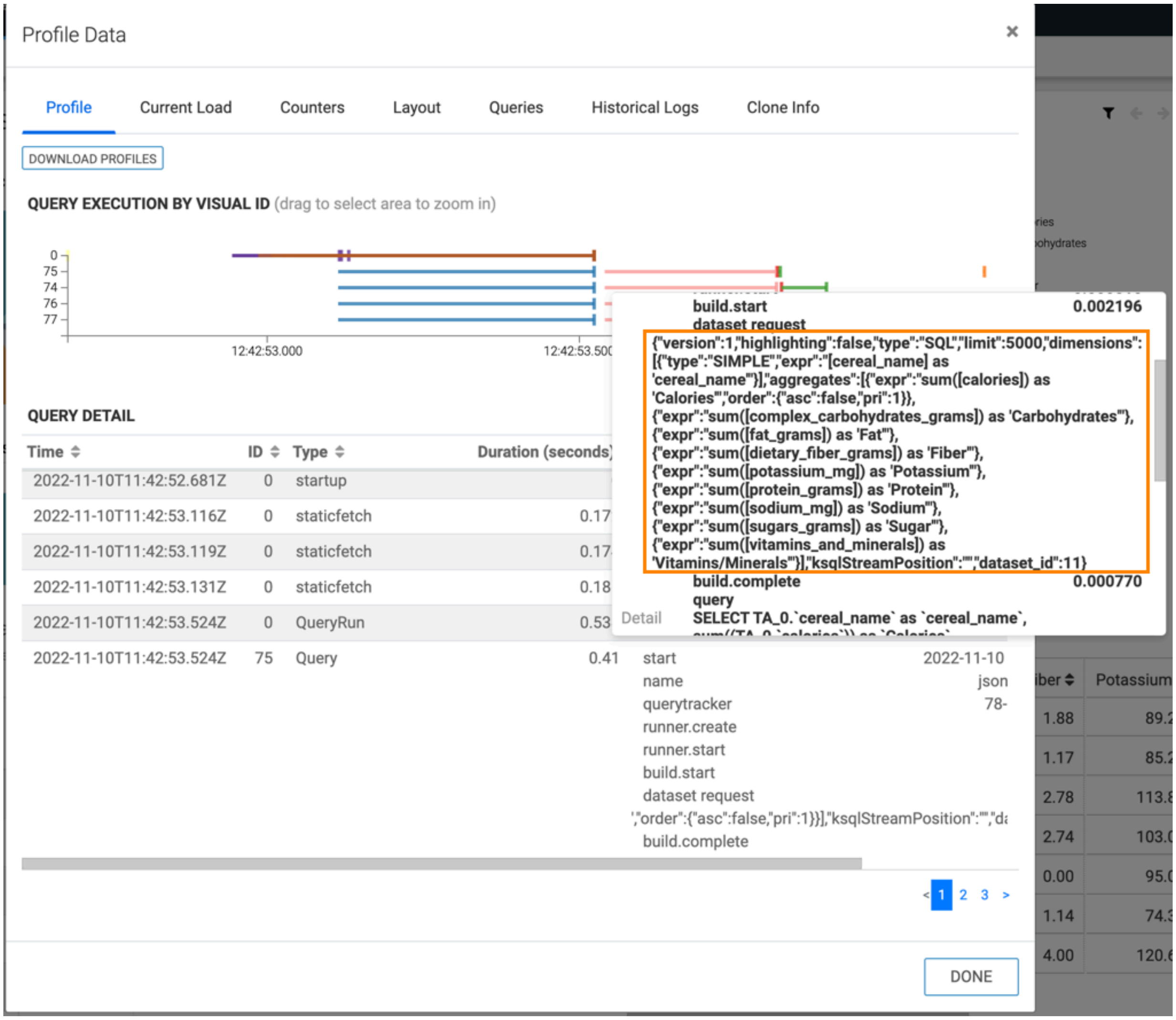
   ```
   ENABLE_DSREQ_PERF_DISPLAY = True
   COMPRESS_DSREQ = True
   ```

2. Restart the CDP Data Visualization service.
3. Open an existing dashboard.

   In this example, we selected Cereals.
4. Use the keyboard shortcut Shift + Control/Cmd + G to open the Performance Profile interface.

5. Hover over a line that represents a visual ID, we selected 75, and view the duration, type, and details of the query that runs against that visual.

   The query uses the same API and includes the details of the data requests. You can use this dataset request as the payload for your Data API call.

   QUERY DETAIL

Time	ID	Type	Duration (seconds)
2022-11-10T11:42:52.681Z	0	startup	
2022-11-10T11:42:53.116Z	0	staticfetch	0.17
2022-11-10T11:42:53.119Z	0	staticfetch	0.17
2022-11-10T11:42:53.131Z	0	staticfetch	0.18
2022-11-10T11:42:53.524Z	0	QueryRun	0.53
2022-11-10T11:42:53.524Z	75	Query	0.41

   ```
   start                     2022-11-10
   name                            json
   querytracker                     78-
   runner.create
   runner.start
   build.start
   dataset request
   ',"order":{"asc":false,"pri":1}}],"ksqlStreamPosition":"","da
   build.complete
   ```

   ```
   build.start                                                         0.002196
   dataset request
   {"version":1,"highlighting":false,"type":"SQL","limit":5000,"dimensions":
   [{"type":"SIMPLE","expr":"[cereal_name] as
   'cereal_name'"}],"aggregates":[{"expr":"sum([calories]) as
   'Calories'","order":{"asc":false,"pri":1}},
   {"expr":"sum([complex_carbohydrates_grams]) as 'Carbohydrates'"},
   {"expr":"sum([fat_grams]) as 'Fat'"},
   {"expr":"sum([dietary_fiber_grams]) as 'Fiber'"},
   {"expr":"sum([potassium_mg]) as 'Potassium'"},
   {"expr":"sum([protein_grams]) as 'Protein'"},
   {"expr":"sum([sodium_mg]) as 'Sodium'"},
   {"expr":"sum([sugars_grams]) as 'Sugar'"},
   {"expr":"sum([vitamins_and_minerals]) as
   'Vitamins/Minerals'"}],"ksqlStreamPosition":"","dataset_id":11}
   build.complete                                                      0.000770
   query
   Detail  SELECT TA_0.`cereal_name` as `cereal_name`,
   ```

Migrating visual artifacts with REST APIs

In addition to UI-enabled export and import, CDP Data Visualization provides REST APIs that give more flexibility in choosing what objects to migrate. While migrating artifacts using the UI allows you to directly specify export of visuals, dashboards, and apps, REST APIs enable you to specify a wider range of migratable objects.

About this task

Note: You do not migrate events and segments separately, but as part of dataset migration. Similarly, you do not migrate thumbnails along with dashboards or visuals that they represent.

Before you begin

Enable and create the relevant data API keys, both for the source and destination machines. For instructions, see *Enabling API keys* and *Creating new API keys*. The examples used in these instructions use the secret API key that appears when you generate the keys for the first time, as in *Creating new API keys*.

Procedure

1. Export the objects by submitting a GET request on the source machine. For instructions, see *Exporting visual artifacts with REST APIs*.
2. Import the objects by submitting a POST request on the target machine. For instructions, see *Importing visual artifacts with REST APIs*.

Related Information

Enabling API keys

Creating new API keys

Exporting visual artifacts with REST APIs

Procedure

1. Authenticate the user using the data API.
2. After authenticating, submit a GET request to the following address for a source machine:source_arcviz_ip/arc/migration/api/export/

   For instructions on import, see *Importing visual artifacts with REST APIs*.

Example

This is a sample Python code snippet for authenticating and submitting the GET request that exports dashboards with IDs 2115 and 2110, and then writing the response to the my_app.json file.

```
export requests
headers = {'AUTHORIZATION':'apikey secret_api_key'}
payload = {'dashboards': '[2115, 2110]', 'filename': 'apitestmigration',
'dry_run':'False'}
r = requests.get('source_arcviz_ip/arc/migration/api/export/', headers=hea
ders, params=payload)
with open('my_app.json', 'w') as f:
        f.write(r.text)
```

Related Information

Importing visual artifacts with REST APIs

Importing visual artifacts with REST APIs

Before you begin

Make sure that the *.json file with information about exported visual artifacts is on the destination machine.

Procedure

1. Authenticate the user using the data API.

2. After authenticating, submit a POST request to the following address for a destination machine: destination_arcviz_ip/arc/migration/api/import/

   For instructions on export, see *Exporting visual artifacts with REST APIs*.

Example

This is a sample Python code snippet for authenticating and submitting the POST request that imports dashboards with IDs 2115 and 2110, which were earlier saved (from a metadata perspective) as my_app.json. This file is uploaded to the Data Visualization server during the import.

```
import requests
headers = {'AUTHORIZATION':'apikey secret_api_key'}
payload = {'dry_run': False, "dataconnection_name":"data_connection"}
files = {'import_file': open('/Users/my_name/Downloads/my_app.json','r')}
r = requests.post('destination_arcviz_ip/arc/migration/api/import/',files=fi
les, data=payload, headers=headers)
print r.status_code # 200 is success
```

Related Information

Exporting visual artifacts with REST APIs