Quickstart: Using client libraries

cloud.google.com/bigquery/docs/quickstarts/quickstart-client-libraries

This page shows you how to get started with the BigQuery API in your favorite programming language.

Before you begin

1. In the Google Cloud Console, on the project selector page, select or create a Google Cloud project.

   Note: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.
   Go to project selector

2. Enable the BigQuery API.
   Enable the API

3. Create a service account:

   1. In the Cloud Console, go to the **Create service account** page.

      Go to Create service account
   2. Select a project.
   3. In the **Service account name** field, enter a name. The Cloud Console fills in the **Service account ID** field based on this name.

      In the **Service account description** field, enter a description. For example, `Service account for quickstart`.

   4. Click **Create and continue**.
   5. Click the **Select a role** field.

      Under **Quick access**, click **Basic**, then click **Owner**.

      Note: The **Role** field affects which resources your service account can access in your project. You can revoke these roles or grant additional roles later. In production environments, do not grant the Owner, Editor, or Viewer roles. For more information, see Granting, changing, and revoking access to resources.
   6. Click **Continue**.
   7. Click **Done** to finish creating the service account.

      Do not close your browser window. You will use it in the next step.

4. Create a service account key:
    1. In the Cloud Console, click the email address for the service account that you created.
    2. Click **Keys**.
    3. Click **Add key**, then click **Create new key**.
    4. Click **Create**. A JSON key file is downloaded to your computer.
    5. Click **Close**.
5. Set the environment variable `GOOGLE_APPLICATION_CREDENTIALS` to the path of the JSON file that contains your service account key. This variable only applies to your current shell session, so if you open a new session, set the variable again.

    Example: Linux or macOS

    Example: Windows

Install the client library

For more on setting up your Python development environment, refer to the Python Development Environment Setup Guide.

```
pip install --upgrade google-cloud-bigquery
```

Import the libraries

For more information, see the BigQuery Python API reference documentation.

View on GitHub Feedback

```
from google.cloud import bigquery
```

Initialize a BigQuery client

Initialize a client to authenticate and connect to the BigQuery API.

Instantiate the bigquery.Client class to create the BigQuery client.

View on GitHub Feedback

```
client = bigquery.Client()
```

Running queries

Query the Stack Overflow public dataset to find the most viewed questions tagged with `google-bigquery`.

```
SELECT
  CONCAT(
    'https://stackoverflow.com/questions/',
    CAST(id as STRING)) as url,
  view_count
FROM `bigquery-public-data.stackoverflow.posts_questions`
WHERE tags like '%google-bigquery%'
ORDER BY view_count DESC
LIMIT 10
```

This query uses standard SQL syntax, which is described in the query reference guide. The client libraries default to standard SQL syntax. See Switching SQL dialects to change SQL dialects.

Running the query

Query using the authenticated BigQuery client.

Use the Client.query() method to start the query.

View on GitHub Feedback

```
query_job = client.query(
    """
    SELECT
      CONCAT(
        'https://stackoverflow.com/questions/',
        CAST(id as STRING)) as url,
      view_count
    FROM `bigquery-public-data.stackoverflow.posts_questions`
    WHERE tags like '%google-bigquery%'
    ORDER BY view_count DESC
    LIMIT 10"""
)results = query_job.result()  # Waits for job to complete.
```

For more examples of running BigQuery queries, see:

- Querying data overview
- How to run interactive and batch queries
- How to write query results to a permanent table

Displaying the query result

Display the query results.

Iterate over the RowIterator to get all the rows in the results. The iterator automatically handles pagination. Each Row exposes the columns by numeric index, column name, or as Python attributes.

View on GitHub Feedback

```
for row in results:
    print("{} : {} views".format(row.url, row.view_count))
```

Learn more about working with data rows in BigQuery:

Complete source code

Here is the complete source code for the sample.

View on GitHub Feedback

```
from google.cloud import
 bigquery


def query_stackoverflow():
    client = bigquery.Client()
    query_job = client.query(
        """
        SELECT
          CONCAT(
            'https://stackoverflow.com/questions/',
            CAST(id as STRING)) as url,
          view_count
        FROM `bigquery-public-data.stackoverflow.posts_questions`
        WHERE tags like '%google-bigquery%'
        ORDER BY view_count DESC
        LIMIT 10"""
    )    results = query_job.result()  # Waits for job to complete.
    for row in results:
        print("{} : {} views".format(row.url, row.view_count))
if __name__ == "__main__":
    query_stackoverflow()
```

Congratulations! You've sent your first request to BigQuery.

How did it go?

What's next

Find out more about our BigQuery API Client Libraries.

Rate and review