Authenticate and connect to a data source with the Conversational Analytics API

This product or feature is subject to the "Pre-GA Offerings Terms" in the General Service Terms section of the Service Specific Terms (/terms/service-terms#1). Pre-GA products and features are available "as is" and might have limited support. For more information, see the launch stage descriptions (/products#product-launch-stages).

Developers can use the Conversational Analytics API (/gemini/docs/conversational-analytics-api/overview), accessed through **geminidataanalytics.googleapis.com**, to build an artificial intelligence (AI)-powered chat interface, or data agent, that answers questions about structured data in BigQuery, Looker, and Looker Studio using natural language.

This page describes how to authenticate to the Conversational Analytics API (#authenticate) and configure connections to your data in Looker (#connect-to-looker), BigQuery (#connect-to-bigquery), and Looker Studio (#connect-to-looker-studio) by using either direct HTTP requests or the SDK. The Conversational Analytics API uses standard Google Cloud authentication methods (/docs/authentication).

Before you begin

Before you can authenticate to the Conversational Analytics API and configure connections to your data, you must complete the prerequisites and enable the required APIs for your Google Cloud project, as described in Enable the Conversational Analytics API (/gemini/docs/conversational-analytics-api/enable-the-api).

Authenticate to the Conversational Analytics API

This section describes how to authenticate to the Conversational Analytics API (through **`geminidataanalytics.googleapis.com`**) by using HTTP and Python methods to obtain the necessary authorization tokens.

HTTP curl (#http-curl)HTTP using Python… Python SDK (#python-sdk)

The following sample Python code demonstrates how to authenticate your Google Account for access to the Conversational Analytics API within Colaboratory:

```
from google.colab import auth
auth.authenticate_user()
```

Connect to Looker with the Conversational Analytics API

To connect to Looker with the Conversational Analytics API, you must provide the following information:

- The URL of your Looker instance
- The specific LookML model (/looker/docs/lookml-terms-and-concepts#model) and Looker Explore (/looker/docs/lookml-terms-and-concepts#explore) that you want to use as a data source

Additionally, the authenticating user or service account must have the required Looker permissions (#required-looker-permissions).

You can then chose to authenticate using either Looker API keys (client ID and client secret) (#looker-api-keys) or an access token (#looker-access-token). Customers using the Looker (Google Cloud core) private IP (/looker/docs/looker-core-private-ip-overview) option must authenticate with an access token (#looker-access-token).

You can connect to only one Looker Explore at a time with the Conversational Analytics API.

Required Looker permissions

The user or service account whose credentials are used for authentication must be granted a Looker role that includes the following permissions for the models that you want to query:

- `access_data` (/looker/docs/admin-panel-users-roles#access_data)
- `gemini_in_looker` (/looker/docs/admin-panel-users-roles#gemini_in_looker)

You can configure these permissions in the **Admin > Roles** section of your Looker instance.

Authentication with Looker API keys

This section describes how to generate the API keys and configure the Conversational Analytics API to connect to Looker by using either direct HTTP requests or the SDK. Customers using the Looker (Google Cloud core) private IP (/looker/docs/looker-core-private-ip-overview) option cannot use this method and should authenticate with an access token (#looker-access-token).

To establish a connection with a Looker instance, you need valid Looker API keys, which are created by Looker and consist of a client ID and a client secret. Looker uses these keys to authorize requests to the Looker API.

To learn more about generating new Looker API keys, see Admin settings - Users (/looker/docs/admin-panel-users-users#api_keys). To learn more about authentication methods and managing Looker API keys, see Looker API authentication (/looker/docs/api-auth).

HTTP using Python... Python SDK (#python-sdk)

After you generate the API keys (client ID and secret), you can configure the Conversational Analytics API to connect to Looker by using Python. The following sample Python code demonstrates how to specify your Looker data source details and your API keys to the Conversational Analytics API.

Tip: Store the Looker client ID (**`looker_client_id`**) and the Looker client secret (**`looker_client_secret`**) as environment variables for improved security.

```
looker_client_id = "YOUR-LOOKER-CLIENT-ID" # @param {type:"string"}
looker_client_secret = "YOUR-LOOKER-CLIENT-SECRET" # @param {type:"strir
```

```
looker_instance_uri = "YOUR-LOOKER-INSTANCE-URI " # @param {type:"string'
lookml_model = "YOUR-LOOKER-MODEL " # @param {type:"string"}
explore = "YOUR-LOOKER-EXPLORE " # @param {type:"string"}

# Looker data source
looker_explore_reference = geminidataanalytics.LookerExploreReference()
looker_explore_reference.looker_instance_uri = looker_instance_uri
looker_explore_reference.lookml_model = lookml_model
looker_explore_reference.explore = explore

credentials = geminidataanalytics.Credentials()
credentials.oauth.secret.client_id = looker_client_id
credentials.oauth.secret.client_secret = looker_client_secret

# Connect to your data source
datasource_references = geminidataanalytics.DatasourceReferences()
datasource_references.looker.explore_references = [looker_explore_referenc
```

Replace the sample values as follows:

- ***YOUR-LOOKER-CLIENT-ID***: The client ID of your generated Looker API key
- ***YOUR-LOOKER-CLIENT-SECRET***: The client secret of your generated Looker API key
- ***YOUR-LOOKER-INSTANCE-URI***: The complete URL of your Looker instance
- ***YOUR-LOOKER-MODEL***: The name of the Looker model that you want to use
- ***YOUR-LOOKER-EXPLORE***: The name of the Looker Explore that you want to use

Authentication with an access token

This section describes how to configure the Conversational Analytics API to connect to Looker using an access token.

To establish a connection with a Looker instance, you need a valid OAuth2 **`access_token`** value, which is created by a successful request to the **`login`** Looker API endpoint.

To learn more about generating an access token, see Looker API authentication (/looker/docs/api-auth) and how to present client credentials to obtain an authorization token

(/looker/docs/reference/looker-api/latest/methods/ApiAuth/login#present-client-credentials-to-obtain-an-authorization-token)

.

Note: The credentials used to generate the **`access_token`** value must have the permissions needed to use the **`Get LookML Model Explore`** (/looker/docs/reference/looker-api/latest/methods/LookmlModel/lookml_model_explore) and **`Run Inline Query`** (/looker/docs/reference/looker-api/latest/methods/Query/run_inline_query) Looker endpoints. The **`develop`** (/looker/docs/admin-panel-users-roles#develop) permission is a minimum requirement.

✓ Using a public IP Using a private IP

HTTP using Python... Python SDKHTTP using JavaScript...
(#python-sdk)

The following sample Python code demonstrates how to define your Looker data source details and your access token to authenticate using the Python SDK.

We suggest storing the Looker access token (`access_token`) as an environment variable for improved security.

```
looker_access_token = "YOUR-TOKEN"
looker_instance_uri = "YOUR-LOOKER-INSTANCE-URI"
lookml_model = "YOUR-LOOKER-MODEL"
explore = "YOUR-LOOKER-EXPLORE"

# Looker data source
looker_explore_reference = geminidataanalytics.LookerExploreReference()
looker_explore_reference.looker_instance_uri = looker_instance_uri
looker_explore_reference.lookml_model = lookml_model
looker_explore_reference.explore = explore

credentials = geminidataanalytics.Credentials()
credentials.oauth.token.access_token = looker_access_token

# Connect to your data source
datasource_references = geminidataanalytics.DatasourceReferences()
datasource_references.looker.explore_references = [looker_explore_referenc
```

Replace the sample values as follows:

- ***YOUR-TOKEN***: The `access_token` value you use to authenticate to Looker
- ***YOUR-LOOKER-INSTANCE-URI***: The complete URL of your Looker instance
- ***YOUR-LOOKER-MODEL***: The name of the Looker model that you want to use
- ***YOUR-LOOKER-EXPLORE***: The name of the Looker Explore that you want to use

Connect to BigQuery with the Conversational Analytics API

To connect to one or more BigQuery tables with the Conversational Analytics API, you must authenticate to the relevant BigQuery project for each table. For each table, provide the following information:

- The BigQuery project ID
- The BigQuery dataset ID
- The BigQuery table ID

With the Conversational Analytics API, there are no hard limits on the number of BigQuery tables that you can connect to. However, connecting to a large number of tables can reduce accuracy or cause you to exceed Gemini's input token limit. Queries that require complex joins across multiple tables might also result in less accurate responses.

This section describes how to configure the Conversational Analytics API to connect to BigQuery by using either direct HTTP requests or an SDK.

HTTP using Python... Python SDK (#python-sdk)

You can use the `auth` SDK from Colaboratory to authenticate to BigQuery by using the credentials of your user that is authenticated to Colaboratory.

The following sample Python code defines a connection to multiple BigQuery tables and demonstrates how to authenticate your Google Account to BigQuery within Colaboratory.

```
from google.colab import auth
auth.authenticate_user()

# BigQuery data source
bigquery_table_reference = geminidataanalytics.BigQueryTableReference()
bigquery_table_reference.project_id = "my_project_id"
bigquery_table_reference.dataset_id = "my_dataset_id"
bigquery_table_reference.table_id = "my_table_id"

bigquery_table_reference_2 = geminidataanalytics.BigQueryTableReference()
bigquery_table_reference_2.project_id = "my_project_id_2"
bigquery_table_reference_2.dataset_id = "my_dataset_id_2"
bigquery_table_reference_2.table_id = "my_table_id_2"

bigquery_table_reference_3 = geminidataanalytics.BigQueryTableReference()
bigquery_table_reference_3.project_id = "my_project_id_3"
bigquery_table_reference_3.dataset_id = "my_dataset_id_3"
bigquery_table_reference_3.table_id = "my_table_id_3"

# Connect to your data source
datasource_references = geminidataanalytics.DatasourceReferences()
datasource_references.bq.table_references = [bigquery_table_reference, big
```

Replace the sample values as follows:

- ***my_project_id***: The ID of the Google Cloud project that contains the BigQuery dataset and table that you want to connect to. To connect to a public dataset (/bigquery/public-data), specify **`bigquery-public-data`**.
- ***my_dataset_id***: The ID of the BigQuery dataset. For example, `san_francisco`.
- ***my_table_id***: The ID of the BigQuery table. For example, `street_trees`.

Connect to Looker Studio with the Conversational Analytics API

To connect to Looker Studio with the Conversational Analytics API, you must first enable the Looker Studio API (#looker-studio-api). This section describes how to configure the

Conversational Analytics API to connect to Looker Studio by using either direct HTTP requests or an SDK.

Enable Looker Studio API

To enable the Looker Studio API, follow the instructions in Enable the API (https://developers.google.com/looker-studio/integrate/api#enable_the_api).

Authenticate to Looker Studio

To connect to Looker Studio with the Conversational Analytics API, you must authenticate to Looker Studio and provide the Looker Studio data source ID.

HTTP using Python... Python SDK (#python-sdk)

After you enable the Looker Studio API, you can authenticate to Looker Studio by using an SDK. The following sample Python code demonstrates how to specify your Looker data source details and authenticate to Looker Studio.

```
datasource_id = "STUDIO-DATASOURCE-ID"

# Looker Studio
studio_references = geminidataanalytics.StudioDatasourceReference()
studio_references.datasource_id = studio_datasource_id

# Connect to your data source
datasource_references = geminidataanalytics.DatasourceReferences()
datasource_references.studio.studio_references = [studio_references]
```

Replace ***STUDIO-DATASOURCE-ID*** with the actual data source ID of the Looker Studio data source that you want to use.

Last updated 2025-09-12 UTC.