Google Search Analytics Import Integration

You can import Google Search Analytics data into Treasure Data. You can filter and group data by dimensions such as page, query, country, or device. You can specify a schedule to periodically import the data.

Prerequisites

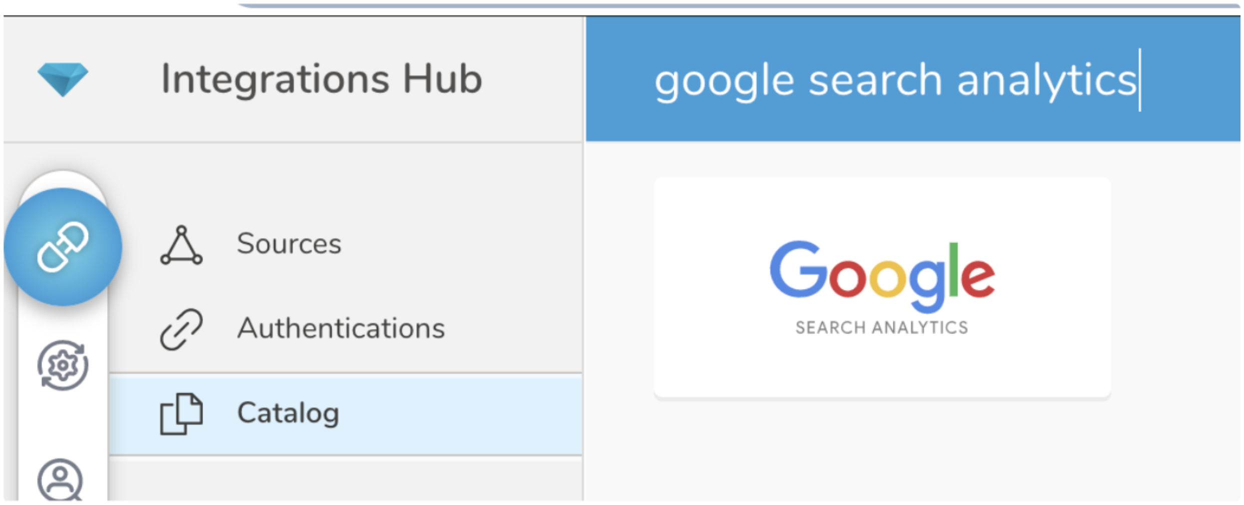
- Basic knowledge of Treasure Data
- Basic knowledge of Google Search Analytics
- A Google Search Console account.

Use the TD Console to Create Your Connection

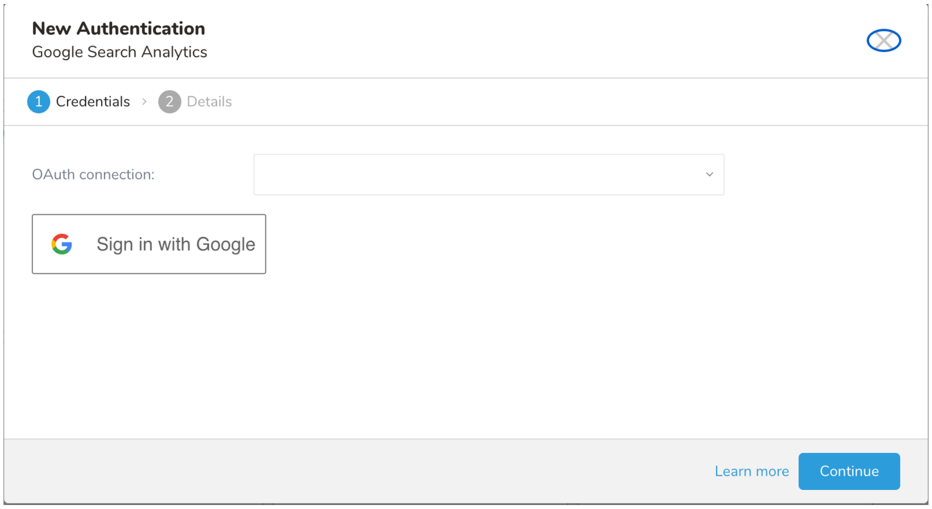
Create a New Connection

When you configure a data connection, you provide authentication to access the integration. In Treasure Data, you configure the authentication and then specify the source information.

Go to **Integrations Hub** > **Catalog** and search and select **Google Search Analytics.**



The following dialog opens

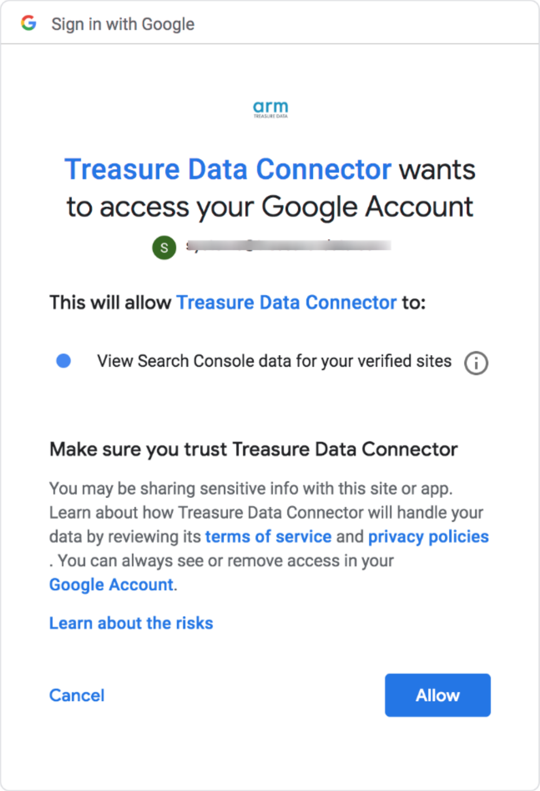


Access to Treasure Data Google Search Analytics requires OAuth2 authentication. The authentication requires that users manually connect their Treasure Data account to their respective Google Search Analytics account.

To authenticate, complete the following procedure:

Select **Click here** to connect to a new account.

Log into your Google account, in the popup window and grant access to the Treasure Data Connector app.



Treasure Data Connector wants to access your Google Account

S [redacted]

This will allow Treasure Data Connector to:

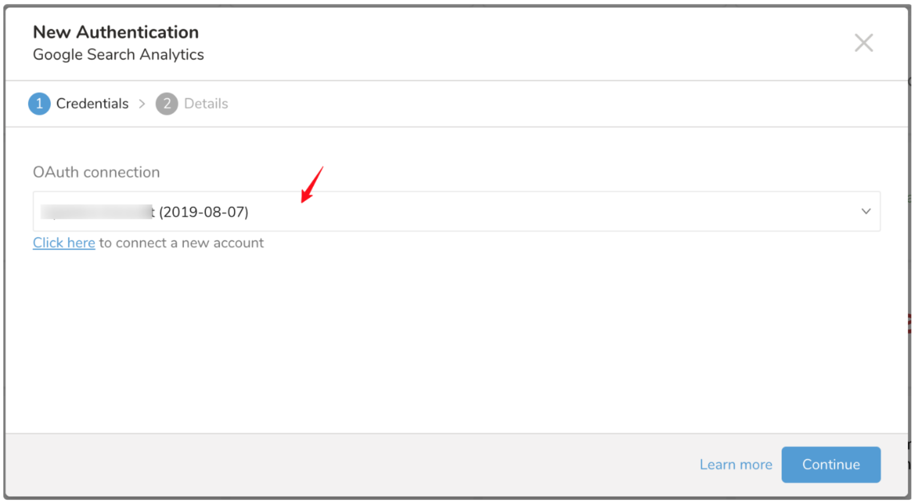
- View Search Console data for your verified sites

Make sure you trust Treasure Data Connector

You may be sharing sensitive info with this site or app. Learn about how Treasure Data Connector will handle your data by reviewing its terms of service and privacy policies . You can always see or remove access in your Google Account.

Learn about the risks

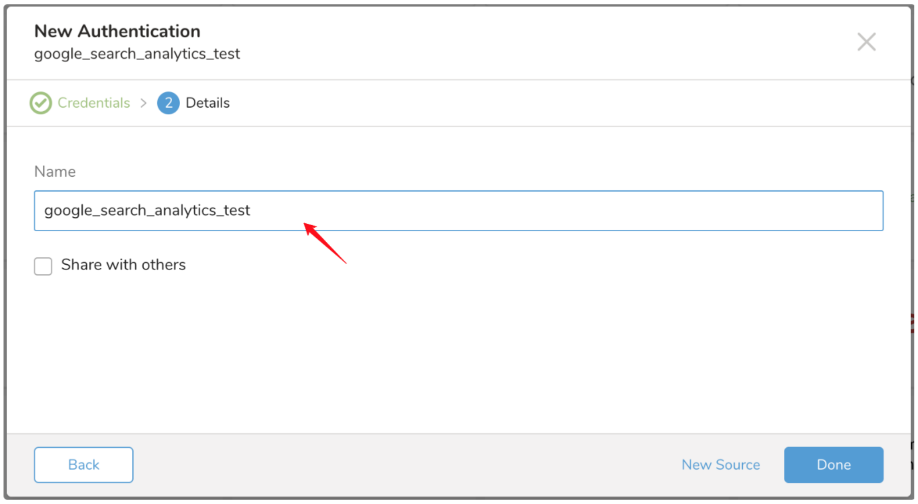
You will be redirected back to TD Console. Repeat the first step (Create a new connection) and choose your new OAuth connection.



OAuth connection: [redacted]t (2019-08-07)

Click here to connect a new account

Name your new OneDrive Connection. Select **Done.**



New Authentication
google_search_analytics_test

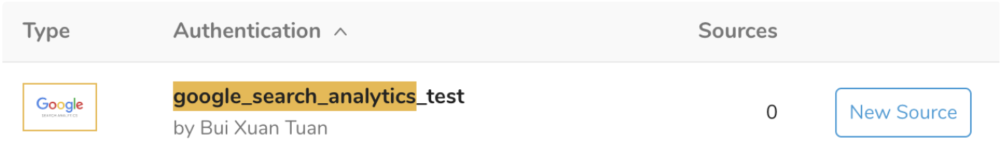
✓ Credentials > 2 Details

Name: google_search_analytics_test

□ Share with others

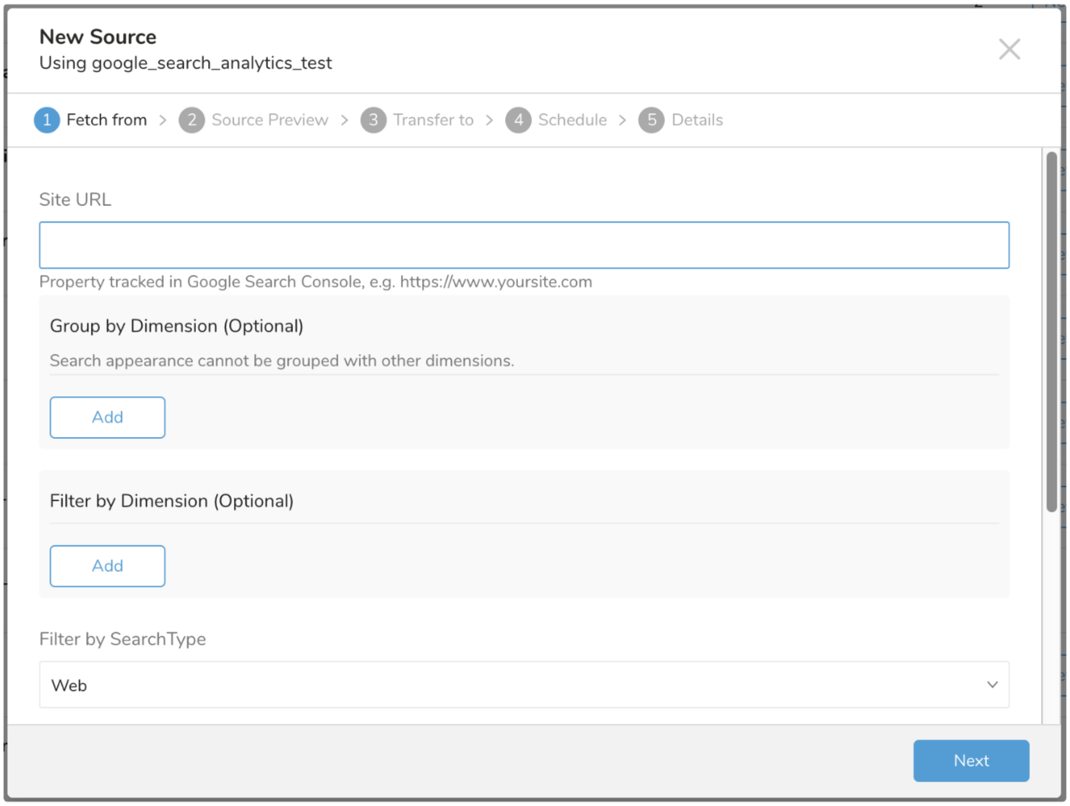
Transfer Your Google Search Analytics Data to Treasure Data

In **Authentications**, configure the **New Source**.



Type	Authentication ^	Sources	
Google	google_search_analytics_test by Bui Xuan Tuan	0	New Source

Complete the details and select **Next**.



New Source
Using google_search_analytics_test

1 Fetch from > 2 Source Preview > 3 Transfer to > 4 Schedule > 5 Details

Site URL

Property tracked in Google Search Console, e.g. https://www.yoursite.com

Group by Dimension (Optional)

Search appearance cannot be grouped with other dimensions.

Filter by Dimension (Optional)

Filter by SearchType: Web

Parameters

Site URL

Property tracked in Google Search Console. Google sees the protocols (http:// or https://), domain, subdomains (for example example.com, m.example.com, and www.example.com) differently, you must enter the exact URL of the domain, subdomain, or path branch from which you want to get data.

If you use Domain properties, use the following format when specifying your domain: sc-domain:example.com.

Group by Dimension

You can group the data you import by dimensions, one or more. Supported Dimensions are: Page, Country, Device, Query, and Search Appearance.

Note that you can not group Search Appearance with other **Dimensions**. Examples of valid dimension groups: Page, Country, Device.

The numbers of selected Dimensions affect the output Schema. For example, if the Dimensions: Country, Device and Query are selected, then the import is Country, Device and Query columns as shown:

```
+----------------+---------------------+---------------+--------------+--------------------+------------
+-----------------+----------------------------------+
| country:string |        query:string | device:string | click:double | impressions:double | ctr:double |
position:double | response_aggregation_type:string |
+----------------+---------------------+---------------+--------------+--------------------+------------
+-----------------+----------------------------------+
|            usa |       a sample data |       DESKTOP |          3.0 |                8.0 |      0.375 
|            1.25 |                       byProperty |
|            usa |       a sample data |        MOBILE |          3.0 |                5.0 |        0.6 
|             1.0 |                       byProperty |
|            twn | a sample data japan |        TABLET |          2.0 |                2.0 |        1.0 
|             1.0 |                       byProperty |
+----------------+---------------------+---------------+--------------+--------------------+------------
+-----------------+----------------------------------+
```

Filter by Dimension

You can filter data in multiple categories. For example, if you are currently grouping data by query, you can add the filters "query contains 'treasure data'".

The Google Search Analytics supports only the “AND” operator currently. Therefore, a filter like:"Country equals'USA'" AND "Device equals'Mobile'" AND "Query contains 'sample data'" returns data, but "Country equals'USA'" AND "Country equals 'JPN'" AND "query contains 'sample data'" won't return data.

Search Type

Filter analytics data by search type. Supported value: Web, Image, or Video. The web is selected by default. The selected value is included in the output schema under column search_type.

Start Date and End Date

The requested date range, **Start Date** and **End Date,** are required. Specify in YYYY-MM-DD format. The **End Date** must be greater than or equal to the **Start Date**. The values are inclusive: start at the beginning of the **Start Date** (**Start Date** at 00:00:00) and end at the end of the **End Date** (**End Date** at 23:59:59). If you want to import on a specific day then set the **Start Date** equal to the **End Date**. For example: Start Date = 2018-06-11, End Date = 2018-06-11.

The data is aggregated by the **Start Date** and **End Date.** If you set the date range greater than 1 day then the value of a click, ctr, impressions, the position is the sum of individual days, over the range of days specified.

The resulting data is in PST time (UTC-8:00).

Data might have a 2-3 day delay, waiting until the data is available in Google Search Console. As a best practice, don't set an End Date that is greater than date when data is available.

Incremental Loading

By enabling Incremental Loading, you can schedule a job to run iteratively. The next iteration of the job run is calculated from the **Start Date** and **End Date** values.

Following is an example of the next iteration of `single day` and `multiple days` time ranges:

Single-day range

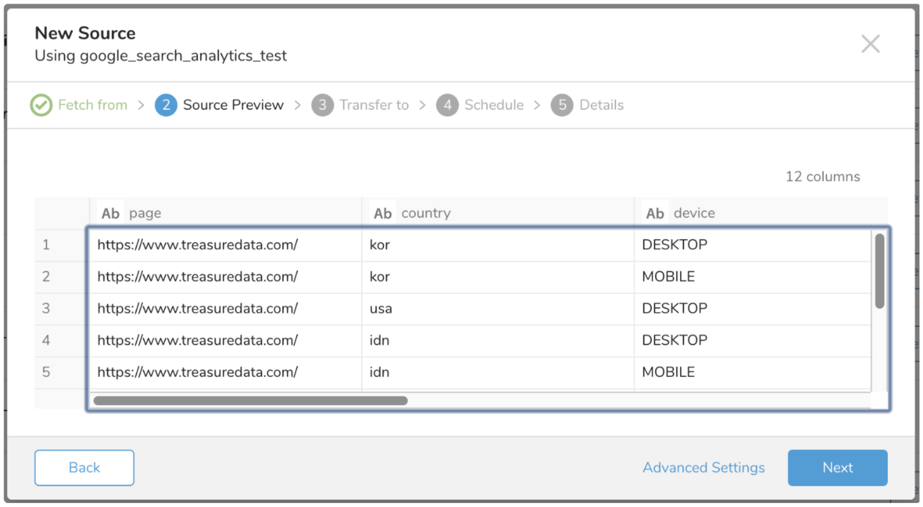
Jobs	Start Date	End Date
First run	2018-01-01	2018-01-01
Second run	2018-01-02	2018-01-02
Third run	2018-01-03	2018-01-03

1 Week range

Jobs	Start Date	End Date
First run	2018-01-01	2018-01-07
Second run	2018-01-08	2018-01-15
Third run	2018-01-16	2018-01-23

Preview and Advanced Settings

You see a Preview of your data similar to the following dialog. Read more about Preview.



New Source
Using google_search_analytics_test

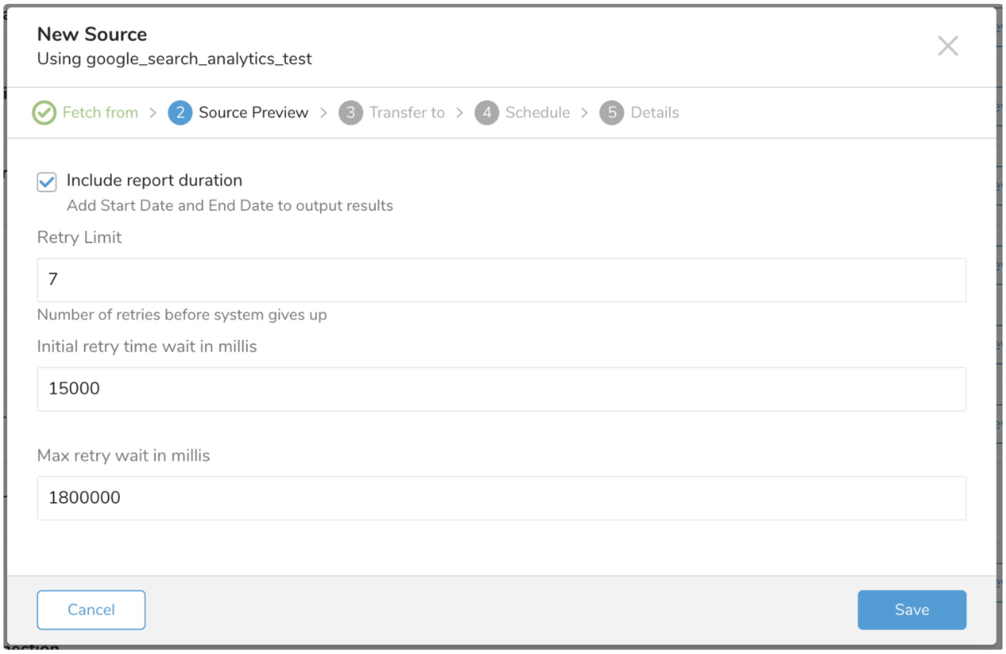
✓ Fetch from > 2 Source Preview > 3 Transfer to > 4 Schedule > 5 Details

12 columns

	Ab page	Ab country	Ab device
1	https://www.treasuredata.com/	kor	DESKTOP
2	https://www.treasuredata.com/	kor	MOBILE
3	https://www.treasuredata.com/	usa	DESKTOP
4	https://www.treasuredata.com/	idn	DESKTOP
5	https://www.treasuredata.com/	idn	MOBILE

Back
Advanced Settings
Next

Select **Advanced Settings** to customize the data connector behavior.



New Source
Using google_search_analytics_test

✓ Fetch from > 2 Source Preview > 3 Transfer to > 4 Schedule > 5 Details

☑ Include report duration
Add Start Date and End Date to output results

Retry Limit
7
Number of retries before system gives up

Initial retry time wait in millis
15000

Max retry wait in millis
1800000

Include Report Duration

The Include Report Duration option affects the output Schema. If the checkbox is selected (by default) then the result of the import job has 2 extra columns (**Start Date** and **End Date**, with the type of string) as shown in the following example:

```
+--------------------------+--------------+--------------------+--------------------+--------------------
+------------------+-----------------+
|             query:string | click:double | impressions:double |         ctr:double |    position:double
|start_date:string | end_date:string |
+--------------------------+--------------+--------------------+--------------------+--------------------
+------------------+-----------------+
|            a sample data |         11.0 |               35.0 | 0.3142857142857143 |  1.342857142857143
|       2018-05-05 |      2018-05-05 |
|             a sampledata |          3.0 |                8.0 |              0.375 |              1.625
|       2018-05-05 |      2018-05-05 |
|      a sample data japan |          2.0 |                2.0 |                1.0 |                1.0
|       2018-05-05 |      2018-05-05 |
|               cdp vs dmp |          1.0 |                3.0 | 0.3333333333333333 | 1.6666666666666665
|       2018-05-05 |      2018-05-05 |
|               cmp vs dmp |          1.0 |                1.0 |                1.0 |                7.0
|       2018-05-05 |      2018-05-05 |
|        a sample treasure |          1.0 |                1.0 |                1.0 |                1.0
|       2018-05-05 |      2018-05-05 |
|               hive guide |          1.0 |                2.0 |                0.5 |                4.5
|       2018-05-05 |      2018-05-05 |
| postgresql elasticcloud  |          1.0 |                4.0 |               0.25 |                8.5
|       2018-05-05 |      2018-05-05 |
|         s3 elasticcloud  |          1.0 |                1.0 |                1.0 |               11.0
|       2018-05-05 |      2018-05-05 |
|            a sample data |          1.0 |                1.0 |                1.0 |                1.0
|       2018-05-05 |      2018-05-05 |
+--------------------------+--------------+--------------------+--------------------+--------------------
+------------------+-----------------+
```

Retry Limit

During the import of data into Treasure Data, there are a number of factors that affect the process, such as network fluctuation, and concurrency limit on Google server. The connector retries import for the number of times you specify.

Initial retry time wait in millis

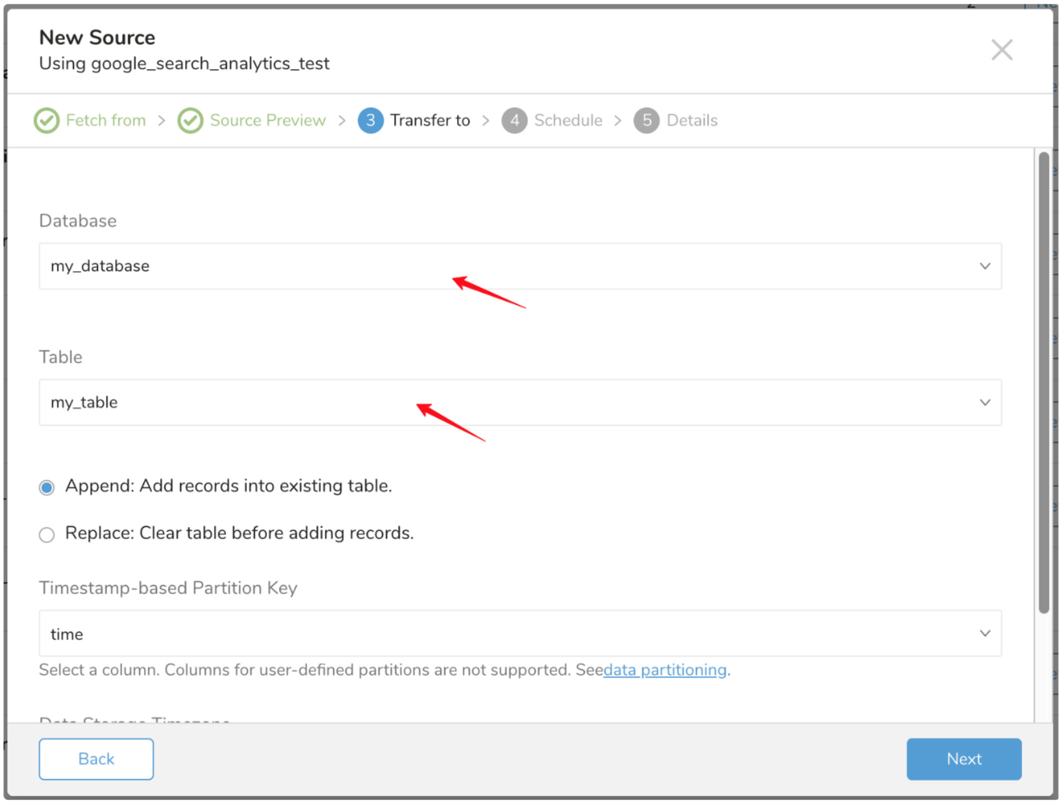
The connector initially waits for this amount of milliseconds that you specify before retrying import. The next retry is 2 * initial retry time, and so on.

Max retry wait in millis

The connector aborts the retry if the amount of wait time reaches this limit that you specify.

Choose the Target Database and Table

Choose an existing or create a new database and table.



New Source

Using google_search_analytics_test

Database

my_database

Table

my_table

● Append: Add records into existing table.

○ Replace: Clear table before adding records.

Timestamp-based Partition Key

time

Select a column. Columns for user-defined partitions are not supported. Seedata partitioning.

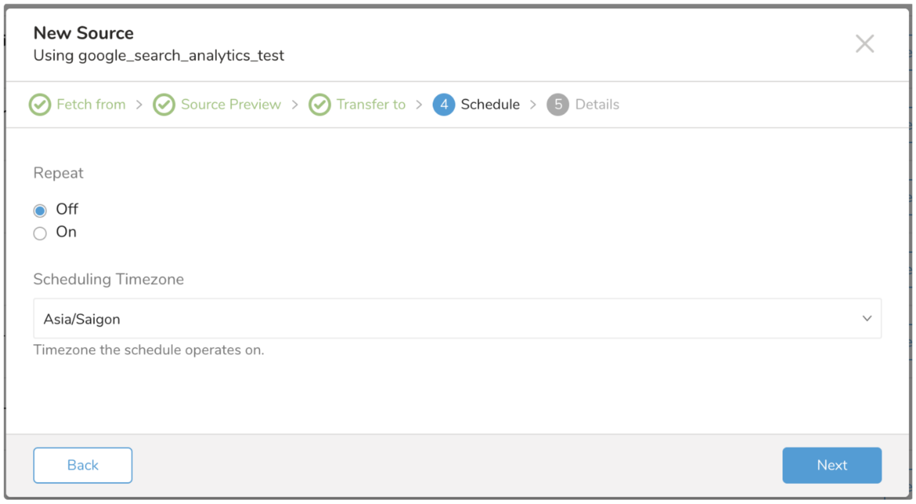
Create a new database and give your database a name. Complete similar steps for **Create new table.**

Select whether to **append** records to an existing table or **replace** your existing table.

If you want to set a different **partition key seed** rather than use the default key, you can specify one using the popup menu.

Scheduling

In the **Schedule** tab, you can specify a one-time transfer, or you can schedule an automated recurring transfer. If you select **Once now,** select **Start Transfer.** If you select **Repeat…** specify your schedule options, then select **Schedule Transfer.**



New Source

Using google_search_analytics_test

Repeat

● Off

○ On

Scheduling Timezone

Asia/Saigon

Timezone the schedule operates on.

Enter the New Source name and select **Done.**

Use Command Line

Install ‘td’ Command

Install the Treasure Data Toolbelt.

Obtain Required Google API Credentials

The connector config for 'td' command requires:

```
* client_id
* client_secret
* refresh_token
```

These parameters can be obtained by the following steps:

Obtain client_id, client secret

If you already have set up OAuth 2 and enabled Google Search Console APIs access, and have client ID and client secret in place, you can skip this step.

To get the client ID and client secret, follow the instructions for Google Search Console APIs for developers: https://developers.google.com/webmaster-tools/search-console-api-original/v3/how-tos/authorizing

You can also use the Google Search Console API Wizard which allows you quickly create a project and turn on the Google Search Console API. Go to Credentials > Create Credentials > OAuth ClientID > Web Application. Enter a name then select **Create**. The next screen shows you the client ID and client secret.

Obtain refresh_token

The remaining credential you need to get is the refresh token. One way to get the refresh token is to use the Google OAuth 2.0 Playground, available here: Google OAuth 2.0 Playground

First from within the OAuth 2.0 Playground, on the top right corner, select the Gear Icon, and make sure to select the Use your own OAuth credentials checkbox. In the OAuth `Client ID` and OAuth `Client secret`, insert the credentials you got from the API console.

At step 1, select the **Search Console API v3**. Then select the "*https://www.googleapis.com/auth/webmasters.readonly*" and select "**Authorize APIs**" and follow the instruction to grant access from Search Console API to your Google Search Console account.

At step 2, Select Exchange authorization code for tokens, which fills the Refresh Token and Access Token fields. The refresh token field is the value used to prepare the Connector configuration in the next steps.

Prepare Configuration and Preview Data

Prepare config.yml as shown:

```
in:
  type: "google_search_analytics"
  client_id: "[Your Client ID]"
  client_secret: "[Your Client Secret]"
  refresh_token: "[Your Refresh Token]"
  site_url: "[Your tracked site]"
  dimensions: ["query"]
  filters:
    - {
        "dimension": "query",
        "operator": "equals",
        "expression": "a sample"
      }
  search_type: web
  start_date: "2018-06-01"
  end_date: "2018-06-01"
  include_report_period: false
out:
  mode: append
filters:
  - {
      type: "add_time",
      to_column: {"name": "time"},
      from_value: {"mode": "upload_time"}
    }
```

For details on available **out** modes, see **Appendix A**

For details of **filters**, see **Appendix B**

Then, you can see a preview of the data using the preview command.

```
$ td connector:preview config.yml
+-------------------------+--------------+--------------------+---------------------+--------------------
+----------------------------------+-------------------------------+
| query:string            | click:double | impressions:double | ctr:double          | position:double    |
response_aggregation_type:string | time:timestamp                |
+-------------------------+--------------+--------------------+---------------------+--------------------
+----------------------------------+-------------------------------+
| "a sample data"         | 34.0         | 117.0              | 0.2905982905982906  | 1.3846153846153846 |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sampledata"          | 19.0         | 54.0               | 0.35185185185185186 | 1.6111111111111112 |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sample data cdp"     | 4.0          | 5.0                | 0.8                 | 1.6                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sample data inc"     | 2.0          | 3.0                | 0.6666666666666666  | 1.0                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sample data inc."    | 1.0          | 1.0                | 1.0                 | 1.0                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sample data pricing" | 1.0          | 2.0                | 0.5                 | 1.5                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sample data, inc"    | 1.0          | 1.0                | 1.0                 | 1.0                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a sample news"         | 1.0          | 1.0                | 1.0                 | 22.0               |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "a-sampledata.com"      | 1.0          | 1.0                | 1.0                 | 1.0                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
| "\"a sample data\""     | 0.0          | 2.0                | 0.0                 | 1.0                |
"byProperty"                     | "2018-06-25 05:28:43.078 UTC" |
+-------------------------+--------------+--------------------+---------------------+--------------------
+----------------------------------+-------------------------------+
```

You do not need to run td connector:guess for the Google Search Analytics connector.

Execute Load Job

Submit the load job. It may take a couple of hours to process depending on the size of the data.

```
$ td connector:issue config.yml --database td_sample_db --table td_sample_table
```

Scheduled Execution

You can schedule periodic data connector execution for incremental import of Search Analytics data.

For the scheduled import, at first run, the Data Connector for Search Analytics imports all of your data specified by "start_date" and "end_date".

On the second and subsequent runs, the connector imports only data that is newer than the last load.

Create the Schedule

You can create a schedule using the td connector:create command. The following are required: the name of the schedule, the cron-style schedule, the database and table where their data will be stored, and the data connector configuration file.

```
$ td connector:create \
    daily_import \
    "10 0 * * *" \
    td_sample_db \
    td_sample_table \
    config.yml
```

The `cron` parameter also accepts three special options: `@hourly`, `@daily`, and `@monthly`.

By default, the schedule is set up in the UTC timezone. You can set the schedule in a timezone using -t or --timezone option. Note that the `--timezone` option supports only extended timezone formats like 'Asia/Tokyo', 'America/Los_Angeles', etc. Timezone abbreviations like PST, CST are *not* supported and may lead to unexpected schedules.

List All Schedules

You can see the list of all current schedule entries with the command td connector:list.

```
$ td connector:list
+--------------+------------+----------+-------+--------------+-----------------+----------------------------+
| Name         | Cron       | Timezone | Delay | Database     | Table           | Config                     |
+--------------+------------+----------+-------+--------------+-----------------+----------------------------+
| daily_import | 10 0 * * * | UTC      | 0     | td_sample_db | td_sample_table | {"in"=>{"type"=>"fac", ... |
+--------------+------------+----------+-------+--------------+-----------------+----------------------------+
```

Show Schedule Settings and History

td connector:show shows the execution settings of a schedule entry.

```
% td connector:show daily_import
Name     : daily_import
Cron     : 0 0 * * *
Timezone : UTC
Delay    : 0
Database : td_sample_db
Table    : td_sample_table
Config
---
in:
  refresh_token: "***"
  client_id: "***"
  client_secret: "***"
  site_url: https://www.sample.com/
  filters:
  - dimension: query
    operator: contains
    expression: treasure
  start_date: '2018-06-11'
  end_date: '2018-06-11'
  dimensions:
  - query
  include_report_period: false
  type: google_search_analytics
filters:
- type: add_time
  to_column:
    name: time
  from_value:
    mode: upload_time
exec: {}
out:
  type: td_internal
  mode: append
  plazma_dataset: "***"

Config Diff
--- {}
```

td connector:history shows the execution history of a schedule entry. To investigate the results of each individual run, use td job <jobid>.

```
% td connector:history daily_import
+--------+---------+---------+--------------+-----------------+----------+---------------------------+----------
+
| JobID  | Status  | Records | Database     | Table           | Priority | Started                   | Duration
|
+--------+---------+---------+--------------+-----------------+----------+---------------------------+----------
+
| 578066 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-18 00:10:05 +0000 | 160
|
| 577968 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-17 00:10:07 +0000 | 161
|
| 577914 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-16 00:10:03 +0000 | 152
|
| 577872 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-15 00:10:04 +0000 | 163
|
| 577810 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-14 00:10:04 +0000 | 164
|
| 577766 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-13 00:10:04 +0000 | 155
|
| 577710 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-12 00:10:05 +0000 | 156
|
| 577610 | success | 10000   | td_sample_db | td_sample_table | 0        | 2018-06-11 00:10:04 +0000 | 157
|
+--------+---------+---------+--------------+-----------------+----------+---------------------------+----------
+
8 rows in set
```

Delete Schedule

td connector:delete removes the schedule.

```
$ td connector:delete daily_import
```

Configuration

The following table provides details on the options available in modes.

Option name	Description	Type	Required?	Default Value
client_id	app client id	string	yes	
client_secret	app client secret	string	yes	
refresh_token	app refresh token	string	yes	
site_url	site url to be tracked	string	yes	
dimensions	search analytics dimension to group result	enum	optional	
filters	filter by dimensions	enum	optional	
search_type	filter by search type. web, image or video	string	optional	web
start_date	start date of request date range	string	yes	
end_date	end date of request date range	string	yes	
incremental	generate next start_date and end_date for schedule import	bool	optional	true
include_report_period	specify if start_date and end_date include in the output result	bool	optional	true
retry_limit	when a retry-able error should give up	integer	optional	7 (times)
retry_initial_wait_millis	initial wait when a retry-able error occurs. Subsequence value will be 2 times of previous value	integer	optional	15000 (15 seconds)
max_retry_wait_millis	max wait for each retry	integer	optional	1800000 (30 minutes)

Appendix

Modes for Out Plugin

You can specify how new data is imported into Treasure data by specifying mode in out section of config.yml.

append (default)

This is the default mode and records are appended to the target table.

```
in:
  ...
out:
  mode: append
```

replace (In td 0.11.10 and later)

This mode replaces data in the target table. Any manual schema changes made to the target table remains intact with this mode.

```
in:
  ...
out:
  mode: replace
```

add_time Filter

The add_time filter plugin for Data Connector allows users to add a new time-based column to the schema either by copying the value from another existing column in the schema or by specifying the value. See document of add_time Filter Plugin for Integrations for more details.

Dimensions

- Page
- Country
- Device
- Query
- Search Appearance

Supported Filter Operators

- Contains
- Equals
- NotContains
- NotEquals

Added New Column

Started from version v0.1.4, a new column name: search_type is added to your job result as we support Search Type filter. The default value is "web" if you do not change the job configuration.