

# Brandwatch Import Integration

This data connector allows you to import Brandwatch Mention objects into Treasure Data.

## Limitations

**API Limitations.** If a case *query* or *query group* contains a large number of mentions, for example, more than 5000 mentions, the configuration takes up to 50 requests to fetch all mentions. Use the *page size* parameter to manage requests.

Continue to the following topics:

- [Prerequisites](#)
- [Using TD Console](#)
- [Use Command Line](#)
- [Scheduled Execution](#)
- [Incremental Loading for Data Extensions](#)

## Prerequisites

- Basic knowledge & access to a Treasure Data account
- Basic knowledge & access to a Brandwatch account

## Using TD Console

### Create a New Connection

Go to Integrations Hub > Catalog. Search and select Brandwatch.



The dialog opens.

Provide your Brandwatch Username and Password information, select **Continue** and give your connection a name:

**New Authentication** Brandwatch ✕

1 Credentials > 2 Details

Username

Password

[Learn more](#) [Continue](#)

## Create a New Source

After creating the connection, you are automatically taken back to Integrations Hub > Catalog. Look for the connection you created and select **New Source**

The dialog opens. Complete the details and select **Next**.

**New Source** Using helentsest ✕

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

Project name

Filter by

query

Query name

Incremental?  
When run repeatedly, attempt to only import new data since the last import

Start date

yyyy-MM-dd'T'HH:mm:ss.SSS z, eg. 2012-07-04T04:19:21.235-07:00

Export data from this time (inclusively).

End date

Next

Next, you see a Preview of your data similar to the following dialog. To make changes, select **Advanced Settings** if you want to change some options such as skipping on errors or rate limits. Otherwise, select **Next**.

Choose an existing or create a new database and table where you want to transfer data to.

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**.

After your transfer has run, you can see the results of your transfer in **Data Workbench > Databases**. A corresponding job appears in the Jobs section.

You are ready to start analyzing your data.

## Use Command Line

Install 'td' Command v0.11.9 or Later

You can install the latest [TD Toolbelt](#).

```
$ td --version
0.15.8
```

## Create Configuration File

Prepare configuration file (for eg: *load.yml*) as shown in the following example with your Brandwatch credential and transfer information.

```
in:
  type: brandwatch
  username: xxxxxxxxxxxx
  password: xxxxxxxxxxxx
  project_name: xxx
  query_name: xxx
  from_date: yyyy-MM-dd'T'hh:mm:ss.SSS'Z'
  to_date: yyyy-MM-dd'T'hh:mm:ss.SSS'Z'
out:
  mode: replace
```

This example shows a dump of Brandwatch mention by executing a query. Besides query, you have another option *query\_group* to fetch mention:

```
in:
  type: brandwatch
  username: xxxxxxxxxxxx
  password: xxxxxxxxxxxx
  project_name: xxx
  query_group_name: xxx
  from_date: yyyy-MM-dd'T'hh:mm:ss.SSS'Z'
  to_date: yyyy-MM-dd'T'hh:mm:ss.SSS'Z'
out:
  mode: replace
```

## Configuration

- *username*: Brandwatch account's username (string, required)
- *password*: Brandwatch account's password (string, required)
- *project\_name*: Brandwatch project which all of query, query group, mentions, etc. belong to (string, required)
- *query\_name*: Brandwatch query name will be executed to fetch mentions (string, optional)
- *query\_group\_name*: Brandwatch query group name will be executed to fetch mentions (string, optional)
  - **Note**: Either *query\_name* or *query\_group\_name* must exist, both must not exist at the same time
- *from\_date*: Specify the date and time to fetch records from (date format: *yyyy-MM-dd'T'hh:mm:ss.SSS'Z'*) (string, required, inclusive)
- *to\_date*: Specify the allowable duration to fetch records (date format: *yyyy-MM-dd'T'hh:mm:ss.SSS'Z'*) (string, required, exclusive)
- *retry\_initial\_wait\_msec*: Parameter that provides the initial wait time (in milliseconds) for each retry logic to call Brandwatch API
- *max\_retry\_wait\_msec*: Parameter that provides the maximum wait time (in milliseconds) for each retry to call Brandwatch API (int, optional)
- *retry\_limit*: Parameter that provides the number of attempts to call the Brandwatch API (int, optional)
- *page\_size*: Parameter that provides the number of mentions fetched per API call
  - **Note**: This parameter is really helpful to overcome the API limitation in case *query* or *query group* contains a large number of mentions e. g. *query X* contains totally 5000 mentions, by default configuration it will take up to 50 requests to fetch all of those mentions, the more requests it makes, the chances it will reach the API limitation will be high, if *page\_size* is changed to *200*, it will take up to only 25 requests with the previous query

## Preview Data to Import (Optional)

You can preview data to be imported using the command *td connector:preview*.

```
$ td connector:preview load.yml
+-----+-----+-----+
| accounttype:string | authorcity:string | authorcitycode:string | ...
+-----+-----+-----+
| individual         | "Atlanta"         | at19                    |
| individual         | "Atlanta"         | at19                    |
+-----+-----+-----+
```

## Execute Load Job

Submit the load job. It may take a couple of hours depending on the data size. Users need to specify the database and table where their data is stored.

It is recommended to specify `--time-column` option, since Treasure Data's storage is partitioned by time (see also [data partitioning](#)). If the option is not given, the data connector selects the first *long* or *timestamp* column as the partitioning time. The type of the column specified by `--time-column` must be either of *long* and *timestamp* type.

If your data doesn't have a time column you can add it using `add_time` filter option. More details at [add\\_time filter plugin](#).

```
$ td connector:issue load.yml --database td_sample_db --table td_sample_table --time-column modifieddate
```

The above command assumes you have already created *database(td\_sample\_db)* and *table(td\_sample\_table)*. If the database or the table do not exist in TD this command will not succeed, so create the database and table manually or use `--auto-create-table` option with `td connector:issue` command to auto create the database and table:

```
$ td connector:issue load.yml --database td_sample_db --table td_sample_table --time-column modifieddate --auto-create-table
```

You can assign Time Format column to the "Partitioning Key" by `--time-column` option.

## Scheduled Execution

You can schedule periodic data connector execution for periodic Brandwatch import. We configure our scheduler carefully to ensure high availability. By using this feature, you no longer need a *cron* daemon on your local data center.

A new schedule can be created using the `td connector:create` command. The name of the schedule, cron-style schedule, the database and table where their data will be stored, and the Data Connector configuration file are required.

```
$ td connector:create \
  daily_Brandwatch_import \
  "9 0 * * *" \
  td_sample_db \
  td_sample_table \
  load.yml
```

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

By default, schedule is setup in UTC timezone. You can set the schedule in a timezone using `-t` or `--timezone` option. The ``--timezone`` option only supports 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 the Schedules

You can see the list of scheduled entries by `td connector:list`.

```

$ td connector:list
+-----+-----+-----+-----+-----+-----+
| Name          | Cron      | Timezone | Delay | Database      | Table          |
Config         |           |          |      |               |               |
+-----+-----+-----+-----+-----+-----+
| daily_brandwatch_import | 9 0 * * * | UTC      | 0     | td_sample_db | td_sample_table | {"type"=>"
brandwatch", ... } |
+-----+-----+-----+-----+-----+-----+

```

## Show the Setting and History of Schedules

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

```

% td connector:show daily_brandwatch_import
Name      : daily_brandwatch_import
Cron      : 9 0 * * *
Timezone  : UTC
Delay     : 0
Database  : td_sample_db
Table     : td_sample_table

```

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

```

% td connector:history daily_salesforce_marketing_cloud_import
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| JobID | Status | Records | Database      | Table          | Priority | Started          | Duration |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 678066 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-28 00:09:05 +0000 | 160     |
| 677968 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-27 00:09:07 +0000 | 161     |
| 677914 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-26 00:09:03 +0000 | 152     |
| 677872 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-25 00:09:04 +0000 | 163     |
| 677810 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-24 00:09:04 +0000 | 164     |
| 677766 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-23 00:09:04 +0000 | 155     |
| 677710 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-22 00:09:05 +0000 | 156     |
| 677610 | success | 10000   | td_sample_db | td_sample_table | 0       | 2017-07-21 00:09:04 +0000 | 157     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
8 rows in set

```

## Delete the Schedule

*td connector:delete* removes the schedule.

```

$ td connector:delete daily_brandwatch_import

```

## Incremental Loading for Data Extensions

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.

In the following example, let's use an 11-day range between the start and end date:

```
Start Date: 2018-03-01T00:00:00Z
End Date: 2018-03-11T00:00:00Z
```

Each job will have the same time range as determined by the period between the start and end dates. The transfer of mentions begins at the completion of the previous job until the period extends past the current date. Further transfers are delayed until a complete period is available, at which the job executes and then pauses until the next period is available.

For example:

- The current date is *2018-04-26*, you pass the incremental loading with *from\_date = 2018-04-01T00:00:00Z* and *to\_date = 2018-04-11T00:00:00Z*
- Cron is configured to run daily at a certain hour

1st runs at 2018-04-26: *from\_date: 2018-04-01T00:00:00Z to\_date: 2018-04-11T00:00:00Z (exclusive, Mention is fetched up to 2018-03-10T23:59:59Z)*

2nd runs at 2018-04-27: *from\_date: 2018-04-11T00:00:00Z to\_date: 2018-04-22T00:00:00Z*

3rd runs at 2018-04-28: it's not able to run because *to\_date* is in the future

4th runs at 2018-04-29: it's not able to run

5th runs at 2018-04-30: it's not able to run

6th runs at 2018-05-01: it's not able to run

7th runs at 2018-05-02: it's not able to run

8th runs at 2018-05-03: *from\_date: 2018-04-22T00:00:00Z to\_date: 2017-05-03T00:00:00Z*