Apptopia Import Integration

This Data Connector allows you to import Apptopia Data Source objects into Treasure Data.

- Prerequisites
- Use TD Console
- Command Line
- Scheduled Execution
- Appendix

Prerequisites

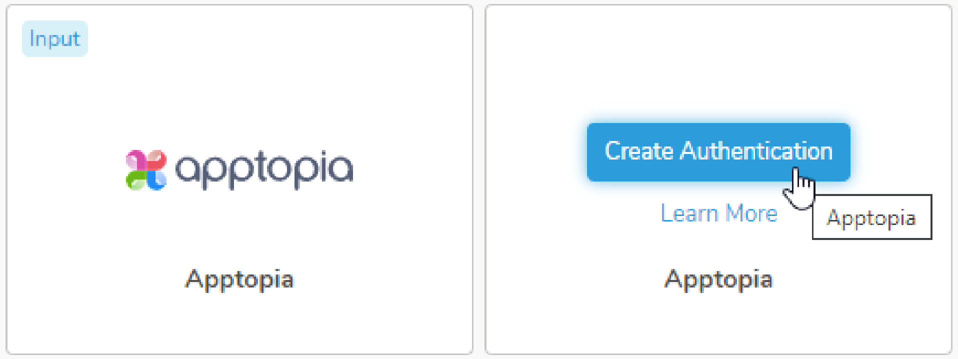
- Basic knowledge of Treasure Data
- Basic knowledge of Apptopia

Use TD Console

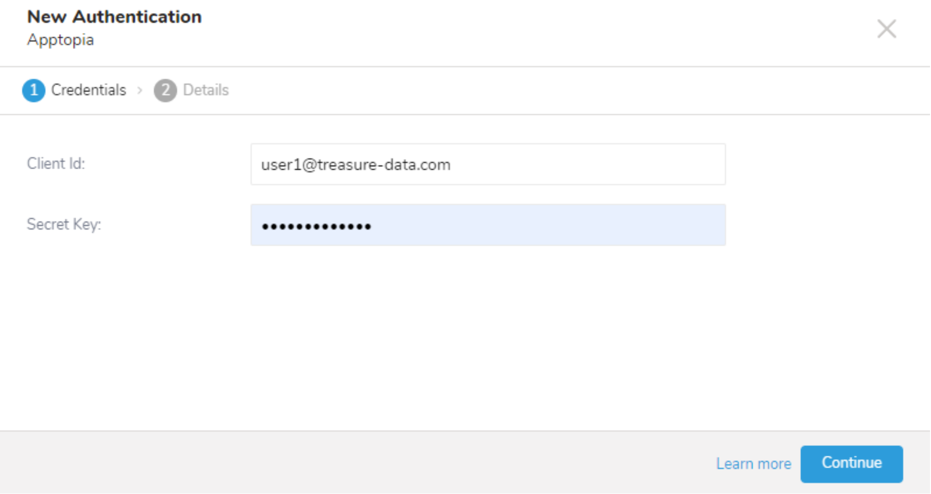
Create a new connection

In Treasure Data, you must create and configure the data connection prior to running your query. As part of the data connection, you provide authentication to access the integration.

1. Open **TD Console**.
2. Navigate to **Integrations Hub** > **Catalog**.
3. Click the search icon on the far-right of the Catalog screen, and enter **Apptopia**.
4. Hover over the Apptopia connector and select **Create Authentication**.



The following dialog opens.



New Authentication
Apptopia

1 Credentials › 2 Details

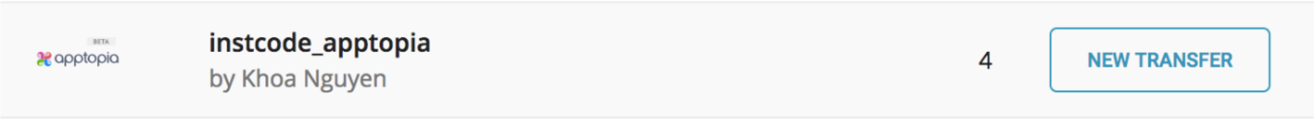
Client Id: user1@treasure-data.com

Secret Key: •••••••••••••

Edit your Apptopia Client ID and Secret Key information. Then click **Continue.** Give your connection a name, and click **Done**.

Create a new transfer

When you select **Create Connection**, you are taken to the Authentications tab. Look for the connection you created and select **New Transfer**.

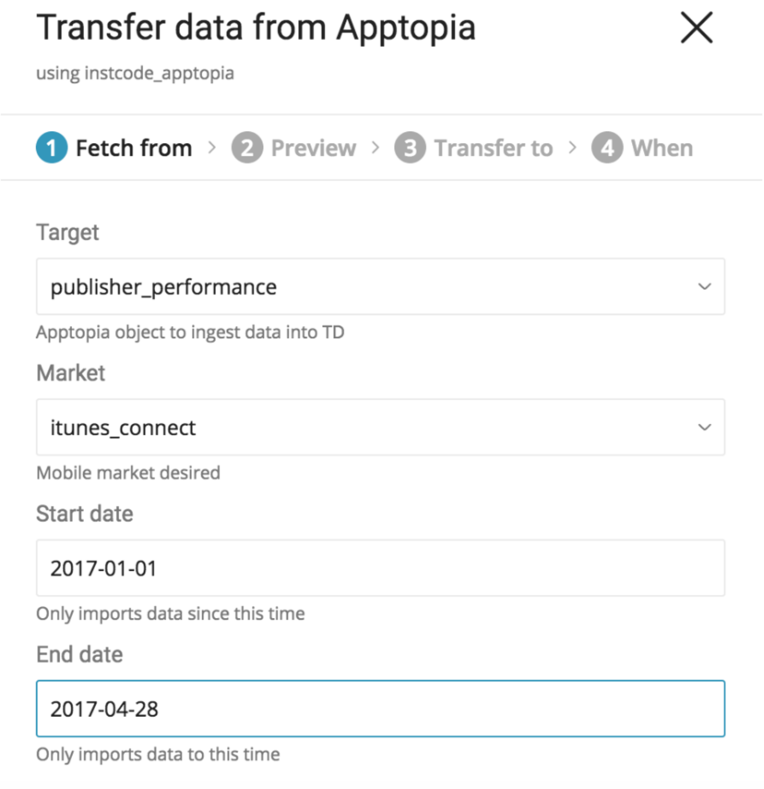


instcode_apptopia

by Khoa Nguyen

4

The following dialog opens. Edit the details and select **Next**.



Transfer data from Apptopia

using instcode_apptopia

1 Fetch from > 2 Preview > 3 Transfer to > 4 When

Target

publisher_performance

Apptopia object to ingest data into TD

Market

itunes_connect

Mobile market desired

Start date

2017-01-01

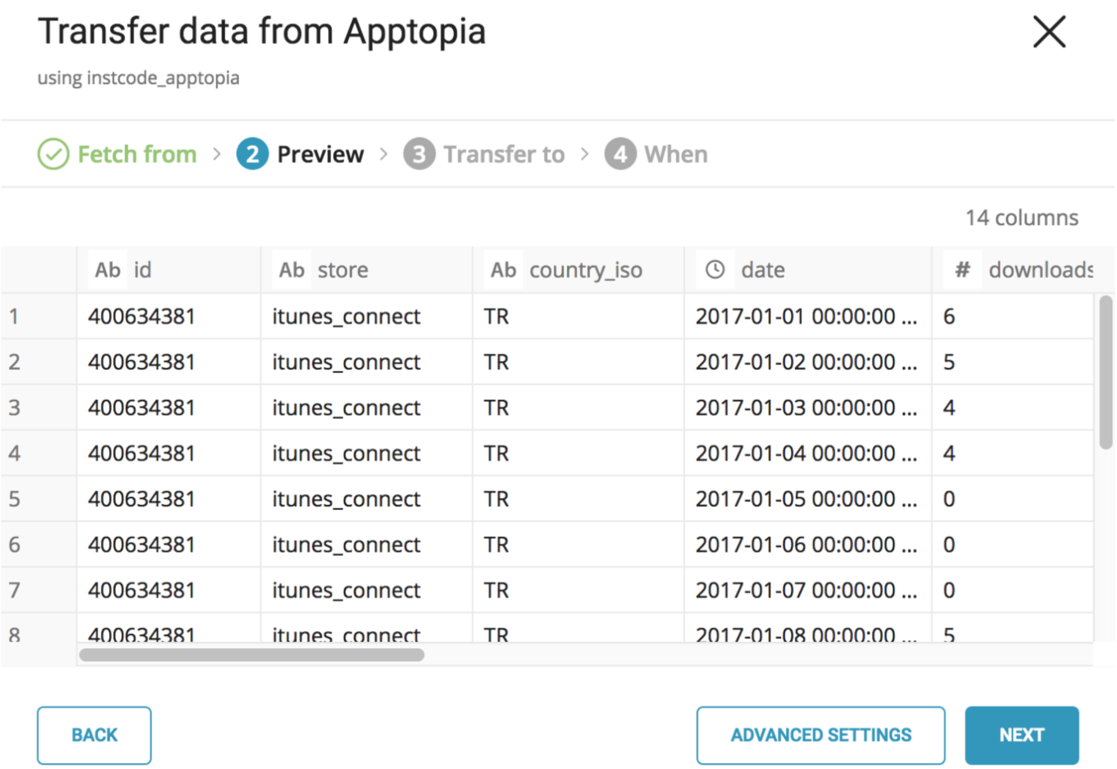
Only imports data since this time

End date

2017-04-28

Only imports data to this time

Preview your data. If you wish to change anything, select **Advanced Settings** or select **Next**.



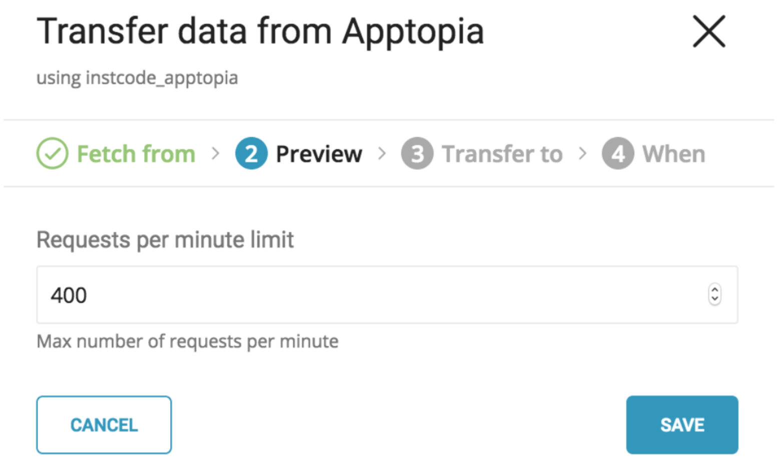
Transfer data from Apptopia

using instcode_apptopia

14 columns

	Ab id	Ab store	Ab country_iso	date	# downloads
1	400634381	itunes_connect	TR	2017-01-01 00:00:00 ...	6
2	400634381	itunes_connect	TR	2017-01-02 00:00:00 ...	5
3	400634381	itunes_connect	TR	2017-01-03 00:00:00 ...	4
4	400634381	itunes_connect	TR	2017-01-04 00:00:00 ...	4
5	400634381	itunes_connect	TR	2017-01-05 00:00:00 ...	0
6	400634381	itunes_connect	TR	2017-01-06 00:00:00 ...	0
7	400634381	itunes_connect	TR	2017-01-07 00:00:00 ...	0
8	400634381	itunes_connect	TR	2017-01-08 00:00:00 ...	5

In Advanced Settings you can change some options such as rate limits:



Transfer data from Apptopia

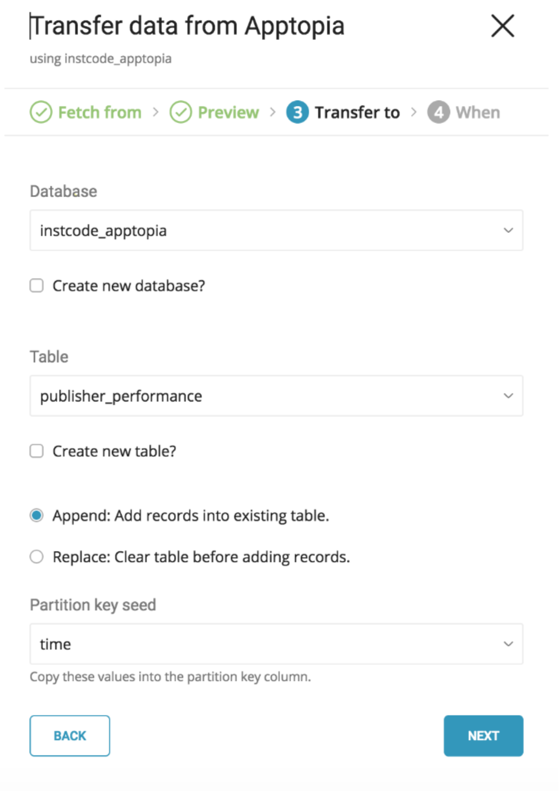
using instcode_apptopia

Requests per minute limit

400

Max number of requests per minute

Select the database and table where you want to transfer the data, as shown in the following dialog:


Transfer data from Apptopia

using instcode_apptopia

Database: instcode_apptopia

□ Create new database?

Table: publisher_performance

□ Create new table?

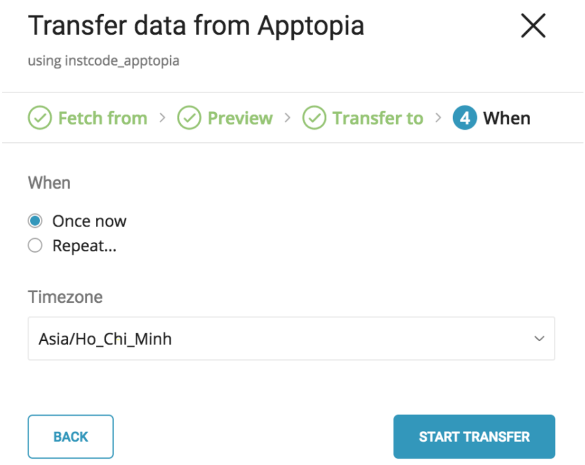
● Append: Add records into existing table.

○ Replace: Clear table before adding records.

Partition key seed: time

Copy these values into the partition key column.

Specify the schedule of the data transfer and select **Start Transfer**:


Transfer data from Apptopia

using instcode_apptopia

When

● Once now

○ Repeat...

Timezone: Asia/Ho_Chi_Minh

You will see the new data transfer in progress, listed under the My Input Transfers tab, and a corresponding job will be listed in the Jobs section.

Command Line

Install ‘td’ command v0.11.9 or later

You can install the newest TD Toolbelt.

```
$ td --version
0.15.0
```

Create Configuration File

Prepare configuration file (for eg: load.yml) with your Apptopia account access information, as follows:

```
in:
  type: apptopia
  client: xxxxxxxx
  secret: xxxxxxxx
  target: publisher_performance (required, see Appendix B)
  store: itunes_connect (required, see Appendix C)
  start_date: 2017-01-01 (required for all targets except `category` and `country`)
  end_date: 2017-02-01 (required for all targets except `category` and `country`)
  requests_per_minute_limit: 300 (optional, 1000 by default, see Appendix F)
out:
  mode: replace
```

This example shows an Apptopia Publisher Performance Data Source:

- client: Apptopia client id.
- secret: Apptopia secret.
- target: Apptopia entity object to be imported.
    - See Appendix for the list of available target.
- store: Mobile market to fetch data
    - See Appendix for the list of available store.
- start_date: From which date (yyyy-MM-dd) product data to be imported. This field is required when fetching product usage (target is product_usage) or product sale (target is product_sales) with in-app-purchase breakdown (breakdown has iap).
- end_date: To which date (yyyy-MM-dd) product data to be imported. This field is optional and is automatically adjusted to maximum 60 days since start_date limited by current date.
- requests_per_minute_limit: Limit number of API calls per minute
    - See Appendix for how to use these options

For more details on available out modes, see the Appendix

Preview Data to Import (optional)

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

```
$ td connector:preview load.yml
+------------+------------------+--------------------+----
| id:string  | store:string     | country_iso:string | ...
+------------+------------------+--------------------+----
| 420233213  | "itunes_connect" | US                 |
| jp.td.app  | "google_play"    | JP                 |
+------------+------------------+--------------------+----
```

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 are stored.

It is recommended to specify --time-column option, because Treasure Data’s storage is partitioned by time (see also data partitioning) If the option is not given, the Data Connector will choose 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 by using add_time filter option. For more details, see add_time Filter Plugin for Integrations.

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

The preceding command assumes that you already created *database(td_sample_db)* and *table(td_sample_table)*. If the database or the table does not exist in TD, the command will not succeed, so create the database and table Database and Table Management 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 updated_date --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 Apptopia 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.

Create the Schedule

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_apptopia_import \
    "10 0 * * *" \
    td_sample_db \
    td_sample_table \
    load.yml
```

The `cron` parameter also accepts the 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 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 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_apptopia_import | 10 0 * * *   | UTC      | 0     | td_sample_db | td_sample_table | {"type"=>"
apptopia", ... }  |
+-----------------------+--------------+----------+-------+--------------+-----------------
+-----------------------------+
```

Show the Setting and History of Schedules

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

```
% td connector:show daily_apptopia_import
Name     : daily_apptopia_import
Cron     : 10 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_apptopia_import
+--------+---------+---------+--------------+-----------------+----------+---------------------------+----------
+
| JobID  | Status  | Records | Database     | Table           | Priority | Started                   | Duration
|
+--------+---------+---------+--------------+-----------------+----------+---------------------------+----------
+
| 578066 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-18 00:10:05 +0000 | 160
|
| 577968 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-17 00:10:07 +0000 | 161
|
| 577914 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-16 00:10:03 +0000 | 152
|
| 577872 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-15 00:10:04 +0000 | 163
|
| 577810 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-14 00:10:04 +0000 | 164
|
| 577766 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-13 00:10:04 +0000 | 155
|
| 577710 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-12 00:10:05 +0000 | 156
|
| 577610 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-11 00:10:04 +0000 | 157
|
+--------+---------+---------+--------------+-----------------+----------+---------------------------+----------
+
8 rows in set
```

Delete the Schedule

td connector:delete removes the schedule.

```
$ td connector:delete daily_apptopia_import
```

Appendix

Modes for Out Plugin

You can specify file import mode in out section of load.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
```

Available Targets

Target	Description
app_metadata	Application metadata
app_performance	Application performance
app_ranks	Application ranking
app_sdks_metadata	Application SDKs metadata
category	List of categories
category_rank_lists	Raw ranks top charts for each category
category_featured_lists	Featured applications for each category
category_new_releases	New app releases for each category
country	Supported countries
publisher_metadata	Publisher metadata
publisher_performance	Publisher performance
sdk_metadata	SDK metadata

Available Markets

Store	Description
itunes_connect	Apple Store
google_play	Google Play Market

Rate Limits

There is requests per minute rate limit in Apptopia. This rate limit is auto refreshed after a certain number of seconds.

If you have multiple transfers under the same Apptopia account, you can control the rate limit usage of each transfer via requests_per_minute_limit in advanced settings as long as the total limit does not exceed your account limit. For example, assume that your account has quotas as 1000 calls/minute, if you create 2 transfers, e.g. app_performance & publisher_performance targets, you could use 600 rpm for app_performance transfer and the rest (400 rpm) for publisher_performance transfer.