Iterable Import Integration

You can connect Iterable to import Campaign metrics or a user's email into Treasure Data.

Prerequisites

- Basic knowledge of Treasure Data
- Iterable API Key

Limitation

- Due to getting the user list rate limit (5 requests/minutes), we do not support all lists import in a single job. User must specify a single list id to import in each job.

About Incremental Data Loading

Incremental loading is the activity of loading only new or updated records from a source into Treasure Data. Incremental loads are useful because they run efficiently when compared to full loads, and particularly for large data sets.

Incremental loading is available for many of the Treasure Data integrations. In some cases, it is a simple checkbox choice and in others, after you select incremental loading you are provided with other fields that must be specified.

Limitations, Supported, Suggestions

- For some integrations, if you choose incremental loading, you might need to make sure that there is an index on the columns to avoid a full table scan.
- Only Timestamp, Datetime, and numerical columns are supported as incremental_columns.
- For the raw query, the incremental_columns is required because it won't be able to detect the Primary keys for a complex query.

About Incremental Loading for Integrations

Treasure Data Incremental loading has 4 patterns (3 types of data connector + 1 workflow `td_load` operator.), then the 3 data connector loading examples are as follows:

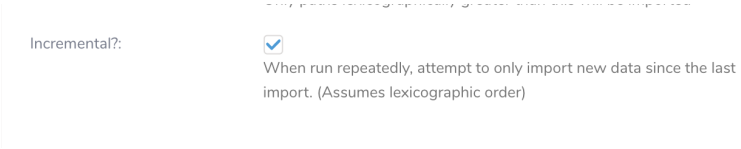
- Cloud storage service (e.g. AWS S3, GCS and etc.)
    - Lexicographic order of file name
- Query (e.g. MySQL, BigQuery and etc.)
    - Date time
- Variable period (Google Analytics, etc)
    - Use `start_date` for loading

Incremental Loading for Connectors

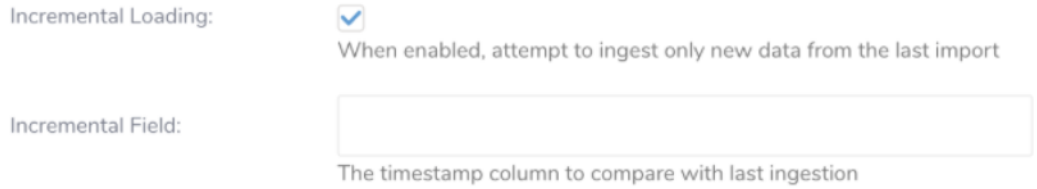
If incremental loading is selected, data for the connector is loaded incrementally.

This mode is useful when you want to fetch just the object targets that have changed since the previously scheduled run.

For example, in the UI:



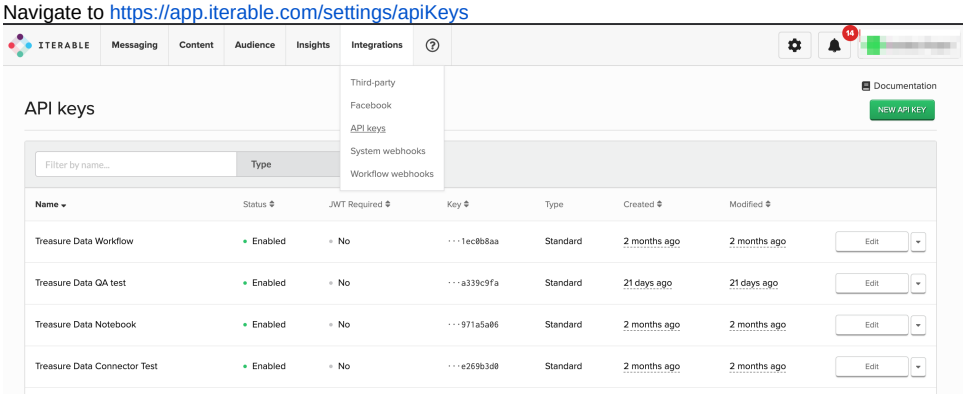
Database integrations, such as MySQL, BigQuery, and SQL server, require column or field names to load incremental data. For example:



Learn more About Database-based Integrations.

Obtaining API Key

1. Navigate to https://app.iterable.com/settings/apiKeys



2. Click **New API KEY**
3. Select **Standard (Server-side)**

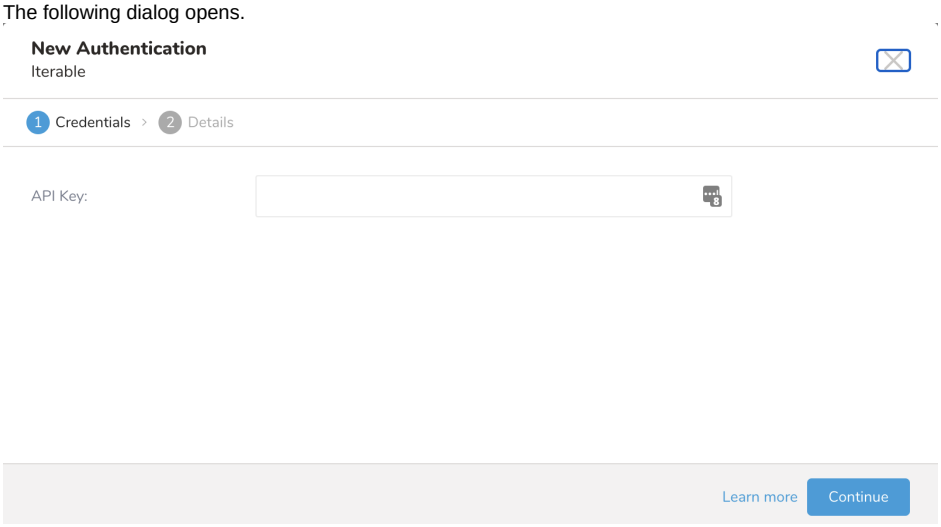
Use the TD Console to Create Your Connection

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. Search for and select Iterable.

4. The following dialog opens.

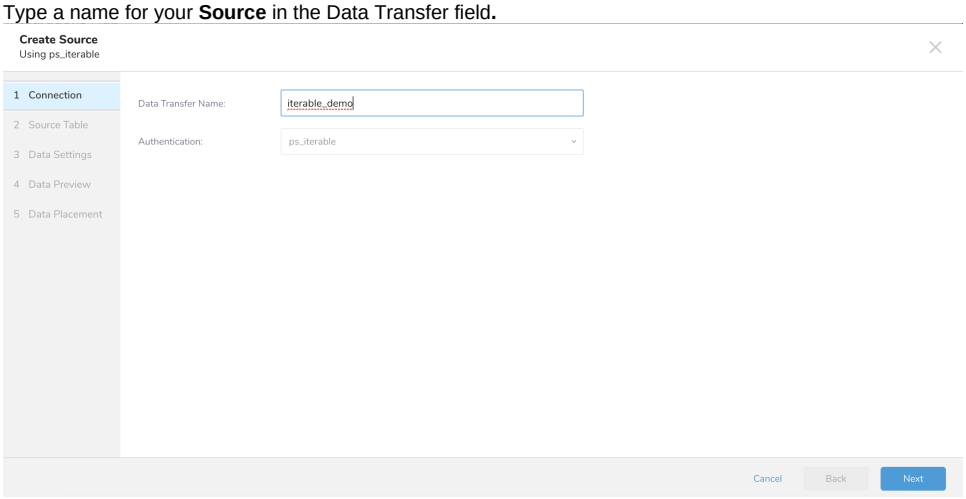


5. Enter your API Key
6. Enter a name for your connection.
7. Select **Continue.**

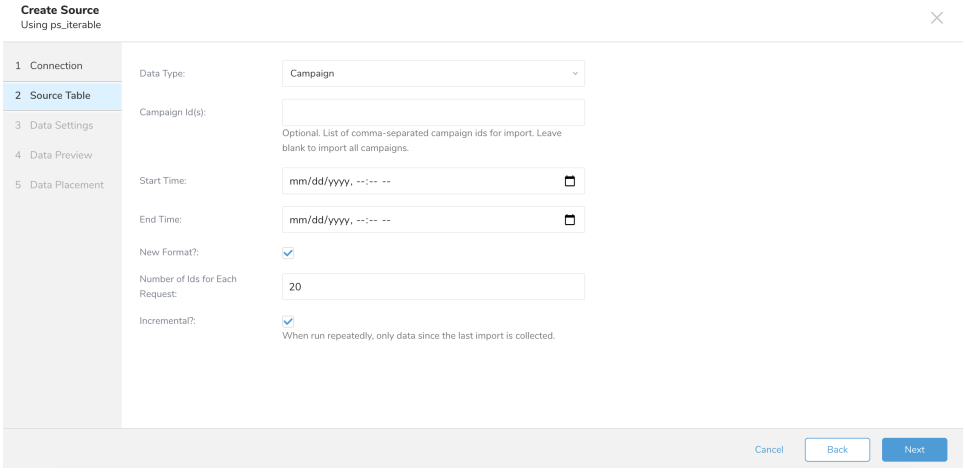
Transfer Your Data to Treasure Data

After creating the authenticated connection, you are automatically taken to Authentications. Search for the connection you created.

1. Select **New Source**.
2. Type a name for your **Source** in the Data Transfer field**.**



3. Select **Next**.
The Source Table dialog opens.



4. Edit the following parameters:

Parameters	Description	
Data Type	Data type to import: ▪ Campaign ▪ List	
Canpaign id (s)	An array of campaign's id, separated by commas. Leave it blank to import all campaigns	
List id	List id to fetch all users belong to it	
Start Time	For UI configuration, you can pick the date and time from the supported browser, or input the date that suits the browser expectation of date-time. For example, on Chrome, you will have a calendar to select Year, Month, Day, Hour, and Minute; on Safari, you need to input the text such as `2020-10-25T00:00`. For CLI configuration, we need a timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds, for example: `"2014-10-02T15:01:23Z"`.	
End Time	For UI configuration, you can pick the date and time from the supported browser, or input the date that suits the browser's expectation of date-time. For example, on Chrome, you will have a calendar to select Year, Month, Day, Hour, and Minute; on Safari, you need to input the text such as `2020-10-25T00:00`. For CLI configuration, we need a timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds, for example: `"2014-10-02T15:01:23Z"`.	
New format	Receive data in New Format if it is true and Old format if is false	
Number of Ids for Each Request	The number of ids for one request. From 1 to 20	
Incremental	Import new data only from the last run. See About Incremental Loading.	

Data Settings

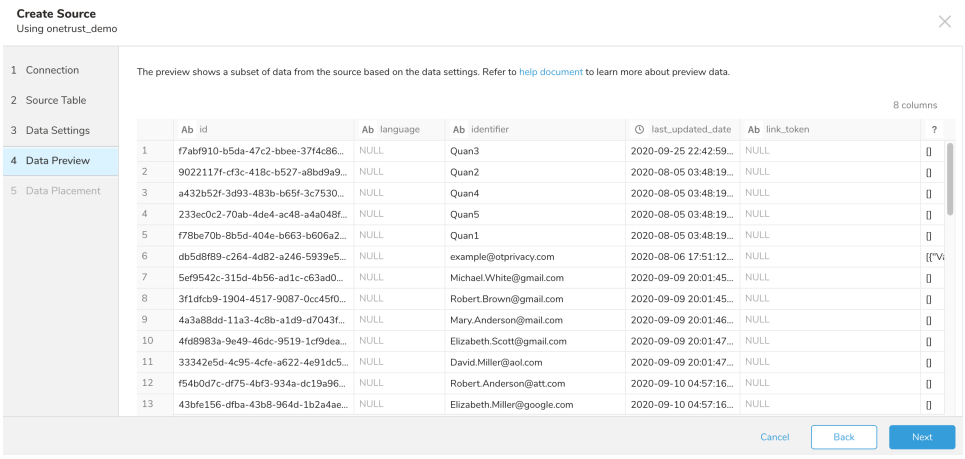
1. Select **Next**.
   The Data Settings page opens.
2. Skip this page of the dialog.

Data Preview

You can see a preview of your data before running the import by selecting Generate Preview.

Data shown in the data preview is approximated from your source. It is not the actual data that is imported.

1. Select **Next**.
   Data preview is optional and you can safely skip to the next page of the dialog if you want.
2. To preview your data, select **Generate Preview**. Optionally, select **Next**.
3. Verify that the data looks approximately like you expect it to.

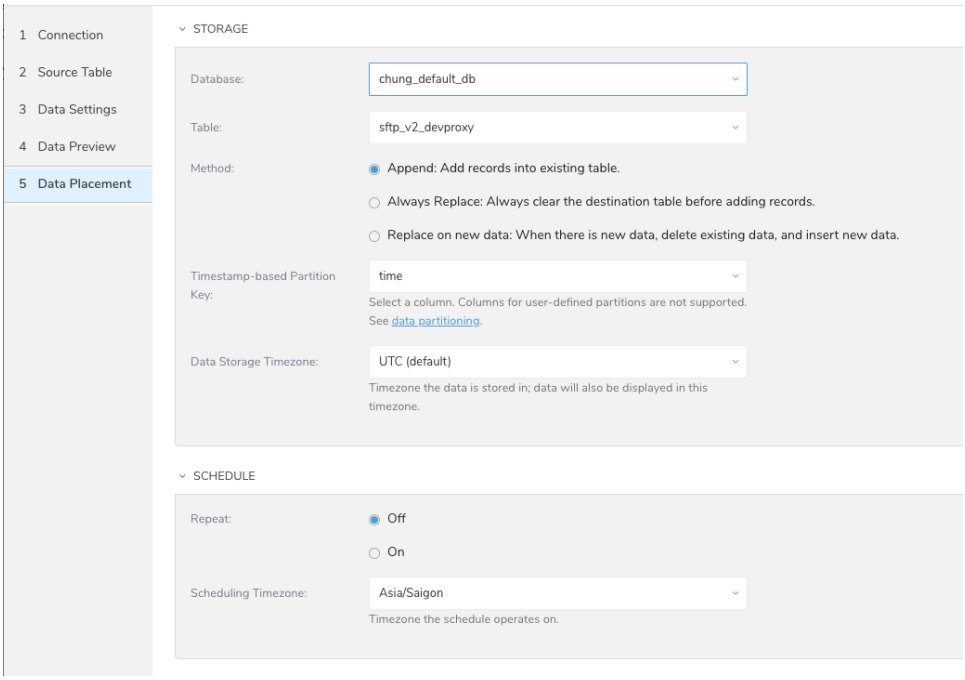


4. Select **Next**.

Data Placement

For data placement, select the target database and table where you want your data placed and indicate how often the import should run.

1. Select **Next.** Under Storage you will create a new or select an existing database and create a new or select an existing table for where you want to place the imported data.



2. Select a **Database** > **Select an existing** or **Create New Database**.
3. Optionally, type a database name.
4. Select a **Table**> **Select an existing** or **Create New Table**.
5. Optionally, type a table name.
6. Choose the method for importing the data.

    - **Append** (default)-Data import results are appended to the table.
      If the table does not exist, it will be created.
    - **Always Replace**-Replaces the entire content of an existing table with the result output of the query. If the table does not exist, a new table is created.
    - **Replace on New Data**-Only replace the entire content of an existing table with the result output when there is new data.
7. Select the **Timestamp-based Partition Key** column.
   If you want to set a different partition key seed than the default key, you can specify the long or timestamp column as the partitioning time. As a default time column, it uses upload_time with the add_time filter.
8. Select the **Timezone** for your data storage.
9. Under **Schedule**, you can choose when and how often you want to run this query.

    - Run once:

        a. Select **Off**.
        b. Select **Scheduling Timezone**.
        c. Select **Create & Run Now**.
    - Repeat the query:

        a. Select **On**.
        b. Select the **Schedule**. The UI provides these four options: *@hourly*, *@daily* and *@monthly* or custom *cron*.
        c. You can also select **Delay Transfer** and add a delay of execution time.
        d. Select **Scheduling Timezone**.
        e. Select **Create & Run Now**.

After your transfer has run, you can see the results of your transfer in **Data Workbench** > **Databases**.