

Salesforce Export Integration

You can connect Salesforce with Treasure Data for greater control over your Salesforce data and better integration with the other business applications in your marketing and sales operations stack. Integrating Salesforce with Treasure Data makes it easy to:

- **Add new features to Salesforce.** For example, you can prevent churn by tracking web usage and receiving alerts when customers' product usage declines.
- **Use Salesforce data to improve other parts of your marketing stack.** For example, you can increase your Facebook Ads ROI by automatically removing new customers from your Facebook Custom Audiences.

For sample workflows on importing data from Salesforce, go to [Treasure Boxes](#).

If you don't have a Treasure Data account yet, [contact us](#) so that we can get you set up.

A front-end application streams data to be collected in Treasure Data via Log/data collector daemon or Mobile SDKs. You can also bulk import your data using Bulk Import from the CLI. A scheduled query is setup in Treasure Data to run periodically on the data and write the result of each query execution into your Salesforce.com Object. The following is a fairly common architecture, for example:



Ranking: What are the “Top N of X?”

Every social/mobile application calculates the “top N of X” (ex: top 5 movies watched today). Treasure Data already handles the raw data warehousing; the “[write-to-Salesforce.com](#)” feature enables Treasure Data to find the “top N” data as well.

Dashboard Application

If you're a data scientist, you need to keep track of a range of metrics every hour/day/month and make them accessible via visualizations. Using this “[write-to-Salesforce.com](#)” feature, you can streamline the process and focus on building visualizations of your query results via Reports and Dashboards on the Salesforce.com organization.

Continue to the following topics:

- [Ranking: What are the “Top N of X?”](#)
- [Dashboard Application](#)

Prerequisites

Support

Use the TD Console to Create Your Connection

- [Create a New Connection](#)
- [Allow TD to Access Salesforce using Salesforce](#)
- [Create New Connection in TD Console](#)

Validate Your Salesforce Connection

- [Integration Parameters for Salesforce](#)
- [Example Query](#)

Optionally Schedule the Query Export Jobs

- 1. [Navigate to Data Workbench > Queries.](#)
- 2. [Create a new query or select an existing query.](#)
- 3. [Next to Schedule, select None.](#)
- 4. [In the drop-down, select one of the following schedule options.](#)
 - [Custom cron... Details](#)
 - 5. (Optional) [If you enabled the Delay execution, you can delay the start time of a query.](#)
- [Execute the Query](#)

Optionally Configure Export Results in Workflow

- [Example Workflow for Salesforce Export](#)

Prerequisites

- Basic knowledge of Treasure Data, including the [TD Toolbelt](#).
- Salesforce.com organization and username, password, and security token for API integration.
- You have "API Enabled" permission.
- Target Salesforce.com Object should exist with read and write permissions.

Support

SFDC output supports these authentication types:

- Credential
- Session

Use the TD Console to Create Your Connection

Create a New Connection

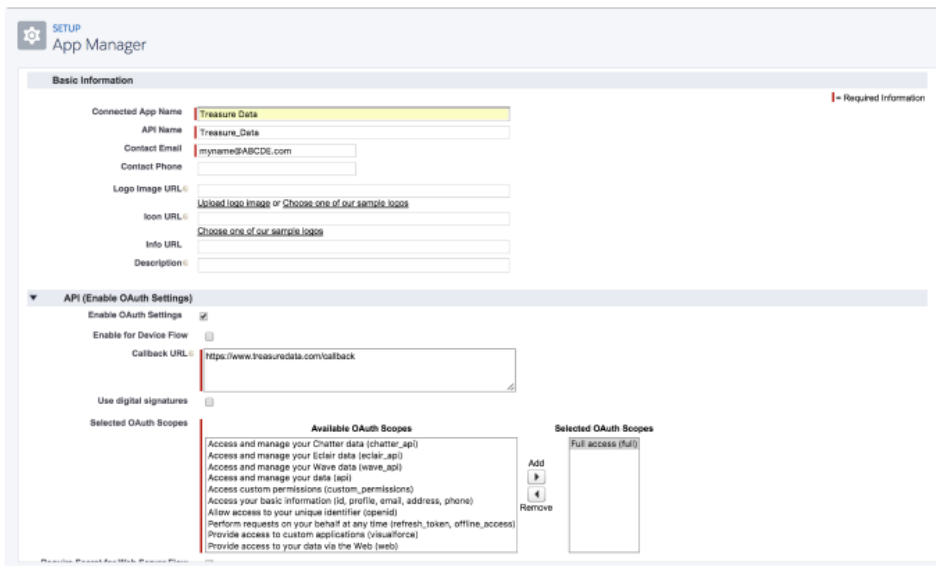
Connecting to Salesforce using the TD Console is quick and easy. Alternatively, create a Salesforce connection using the command line. The import integration supports credentials; you need a client ID and client secret to authenticate using credentials.

Allow TD to Access Salesforce using Salesforce

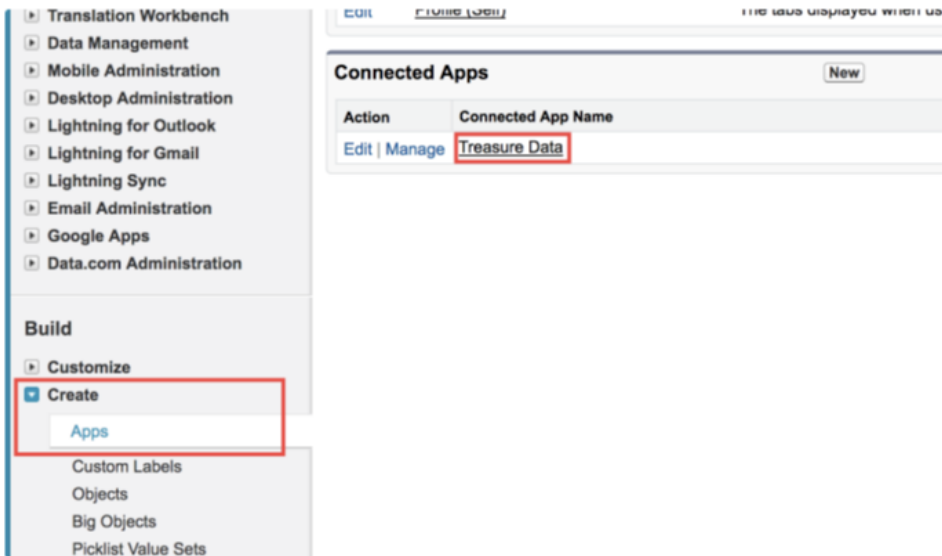
These instructions guide you to locate the client ID and client secret that is necessary to authenticate using credentials.

1. Go to Setup > Apps > App Manage.
2. Select New Connected App.

The steps may vary depending on the version of Salesforce you are running. In Spring 19 classic UI: **Setup > Build > Create > Apps > Connect** **d Apps > New**.



3. Navigate to **Setup > Build > Create (Apps)**, and validate all your connected apps:



4. Select your app name to go to the page where you can view and manage all information about your connected app.
5. Write down or copy your **Consumer Key** (client_id) and **Consumer Secret** (client_secret).

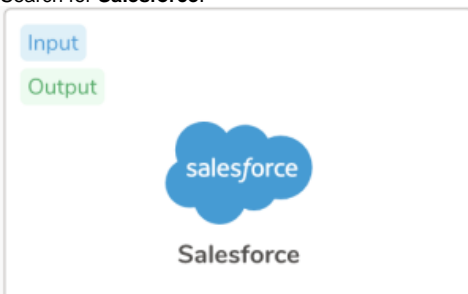


6. For secure account access, get a Salesforce Security Token. If you don't have a security token, go to **Account > Settings > Reset My Security Token** and select **Reset Security Token**. You'll receive your Security Token by email.

In Spring 19 classic UI: **My account > My Settings > Personal > Reset My Security Token**.

Create New Connection in TD Console

1. Open **TD Console**.
2. Navigate to **Integrations Hub > Catalog**.
3. Search for **Salesforce**.



- 4.

1. To authenticate with your credentials, enter your username (your email) and password, as well as your Client ID, Client Secret, and Security Token.
2. In the dialog box, enter login.salesforce.com as the login URL. Remove unnecessary letters from Login URL parameter.

New Authentication

Salesforce

1 Credentials > 2 Details

Login URL:

Authentication method: Authentication by "Session ID" currently works only for export

Username: Username for Force.com

Password: Password for Force.com

Client ID: Client ID for your application. Required for import from Salesforce

Client secret: Client secret for your application. Required for import from Salesforce

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

New Authentication

Salesforce Legacy

1 Credentials > 2 Details

Login URL:

Authentication method: Authentication by "Session ID" currently works only for export

OAuth connection: [Click here](#) to connect a new account

Initial retry delay: The first time we fail to connect, wait this many seconds before retrying (with exponential backoff)

Retry limit:

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

5. Select **Continue**.

6. Give your connection a descriptive name and select **Create Connection**.

Validate Your Salesforce Connection

To validate the user permission, make sure to use Salesforce to validate:

- Authority: check the Salesforce import integration connection steps.
- Allowed access to Salesforce from Treasure Data: sometimes configuring this requires knowing and configuring your TD static IP address. Please contact support if you require the static IP address information.

Without properly configured authority and access, you might encounter access restriction errors. For example:

```
Response not 2xx:  
400 Bad Request { "error": "invalid_grant", "error_description": "authentication failure" }
```

Define your Query

Select a database and table from the popup menus. Enter your query and select Output results.

To avoid any issues with result export, define column aliases in your query such that resulting column names from the query match the Salesforce field names for default fields and API names (usually ending with __c) for custom fields.

1. Complete the instructions in [Creating a Destination Integration](#).
2. Navigate to **Data Workbench > Queries**.
3. Select a query for which you would like to export data.
4. Run the query to validate the result set.
5. Select **Export Results**.
6. Select an existing integration authentication.

Choose Integration ✕

Use Existing Integration

Search...

00_2977_box_connection_1 box

00_297_box_connection_2 box

00_mailpublisher_shirai mail_publisher_smart

7. Define any additional Export Results details. In your export integration content review the integration parameters. For example, your Export Results screen might be different, or you might not have additional details to fill out:

Export Results ✕

Lookup Field:

Name of field for dedup (default to email)

Retry Limit:

Retry Initial wait in
Milliseconds:

Retry Max wait in
milliseconds:

Max http waiting time in
milliseconds:

Max upload chunk size (in
bytes):

Batch max wait in

8. Select **Done**.

9. Run your query.
10. Validate that your data moved to the destination you specified.

Integration Parameters for Salesforce

Export Results
✕

Integration: **meg_salesforce_v2**

Object:

Mode:

Concurrency mode:
This should usually be parallel.

Retry limit:

Batch size:

Back
Done

Parameter	Description
Object	Salesforce.com Object
Mode	<ul style="list-style-type: none"> The append mode is the default that is used when no mode option is provided in the URL. In this mode, the query results are appended to the object. With the truncate mode, the system first truncates the existing records in the Salesforce.com Object and moves them into the Trashbin, then inserts the query results. With the update mode, a row is inserted unless it would cause a duplicate value in the external key columns specified in the "unique" parameter. In such a case, an update is performed instead. The "unique" parameter is required with this mode and must be defined as an external key when used with the update mode. <p>With the update mode, a row is inserted unless it would cause a duplicate value in the external key columns specified in the "unique" parameter. In such a case, an update is performed instead. The "unique" parameter is required with this mode and must be defined as an external key when used with the update mode.</p>
Concurrency mode	<p>The concurrency_mode option controls how the data is uploaded to the Salesforce.com organization. The default mode is parallel.</p> <p>With the parallel method, data is uploaded in parallel. This is the most reliable and effective method for most situations.</p> <p>if you see "UNABLE_TO_LOCK_ROW" in an error message, try concurrency_mode=serial instead.</p>
Retry Limit	<p>This option sets the number of attempts made to write the result to the configured Salesforce.com destination, if errors occur. If the export fails more than the set number of retries, the query fails.</p> <p>The default number of retries is <code>retry=2</code>.</p>
Batch size	Splits the records in the result of a query in chunks of 10000 records by default and bulk uploads one chunk at a time.

Example Query

Database: Type: [more settings »](#)

Result Export: Scheduling:

[Presto UDFs](#) [Presto Performance Tuning](#) [Presto Query Templates](#) [Presto Known Limitations](#)

```
1 SELECT code as Code__c, COUNT(1) as Count__c FROM www_access GROUP BY code
```

```
SELECT identifier_type, identifier  
FROM table my_table
```

Optionally Schedule the Query Export Jobs

You can use Scheduled Jobs with Result Export to periodically write the output result to a target destination that you specify.

1. Navigate to **Data Workbench > Queries**.
2. Create a new query or select an existing query.
3. Next to **Schedule**, select None.

Schedule: **None**

4. In the drop-down, select one of the following schedule options.

Schedule Query [X]

Schedule

- None
- Custom cron...
- @daily (midnight)
- @hourly (:00)
- None

Drop-down Value	Description
Custom cron...	Review Custom cron... details .
@daily (midnight)	Run once a day at midnight (00:00 am) in the specified time zone.
@hourly (:00)	Run every hour at 00 minutes.
None	No schedule.

Custom cron... Details

Schedule Query
✕

Schedule
Cron ?

The TD_SCHEDULED_TIME UDF returns the time of the job's scheduled run formatted as a Unix timestamp integer.

Timezone

America/Los_Angeles

Delay execution

A delay ensures all buffered events are imported before running the query. Doesn't affect TD_SCHEDULED_TIME().

Cancel
Schedule

Cron Value	Description
0 * * * *	Run once an hour
0 0 * * *	Run once a day at midnight
0 0 1 * *	Run once a month at midnight on the morning of the first day of the month
""	Create a job that has no scheduled run time.

```

* * * * *
- - - - -
| | | | |
| | | | +----- day of week (0 - 6) (Sunday=0)
| | | +----- month (1 - 12)
| | +----- day of month (1 - 31)
| +----- hour (0 - 23)
+----- min (0 - 59)

```

The following named entries can be used:

- Day of Week: sun, mon, tue, wed, thu, fri, sat
- Month: jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec

A single space is required between each field. The values for each field can be composed of:

Field Value	Example	Example Description
a single value, within the limits displayed above for each field.		

a wildcard '*' to indicate no restriction based on the field.	'0 0 1 * *'	configures the schedule to run at midnight (00:00) on the first day of each month.
a range '2-5', indicating the range of accepted values for the field.	'0 0 1- 10 * *'	configures the schedule to run at midnight (00:00) on the first 10 days of each month.
a list of comma-separated values '2,3,4,5', indicating the list of accepted values for the field.	0 0 1,11,21 * *'	configures the schedule to run at midnight (00:00) every 1st, 11th, and 21st day of each month.
a periodicity indicator '* /5' to express how often based on the field's valid range of values a schedule is allowed to run.	'30 */2 1 * *'	configures the schedule to run on the 1st of every month, every 2 hours starting at 00:30. '0 0 */5 * *' configures the schedule to run at midnight (00:00) every 5 days starting on the 5th of each month.
a comma-separated list of any of the above except the '*' wildcard is also supported '2, */5,8-10'.	'0 0 5, * /10,25 * *'	configures the schedule to run at midnight (00:00) every 5th, 10th, 20th, and 25th day of each month.

5. (Optional) If you enabled the Delay execution, you can delay the start time of a query.

Execute the Query

Save the query with a name and run, or just run the query. Upon successful completion of the query, the query result is automatically imported to the specified container destination.



Scheduled jobs that continuously fail due to configuration errors may be disabled on the system side after several notifications.

Optionally Configure Export Results in Workflow

Within Treasure Workflow, you can specify the use of this data connector to export data.

Learn more at [Using Workflows to Export Data with the TD Toolbelt](#).

Example Workflow for Salesforce Export

```
timezone: UTC

schedule:
  daily>: 02:00:00

sla:
  time: 08:00
  +notice:
    mail>: {data: Treasure Workflow Notification}
    subject: This workflow is taking long time to finish
    to: [meg@example.com]

_export:
  td:
    dest_db: dest_db
    dest_table: dest_table

+prepare_table:
  td_ddl>:
    create_databases: ["${td.dest_db}"]
    create_tables: ["${td.dest_table}"]
    database: ${td.dest_db}

+load:
  td_load>: config/daily_load.yml
  database: ${td.dest_db}
  table: ${td.dest_table}
```