

Create Treasure Data Package – instruction

Contents

1.	How to prepare CLI to use with Treasure Data	2
2.	Package structure	3
3.	Tables.....	4
a.	Types:	5
b.	How to add to script file	5
4.	Connectors	7
a.	How to create YML file	7
b.	How to add to script file	8
5.	Workflows	10
a.	How to create YML file	10
b.	How to add to script file	10

1. How to prepare CLI to use with Treasure Data

- Download and install TD toolbelt
 - <https://toolbelt.treasuredata.com/>
- Run CMD/PowerShell and check that the toolbelt has been installed.
 - **td version**

```
C:\Users\mariusz.drobot>td --version
0.16.3
```

- Set endpoint
 - **td -e https://api.eu01.treasuredata.com account -f**
 - type email and TD password
- Set apikey
 - **td apikey:set xxx -f**
 - xxxxxxxx is an apikey which you can find in console in the MyProfile site
- Check if apikey works
 - **td db:list**
- Install digdag plugin to management workflows via CLI td
 - **td wf**
 - Confirm installation

2. Package structure

- cleanup.sh -> rollback script
- create_connectors.sh -> create connectors script
- create_databases.sh -> create databases script
- create_tables.sh -> create tables script
- create_workflows.sh -> create workflows script
- delete_connectors.sh -> delete connector which should be recreate in the package
- delete_tables.sh -> delete tables which should be recreate in the package
- install.sh -> MAIN shell script
- run_scripts.sh -> run required query before/after deployment

- 01_tables -> folder with DDL folders
 - ddl_db_l1_bi -> folder with DDL files
 - bounces.sql -> DDL file
 - clicks.sql -> DDL file
 - emailsendlog.sql -> DDL file
 - journey.sql -> DDL file
 - journeyactivity.sql -> DDL file
 - opens.sql -> DDL file

- 02_connectors -> folder with YML files
 - con_db_l0_sfmc_sfmc_bi_emailsendlog.yml -> YML configuration of connector
 - con_db_l0_sfmc_sfmc_bi_journey.yml -> YML configuration of connector
 - con_db_l0_sfmc_sfmc_bi_journeyactivity.yml -> YML configuration of connector
 - con_db_l0_sfmc_sfmc_bi_sent.yml -> YML configuration of connector
 - con_db_l0_sfmc_sfmc_bi_sms.yml -> YML configuration of connector

- 03_workflows -> folder with projects
 - proj_sfmc_src_bi -> folder with .sql and .dig files
 - qr_delta_success_l2_monitoring.sql -> .sql used in workflow
 - qr_delta_success_monitoring.sql -> .sql used in workflow
 - qr_email_L1_bounces.sql -> .sql used in workflow
 - qr_email_L1_clicks.sql -> .sql used in workflow
 - wf_l0_stg_email_bi.dig -> .dig contains all details of workflow
 - wf_l0_stg_ga_email_history.dig -> .dig contains all details of workflow

- 04_scripts -> folder with additional query
 - delete_statements_qa_email_sms.sql -> sql to query after/before deploying
 - insert_db_stg_bi_technical.delta_load_log_dce.sql -> sql to query after/before deploying

3. Databases

a. How to add to script file

- In the create_databases.sh file you should add:
 - **td database:create <database_name>**
- In the cleanup.sh file you should add:
 - **td database:delete <database_name>**

4. Tables

a. Types:

In the package table is create by:

- Running DDL files with schema:
 - We can get schema using **td table:show <db> <table>** and based on it create DDL file.
- Using **td table:create <db> <table>** without schema
 - Required for LO SFMC tables

b. How to add to script file

- In the create_tables.sh file you should add:
 - If you have DDL file:
 - **td query -d <db_name> -t presto -w -q ./01_tables/ddl_<db_name>/<table_name>.sql**
 - If you want create table without schema:
 - **td table:create <db_name> <table_name>**
- In the cleanup.sh file you should add:
 - **td table:delete <db_name> <table_name> -f**
- If you want recreate table, you have to delete current table, to do this, add statement in the delate_tables.sh file:
 - **td table:delete <db_name> <table_name> -f**

DDL should be inside 01_tables/ddl_<db_name> folder

c. Example DDL:

```
CREATE TABLE "td-presto".db_l2_bi.sendouts(  
  record_id bigint,  
  createddate bigint,  
  createdby varchar,  
  lastmodifieddate bigint,  
  lastmodifiedby varchar,  
  sendout_id varchar,  
  objectid varchar,  
  folderid varchar,  
  messagename varchar,  
  description varchar,  
  src_sfmc_createddate bigint,  
  src_sfmc_modifieddate bigint,  
  ja_emailname varchar,  
  sj_emailname varchar,  
  email_sendlog_dev_errorcode bigint,  
  journey_id varchar,  
  journey_name varchar,  
  batchid bigint,  
  eventdate bigint,  
  jobid bigint,  
  subscriberid varchar,  
  subscriberkey varchar,  
  customer_key varchar,  
  sms_campaignname varchar,  
  type varchar  
)
```

5. Connectors

a. How to create YML file

Creating YML configuration based on connectors in DEV environment:

td connector:show <connector_name>

if you type

td connector:show <connector_name> >> <connector_name> .txt

configuration will be saved to txt file.

Credentials should be replacing according to the table below

Source	Credentials in YML file from DEV	Type	Target
GA	json_keyfile: content: "****" project_id: ga-data360	Json_key	json_keyfile: content: { "type": "\${aws_secrets-ga-type}", "project_id": "\${aws_secrets-ga-project_id}", "private_key_id": "\${aws_secrets-ga-private_key_id}", "private_key": "\${aws_secrets-ga-private_key}", "client_email": "\${aws_secrets-ga-client_email}", "client_id": "\${aws_secrets-ga-client_id}", "auth_uri": "\${aws_secrets-ga-auth_uri}", "token_uri": "\${aws_secrets-ga-token_uri}", "auth_provider_x509_cert_url": "\${aws_secrets-ga-auth_provider}", "client_x509_cert_url": "\${aws_secrets-ga-client_cert_url}" } project_id: ga-data360
GCM	json_keyfile: content: "****" project_id: doubleclick-bigquery-dts	Json_key	json_keyfile: content: { "type": "\${aws_secrets-gcm-type}", "project_id": "\${aws_secrets-gcm-project_id}", "private_key_id": "\${aws_secrets-gcm-private_key_id}", "private_key": "\${aws_secrets-gcm-private_key}", "client_email": "\${aws_secrets-gcm-client_email}", "client_id": "\${aws_secrets-gcm-client_id}", "auth_uri": "\${aws_secrets-gcm-auth_uri}", "token_uri": "\${aws_secrets-gcm-token_uri}", "auth_provider_x509_cert_url": "\${aws_secrets-gcm-auth_provider}", "client_x509_cert_url": "\${aws_secrets-gcm-client_cert_url}" } project_id: doubleclick-bigquery-dts

SFMC	client_id: 0iwy9hwtg75bbqo72fmp1rv5 client_secret: "***" target: data_extension page_size: 50 data_extension_names: - <data_extension_name>	secret_id, secret_key, data extension name	client_id: \${aws_secrets-sfmc_l1-client_id} client_secret: \${aws_secrets-sfmc_l1-client_secret} target: data_extension page_size: 50 data_extension_names: - <data_extension_name>
------	--	--	---

YML configuration must start from **in:** statement and end on **out:** statement

In statement contains all details about connectors

Out statement contains all TD server details:

Example:

```

out:
type: td
apikey: ${jenkins-td-eu-api_key}
endpoint: ${jenkins-td-eu-api_url}
default_timezone: UTC
mode: replace

```

b. How to add to script file

- In the create_connectors.sh file you should add:
 - **td connectors:create <con_name> "" <db_name> <table_name> ./02_connectors/<config_file>.yml**
- In the cleanup.sh file you should add:
 - **td connector:delete <con_name>**
- If you want recreate connector, you must delete current connector, to do this, add statement in the delate_connectors.sh file:
 - **td connector:delete <con_name>**

c. Example:

```
---
in:
  client_id: ${aws_secrets-sfmc_l1-client_id}
  client_secret: ${aws_secrets-sfmc_l1-client_secret}
  target: data_extension
  page_size: 50
  data_extension_names:
    - DataView_Journey
  shared_data_extension: false
  incremental: false
  maximum_retries: 7
  initial_retry_interval_millis: 1000
  maximum_retry_interval_millis: 120000
  timeout_millis: 300000
  sleep_between_requests_millis: 5000
  type: salesforce_marketing_cloud
filters:
- type: add_time
  to_column:
    name: time
    type: timestamp
  from_value:
    mode: upload_time
- type: rename
  rules:
    - rule: upper_to_lower
    - rule: character_types
      pass_types: [ "a-z", "0-9" ]
      pass_characters: " "
      replace: " _"
exec: {}
out:
  type: td
  apikey: ${jenkins-td-eu-api_key}
  endpoint: ${jenkins-td-eu-api_url}
  default_timezone: UTC
  mode: replace
```

6. Workflows

a. How to get files from DEV

- Download project from DEV:
 - `td wf download <project_name>`
- Copy required .dig and .sql files from project folder to `03_workflows/<project_name>/` in the package

If in .SQL or .DIG file you use some credentials, it should be replacing according to the table in the connector chapter.

If in workflow you trigger connectors, you must change name of them according name in `/02_connectors/` folder.

b. How to add to script file

- In the `create_workflows.sh` file you should add:
 - `digdag push <project_name> -e ${jenkins-td-eu-endpoint_url} -X client.http.headers.authorization='TD1 ${jenkins-td-eu-api_key}' -r `date -u +%Y-%m-%dT%H:%M:%SZ` --project ./03_workflows/<project_name>`
- In the `cleanup.sh` file you should add:
 - `digdag delete <project_name> -e ${jenkins-td-eu-endpoint_url} -X client.http.headers.authorization='TD1 ${jenkins-td-eu-api_key}' -r `date -u +%Y-%m-%dT%H:%M:%SZ` --force`
- If you want recreate workflow, you don't have to delete current version because during deployment old project will be overwrite by new one