Firebase Cloud Messaging Export Integration

Firebase is a mobile- and web application development platform that helps you deliver richer app experiences. You can send direct notifications to your mobile app and website through Firebase Cloud Messaging.

The Realtime Database and Cloud Firestore can stock document-structured data and synchronize the corresponding apps in milliseconds whenever a data transformation occurs. The app and its database listen to each other, providing your customers with a more reactive app experience. Using Firebase **Cloud Functions you** can write backend code to respond to events happening in the Firebase platform without having to deal with any servers.

Prerequisites

- Basic knowledge of Treasure Data, including the TD Toolbelt.
- A Google Firebase Project.
- For substitution, NULL values will be truncated to "" (empty string). If the substitution is for a numeric column, there will be a warning and the record is skipped.
- Because the messages are sent individually, we will not revert the whole sessions. The error messages are logged to the TD Console, but the job will continue.
- Strong knowledge of the Firebase message template structure.

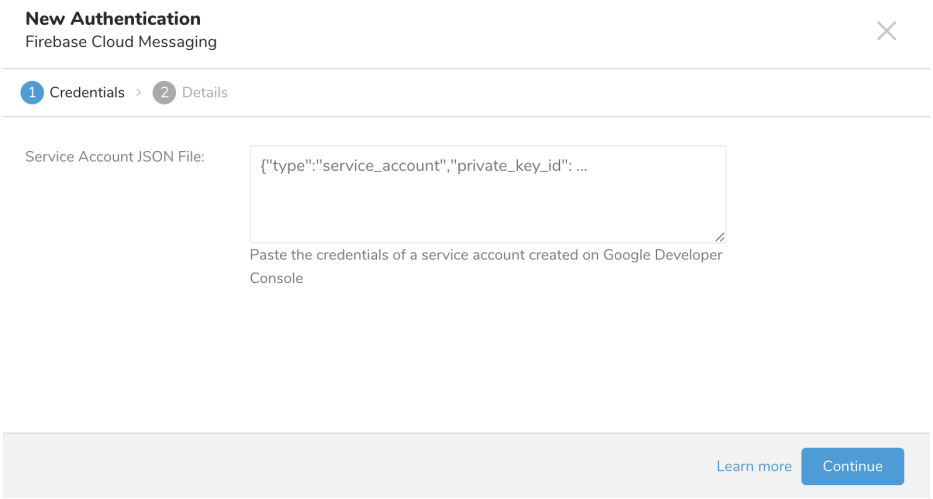
Use the TD Console to Create Your Connection

Create a New Connection

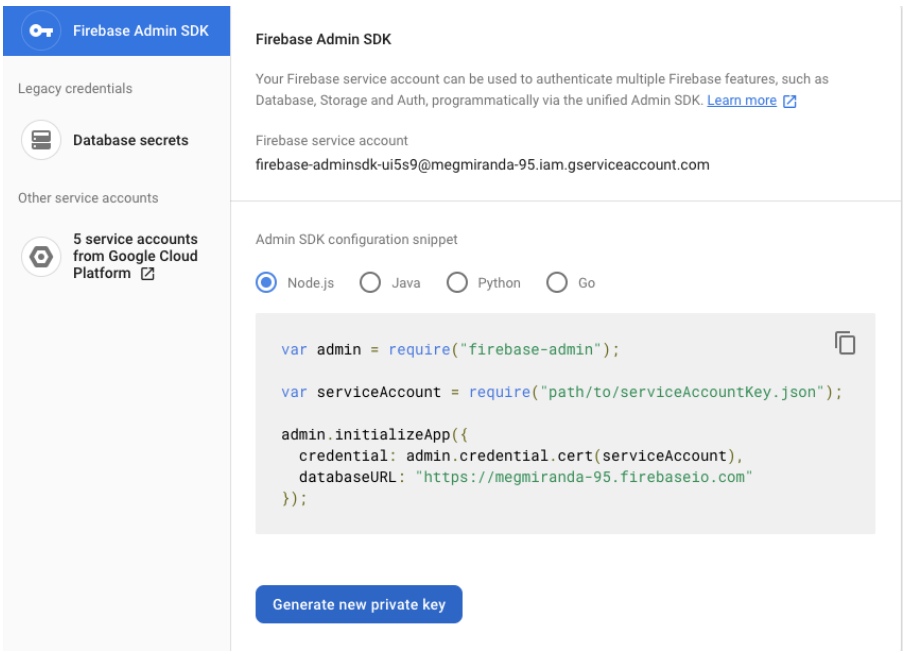
When you configure a data connection, you provide authentication to access the integration. In Treasure Data, you configure the authentication and then specify the source information.

1. Open TD Console.
2. Navigate to Integrations Hub > Catalog.
3. Search for and select Firebase.

The following dialog opens.



4. Type or paste the credential from your Google Developer Console service account.



For example:

```
{
  "type": "service_account",
  "project_id": "megmida-95",
  "private_key_id": "e793538838c9daa051b2b2f48eea31e10cf46f75",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADA.........br8RqYJePR6sU+68QfAF\n-----END
PRIVATE KEY-----\n",
  "client_email": "firebase-adminsdk-ui5s9@megmida-95.iam.gserviceaccount.com",
  "client_id": "114220230186840911629",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-
ui5s9@megmida-95.iam.gserviceaccount.com"
}
```

5. Select **Done.**

Example Service Account JSON File

A sample JSON file:

```
{
"type": "service_account",
"project_id": "[PROJECT-ID]",
"private_key_id": "[KEY-ID]",
"private_key": "-----BEGIN PRIVATE KEY-----\n[PRIVATE-KEY]\n-----END PRIVATE KEY-----\n",
"client_email": "[SERVICE-ACCOUNT-EMAIL]",
"client_id": "[CLIENT-ID]",
"auth_uri": "https://accounts.google.com/o/oauth2/auth",
"token_uri": "https://accounts.google.com/o/oauth2/token",
"auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
"client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/[SERVICE-ACCOUNT-EMAIL]"
}
```

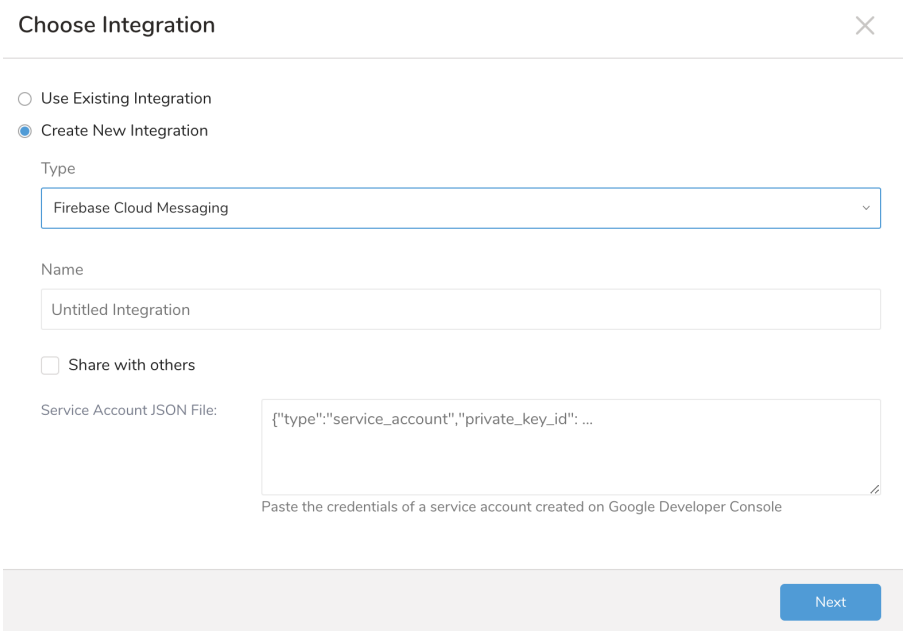
- See Firebase document to create a new service account key.

Configure Export Results in Your Data Connection

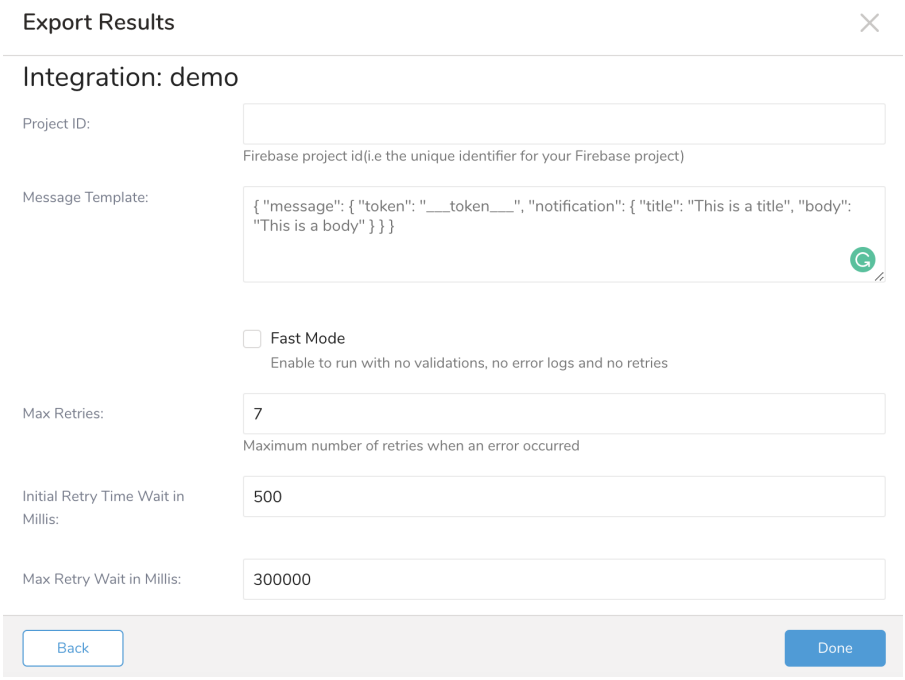
In this step, you create or reuse a query. In the query, you configure the data connection.

Configure the Connection by Specifying the Parameters

1. Open the TD Console.
2. Navigate to **Data Workbench** > **Queries**.
3. Select the query that you plan to use to export data.
4. Select **Export Results** located at top of your query editor. The Choose Integration dialog opens. You have two options when selecting a connection to use to export the results, using an existing connection or creating a new one.



5. Use an existing connection
    a. Type the connection name in the search box to filter.
    b. Select your connection
6. Specify the export details.

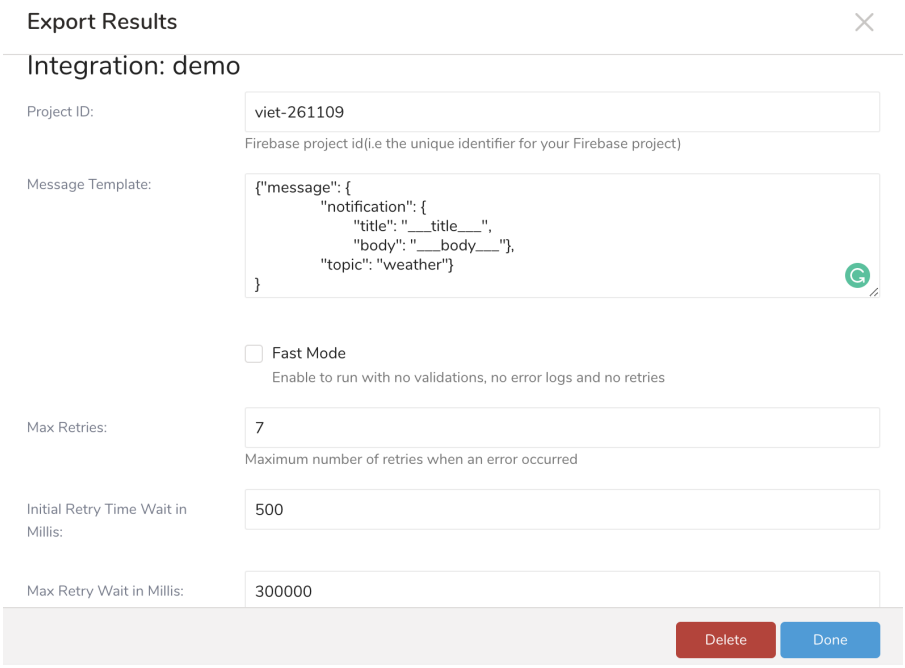


- **Project Id:** project ID of your Firebase project.
- **Message Template:** the connector replaces all substitution columns and aliases (___column_name___) in the template to the individual messages to send Firebase Message. All substitution columns and aliases exist in the export data.
- **Fast Mode:** Enable parallel run with no validations, no errors logs and no retries. Default: false
- **Max retries** (optional): Number of retries before the system gives up. Default: 7.
- **Initial retry time wait in milliseconds** (optional): The time, in milliseconds, between the first and second attempt. Default: 500, which is equivalent to 0.5 seconds.
- **Max retry wait in milliseconds** (optional): The time, in milliseconds, between the second and all subsequent attempts. Default: 300000, which is equivalent to 5 minutes.

The following is a sample template configuration:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
{
        "message": {
                "notification": {
                        "title": "___title___",
                        "body": "___body___"
                },
                "topic": "weather"
        }
}
```



Example of a Query to Populate Firebase

From Treasure Data, run the following query and export results to a connection for Firebase:

```
Select 'test_body' as body, 'test_title' as title
```

Optionally Use Scheduled Jobs for Export

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

Optionally Configure Export Results in Workflow

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

```
timezone: UTC

_export:
  td:
    database: sample_datasets

+td-result-into-firebase:
  td>: queries/sample.sql
  result_connection: your_connections_name
  result_settings:
    service_account_json: '{"type":"service_account","project_id":"project_id","private_key_id":"####"}'
    project_id: project_id
    message_template: {"message":{"notification":{"title":"___title___","body":"___body___"},"topic":"weather"}}
    fast_mode: true
    retry_count: 7
    retry_initial_wait_millis: 500
    max_retry_wait_millis: 300000
```

Optionally, read more information on using data connectors in workflow to export data.