IdentitySync

Overview

IdentitySync is Gigya's robust ETL solution (Extract, Transform, Load) that offers an easy way to **transfer data in bulk** between platforms.

With IdentitySync, you can:

- Take all the permission-based social and profile identity information stored at Gigya and channel it into another platform, such as an ESP, CRM, or marketing automation system.
- Get up-to-date data from a 3rd party platform, such as a user newsletter subscription status, survey responses or account balance, and sync existing Gigya user profiles or create new ones ad-hoc.
- Import from one Gigya site to another

IdentitySync is the engine that runs Gigya integrations with:

- ESP marketing systems
- Customer relationship management systems (CRM)
- Data management platforms (DMPs)
- Any file based integrations, using interim platforms such as SFTP, Azure and Amazon.
- Other platform types, by dynamically writing data to an external endpoint

IdentitySync jobs can be carried out on a one-time basis, for example if migrating data, or they can be scheduled to run on a regular basis in order to keep your platforms synchronized.

IdentitySync APIs use the **idx** namespace.

> IdentitySync is a premium platform that requires separate activation. If it is not part of your site package please contact your Gigya Customer Engagement Executive or contact us by filling in a support form on our site. You can also access the support page by clicking "Support" on the upper menu of Gigya's site.

Use Cases

IdentitySync gives you the flexibility to use your data in any way you need. For example, with IdentitySync, the following scenarios are supported:

Admin Activities

- Retrieve all accounts that have remained unverified or unregistered for over a week (**isVerified**==false or **isRegistered**==false and **created**>'one week ago'), export the relevant email addresses to an ESP, from which to send follow-up emails.
- As a sports club, regularly import accounts from an external ticketing system, thus fortifying your fanbase.
- Query the audit log to retrieve deleted users, and use a batch job to other external systems so that they can be deleted from there, too.

User Segmentation and Progressive Profiling

- Set up data fields for segmenting users according to certain types of behavior in your site - such as Loyalty interactions, purchases or page visits (people who like and share content related to tabi socks, or have purchased said socks) then use an IdentitySync job to send only users that match these criteria to a marketing system for a targeted campaign (50% off in our summer sock sale).
- Use a Gigya-toGigya IdentitySync job to query users by Facebook likes stored in their profiles - for example, people who like vampire and zombie related content - and to plant a value (e.g. "horrorFic") in a Gigya data field. Then launch a gruesome Halloween marketing campaign targeting these users.
- Use an IdentitySync job to initialize default data to a Boolean field (e.g. set to null), and using a custom component that is activated according to this value, trigger a progressive profiling screen that requests more information from site visitors.

Main Features

For full, up-to-date details of the service's capabilities, see the Component Repository.

Main Supported Data Sources/Targets	Sample Transformations	Main File Formats
• Gigya accounts and email accounts • Gigya Data Store • FTP • SFTP • Amazon S3 cloud • Azure • ESP, DMP, CRM Platforms • Other platforms, using the generic API writer	• Reordering, renaming, and removing fields • Replacing strings within field values using **regex** • Using **JEXL** expressions to create new fields based on the values of existing fields • Flattening objects (with some limitations) • Flattening an array field into a string field • PGP encryption and decryption (note that GPG is not supported)	Main file formats supported: • DSV • JSON • Salesforce, Mailchimp, Krux and other formats • GZIP, LZO

Building Blocks

Each IdentitySync job runs a **dataflow**. The building blocks of the dataflow are dedicated **components**. A component is a pre-configured unit that is used to perform a specific data integration operation. The components include readers, writers, transformers and lookups. Each component is responsible for performing a single task, such as:

- Extracting accounts from Gigya based on specific parameters
- Changing some field names
- Creating a CSV file
- Uploading a file to a given FTP
- Writing data directly to a target platform or sending it to a generic API endpoint

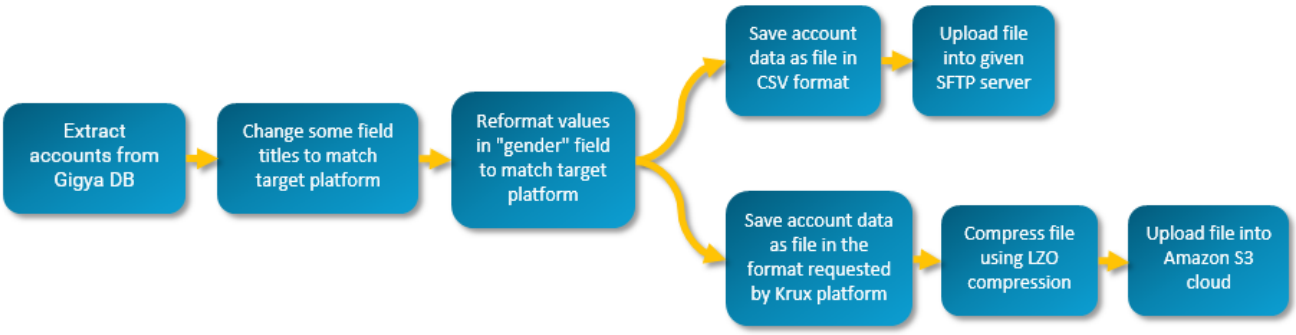
Components can be added to the dataflow, removed or changed as needed.

For detailed information, visit the Component Repository and see sample Dataflow Templates.

Dataflow Example

The following chart is a visualization of a dataflow in IdentitySync. This dataflow exports user accounts from Gigya to a partner platform (Krux).

Each step in the dataflow runs a separate component.



> The above flow demonstrates a split dataflow. Dataflows are split using the **next** parameter. For more details, see Customize the Dataflow.

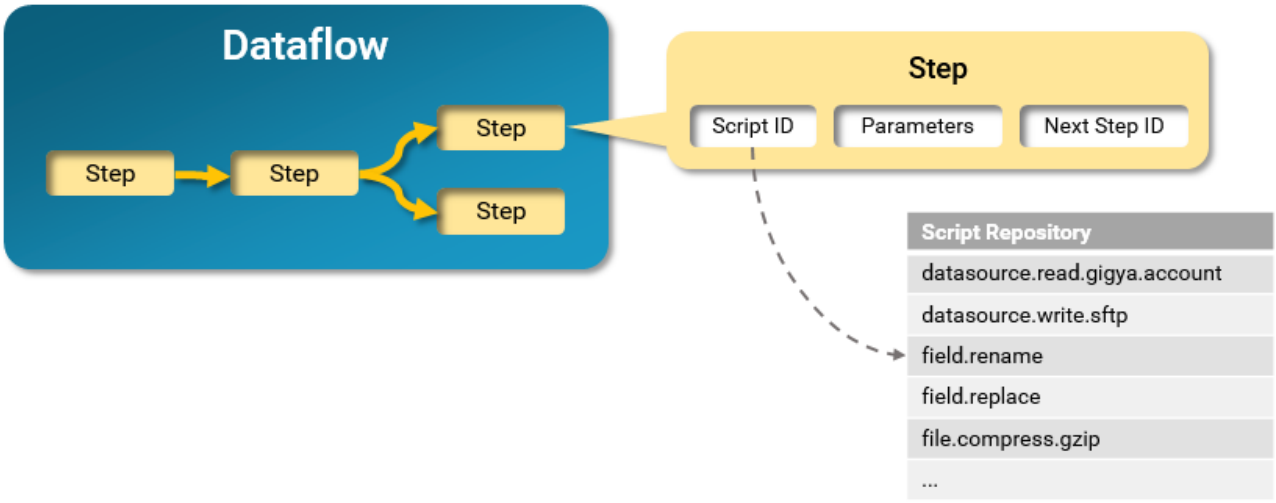
Click to view a sample JSON dataflow:

```
{
  "name":"NextDigital Krux",

  "description":"enter your dataflow description here...",

  "steps":[

     {
        "id":"account",
        "type":"datasource.read.gigya.account",
        "params":{
           "fields":"UID,profile.gender,profile.age",
           "whereClause":"data.internalKruxIdSyncResponseCode=true"
        },
        "next":["rename"]
     },
```

.

Using IdentitySync

Object Model



Dataflow

A dataflow is a complete definition for a transfer of information between Gigya and a third-party platform in a specific direction.

The dataflow includes all the necessary information about where the data is extracted from, which data is extracted, how the data is processed, and where the data is transmitted.

For details, see Dataflow Object. For samples, see Dataflow Templates.

Step

Steps are the building blocks of the dataflow. Each step is a call to a component that performs a specific task, such as extracting accounts from Gigya, or compressing a file in GZIP format. The step calls the component with specific parameters, and the output is passed on to the next step in the dataflow for further processing. For example, "datasource.read.gigya.account" is a component that searches the Gigya account database and returns all accounts that match specific parameters (an SQL-like search query). This component will typically be called in the first step in a dataflow that exports accounts from Gigya to a partner platform. Each step includes the following attributes:

- **id**: the unique identifier of the step within a given dataflow. Each step has to have an ID so it can be called by other steps in the "next" attribute.
- **type:** the ID of the IdentitySync component used in this step (see Component Repository).
- **params**: an object defining the parameters passed in this IdentitySync component.
- **next**: an array containing one or more IDs of the next step(s) to be carried out.
    - A step to which no other step refers in the **next** attribute is automatically considered the entry point of the dataflow.
    - Steps which do not have a **next** attribute are automatically considered end-points of the dataflow.

- Assign multiple values to a **next** attribute to split the dataflow. See example.

Step Structure

```
{
         "id":"dsv", // A name you assign to this step that serves as a
unique identifier in this dataflow.
         "type":"file.parse.dsv", // The component run in this step.
         "params":{ // The parameters and values, expressions etc. used in
this step.
            "columnSeparator":","
         },
         "next":[ // The next step to be run once this step completes.
            "rename"
         ]
      },
```

For examples of end-to-end dataflows, see Export from Gigya to SFTP and Import from SFTP to Gigya.

Data File Example

The following is an example of a data file in DSV format. The quotes around each field can be removed.

```
"UID","email","firstname"
"_gid_XeCEe4oZYgvn83np9DPA+g==","sample_mail@something.com","John"
"_gid_XeCEe4oZYgvn83np9DPA+g==","","Jane"
```

Handling Errors

IdentitySync includes a built-in capability for separating failed records and writing them to a file, so that they may be reviewed and handled, and fed back into the flow.

Handling failed records is done by adding an additional step after a "writer" step, for writing to a separate file the records that did not complete the flow successfully. For detailed instructions, follow the implementation flow below (under Edit the Dataflow). For a code sample of a flow that writes failed records to SFTP, see the Component Repository.

> Note that IdentitySync jobs are scheduled in UTC time. Therefore, the platform participating in the flow should be set to the UTC timezone to ensure that file requests are handled properly.

Implementation

To create an integration based on IdentitySync, complete the following process:

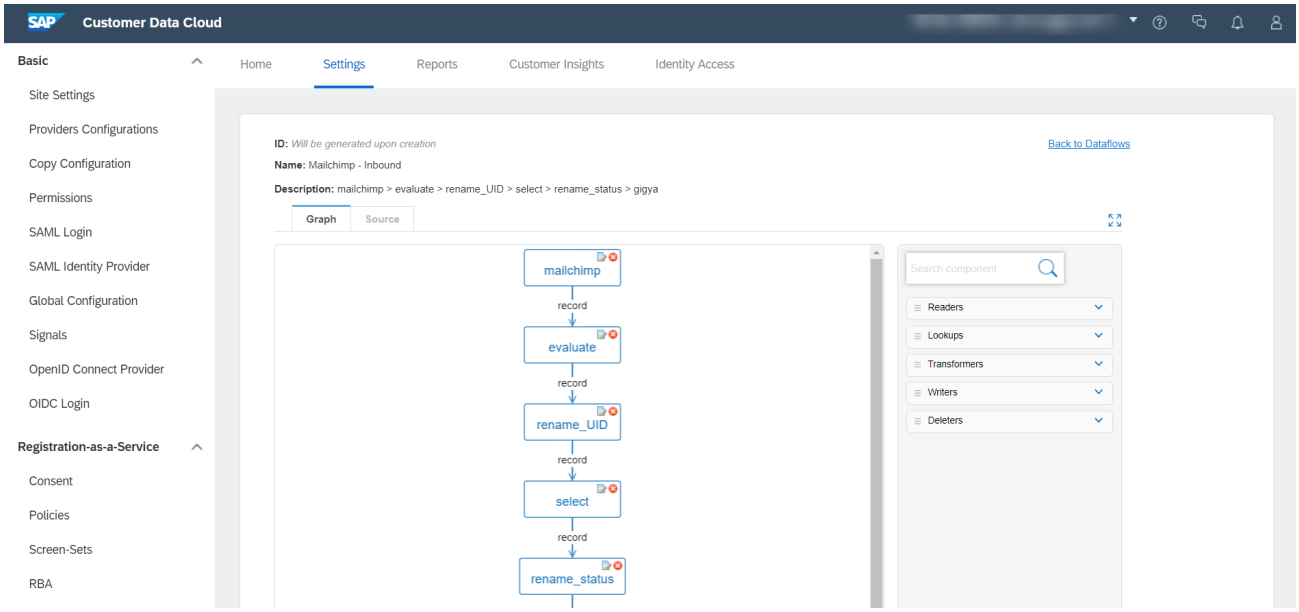
1.	2.	3.
Create Dataflow	**Edit the Dataflow**	**Schedule the Dataflow**

IdentitySync is a premium platform that requires separate activation. If it is not part of your site package please contact your Gigya Customer Engagement Executive or contact us by filling in a support form on our site. You can also access the support page by clicking "Support" on the upper menu of Gigya's site.

Unable to render {include} The included page could not be found.

1. Create Data Flow

1. Open IdentitySync Data Flows in Gigya's Console. Make sure your are signed in and have selected the relevant site. The IdentitySync dashboard may also be accessed by clicking **Settings** in the upper menu and then **IdentitySync Data Flows** in the left menu.
2. In the dashboard, click **Create Data Flow**.
3. In the **Create Data Flow** window, select the data flow integration from the dropdown. If the flow you wish to create is not available in the dropdown, select any available flow: it is customized in the next steps. For more information, see Dataflow Templates.
4. Select the data flow template: the direction of the flow, whether from or into Gigya. Note that at the bottom of this window, you can see an outline of the flow that will be created (e.g., Account > rename > dsv > gzip >sftp).
5. Click **Continue**. As a result, the **IdentitySync Studio** screen opens in the dashboard.



2. Edit the Data Flow

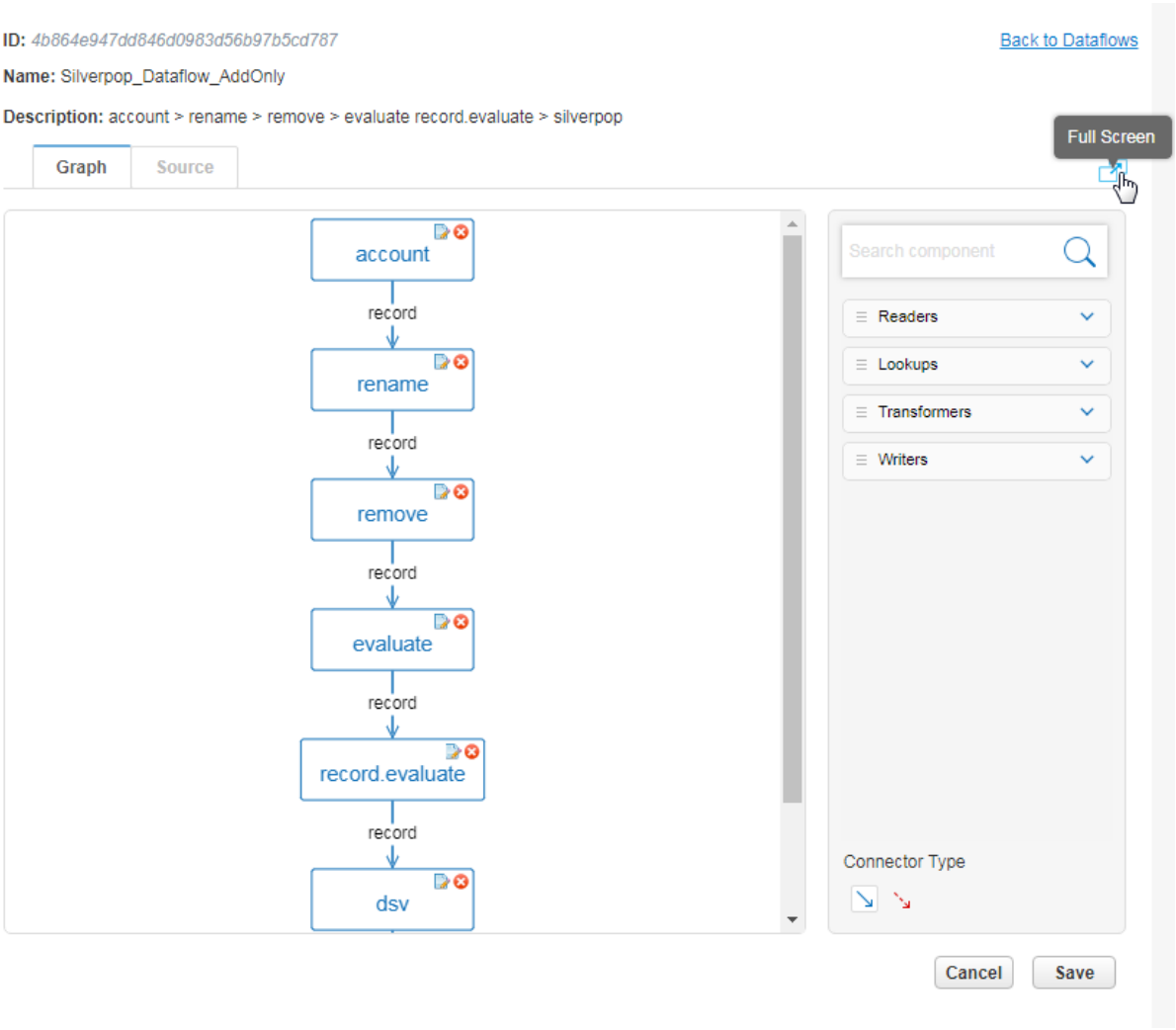
The data flow you created is built of the required steps for data transfer between Gigya and the selected vendor. Use the Component Repository t o understand the structure and parameters required in each step.

Using IdentitySync Studio, you can:

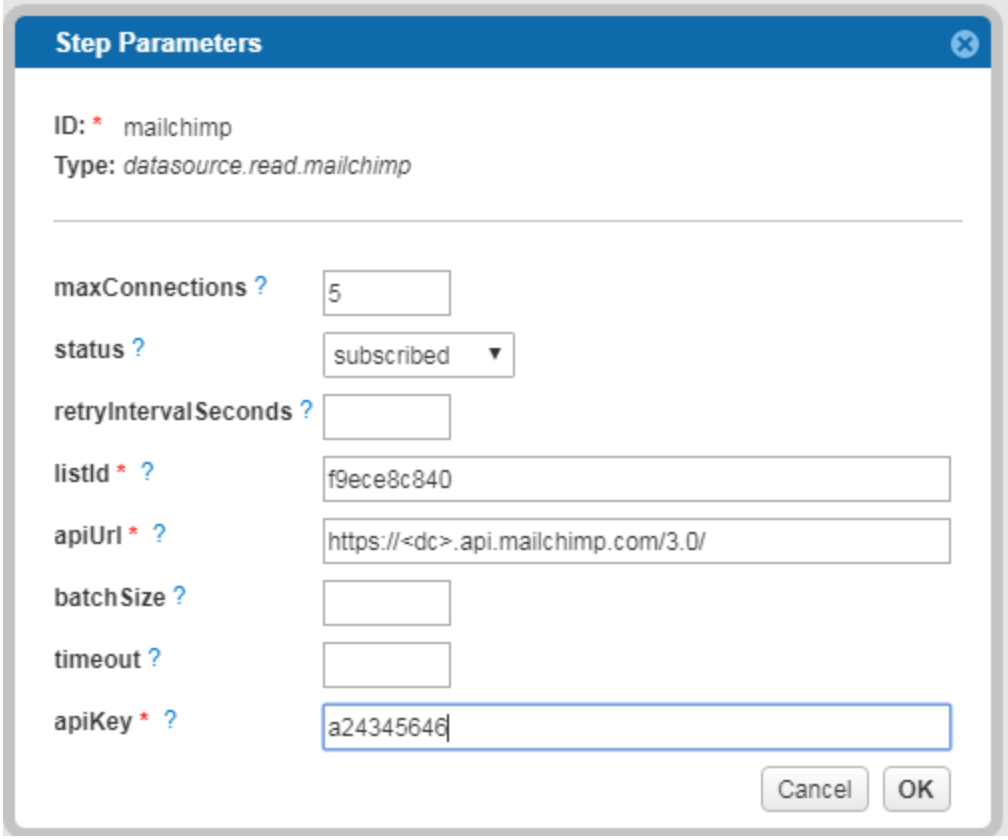
- Specify passwords, IDs, API keys etc. required for accessing each system and customer database.
- Add the names of fields included in the data flow.
- Flatten fields, remove non-ASCII strings, specify the compression type, parse in JSON or DSV format, etc.
- Map fields and extract array data, for example using field.array.extract.
- Change the name of the data flow.
- Split a data flow, for example if you want to create two duplicate files and upload each file into a different destination. To do so, simply drag and drop the relevant step into the flow, and add connecting arrows as needed. In the code for the flow, this will be expressed in the **next** attribute, where you will find reference to the next two steps rather than just one. For a sample dataflow which employs this method, see the Epsilon Dataflow.
- Add Custom Scripts.
- Write failed records, that did not complete the flow successfully, to a separate file for review.

To do so:

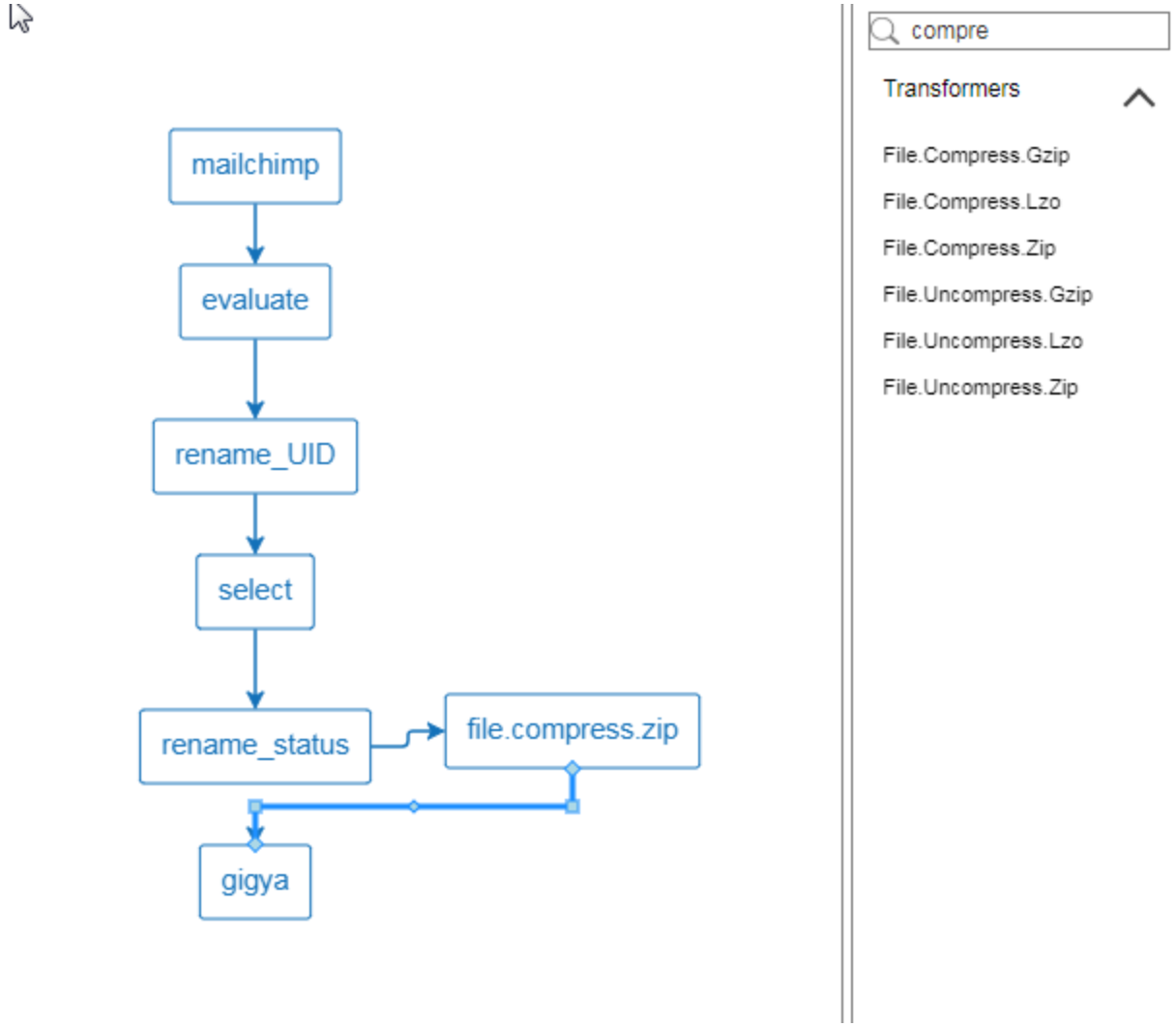
1. If it's more convenient, you can work in full screen mode by clicking the full-screen toggle on the top rigt corner.



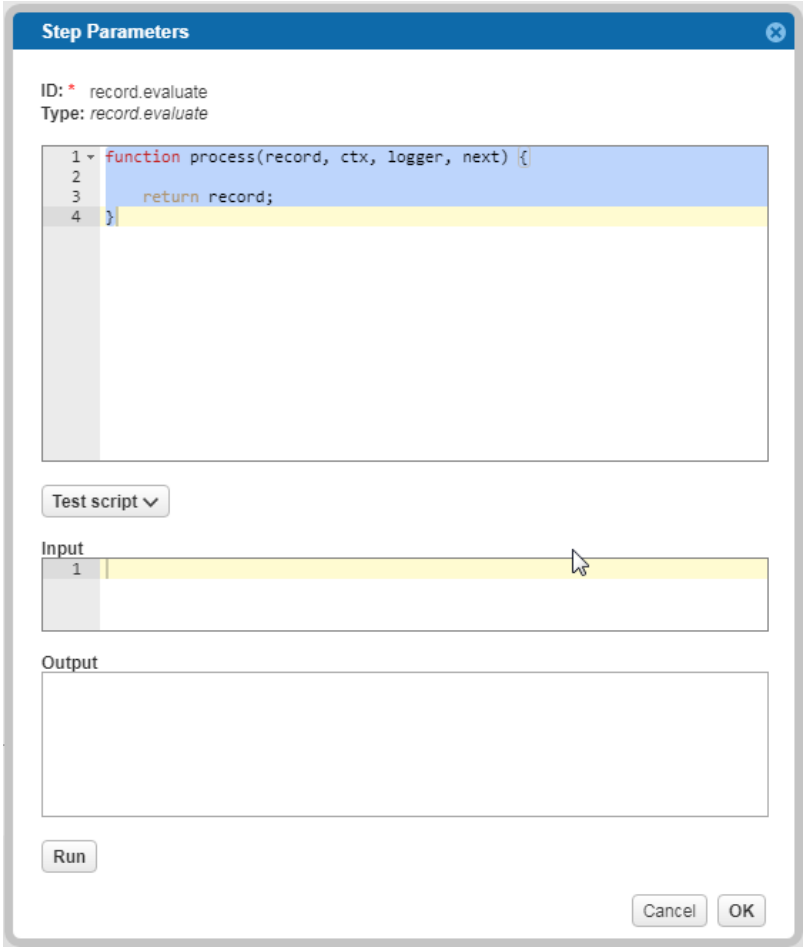
2. Double-click any of the steps to add or edit its parameters. Click OK when finished.



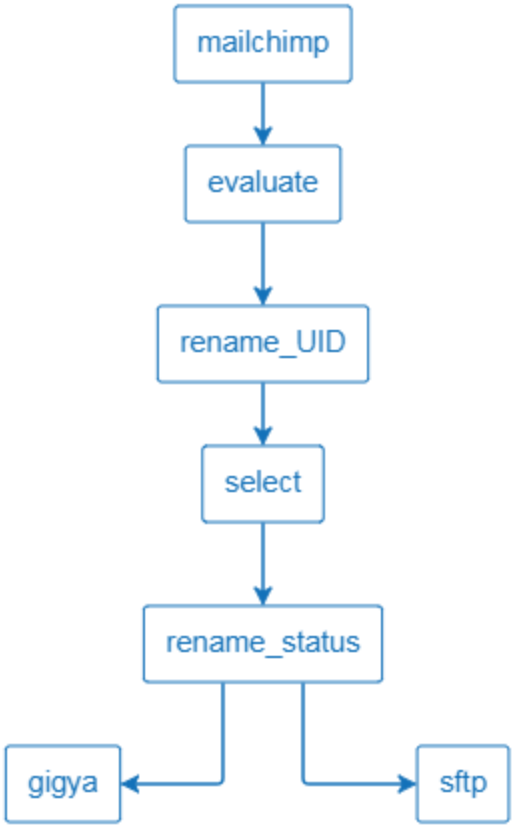
3. To add a new step, start typing its name in the **Search component** box. Drag the step from the list of components into the canvas.
4. Drag arrows from/to the new step and from/to existing steps, to include it in the correct place in the flow. Make sure the "Success path" arrow is selected, under **Connector Type**.



5. To add a custom step, locate the **record.evaluate** step in the list of components and drag it to the canvas.
6. Double click the custom step to open a JavaScript editor. Click **Test script** to validate the code. For a full explanation of custom steps, see IdentitySync Custom Scripts.

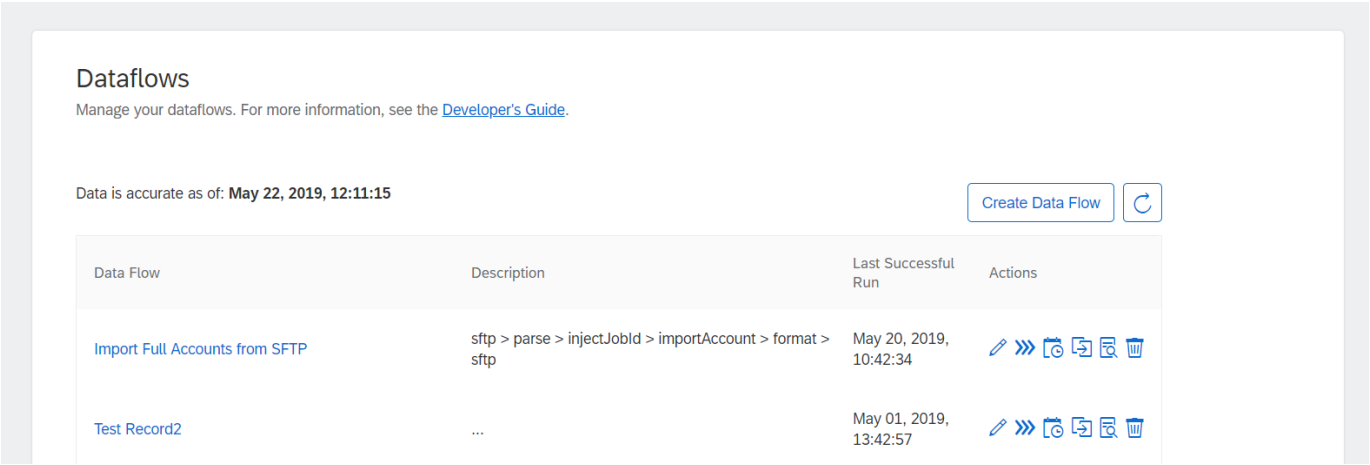


7. To split the data flow (for example to write to two target platforms), add the relevant step (e.g. another "write" step) and draw arrows accordingly:



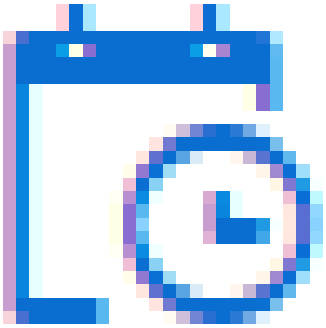
8. Handling failed records: You can add additional steps after a "writer" step, for writing to a separate file the records that did not complete the flow successfully. To do so:
    a. Add the relevant components to the flow (for example, a file.format step to write the records to a file, and a writer to write the file to the relevant destination).
    b. Under **Connector Type**, select the "Error path" connector.
    c. Draw a connection from the original writer, to which successful records will be written, to the next step that handles failed records (e.g., the file.format step).
    d. Under **Connector Type**, select the "Success path".
    e. Connect the next steps that handle the failed records (e.g., the writer) using the "Successful path" connector.
9. Delete a step by selecting it and hitting the **Delete** button on your keyboard.
10. If necessary, click **Source** to review the data flow code , and edit the code as needed.
11. Click **Save**.

Your dashboard should now look something like this:

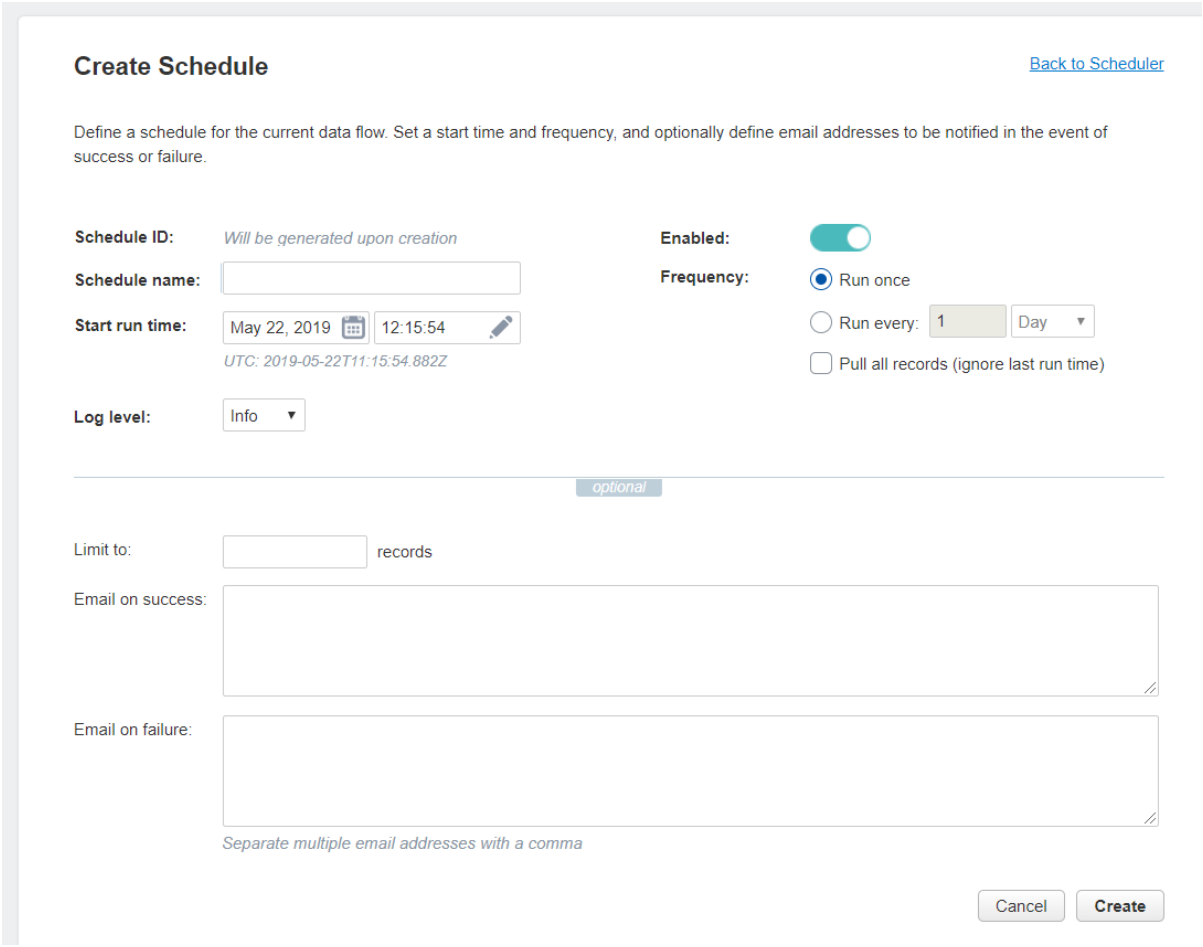


Actions

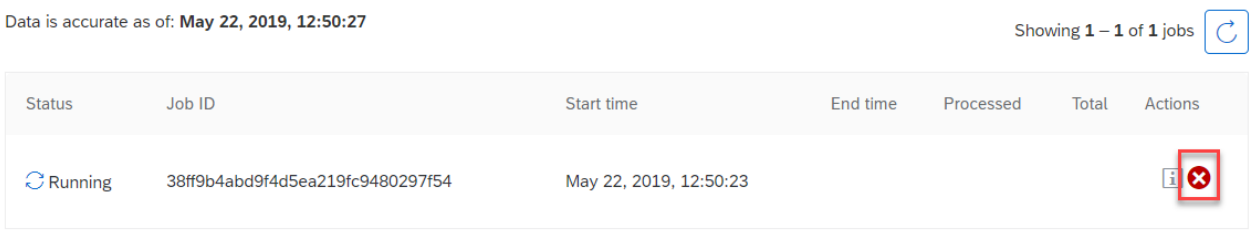
The following actions are available:

Icon	Action	Description
	Edit	Opens the current data flow in IdentitySync Studio and change any of its attributes, steps and parameters.
	Run Test	Runs the data flow once on 10 records for test purposes. If the test was successful, after refreshing the dashboard, you will see the timestamp of the test run under **Last Successful Run**. Use the **Status** button to view the details of the run. See Job History section on this page.
	Schedule	See Schedule the Dataflow.
	Duplicate	Useful for creating a new data flow based on a flow which has already been customized, if you wish to create a similar flow with slight variations.
	Status	Displays the status of the current jobs running in your IdentitySync configuration. See Job History section on this page.
	Delete	Deletes this data flow.

3. Schedule the Dataflow



1. Under **Actions**, click
   
   (schedule) to open the Scheduler.
2. Click **Create Schedule**.
3. Configure the schedule:
   - Enter a schedule name
   - Change the start time as needed
   - Choose whether to run once or at scheduled intervals
   - "Pull all records" should usually be selected only in the first run, when migrating records from one database to the other, and in any case should be used with caution. If the checkbox is not selected, and this is the first time this dataflow is run, records will be pulled according to the following logic:
     - If the dataflow is set to run once, all records from the last 24 hours will be pulled.
     - If the dataflow is recurring, records will be pulled according to the defined frequency. For example, if the dataflow is set to run once a week, the first time it is run, it will pull all records from the previous week.
   - (Optional) Enter the email adress(es) for success and failure of the dataflow run.
   - (Optional) Limit to a specific number of records. This is usually used for test runs: when running a test from the dashboard, a one-time schedule is created which runs immediately for 10 records.
4. Click **Create**, and, once you are back in the Schedule dashboard, click the **Refresh** button.
5. The status of the scheduling is indicated in the **Status** column.
6. You can stop a job mid-run by clicking the **Stop** icon under **Actions**:



- The dashboard creates a Scheduling Object.
- **Last Successful Run** corresponds with the **lastRuntime** parameter in the Dataflow Object.
- The **Scheduled Next Run** displays the newest date defined in the scheduler. Therefore, it's possible to see a past date here, if no more recent dates were configured.

Unable to render {include} The included page could not be found.

Test and Monitor

Test Run

Test the data flow by clicking

(run test) under Actions. This creates an immediate one-time run for 10 records. If the run was successful, after refreshing the dashboard (with the **Refresh** button) you will see its timestamp under **Last Successful Run**.

Notification Email

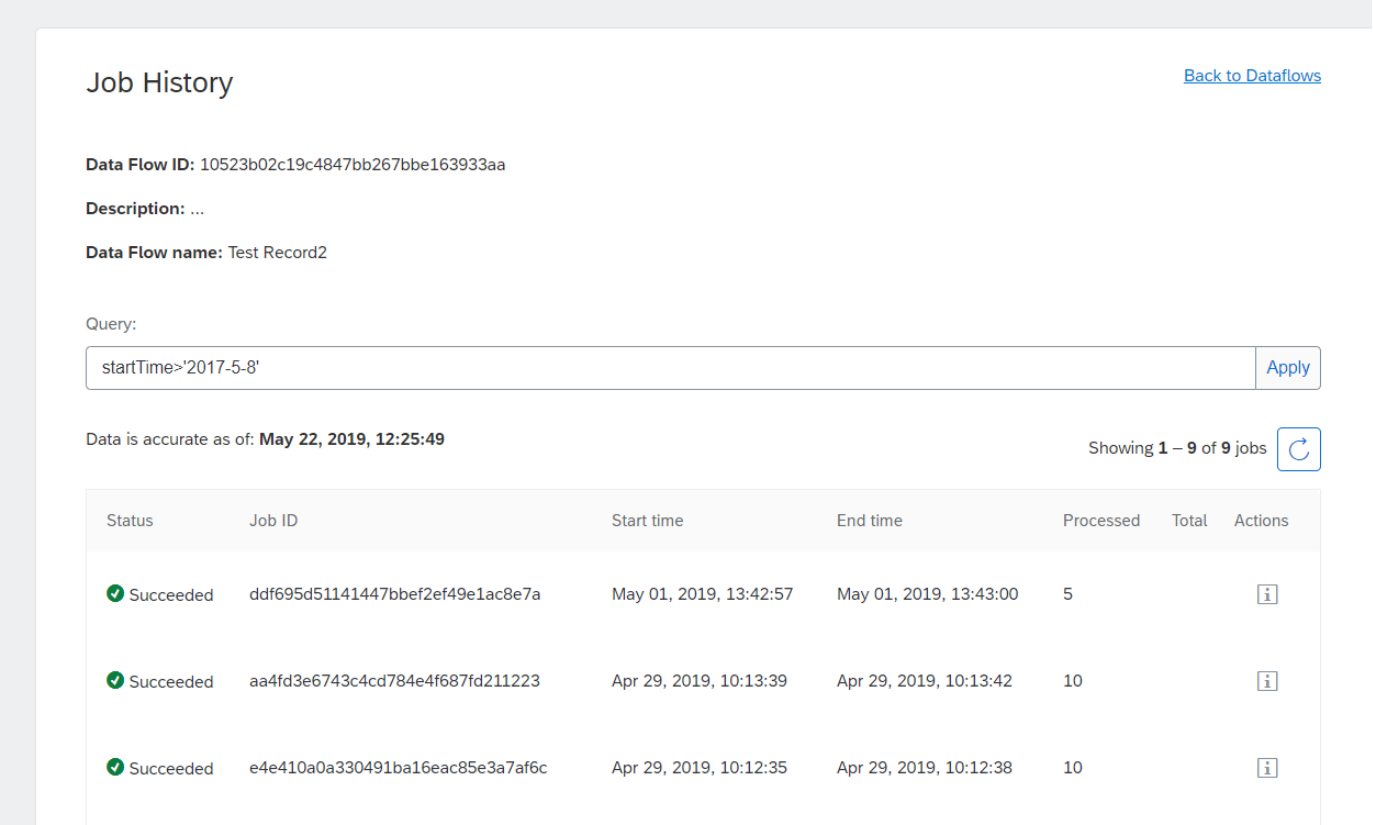
When scheduling the dataflow, you can enter email addresses to which a success and/or failure notification will be sent. We recommend adding idx-failure-jobs@gigya-inc.com to the list of failure notification email addresses, so that Gigya will receive feedback of system health.

Job History

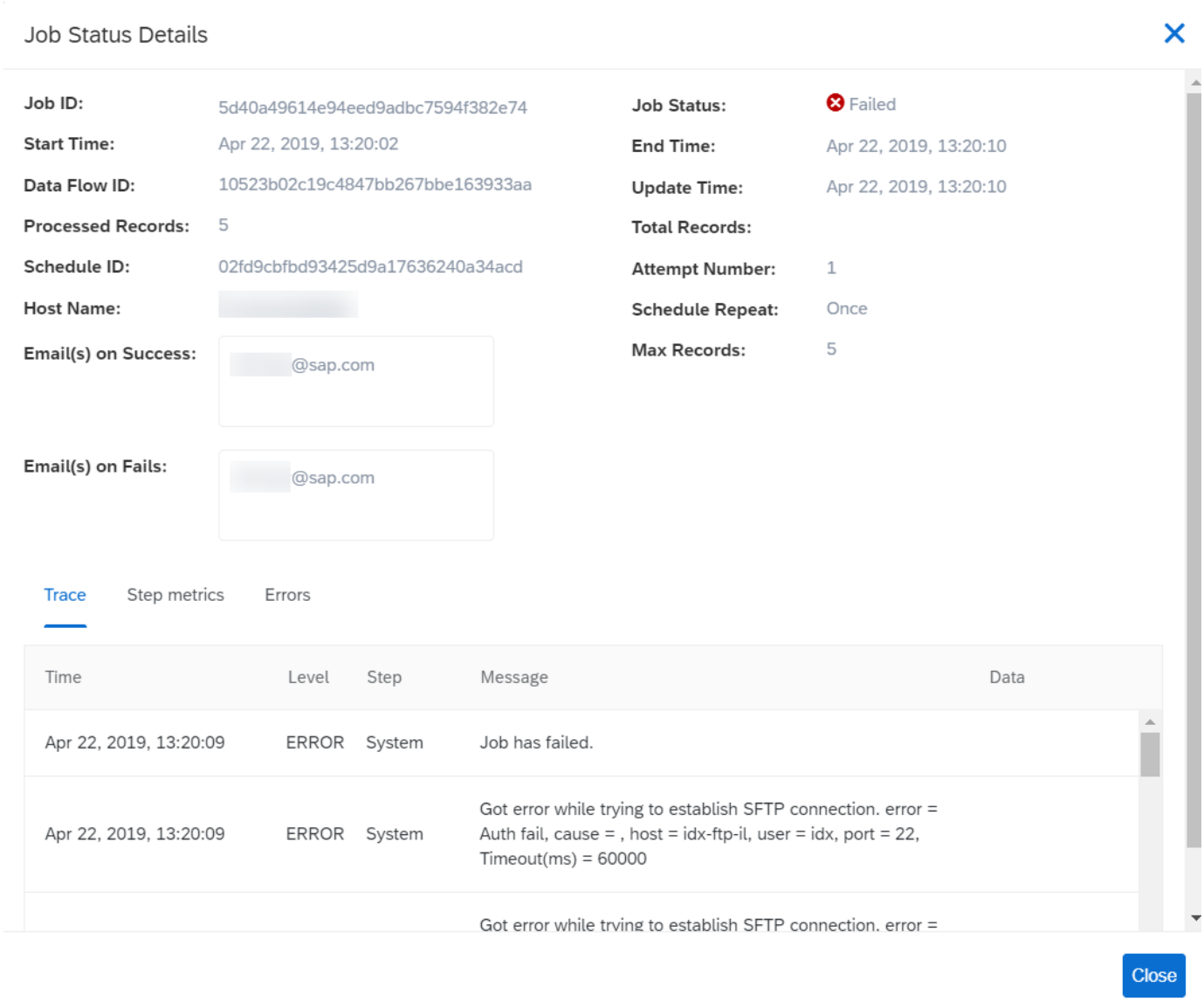
You can monitor data flows by reviewing previous runs (jobs). The job history displays the status of each run, its start and end times, and the number of records for which the data flow was completed successfully (under **Processed**).

Under **Actions**, click the Status button

to open the Job History screen.

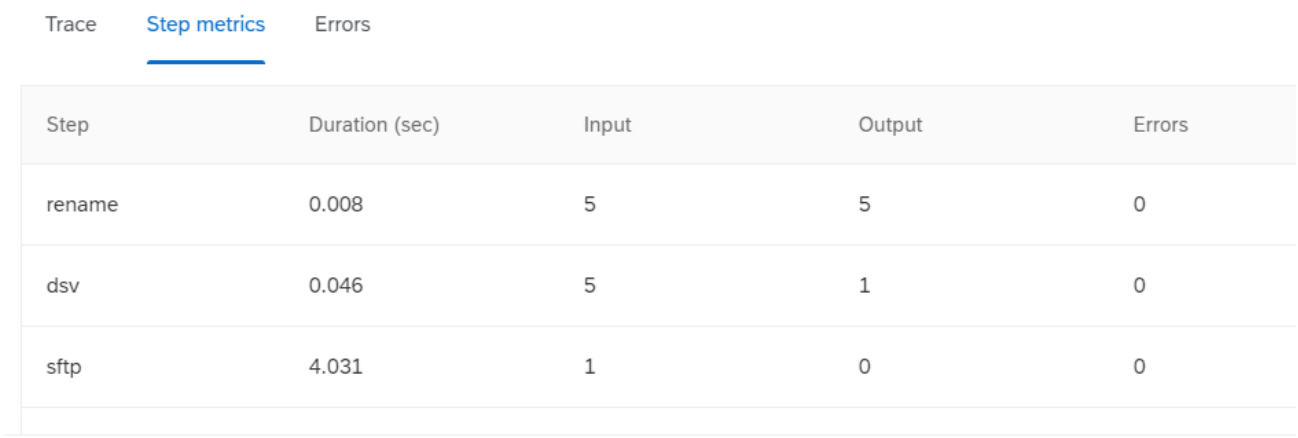


For advanced monitoring and debugging, click the info icon for the relevant job under **Details**, and the **Job Status Details** screen opens.

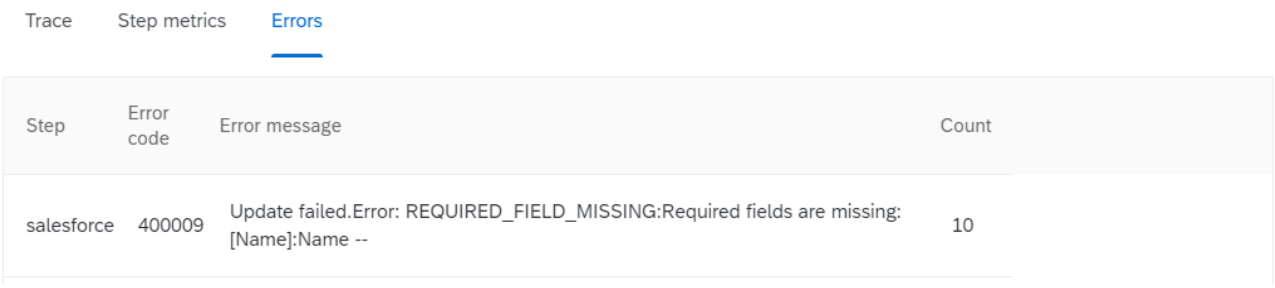


Note the tabs that display the following detailed information:

- **Trace**: Contains a detailed trace of the job execution, including a log level and timestamp of each step.
- **Step metrics**: Displays the following metrics for each step: Duraion, Input, Output and Errors. Using step metrics, you can find out what were the bottlenecks of a job that took a long time to run, review performance issues, and monitor the number of records that completed the flow.



- **Errors**: Displays details of the errors that occurred during the job execution.



Copying Accounts From One Site to Another

IdentitySync gives you the option of copying the account database from one Gigya site to another, using the read from Gigya and write to Gigya components. When doing so:

- The job should be set up on the target site.
- On the source site, you should create an application, and use the credentials in the dataflow. For more information, see Application Keys.

- Filter the accounts to be extracted and written using the WHERE clause in the datasource.read.gigya.account component.

Click to view a sample dataflow for copying site accounts

```
{
    "name": "Copy accounts between 2 different sites",
    "steps": [
        {
            "id": "account",
            "type": "datasource.read.gigya.account", // Read from source
site
            "params": {
                "select":
"UID,profile.email,profile.firstName,profile.lastName", // The Gigya
fields to extract
                "batchSize": 300,
                "from": "accounts",
                "deltaField": "lastUpdatedTimestamp",
                "maxConcurrency": 1,
                "apiKey": "...", // The source site API key
                "userKey": "...", // The user key of the application
created on the source site
                "secret": "..." // The secret key of the application
created on the source site
            },
            "next": [
                "write"
            ]
        },
        {
            "id": "write",
            "type": "datasource.write.gigya.account", // Write to the
target site
            "params": {
                "maxConnections": 10
            }
        }
    ]
}
```

Unable to render {include} The included page could not be found.

IP Whitelisting

Depending on your networking policies, you may have to add the IPs of IdentitySync servers to a whitelist in order to allow IdentitySync to upload/pull information.

The full list of Gigya IPs are listed here. In addition, the following addresses are related to IdentitySync:

EU Data Center:

- 46.51.204.12
- 54.76.191.69

US Data Center:

- 52.204.240.189
- 18.204.248.129

AU Data Center:

- 54.66.139.77
- 54.66.141.200

CN Data Center:

- 101.132.236.215

RU Data Center

- 95.213.253.43
- 95.213.238.43