TIBCO Flogo® Enterprise

Quick Start

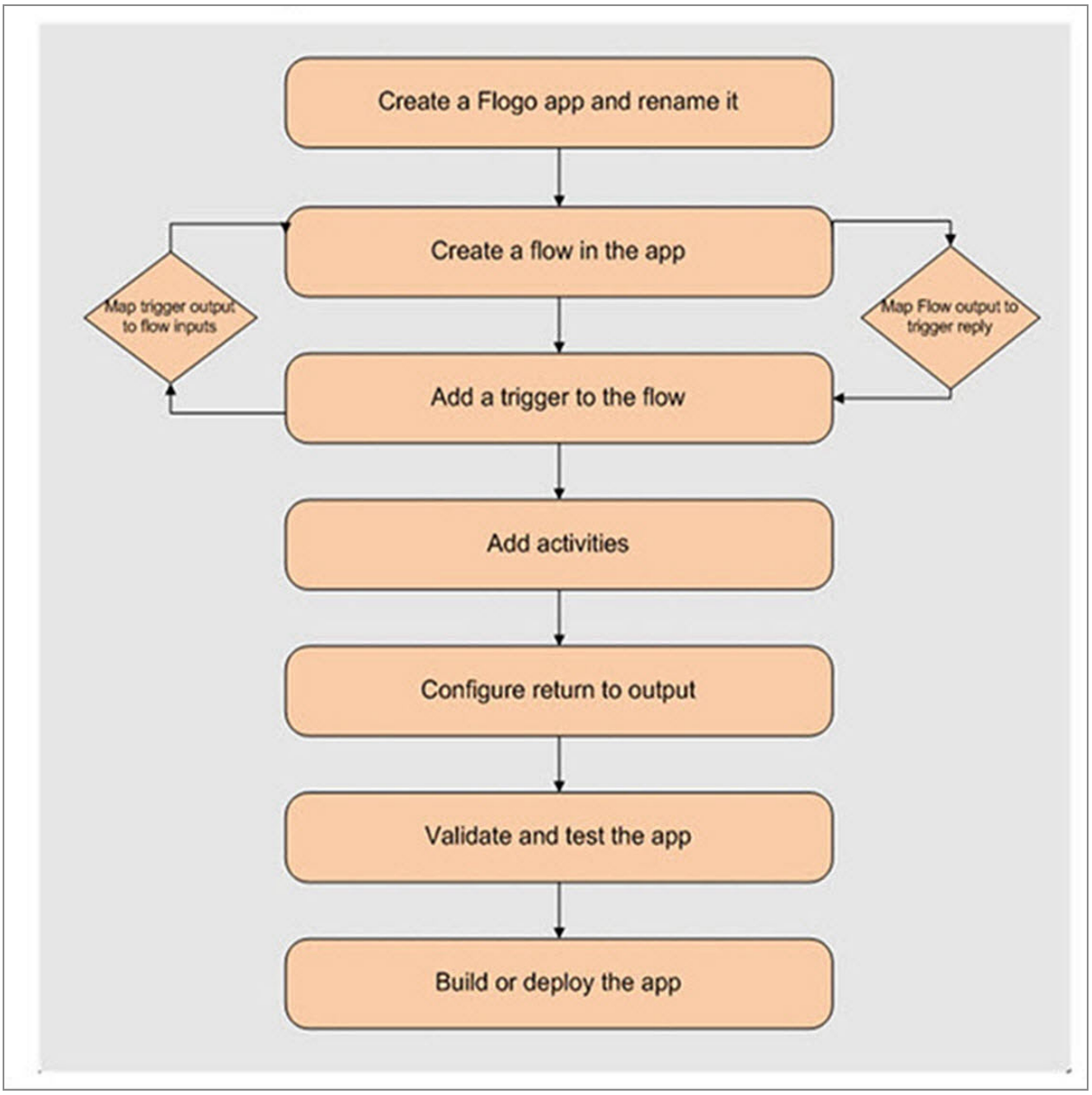
Version 2.25.7 | July 2025

Contents

Quick Start

The purpose of this section is to assist new users in understanding the basic workflow of the TIBCO Flogo app and also give a demo about the configuration and deployment of an app.

This is the workflow for creating a Flogo® app and deploying it to TIBCO Cloud™.


Create a Flogo app and rename it
Create a flow in the app
Map trigger output to flow inputs
Map Flow output to trigger reply
Add a trigger to the flow
Add activities
Configure return to output
Validate and test the app
Build or deploy the app

Step 1: Create a Flogo app and rename it

Begin with creating a Flogo app, that invokes a Rest service and generates a log message, that shows the status of the invoked Rest service. We further rename the new app as `Rest_Service_app`.

Creating a Flogo App

Procedure

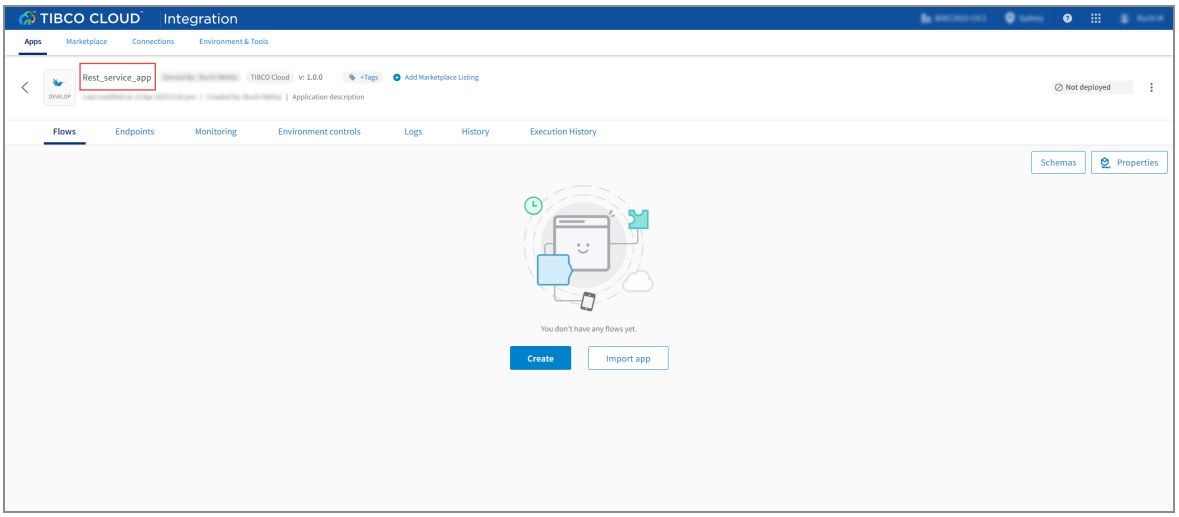
1. Click **Apps** > **Create/Import**.
2. Then, go to **Quickstart** > **All app types** > **Apps**, and click **Create a Flogo app**.

   A Flogo app is created with the default name in the `New_Flogo_App_<sequential_app_number>` format.

Renaming the new Flogo app

Procedure

1. Change the app name to `Rest_service_app`.
2. Click the default app name to make it editable, type the new name, and click anywhere outside to save the changes.
3. The name of the app changes to `Rest_service_app`.


TIBCO CLOUD
Integration
Apps
Marketplace
Connections
Environment & Tools
Rest_service_app
TIBCO Cloud
v: 1.0.0
+Tags
Add Marketplace Listing
DEVELOP
Application description
Not deployed
Flows
Endpoints
Monitoring
Environment controls
Logs
History
Execution History
Schemas
Properties
You don't have any flows yet.
Create
Import app

Step 2: Create a flow in the app

A flow is used to implement the business logic as a process. A flow can consist of one or more activities that perform a specific task. Activities are linked together to create the logic of the flow. It is achieved by defining the order of execution of those activities. Also, it can contain conditional logic for branching. Every app must have at least one flow and in most cases, a trigger that initiates the flow. Here, a trigger is the entry point for events that receive data from external sources. A trigger can be a subscriber on an MQTT topic, Apache Kafka® topic, HTTP REST interface, or a specific IoT sensor.

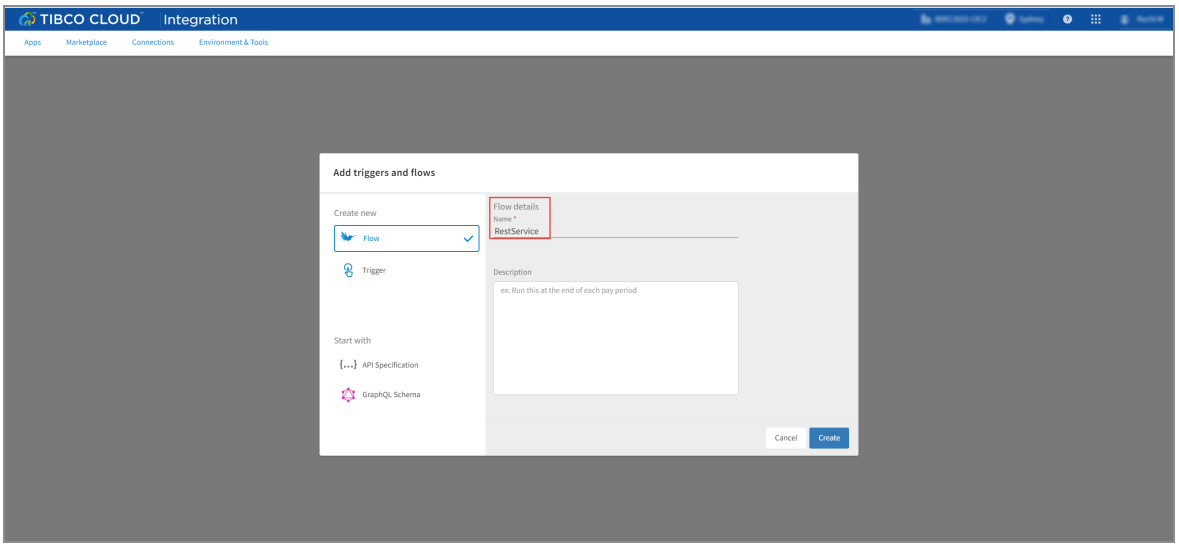
In this step, we go through the creation of a new flow attached to a REST trigger. The **ReceiveHTTPMessage** REST trigger listens for incoming rest requests to invoke the rest service and generates a log message about the status of the invoked rest service.

Creating a Flow

Procedure

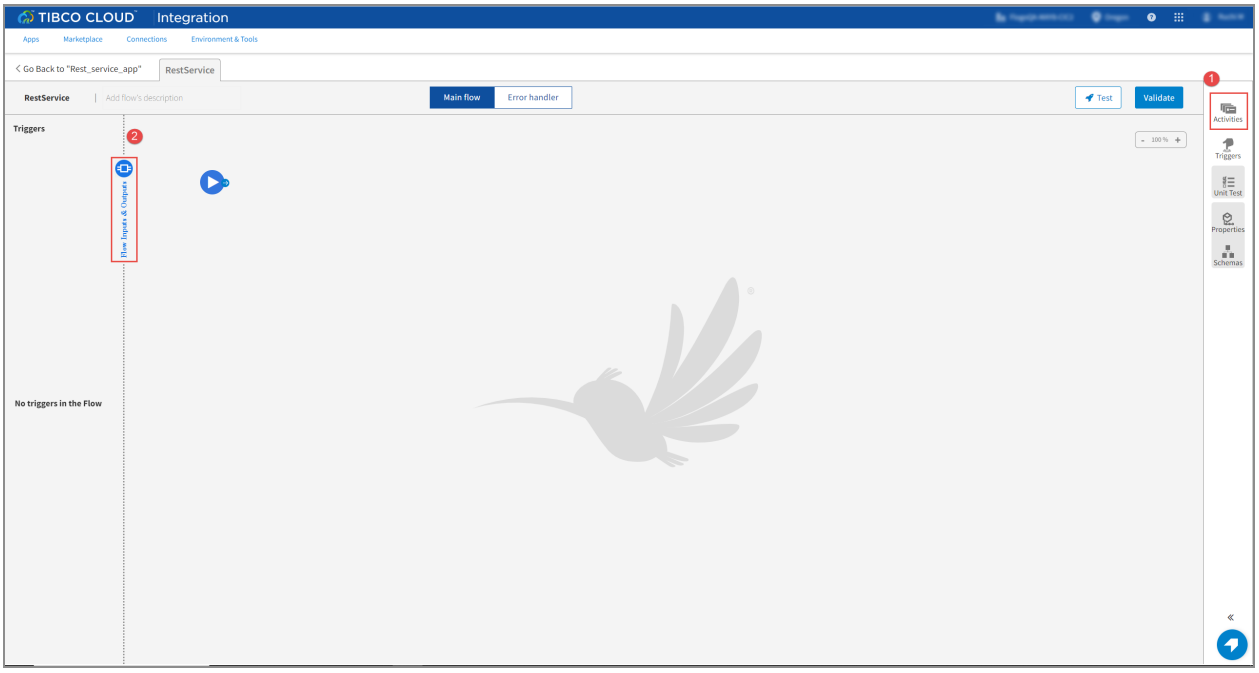
1. On the Flows page, click **Create**.
2. In the **Flow details** section, provide the following details and click Create:

   Name: RestService, **Description:** Optional description of the flow.



After the flow is created, an empty flow appears on the following screen. Here, you have two options for starting the process -

- Choose the **Triggers** option to configure the triggers of the process.
- Choose the **Flow Input & output** option to configure the flow of the process.



Note: Choosing the **Trigger** option or the **Flow input & output** option is entirely up to you. Only for the sake of maintaining the procedure, the next step chosen is adding a trigger.

Step 3: Add a trigger to the flow

Triggers receive events from external sources such as Kafka, Salesforce, and GraphQL. The handlers residing in the triggers, dispatch events to the flows. Flogo provides a set of out-of-the-box triggers and connectors for receiving events from external systems. In this guide, a **ReceiveHTTPMessage** REST trigger is used to listen to incoming REST requests.

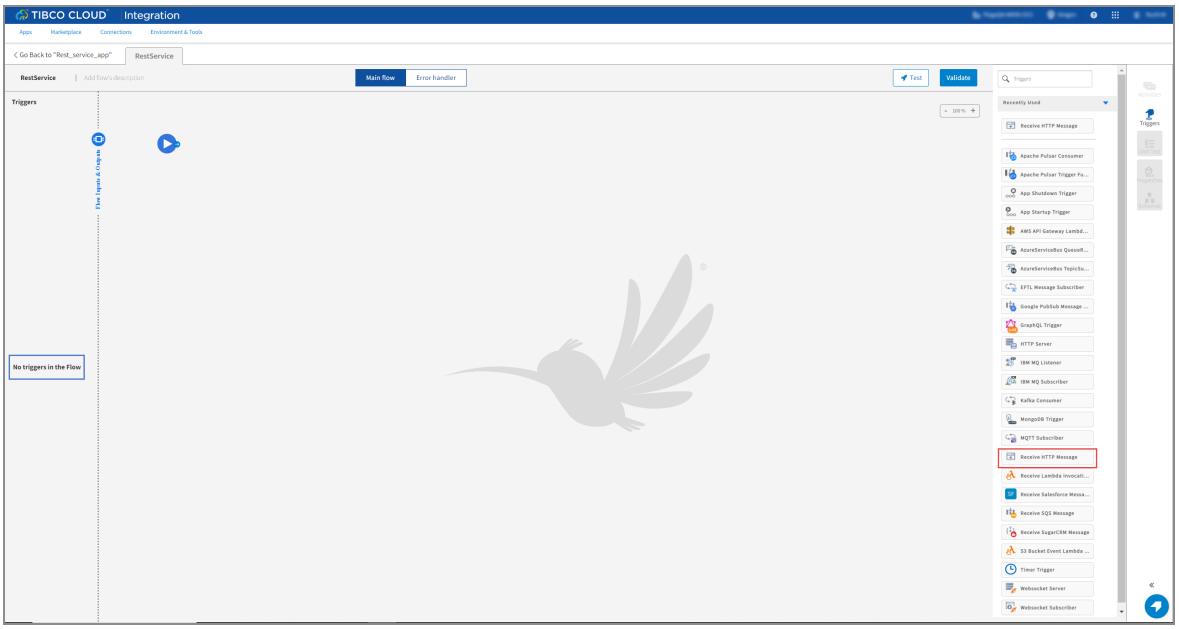
Adding a trigger to the flow

Procedure

1. On the sidebar, click the **Triggers** option. The **Triggers** palette lists the triggers that serve a specific use case and each of them has a unique set of configurations.

   For more information about triggers listed in the catalog, refer to the "Triggers" section of *TIBCO Flogo® Enterprise Activities, Triggers, and Connections*.

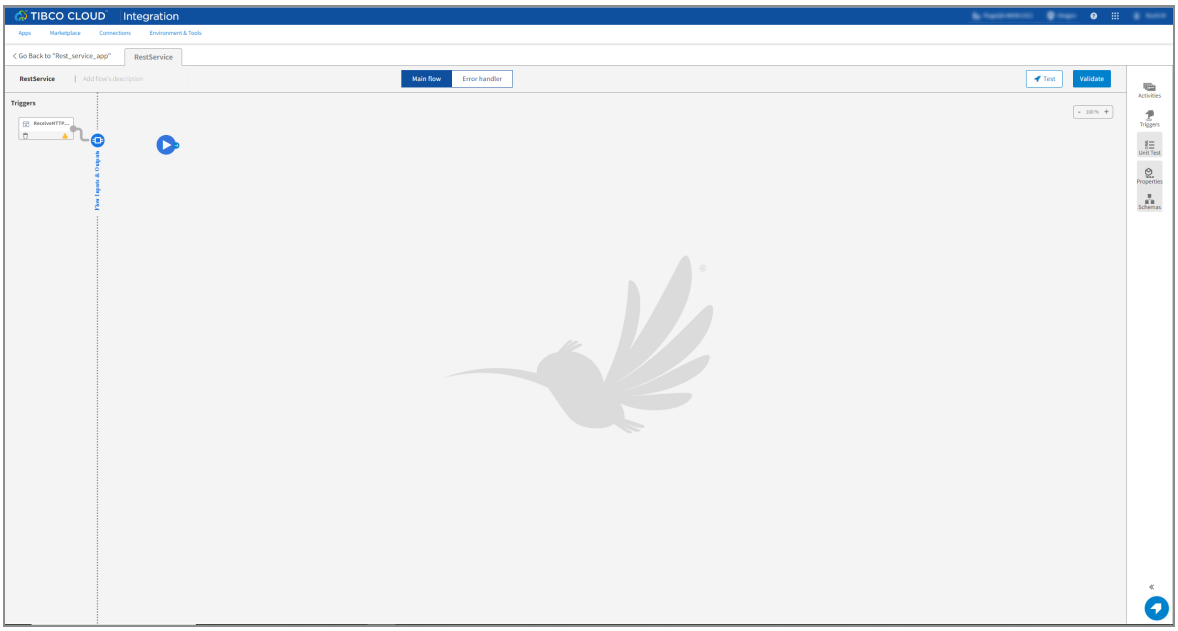
2. When you click the **Triggers** option, you see the **Triggers** palette. On the left side, you also see a column that displays the message "No triggers in the Flow". On the **Triggers** palette, drag the **Receive HTTP Message** trigger in the column on the left side.



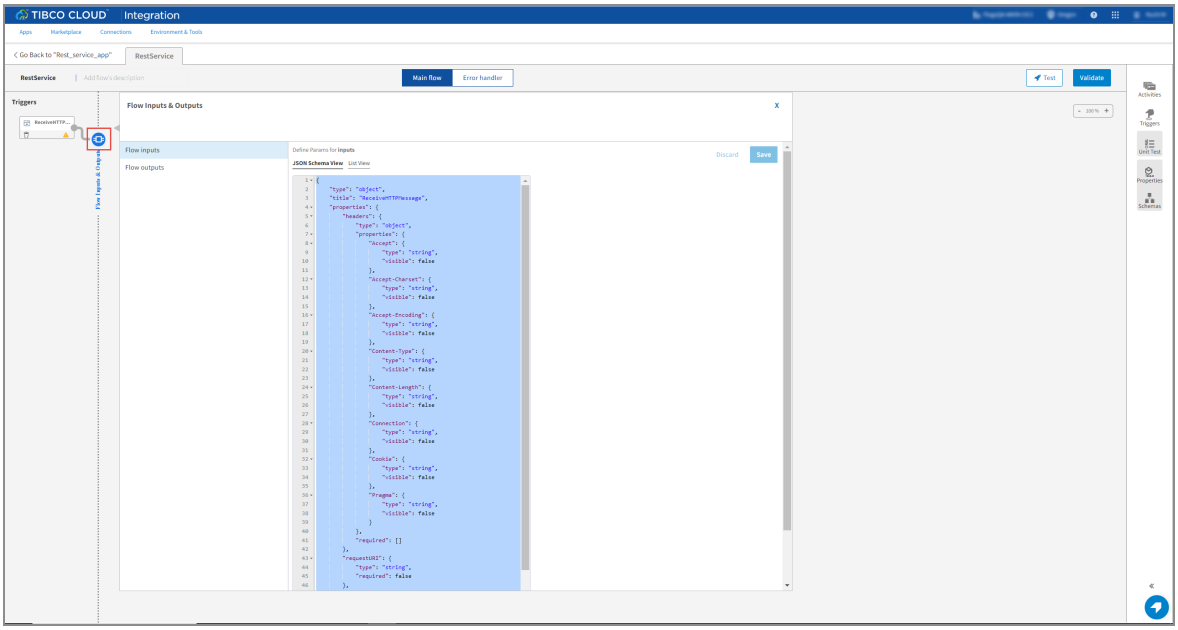
3. In the trigger configuration dialog of the **Receive HTTP Message** trigger, perform the following tasks:
    a. Enter/**abc** in the **Resource path field**.
    b. Click **Continue**.
4. When prompted, select **Copy Schema**.

> **Tip:** You can define a JSON or Avro schema such that it is available for reuse across an app. An app-level schema can be used in any flow, Activity, or trigger configuration where a schema editor is provided. You can simply pick an existing schema from a list.
>
> Currently, Flogo only supports the JSON and Avro types of schemas. For more information about defining, editing, or using the app-level schema, refer to the "App Schemas" section of the *TIBCO Flogo® Enterprise User Guide*.

The trigger configuration dialog automatically closes and you see the following screen.



Click the  icon to view the **Flow Inputs & Outputs** tab as seen in the following screen. Click the icon again to collapse it.



You can also create a trigger before attaching a flow to it. For more information, refer to the "Creating a Trigger without a Flow" of the *TIBCO Flogo® Enterprise User Guide*.

Step 4: Map trigger output to flow input

Flogo provides a graphical data mapper to map data between the activities within a flow, and between the trigger and the flows attached to the trigger within an app. Use the Flow inputs & outputs tab to configure the input to the flow and the flow output.

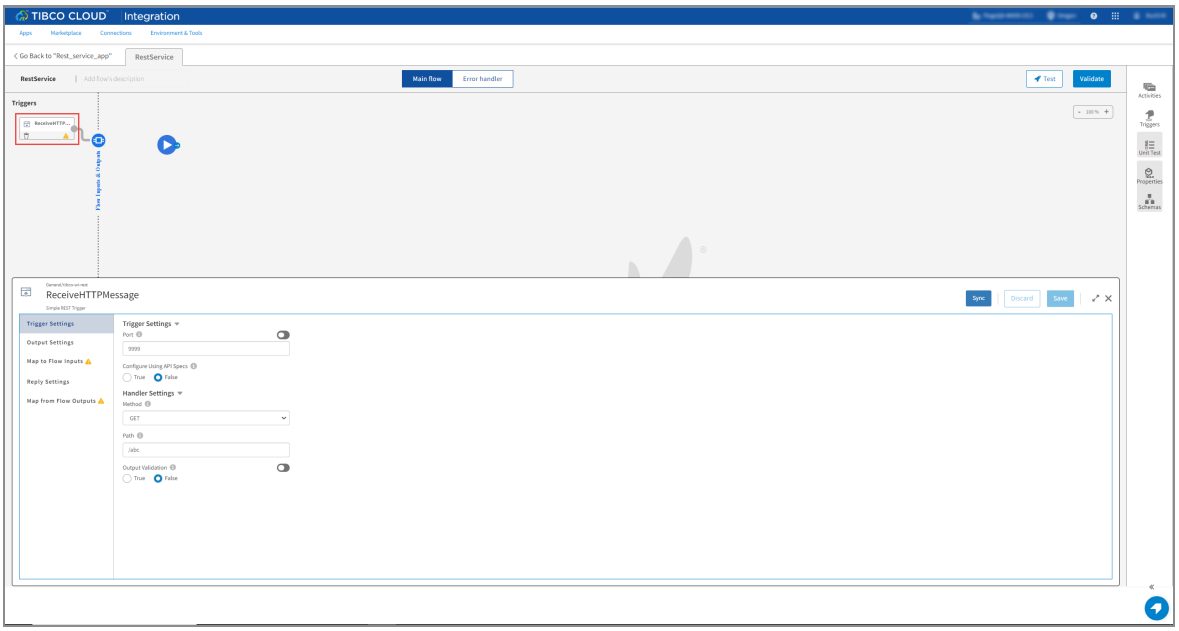
Use the mapper to enter the flow or Activity input values manually or map the input schema elements to output data of the same data type from preceding activities, triggers, or the flow itself. For more information, refer to the "Data Mappings" section of the *TIBCO Flogo® Enterprise User Guide*.

When a REST service is invoked, the data from the request is produced as output by the ReceiveHTTPMessage REST trigger. For the request to be processed, this output must be used by the flow in the form of flow input. Hence, you must map the trigger output to the flow input.

Mapping the trigger output to the flow input

Procedure

1. Click the REST trigger icon to open its configuration dialog.



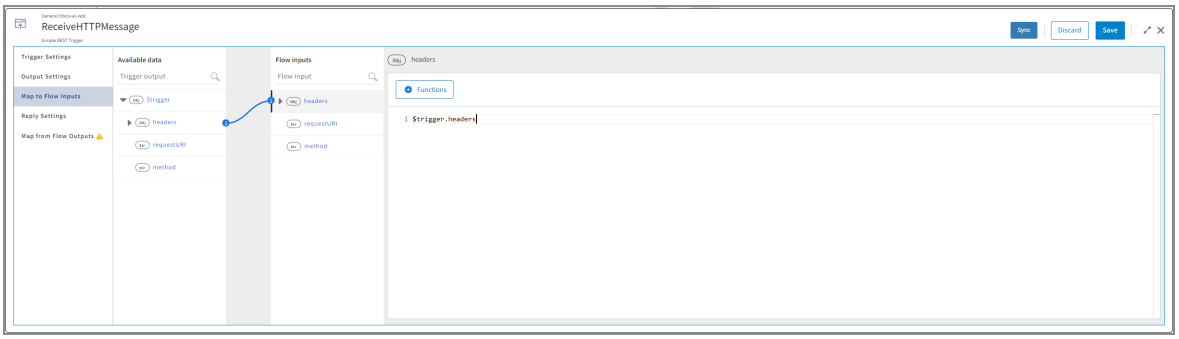
In the configuration dialog, multiple tabs are displayed in a column on the left. **Trigger Settings** is selected by default. When you select the **Map to Flow Inputs** tab, the **Available data** and **Flow inputs** panes are displayed. Flow inputs are the list of data entering the flow to define. The flow input value can be entered manually or by mapping the data available in the available data pane. The trigger outputs are also included in this pane.

2. In the **Flow inputs** column, select **headers** to start the mapping. The headers text editor on the right of flow inputs appears empty initially.

   To map the **trigger output headers** to the **flow input headers**:

   a. Drag headers from the Available data pane to the headers in the Flow inputs pane. Alternatively, click headers from the Flow inputs pane, then drag headers from the Available data pane, into the text editor.

   b. The text editor now displays **$trigger.headers** and a connection line between the two panes. This indicates that you have successfully mapped the trigger output headers to the flow input header. The numbers at the end of the connection line indicate the total number of mappings for the selected element.


ReceiveHTTPMessage
Sync
Discard
Save
Trigger Settings
Output Settings
Map to Flow Inputs
Reply Settings
Map from Flow Outputs
Available data
Trigger output
$trigger
headers
requestURI
method
Flow inputs
Flow input
headers
requestURI
method
headers
Functions
1 $trigger.headers

Step 5: Map flow output to trigger reply

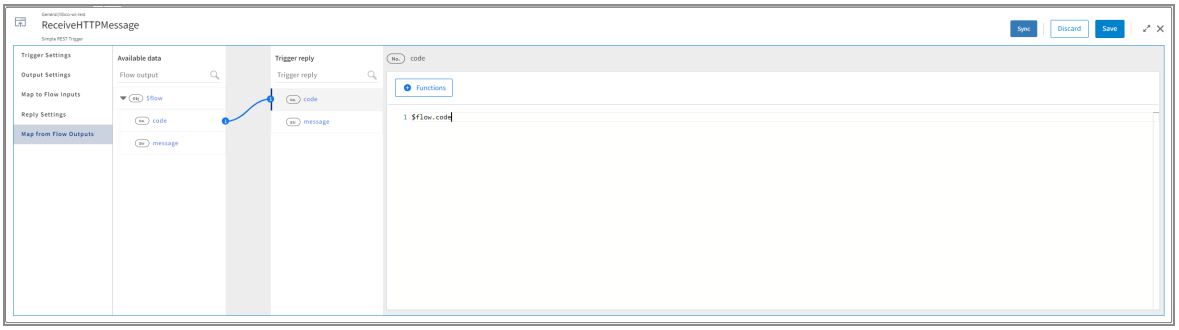
When the execution of the flow is complete, the output must be sent back to the trigger to send a reply to the REST request initiator. Hence, the flow output data must be mapped to the trigger reply. Then, it returns the result of the flow execution to the REST request initiator.

Mapping the flow output to the trigger reply

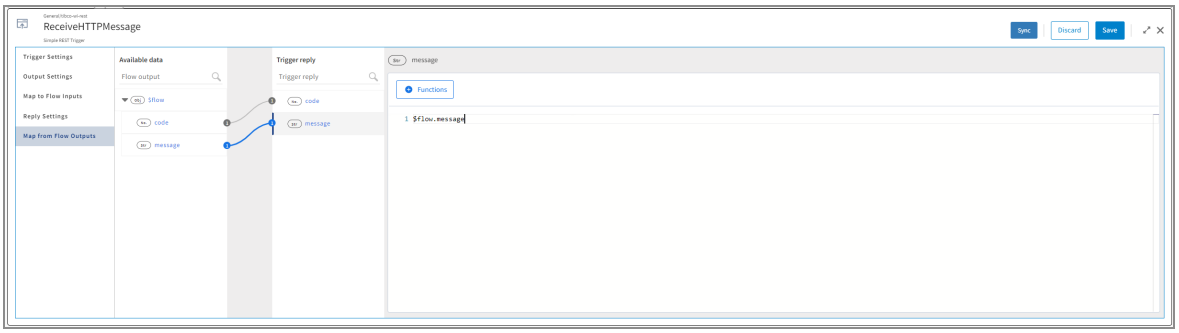
On the left pane, click the **Map from Flow Outputs** tab to configure the mapping of the **trigger reply** field. The Available data and Trigger reply panes are displayed. To see all the flow outputs available for the mapping, expand **$flow**. You can map the **code** and **data** trigger replies to the flow outputs.

Procedure

1. Drag **code** from **Available data** to the **code** in the **Trigger reply** pane. The **$flow.code** string appears in the code text editor, which means you have successfully mapped the code in Trigger reply to the code in Available data.



2. Similarly, map the **message** field from **Trigger reply** with the **message** field from **Available data**.



3. Click **Save** and close the trigger dialog. The trigger is now configured and activities can be added to this blank flow.

Step 6: Add invoke REST Service and Log Message Activity to the flow

Activities perform specific tasks within the flow. A flow typically contains multiple activities. Configure three simple activities such as:

- The Invoke REST Service Activity to invoke an external service,
- The Log Message Activity to register a message about the status of the first Activity,
- The Return Activity to produce the results of the flow that are run.

Tip: Each Activity in the Add Activity dialog realizes specific action in the flow and each action has a specific configuration. For more information on specific activities, refer to the "Activities" section of *TIBCO Flogo® Enterprise Activities, Triggers, and Connections*.

To invoke a rest service, configure the Activity and map the status of the same to the Log Message Activity.

Configuring Invoke Rest service Activity

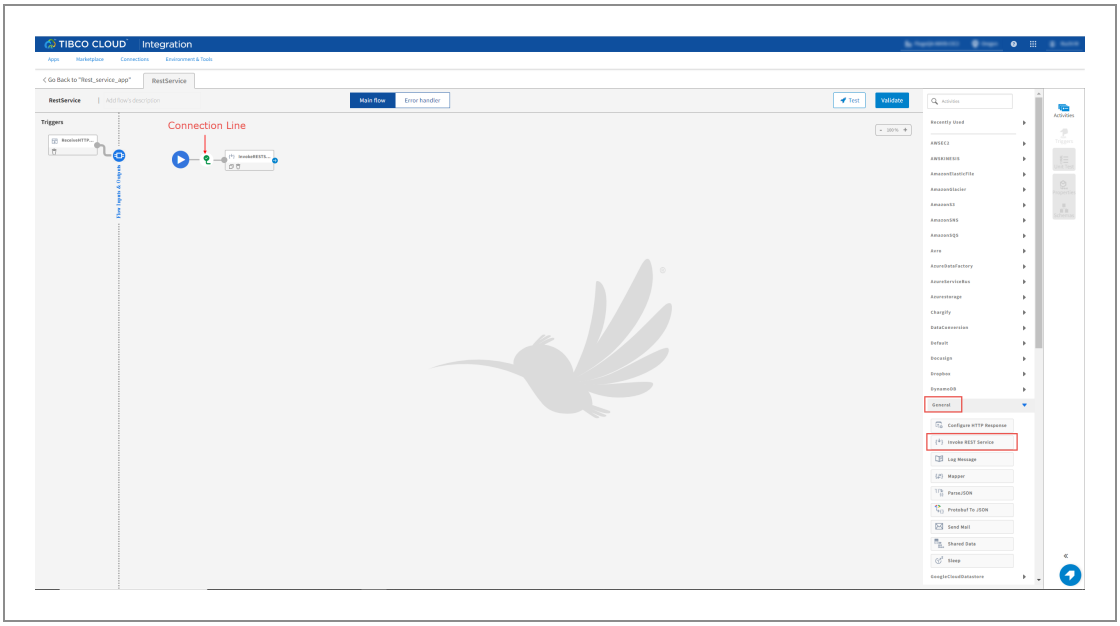
Procedure

1. On the right sidebar, click **Activities**.

   Note: The  represents the **StartActivity**. The subsequent Activities can be placed anywhere in the canvas.

2. You see the **Activity** palette open, that has a list of different Activites to choose from. Go to the **General** category and select the **Invoke REST Service** Activity from the group of Activities listed there. Drag this Activity from there to anywhere in the canvas. Drop a connection line between the **StartActivity** and the **Invoke REST**

Service Activity. This represents the order in which the Activities are performed.



Click the **Invoke Rest Service Activity** tile to open the configuration window. It opens with the Settings tab selected by default.

3. On the **Settings** page, in the **URL** field, provide the URL for the Rest API that is to be invoked. Then, click **Save** and close the configuration window. Here, the **URL** is set as `https://reqres.in/api/users/2`.

Adding a Log Message Activity

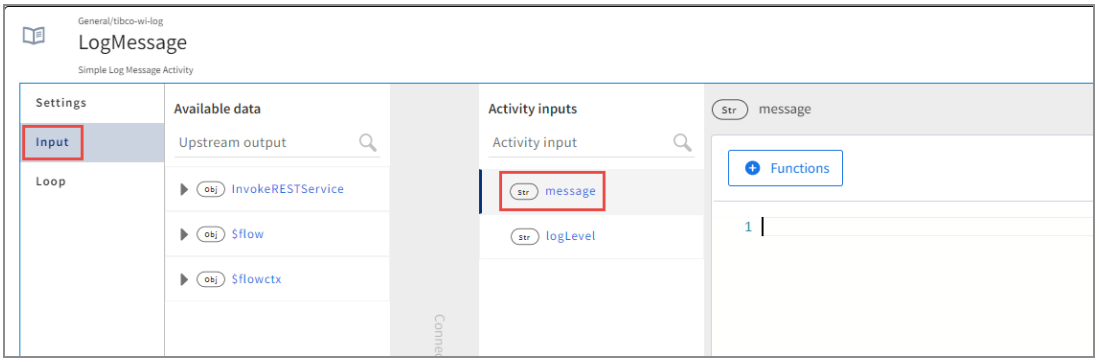
Procedure

1. Beside the Invoke REST Service Activity, we add the **LogMessage** Activity. To do so, go to the **Activity** palette and click the **General** category. A dropdown displays which shows the list of activities that are grouped under this tab. Drag the **LogMessage** Activity from there to anywhere in the canvas. Add the connection line between the two Activities.
2. Click the **LogMessage** Activity tile to display the configuration window. It opens with the Settings tab selected by default.
3. To configure the LogMessage Activity with a message to log when it receives an incoming request from the ReceiveHTTPMessage trigger about the status of the

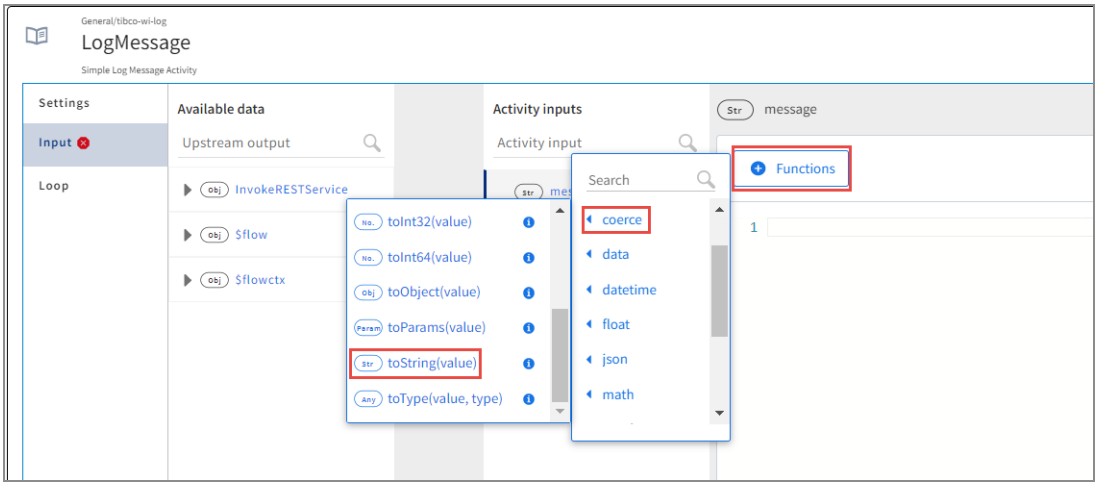
Invoke rest service:

A. Click the **Input** tab. The **Available data** and **Activity inputs** columns are displayed.

B. Select the **message** in the Activity Inputs column. Configure the message to print in the logs. This opens the text editor of the message.

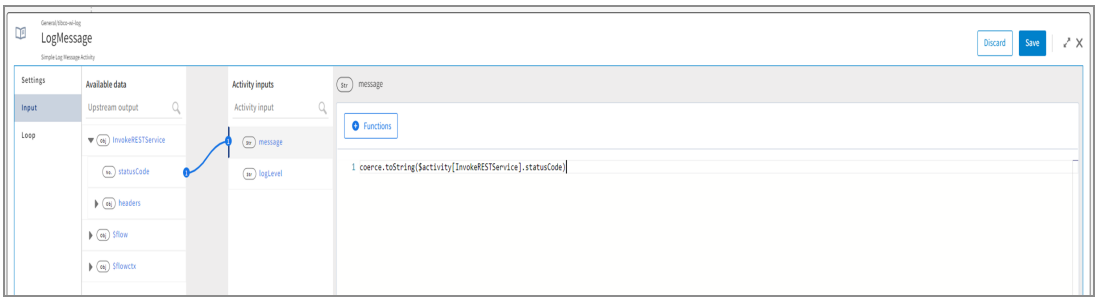


C. Configure this Log Message Activity to log the status code received from the **InvokeRESTService** Activity. To use a function in the editor, click **Functions**, expand the string and select **coerce.tostring(value).**



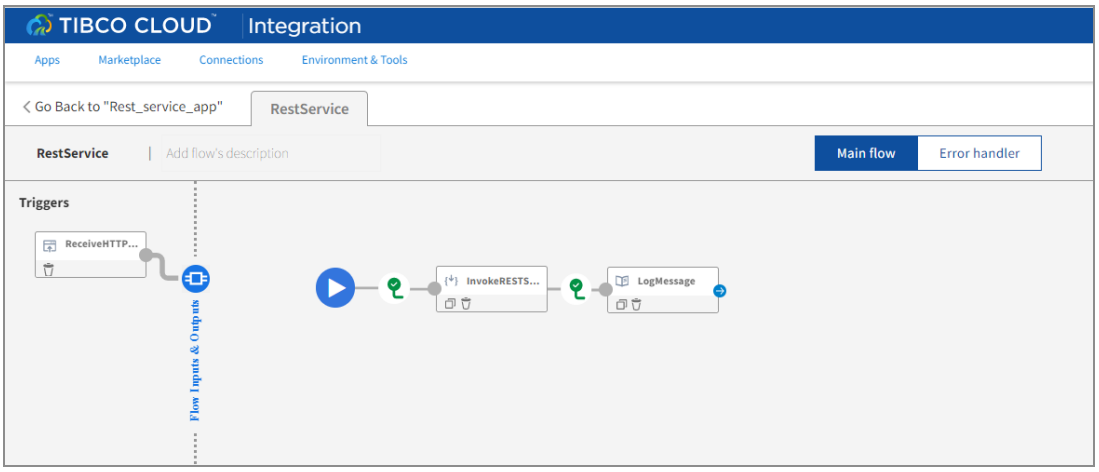
D. The newly added function appears in the editor with a default placeholder (value) as an argument. Replace the placeholder with the status code:

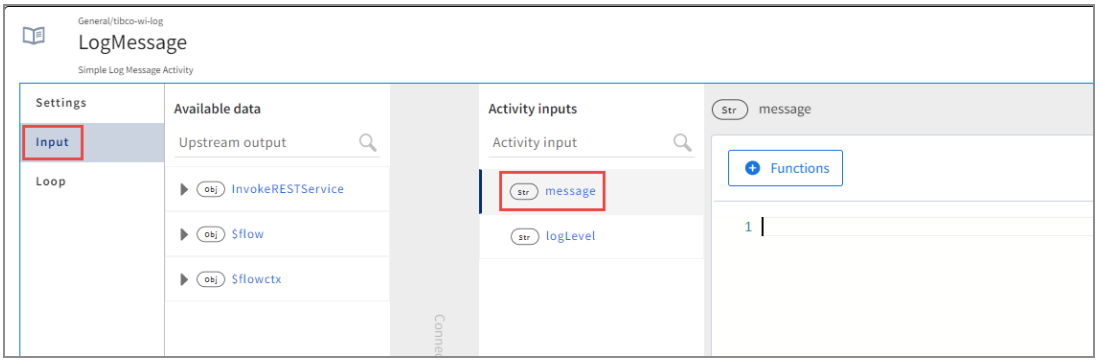
In the Available data pane, expand **InvokeRESTService**, drag **StatusCode** and drop it in place of **Value**, and click **Save**. Alternatively, you can also select the placeholder and double click on **StatusCode** in Available data and save it.



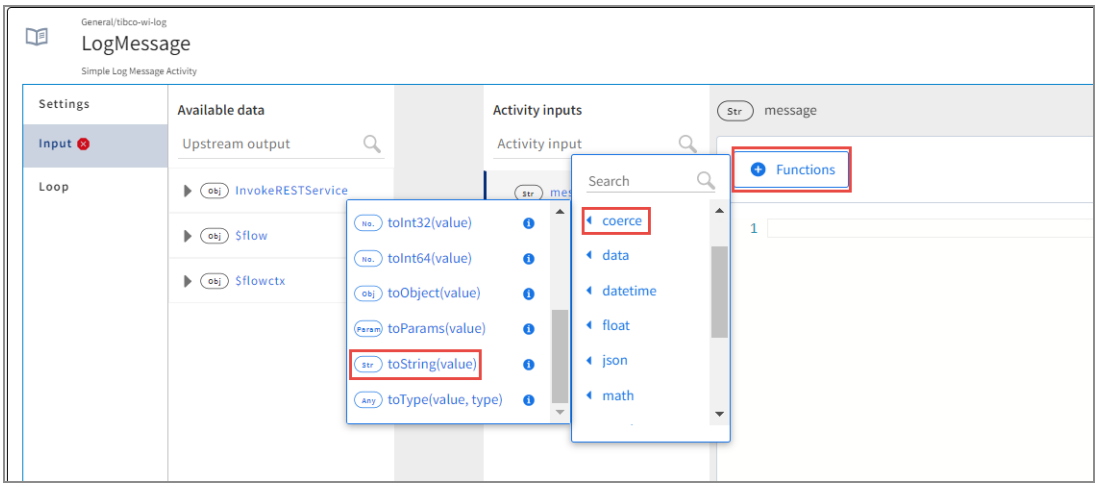
E. Close the **LogMessage** dialog box.

Now, the flow has two configured activities:



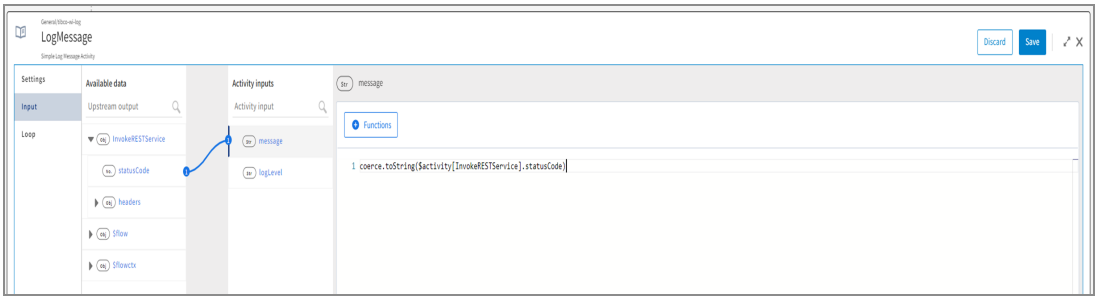


F. Configure this Log Message Activity to log the status code received from the **InvokeRESTService** Activity. To use a function in the editor, click **Functions**, expand the string and select **coerce.tostring(value).**



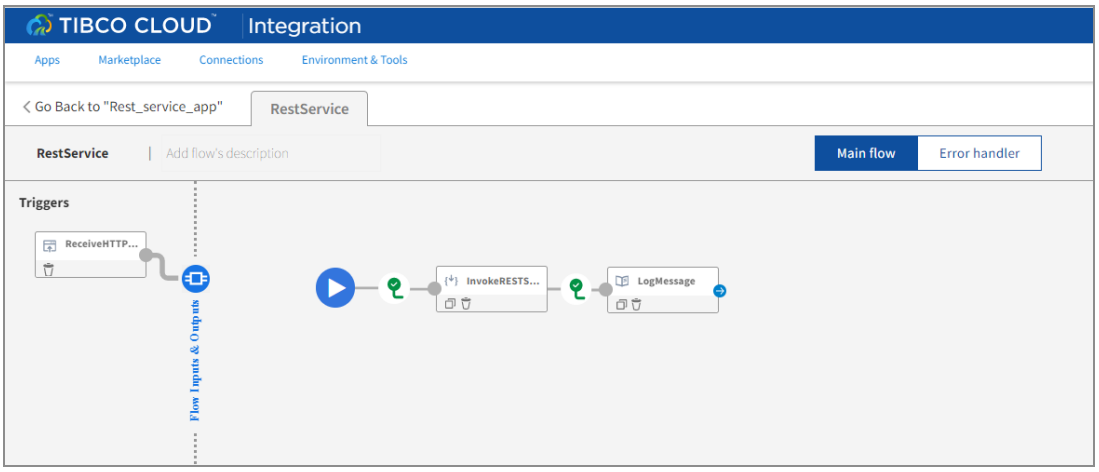
G. The newly added function appears in the editor with a default placeholder (value) as an argument. Replace the placeholder with the status code:

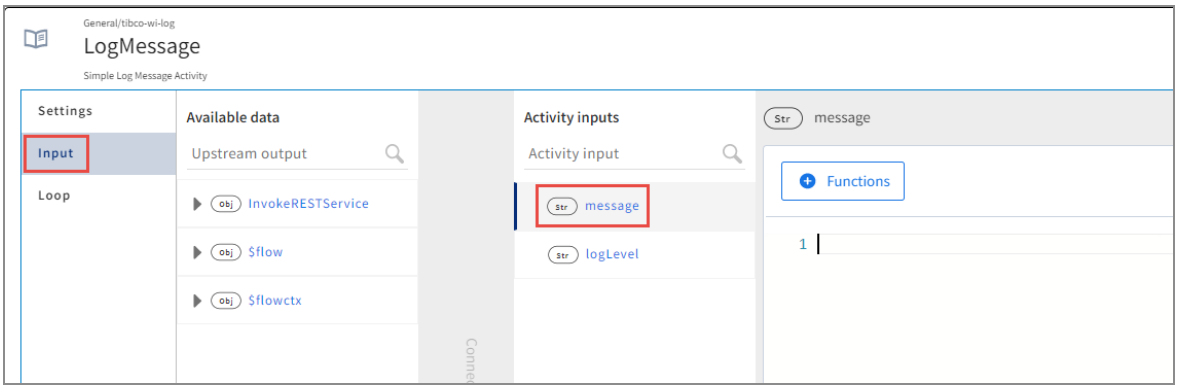
   In the Available data pane, expand **InvokeRESTService**, drag **StatusCode** and drop it in place of **Value**, and click **Save**. Alternatively, you can also select the placeholder and double click on **StatusCode** in Available data and save it.



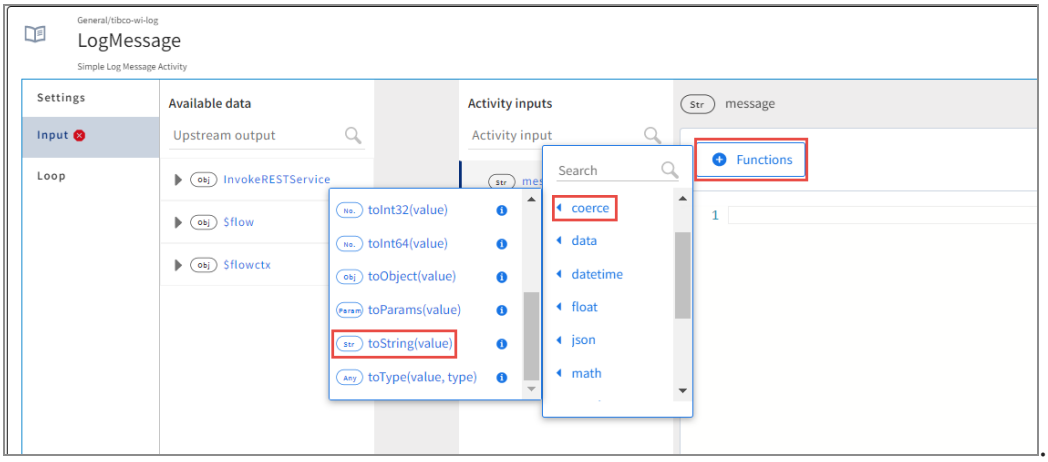
H. Close the **LogMessage** dialog box.

   Now, the flow has two configured activities:



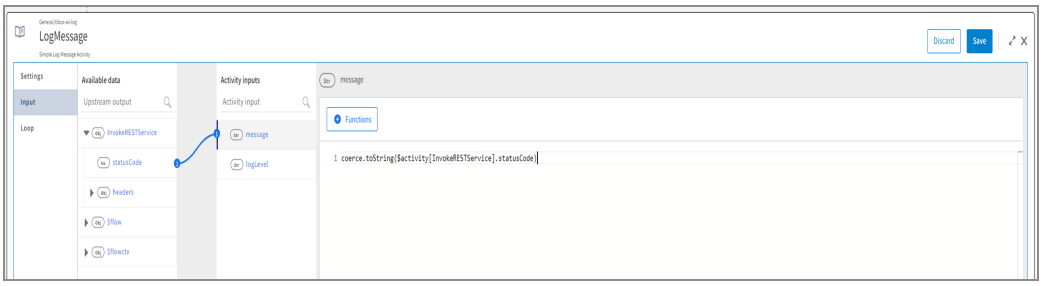


I. Configure this Log Message Activity to log the status code received from the **InvokeRESTService** Activity. To use a function in the editor, click **Functions**, expand the string and select **coerce.tostring(value)**.



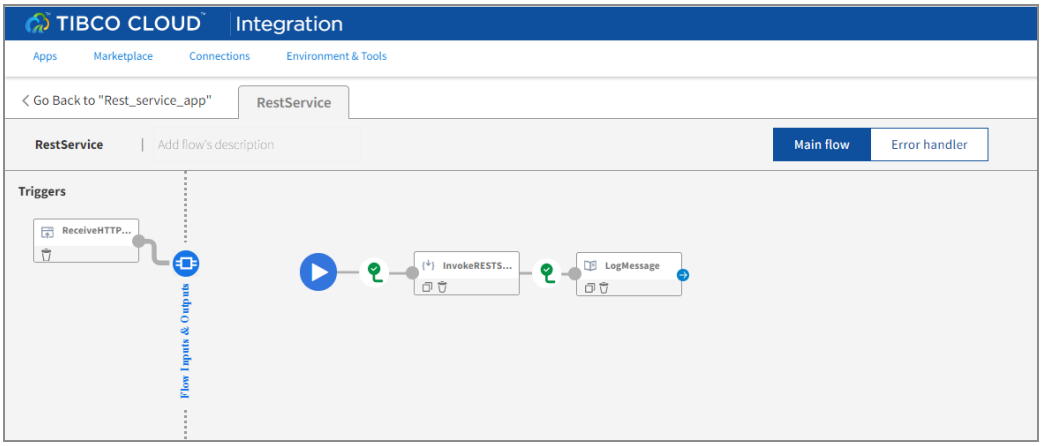
J. The newly added function appears in the editor with a default placeholder (value) as an argument. Replace the placeholder with the status code:

   In the Available data pane, expand **InvokeRESTService**, drag **StatusCode** and drop it in place of **Value**, and click **Save**. Alternatively, you can also select the placeholder and double click on **StatusCode** in Available data and save it.



K. Close the **LogMessage** dialog box.

   Now, the flow has two configured activities:



Step 7: Configure a Return Activity to display the status of the trigger

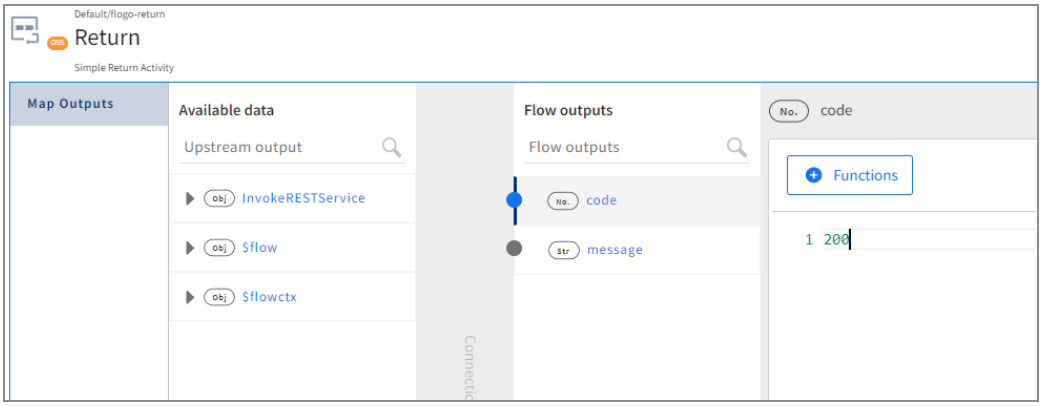
Adding a Return Activity

Procedure

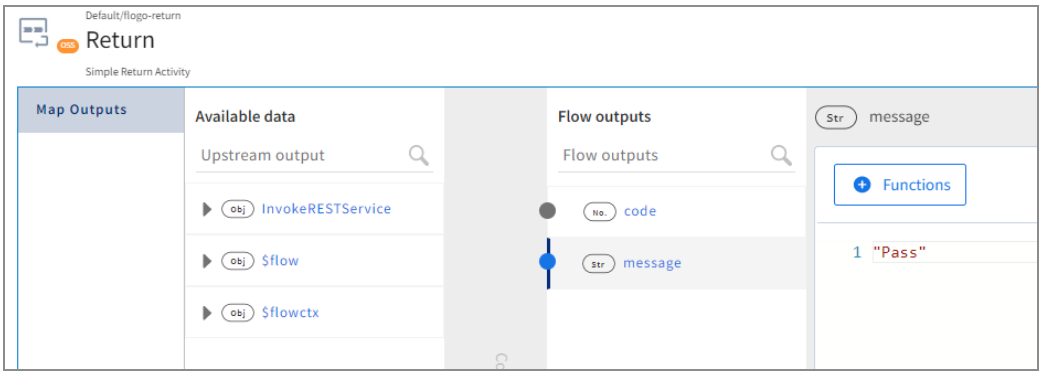
1. Open the **Activities** palette. Go to the **Default** category and select the **Return** Activity from the group of Activities listed there. Drag this Activity from there to anywhere in the canvas. Drop a connection line between the **LogMessage** and the **Return** Activities.
2. Click the **Return** Activity tile to open the configuration window. Configure the **Return** Activity to produce the flow results.

   To configure the Return Activity:

   A. Click **code** under **Flow outputs** to open the mapper and type `200` in the editor manually to assign a constant value to **code** as an HTTP success code.



   B. Select the **message** in the **Flow Outputs** column manually to assign a constant value. Type '**Pass**' which states the output of the flow result on successful execution.



C. Click **Save** and then close the dialog.

Tip: Flow execution branch as a business logic can be more complex by using conditional branching, loops, and conditional mapping to run actions based on conditions.

For more information on flow execution branch settings, refer to the "Creating a Flow Execution Branch", "Using the Loop Feature in an Activity", and "Mapping Data by Using `if/else` Conditions" sections of *TIBCO Flogo® Enterprise User Guide*. After you design a flow, use the Flow Tester to test the flow.

Step 8: Validate and test the app

The app is now ready. Before you build the app or push the app to the cloud, validate all the flows to confirm that there are no errors or warnings. To do so, click **Validate**. For more information, see the "Validating your App" section in the *TIBCO Flogo® Enterprise User Guide*.

Flogo validates each flow and Activity within the flow. If there are any errors or warnings, you see the respective icons next to the flow name or Activity tab that contains the error or warning.

A success message is displayed after successful validation.

Further, to test the configured app, see the "Testing Flows from the CLI" section in *TIBCO Flogo® Enterprise User Guide*.

Step 9: Build the App Binary or Push the app to the TIBCO Cloud

After you have created your app, you can build it anytime. When you build the app, its deployable artifact gets created. You can download it to your local machine. Each operating system has its build target. Select the right target for your operating system when building the app. You can use the built artifact to run the app. The app binary can be also used for container deployment or serverless deployment. For more information, see the "Container Deployments" and "Serverless Deployments" sections in the *TIBCO Flogo® Enterprise User Guide*.

Building the app binary

Before you begin

Make sure you have the app for which an app executable needs to be created must have a trigger and a flow in it. If the app does not have a trigger and flow, the app executable is not created.

Procedure

1. Open the **Apps** page.
2. Click the app for which you want to build an app executable. The page for the selected app opens.
3. On the page that opens, click **Validate** and resolve errors if any.
4. Open the shortcut ⋮ menu, click **Build app**, and select a build target option that is compatible with your operating system (such as Darwin/amd64 for Macintosh).

> **Note:** If you have created or pushed an app using tibcli or platform API, the **Build App** option is not displayed as the apps are read-only.

The following build target options are available:

- Macintosh: **Darwin/amd64**
- 64-bit Linux: **Linux/amd64**
- 32-bit Linux: **Linux/x86**
- Microsoft Windows: **Windows/amd64**

The app begins to build. When it is built, the deployable artifact is downloaded to your local machine.

5. To confirm whether the app executable is built successfully, go to the **History** tab and check whether **Action** is displayed as **APP BUILD**.

Running the App

On Macintosh and Linux

1. Open a terminal.
2. Run:

   ```
   chmod +x <app-file-name>
   ```

3. Run:

   ```
   ./<app-file-name>
   ```

On Microsoft Windows

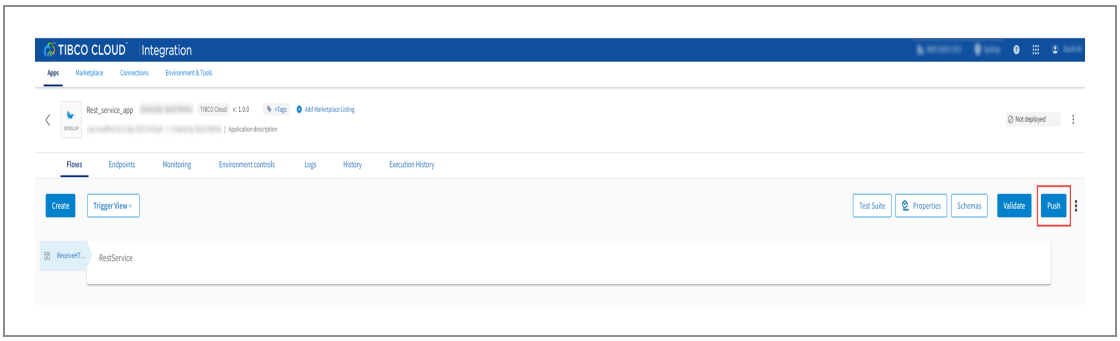
At the command prompt, run:

```
app-file-name.exe
```

The other way to use the built app is to push it to TIBCO Cloud.

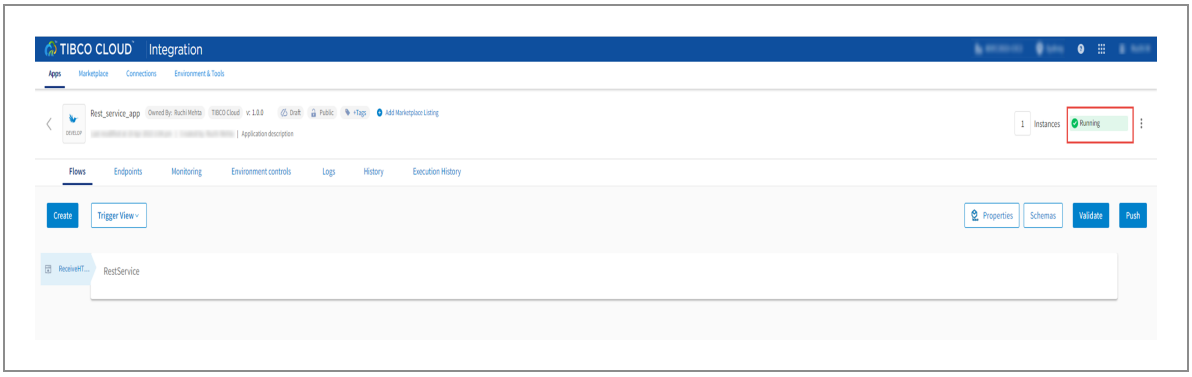
Pushing your app to TIBCO Cloud

Click the back arrow on the top-left corner or click the **Apps** tab to open the app details page and then, click **Push**.



After the app has finished deploying, you see the following status:

The app is in the **stopped** state as there are no instances of the app running. To run the app, hover over the zero to display the up and down arrows above and below it. Click the up arrow above the zero such that the zero turns to 1, and click **Scale**. You see the running status once the instance starts.



TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for TIBCO Flogo® Enterprise is available on the TIBCO Flogo® Enterprise Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.