



TIBCO Flogo[®] Enterprise

Activities, Triggers, and Connections

Version 2.25.2 | January 2025



Contents

Contents	2
General Category Triggers, Activities, and Connections	4
Triggers	4
Timer Trigger	5
REST Trigger - ReceiveHTTPMessage	8
GraphQL Trigger	15
Receive Lambda Invocation	18
AWS API Gateway Lambda Trigger	19
S3 Bucket Event Lambda Trigger	23
App Startup Trigger	25
App Shutdown Trigger	26
Activities	27
ConfigureHTTPResponse	27
Run JavaScript	29
GRPC Invoke	32
InvokeLambdaFunction	34
InvokeRETSservice	36
LogMessage	47
Mapper	49
ParseJSON	50
ReplyToTrigger	51
ProtobufToJSON	51
No-Op	52
ReplyToHTTPMessage	53
SendMail	54
SharedData	56
Sleep	58

JSONToXML	60
XMLToJSON	60
Connections	62
HTTP Client Authorization Configuration	62
TIBCO Documentation and Support Services	66
Legal and Third-Party Notices	68

General Category Triggers, Activities, and Connections

The **General** category is available by default in all flows. It consists of activities, triggers, and connections that are commonly used by any flow in the app. A trigger initiates the flow in which it is used. Activity is used to perform a task. A connection is used to connect an app to various services.

Triggers

In addition to the triggers available for general use, Flogo supports triggers that are originally created in Project Flogo™. These triggers are marked with an OSS tag, enabling seamless import of apps created in Project Flogo.

If you are creating an app in Flogo, it is preferable to use the general-purpose triggers (the triggers that do not have an OSS tag on them) as they have richer functionality.

Refer to <https://github.com/project-flogo/contrib> for details on the triggers that are marked with an OSS tag.

Trigger configuration fields are grouped into **Trigger Settings** and **Handler Settings**. A single trigger can be associated with multiple handlers.

- **Trigger Settings** - these settings are common to the trigger across all flows that use that trigger. When **Trigger Settings** are changed, the change applies to all flows that are attached to the trigger. A warning message is displayed asking you to confirm the changes before they are committed.
- **Handler Settings** - these settings apply to a specific flow attached to the trigger. Hence, each flow can set its values for the **Handler Settings** fields in the trigger. To do so, open the flow and click on the trigger to open its configuration dialog box. Click the **Settings** tab and edit the fields in the **Handler Settings** section.

i Note:

- You cannot create a flow branch from a trigger.
- You can create the trigger at the time of flow creation or create a blank flow to begin with and attach the flow to one or more triggers at a later time after the flow has been created. If you anticipate that you might need to attach the flow to multiple triggers, be sure to create a blank flow and attach it to the triggers as needed.

For triggers that have an output, the output from the trigger becomes the input to the flow. Likewise, the output from the flow becomes the reply from the trigger.

When using the Lambda, S3, or Gateway triggers, keep the following in mind:

- You can only have one trigger. The Lambda trigger supports only one handler per trigger, it can have only one flow attached to it. The S3 and Gateway triggers support multiple handlers (flows), so you can have multiple flows in the app that are attached to the same S3 or Gateway trigger.
- An app that has one of these triggers cannot contain any other trigger.
- You can also have blank flows in the app, which can serve as subflows for the flows that are attached to one of these triggers.

Timer Trigger

Use the **Timer Trigger** as a process starter when creating flows designed to be activated without external input. It is useful when you want your flow to run at certain time intervals. You can also configure the **Timer Trigger** to activate the flow multiple times at a specified interval or to set a **Cron Job**.

Trigger Settings

Field	Description
Handler Settings	
Scheduler	<ul style="list-style-type: none"> • Timer: Runs the flow at the specified time or interval

Field	Description
Options	<ul style="list-style-type: none"> • Cron Job: Provides more customization options for scheduling the runs
Additional information about fields available when you select the scheduler option as Timer	
Start Time	<p>Use the calendar to set start date and time. When configuring an app property for the Start Time, use the RFC 3339 date and time formats.</p> <p>Format with UTC offset:</p> <p>YYYY-MM-DDTHH:MM:SS+00:00</p> <p>OR</p> <p>Format without UTC offset:</p> <p>YYYY-MM-DDTHH:MM:SSZ</p> <p>Here,</p> <p>T is used as a separator between date (YYYY-MM-DD) and time (HH:MM:SS). Replace it with a space, if needed.</p> <ul style="list-style-type: none"> • 2021-04-12T23:20:50.52+00:00 • 2021-11-22T09:16:47Z • 1996-12-19T16:39:57-08:00 • 1990-12-31 23:59:60+05:00 <p>Note: Start Time by Default is Blank. If there is no date and time mentioned, the flow runs as soon as you push the changes or app binary.</p>
Use these fields in combination to define the schedule for recurring runs.	
Repeating	When Repeating is enabled, the flow is triggered at the same time as the first run at the specified time interval. (Default: False)
Delayed Start	Delayed Start delays only the first execution. The successive runs are triggered at the specified time interval. (Default: False)

Field	Description
Time Interval	This is a period between two successive runs.
Interval Unit	This is a unit specified for Time Interval .

Additional information about fields available when you select the scheduler option as **Cron Job**

Cron Expression You can enter any **Cron Expression** manually in this field.

The standard cron expression should be **CRON_TZ=IANA TimeZone * * * * ***

Here,

CRON_TZ=IANA TimeZone is used to set the specific time zone in which the cron job is run.

*	*	*	*	*
Minutes	Hours	Day of Month	Months	Day of Week

Here are some examples for you.

Cron expression	Schedule
* * * * *	Every minute
0 * * * *	Every hour
0 0 * * *	Every day at 12:00 AM
0 0 * * 4	At 12:00 AM, only on Friday
	Here, number 0-6 stands for the days of the week starting with Sunday as 0.
0 0 1 * *	At 12:00 AM, on day 1 of the month

Field	Description
Cron Expression Builder	<p>If you use the Cron Expression Builder, this field is auto-populated as you build the expression.</p> <p>Simple cron expressions can be built using the Cron Expression Builder. You can use the different tabs to define the frequency: Minutes, Hours, Day of Week, Day of Month, Months.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note:</p> <ul style="list-style-type: none"> • Expressions built using Cron Expression Builder can be modified manually. • If there is no time zone mentioned while building a cron expression through app property, by default the time zone of the system is considered and there might be no logs generated for the executed flow. • UI validation for the built cron expression is not supported. </div>

Map to Flow Inputs

Use this tab to map the trigger output to the flow input.

REST Trigger - ReceiveHTTPMessage

Use the **ReceiveHTTPMessage** REST trigger when creating flows that are going to be invoked by an external REST call. The **ReceiveHTTPMessage** trigger exposes your flow as an API, making it accessible to other apps running on TIBCO Cloud™ or elsewhere. This trigger must be configured to set up the fields for a request that the server receives from a REST client.

Note:

- If you add or delete path or query parameters in the trigger, you must click **Sync** for the changes to be propagated to the flow input schema
- REST trigger does not support authentication and authorization headers.


Trigger Settings



Field	Description
Trigger Settings	
Port	<p>By default, the trigger listens on port 9999. You can change this to use another open port.</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p>Important: If the app has multiple triggers that require a port to be specified, specify a unique port number for each trigger. Two triggers in the same app cannot run on the same port.</p> </div>
Configure Using API Specs	<p>While creating a REST trigger, you can configure it by uploading an API specification file.</p> <p>To do this, click True (By default, it is set to False) and specify the following:</p> <p>API Spec: Click Browse and then select the specification file to be used for configuring the trigger. Supported specifications are Swagger Specification 2.0 and OpenAPI Specification 3.0.</p>
Secure Connection	<p>By default, it is set to False. If you set this field to True, you can create a secure endpoint by providing Server Key and CA or Server Certificate.</p> <p>Server Key - A PEM encoded private key file</p> <p>CA or Server Certificate - A PEM encoded CA or self-signed server certificate file</p>
Handler Settings	
Path	<p>The resource path for the operation.</p> <p>If you upload an API specification file, select a path from the drop-down list. The path parameters are parsed from the API spec file and the data types displayed are string, integer, or boolean as specified in the file.</p> <p>For manual configuration, the data type for resource path is string.</p> <p>By default, the path displayed here is the resource path you had entered</p>

Field	Description
	<p>when you created the flow. The Path field is editable if you have not uploaded an API specification file. For example, if you want to add a path parameter for a GET operation, you can do so by editing the resource path in the GET flow. If you edit the path in the Path field for a particular REST operation flow, the edited resource path applies only to the flow in which it was edited.</p> <p>Two resource paths with the same base path should not contain the path parameters at the same location. For example, you cannot have the following paths in the same app:</p> <ul style="list-style-type: none"> • <code>/books/{ISBN}/Author/{releaseDate}</code> and <code>/books/{ISBN}/Author/releaseDate</code> is considered the same from a routing perspective. <p>In these two paths, the dynamic ISBN value causes a conflict during path resolution.</p> <ul style="list-style-type: none"> • <code>/books/{ISBN}/{releaseDate}</code> and <code>/books/{ISBN}/Author</code> in the same app is not supported. <p>Although the two paths seem different, when a message comes in, the router mechanism cannot determine which path to call (the one with the parameter or the one without) because the actual value has been substituted for the parameter.</p> <ul style="list-style-type: none"> • Resource path with two different path parameters at the same URL subsection. For example, <code>/0.6/api/account/{account}/orderhistory/{orderhistory}/branch/{branch}</code> and <code>/0.6/api/account/{AccountKey}/Price?ProductList={ProductList}</code> <p>In these paths even though the paths differ after the base path (<code>/0.6/api/account/</code>), there is a conflict when resolving the <code>{account}</code> and <code>{AccountKey}</code> values.</p> <ul style="list-style-type: none"> • Multiple REST resources with the same base path and the same number of path parameters. For example, <code>/resource/{id}</code> and <code>/resource/{id1}</code> • <code>/messages/{messageid}/comments/{commentid}</code> and <code>/messages/{messageid}/likes/{likeid}</code>

Field	Description
	where the paths differ after {messageid}.
Method	The REST operation which the flow implements. Supported HTTP methods are: GET, PUT, POST, DELETE, and PATCH.
Request Type	<p>This field is displayed only when Method is POST, PUT, or PATCH. Select one of the following types from the dropdown list:</p> <ul style="list-style-type: none"> • application/json • application/x-www-form-urlencoded • multipart/form-data <p>Default: application/json</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note: If you create a Flogo app using an API specification file having Request Type as application/x-www-form-urlencoded or multipart/form-data, then you must click Sync to update the request parameters with the Flow Input.</p> </div>
Output Validation	<p>When set to True, the incoming data (body, headers, and query parameters) is validated against the configured JSON schema.</p> <p>Default: False</p>

Output Settings

Field	Description
Query Parameters	<p>Query parameters to be appended to the path. To add the query parameters, click  and press Enter to save your changes.</p> <ul style="list-style-type: none"> • parameterName: Name of the query parameter. • type: The data type of the query parameter. Supported types are string, number, and boolean. • repeating: Set to True if more than one value is expected for the


Field	Description
	<p>query parameter.</p> <ul style="list-style-type: none"> • required: Set to True if the query parameter is a required configuration. The trigger reports an error if no values are provided to the required query parameter.
Path Parameters	Path parameters that are appended to the path.
Headers	<p>Header values for the trigger. To add the header parameters, click  and press Enter to save your changes.</p> <ul style="list-style-type: none"> • parameterName: Name of the header parameter. • type: The data type of the header parameter. Supported types are string, number, and boolean. • repeating: Set to True if more than one value is expected for the HTTP header. • required: Set to True if the header parameter is a required configuration. The trigger reports an error if no values are provided to the required header parameter.
Request Schema	<p>Enter a request schema here. This field is visible only if you selected the POST, PUT, PATCH, or DELETE method on the Settings tab.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note:</p> <ul style="list-style-type: none"> • For the DELETE method, specifying the request schema here is supported for manual configuration only and not when configured with Tibco Cloud Mesh and API specification. • If you selected <code>application/x-www-form-urlencoded</code> as the Request Type on the Settings tab, the default schema is set here. You can edit the default schema or specify your schema. If you specify your schema, it must be a name-value string pair. </div>
Multipart Data	<p>This field is displayed in place of the Request Schema field if you select <code>multipart/form-data</code> as the Request Type in the Trigger Settings. Click the Add row  button to add the parameters.</p>

Field	Description
	<ul style="list-style-type: none"> • Name: Parameter name. • Type: Supported types are <code>string</code>, <code>object</code>, and <code>filecontent</code>. • Required: Check the box if the parameter is a required configuration. • Schema: Enter the JSON schema in this field if the Type is an <code>object</code>. <p>Note: The file content received by the trigger is converted into a byte array. This is passed to the Activity as a Base64-encoded value in an array. If you want to fetch the content of the file, coerce the first element of the array to <code>string</code>.</p>

Map to Flow Inputs

Use this tab to map the trigger output to the flow input.

Reply Settings

Field	Description
Configure Response Codes	<p>Use this field to configure response codes.</p> <p>Default: False (See "Reply Data Schema" in this table.)</p> <p>To specify a response code, select True and click . Enter the following details:</p> <ul style="list-style-type: none"> • Code: Enter the response code. • Type: Select the type of response expected for the Code. Supported types are String and Object. • Response Body: If the Object is selected as the Type, enter the JSON schema in the Response Body column. For String, you need not enter anything in the Response Body column. • Response Headers: The header parameters for the reply are in JSON data format.

Field	Description
-------	-------------

- **Actions:** The actions displayed change based on the type of the response code.
 - **Edit, Delete:** For an **Object** type of response, you can edit the details or cancel it.
 - **Save, Cancel:** For a **String** type of response, you can save or cancel the changes.

The response codes is displayed on the Map from the Flow Outputs tab.

Note: The REST reply data type is by default set to data type any. To configure the reply to an explicit data type, see the "Configuring the REST Reply" section in the *TIBCO Flogo® Enterprise User Guide*.

For multiple response codes, use the **ConfigureHTTPResponse** Activity in the flow to map **Response Body** and **Response Headers** from the REST trigger with the **Input** in the Activity. To configure **ConfigureHTTPResponse** Activity, see [ConfigureHTTPResponse](#).

The image shows the **Reply Settings** with multiple response codes.



Caution: If you modify the response code schema in the table, the corresponding **ConfigureHTTPResponse** activities within that flow do not change appropriately. This happens specifically when removing fields from the **Reply Settings** tab. Redo the mappings for the **ConfigureHTTPResponse** activities.

Reply Data Schema

Note: This field is displayed only when **Configure Response Codes** is set to **False**.

The schema is used for the reply data of the trigger. Be sure to use straight quotes for element names and values in the schema.

Map from Flow Outputs

Map the flow output to the trigger reply on this tab.

Note:

- To update these settings for a trigger configured from the Swagger 2.0 or OpenAPI 3.0 specification, update the API specification file and upload it to the **Trigger Settings**. Do not update the settings as manual updates are removed.
- If you are using a REST trigger in your app:
The endpoint URL contains the app name only if the app has one trigger.
The endpoint URL contains the app name and trigger name for more than one trigger.

If you add a REST trigger to an existing app, you must reconfigure the client app.

GraphQL Trigger

The **GraphQL Trigger** lets the Flogo app act as the GraphQL server. To use this trigger, you simply upload your GraphQL schema and Flogo Enterprise automatically creates the flows corresponding to each query or mutation field in your schema.

Trigger Settings

Field	Description
Trigger Settings	
Port	The port on which the trigger listens to requests. By default, it is set to 7879. You can change this to use any other open port. This field can also be set using an app property.
	Important: If the app has multiple triggers that require a port to be specified, make sure that the port number is unique for each trigger. Two triggers in the same app cannot run on the same port.

Field	Description
Path	The HTTP resource path for the operation. By default, it is set to <code>/graphql</code> , but you can change it to any string that is meaningful to you. It is the single endpoint that GraphQL queries and mutations use to access data from the multiple resources on the server. This field can also be set using an app property.
GraphQL Schema File	The file that contains the GraphQL schema is used to create the flow. The file has a <code>.gql</code> or <code>.graphql</code> extension. <p>Note: If you have made changes to the GraphQL schema file that you uploaded when creating the flow or trigger, you must propagate the changes to the flow input and flow output. To do this, after you select the updated schema file in this field, click Sync on the top right corner.</p>
Schema Introspection	By default, it is set to True . You can use it to get the schema field details for GraphQL. If you set it to False , it disables introspection and you cannot fetch the schema field details.
Secure Connection	By default, it is set to False . If you set this field to True , you can create a secure endpoint by providing Server Key and CA or Server Certificate . <p>Server Key - A PEM encoded private key file</p> <p>CA or Server Certificate - A PEM encoded CA or self-signed server certificate file</p>
Handler Settings	
GraphQL Operation	The type of GraphQL operation the flow should represent. You can select either Query or Mutation
Resolver for	This field is populated based on the type of GraphQL Operation that you selected. If you selected Query , the Resolver For lists the field names under the query type in the schema. If you select Mutation , the drop-down menu lists the field names under the mutation type in the schema.

Output Settings

Field	Description
Headers	You can define the JSON schema of the key value pair for Headers . This field is available inside the flow.

Note: **Headers** values are parsed as a string.

Map to Flow Inputs

You can map the trigger output to flow input on this tab. The tab contains an element, arguments, which contains a list of fields or objects that match the input arguments of the **Resolver For** field in the GraphQL schema. **Fields** and **Headers** are available to map inside **Map to Flow Inputs**. **Fields** contains the fieldName string and field array passed for the given **Query** or **Mutation**. The following structure is an example of **Fields**:

```
{"fieldName":"employee","fields":[{"fieldName":"id"}, {"fieldName":"name"}]}
```

Map from Flow Outputs

You can map the flow output to the trigger reply on this tab. The tab contains a child element, data, which contains either a simple type or an object that matches the output type of the **Resolver For** field in the GraphQL schema. If the output type of the field is an interface type, the data contains a single field of type any. You can use the **Error** field to map the trigger reply setting for any error in the flow. Its value is parsed as a string.

i **Note:** To avoid any runtime exception, you can define your data and error mapping using the following formats:

- Data: `isDefined($flow.data) ? $flow.data : coerce.toObject('{}')`
- Error: `isDefined($flow.error) ? $flow.error : ''`

Receive Lambda Invocation

Use the **Receive Lambda Invocation** trigger for AWS to start your flow as a Lambda function. The **Receive Lambda Invocation** trigger can be configured only in blank flows. It must not be used with flows that are created with another trigger.

Trigger Settings



Note:

- An app can contain only one Lambda trigger. An app that has a Lambda trigger cannot contain any other triggers including another Lambda trigger. Also, as the Lambda trigger supports only one handler per trigger, it can have only one flow attached to it. However, the apps that contain a Lambda trigger can contain blank flows that can serve as subflows for the flow attached to the Lambda trigger.
- For overriding app properties, use the FLOGO_APP_PROPS_JSON environment variable only. You cannot override app properties using the FLOGO_APP_PROPS_ENV environment variable.

Field	Description
AWS Connection Name	Name of the AWS connector connection you want to use for the flow.
Execution Role Name	(optional) ARN of the role to be used to execute the function on your behalf. The role must be assumable by Lambda and must have CloudWatch logs permission execution role.

Output Settings

Enter the payload schema for the request received on the Lambda function invocation on AWS.

Map to Flow Inputs

Map the trigger output to the flow input on this tab.

Field	Description
Function	Information about the Lambda function
Context	Envelope information about this invocation
Identity	Identity for the invoking users
ClientApp	Metadata about the calling app
API Gateway Request	Displays the elements in the payload schema that you entered on the Output Settings tab. The elements are displayed in a tree format.

Reply Settings

Enter a schema for the trigger reply in the Reply Data text box.

Map from Flow Outputs

Map the flow output to the trigger reply on this tab.

AWS API Gateway Lambda Trigger

Use the **AWS API Gateway Lambda** trigger to start your flow as a Lambda function using an API Gateway.

Trigger Settings


Note:


- For overriding app properties, use the FLOGO_APP_PROPS_JSON environment variable only. You cannot override app properties using the FLOGO_APP_PROPS_ENV environment variable.
- You can have only one Gateway trigger in an app. An app that has a Gateway trigger cannot contain any other triggers including another Gateway trigger. The Gateway trigger supports multiple handlers (flows), so you can have multiple flows in the app that are attached to the same Gateway trigger. You can also have blank flows in the app, which can serve as subflows for the flows that are attached to the Gateway trigger.

Field	Description
AWS Connection Name	(Mandatory) Name of the AWS connection that you want to use for deploying the flow as a Lambda connection.
Execution Role Name	<p>Permission of the Lambda function to execute. The role must be assumable by Lambda and must have CloudWatch logs permission execution role. A default role with the AWS predefined <code>AWSLambdaBasicExecutionRole</code> permission is created if you do not specify anything in this field.</p> <p>By default, Cloud watching is enabled.</p>
API Deploy Stage	<p>Name of the API Gateway deployment stage.</p> <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;">Note: Spaces and special characters are not allowed.</div> <p>You can use this field to create different deployment stages such as Testing, Production, and so on. You can also create this stage on the AWS console after deployment.</p>
Method	The operation which the flow implements. For example - GET, PUT, POST, or DELETE. You can select only one method at a time.

Field	Description
Path	The resource path for the operation. By default, the path displayed here is the resource path you had entered when you created the flow.
Output Validation	When set to True , the incoming data (body, headers, and query parameters) is validated against the configured JSON schema. Default: False

Output Settings

Field	Description
Query Parameters	<p>Query parameters to be appended to the path. To add the query parameters, click  and press Enter to save your changes.</p> <p>parameterName: Name of the query parameter.</p> <p>type: The data type of the query parameter. Supported types are string, number, or boolean.</p> <p>repeating: Set to True if more than one value is expected for the query parameter.</p> <p>required: Set to True if the query parameter is a required configuration. The trigger reports an error if no value(s) are provided to the required query parameter.</p> <p>Note: By default, query parameters are not mandatory. When you create an API Gateway, you must explicitly navigate to the API Gateway console and change the settings.</p>
Path Parameters	<p>Path parameters are appended to the resource path. Add the path parameters by using '{ }' in the Path field in the trigger configuration as follows:</p> <pre style="background-color: #e6f2ff; padding: 10px; margin: 10px 0;">/path/{pathparam1}/{pathparam2}</pre> <p>Path parameters are of the <code>string</code> data type only.</p>

Field	Description
Headers	<p>Header values for the trigger. To add the header parameters, click  and press Enter to save your changes.</p> <p>parameterName: Name of the header parameter.</p> <p>type: The data type of the header parameter. Supported types are string, number, or boolean.</p> <p>repeating: Set to True if more than one value is expected for the HTTP header.</p> <p>required: Set to True if the header parameter is a required configuration. The trigger reports an error if no value(s) are provided to the required header parameter.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>Note: By default, header parameters are not mandatory. When you create an API Gateway, you must explicitly navigate to the API Gateway console and change the settings.</p> </div>

Map to Flow Inputs

Map the trigger output to the flow input on this tab.

Field	Description
Function	Information about the Lambda function
Context	Envelope information about this invocation
Identity	Identity for the invoking users
ClientApp	Metadata about the calling app
API Gateway Request	API Gateway default schema which can be mapped with the schema of AWS API Gateway Lambda trigger

Reply Settings

Field	Description
Reply Data Schema	The schema is used for the reply data of the trigger. Be sure to use straight quotes for element names and values in the schema.

Map from Flow Outputs

Map the flow output to the trigger reply on this tab.

Field	Description
code	HTTP code
body	Body of the reply

S3 Bucket Event Lambda Trigger

Use the **S3 Bucket Event Lambda** trigger to trigger a Lambda function when a supported event occurs on the associated S3 bucket.

Trigger Settings

Note:

- Creating a new event or updating an existing event in the S3 Bucket Event Lambda trigger and re-pushing the app deletes existing Events on AWS S3.
- You can have only one S3 trigger in an app. An app that has an S3 trigger cannot contain any other triggers including another S3 trigger. The S3 trigger supports multiple handlers (flows), so you can have multiple flows in the app that are attached to the same S3 trigger. You can also have blank flows in the app which can serve as subflows for the flows that are attached to the S3 trigger.
- For overriding app properties, use the FLOGO_APP_PROPS_JSON environment variable only. You cannot override app properties using the FLOGO_APP_PROPS_ENV environment variable.

Field	Description
AWS Connection Name	(Mandatory) Name of the AWS connection that you want to use for deploying the flow.
Execution Role Name	Permission of the Lambda function to execute. The role must be assumable by Lambda and must have CloudWatch logs permission execution role. By default, Cloud watching is enabled.
Bucket	Name of the S3 bucket with which the trigger is to be associated. This bucket must be an existing one.
Event name	Name of the S3 bucket event notification.
Event list	A list of operations to be performed on the S3 bucket. Supported operations are POST, PUT, COPY, and DELETE.
Object prefix filter	(Optional) The prefix is to be used to filter the S3 bucket.

Field	Description
	For example, images/
Object suffix filter	(Optional) The suffix is to be used to filter the S3 bucket. For example, .jpg

Map to Flow Inputs

Map the flow output to the trigger reply on this tab. The tab displays the following fields.

Field	Description
Function	Information about the Lambda function
Context	Envelope information about this invocation
Identity	Identity for the invoking users
ClientApp	Metadata about the calling app
S3Event	Default schema of S3 bucket event trigger. It can be mapped with the flow input to pass the key values to the flow.

App Startup Trigger

Use this trigger to run flows before other triggers in the app start. It can be used to specify initialization logic that is specific to an app. For example, this trigger can be used for:

- Setting data or cache for later use in other flows
- Initialization of a database (insertion or extraction of data from tables)

Design considerations

- You can add one or more App Startup triggers to an app.

- You can add the App Startup trigger along with the Receive Lambda trigger, AWS API Gateway trigger, and S3 bucket Lambda trigger. You cannot add any other trigger along with the Receive Lambda trigger, AWS API Gateway trigger, and S3 bucket Lambda trigger.
- The trigger supports multiple handlers. So, you can configure more than one flow in a trigger.
- The flows are executed in the order in which they are configured in the trigger.
- The trigger is executed for all instances of the app. For example, if you scale up to multiple instances of the app, the trigger is executed on each scale-up.
- If a startup flow fails, the engine is terminated.
- In container deployments, the collective execution time of all flows configured to this type of trigger must not exceed the startup time set for the app.

App Shutdown Trigger

Use this trigger to run flows after all other triggers in the app are successfully stopped. It can be used to specify shutdown logic that is specific to an app. For example, this trigger can be used for:

- Cleaning up data or cache
- Deleting tables from a database

Design considerations

- You can add one or more App Shut down triggers to an app.
- You can add the App Shut down trigger along with the Receive Lambda trigger, AWS API Gateway trigger, and S3 bucket Lambda trigger. You cannot add any other trigger along with the Receive Lambda trigger, AWS API Gateway trigger, and S3 bucket Lambda trigger.
- The trigger supports multiple handlers. So, you can configure more than one flow in the trigger.
- The flows are executed in the order in which they are configured in the trigger.
- The trigger is executed for all instances of the app. For example, if you scale down

multiple instances of the app, the trigger is executed on each scale down.

- If an app is forcefully shut down, the trigger and subsequent flows are not executed. This trigger is executed only when an app is gracefully shut down.
- In container deployments, the collective execution time of all flows configured to this type of trigger must not exceed the graceful-stop time set for the app.
- You must exercise caution while defining a flow in the App Shutdown trigger. For example, when an app is scaled to more than one instance, cleanup is done while shutting down one instance might impact other running instances.

Activities

In addition to the activities available for general use, Flogo Enterprise supports activities that are originally created in Project Flogo. These activities are marked with an OSS tag enabling seamless import of apps created in Project Flogo. The Project Flogo activities are placed under the **Default** category.

If you are creating an app in Flogo Enterprise, it is preferable to use the general-purpose activities (the activities that do not have an OSS tag on them), as they have richer functionality.

Refer to <https://github.com/TIBCOSoftware/flogo-contrib> for details on the activities that are marked with an OSS tag.

The available activities are placed under the following categories:

- Default
- General

You can create a flow branch from any Activity except the **Return** Activity.

For quick access, activities used recently for the flow configuration are listed on the **Recently Used** tab. Eight of the last-used activities can be accessed from this tab. The activities listed on this tab are specific to the user and browser.

ConfigureHTTPResponse

This Activity is used to configure HTTP response codes that you want to use in your REST reply.

When using this Activity, keep the following considerations in mind:

- The flow in which you want to add the ConfigureHTTPResponse Activity must have a ReceiveHTTPMessage trigger.
- Do not use this Activity with subflows.

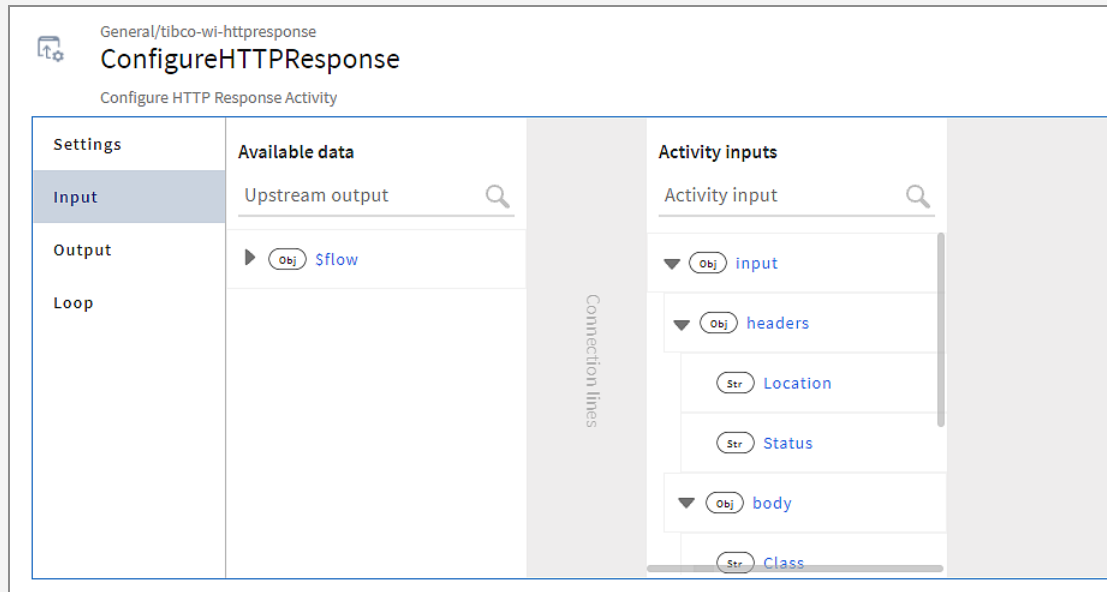
Settings

Field	Description
Trigger Name	The list of triggers to which this flow is attached. This field is displayed only when the flow is attached to multiple REST triggers. Select a trigger in which the code that you want to use is configured.
Code	List of response codes that you have configured in the selected trigger. Select a response code that you want to use.

Input

Displays the schema for the code that you selected on the **Settings** tab. Map the elements in the schema in the mapper or manually enter the values that you want to send as a response.

- i Note:** For multiple response codes in the [ReceiveHTTPMessage](#) REST trigger, map the **Response Body** and **Response Headers** from the trigger with the **Input** in this Activity. The image shows the **Input** with **headers** and **body** mapped from the REST trigger response codes.



Output

Displays the schema for the code in a read-only tree format.

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

Run JavaScript

This activity runs a JavaScript code using the specified input parameters and returns the result in the output.

Use this activity to write complex logic in JavaScript, which might not be straightforward to achieve using the mapper. For example, you can easily filter arrays based on some conditions or loop through arrays using `forEach` and other useful JavaScript functions.

This activity supports both ECMAScript 5-compatible and ECMAScript 6-compatible functions. For more information, see [ECMAScript 5-compatible functions](#) or [ECMAScript 6-compatible functions](#).

Important Considerations

- You might see some code suggestions in the editor that are not valid for ECMAScript 5-compatible or ECMAScript 6-compatible functions. Do not implement these suggestions, as they might cause the activity to return errors.
- For ECMAScript 6-compatible functions, the following are not supported.
 - Classes
 - Promises
 - Default parameters
 - Function REST parameter
 - Arrow functions

Settings

Field	Description
Javascript Code	<p>Specify the JavaScript code to be run in the following format:</p> <pre>var result, parameters <JavaScript code></pre> <p>Where:</p> <ul style="list-style-type: none">• The <code>result</code> variable must be defined for the output of the JavaScript.• The <code>parameters</code> variable must be defined for the input of the JavaScript. Set the input parameters on the Input Settings tab. Set the output parameters on the Output Settings tab.

Input Settings

Field	Description
Script Input Parameters	<p>Configure a schema for one or more input parameters to the JavaScript. The elements of this schema are available for mapping on the Input tab.</p> <p>Use app-level schema: Click Use app-level schema and select a schema that you might have defined earlier.</p> <p>Change: To change the schema, click Change.</p> <p>Syntax to access the input parameter value defined in the Javascript code:</p> <pre>parameters.<parameter_name></pre> <p>For example, if you have defined the input parameter as <code>foo</code> in the JavaScript code, use the following syntax to access the value:</p> <pre>parameters.foo</pre>

Input

The **Input** tab displays the schema that you entered in the **Input Settings** tab in a tree format. Map the elements in the schema using the mapper or manually enter the value for the element in the mapper.

Output Settings

Field	Description
Script Output	<p>Configure a schema for one or more output parameters of the JavaScript.</p> <p>Use app-level schema: Click Use app-level schema and select a schema that you might have defined earlier.</p>

Field	Description
	<p>Change: To change the schema, click Change.</p> <p>Syntax for setting the value of the output parameter in the Javascript code:</p> <pre>result.<parameter_name></pre> <p>Example:</p> <pre>result.foo = bar</pre>

Output

The **Output** tab displays the output parameters from the schema that you entered in the **Output Settings** tab. The output parameters are displayed in the `result` object in a tree format.

Field	Description
error	Flag indicating whether an error occurred while running the JavaScript.
errorMessage	The error message.
result	The output of the JavaScript code, indicates successful execution of the JavaScript.

GRPC Invoke

The gRPC Invoke Activity is an implementation of a gRPC client. Use this Activity to make outbound gRPC calls.

Settings

Field	Description
Host URL	URL used to connect to the gRPC server. The name or IP address of the host on which your .proto file resides. Be sure to append the port number on which the service is running.
Secure Connection	<p>By default, it is set to False. If you set this field to True, you can create a secure endpoint by providing Server Key and CA or Server Certificate.</p> <p>Server Key - A PEM encoded private key file.</p> <p>CA or Server Certificate - A PEM encoded CA or self-signed server certificate file.</p>
Proto File	<p>A file with .proto extension that contains the methods and services definition that Flogo Enterprise uses to create the flows. Each flow corresponds to one method in the .proto file. Currently, importing a .proto file into another .proto file is not supported.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note:</p> <ul style="list-style-type: none"> • The gRPC activity do not support options in the .proto file. If your .proto file contains any options, be sure to remove the options in the .proto file before using it. • You must not use the same gRPC .proto file for the gRPC activities in the same app. The package names for the gRPC activities must be unique. </div>
Service Name	Name of the service you want to invoke. The service is defined in the .proto file that you have selected.
Method	Name of RPC method in the selected .proto file. Each method in the .proto file is a gRPC request which is represented by a separate flow.

Input

Field	Description
Activity Input	The Input tab lists the parameters for the method that you chose on the Settings tab so that you can either enter or map their values in the mapper.

Output

The **Output** tab displays a read-only schema of the Activity output (the response that is configured for the selected method).

Iterator

For details on using the iterator, see the "Using the Iterator in an Activity" section in the *TIBCO Flogo® Enterprise User Guide*.

InvokeLambdaFunction

Use this Activity to invoke a specific Lambda function.

Settings

Field	Description
AWS Connection Name	Select an AWS connection.
ARN (Optional)	Amazon Resource Name.

Field	Description
	<p>Note:</p> <ul style="list-style-type: none"> You can also specify the ARN on the Input tab. If you specify the ARN on both the tabs, the ARN on the Input tab is used. You must specify the ARN on at least one tab; otherwise, the Activity returns an error at runtime.

Input Settings

Field	Description
Payload Schema	Enter a JSON request schema for your payload that is used to invoke the Lambda function.

Input

The payload schema that you entered on the **Input Settings** tab is displayed in a tree format on the **Input** tab. Map the elements in the schema using the mapper or enter values for the element by manually typing the value in the mapper.

Field	Description
LambdaARN	Amazon Resource Name.
	<p>Note:</p> <ul style="list-style-type: none"> You can also specify the ARN on the Settings tab. If you specify the ARN on both the tabs, the ARN on the Input tab is used. You must specify the ARN on at least one tab; otherwise, the Activity returns an error at runtime.

Output Settings

Field	Description
Result Schema	The schema for the result that is expected from the Lambda function invokes the request.

Output

The Output tab displays the result schema you entered on the **Output Settings** tab in a tree format.

InvokeRESTService

This Activity is used to request a REST service; it also accepts the reply returned by the service.

Settings

Field	Description
API Spec	<p>(Optional) Click Browse and browse to the file location on the machine. Select a JSON file.</p> <p>Supported specifications are Swagger Specification 2.0 and OpenAPI Specification 3.0.</p>
Resource Path	<p>Note: This field is only displayed if you upload a JSON file in the API Spec field.</p> <p>All resource paths available in the JSON file (that is, the Swagger 2.0 or OpenAPI 3.0 specification file you uploaded) are listed in the dropdown. Depending on the resource path you select, the supported operations are listed in the Method field.</p>
Enable	Select True if you want to enable authentication and authorization for

Field	Description
Authentication	<p>your apps using the HTTP Client Authorization Configuration connection.</p> <p>Default: False</p>
Authentication Connection	<p>Note: This field is displayed only if the Enable Authentication field is set to True.</p> <p>Select a connection that you have set up from the dropdown list.</p> <p>For information on setting up a connection, refer to HTTP Client Authorization Configuration.</p>
Method	<p>Select an operation for the request. For example: GET, POST, PUT, DELETE, or PATCH.</p>
URL	<p>An absolute path to the REST service that you want to invoke. For example: <code>http://acme.com</code> or <code>https://acme.com</code>.</p> <p>Note: If you upload an OpenAPI 3.0 JSON specification file in the API Spec field, the URL is a dropdown list. This lists the server URLs mentioned in the JSON file. Select a server URL from the list.</p> <p>Tip: To override dynamically the path provided in the URL, you can enter the URL as: <code>http://<host-url>:port/{path}</code> Here, <code>{path}</code> is the parameter that can be modified. You can map this parameter in the Input section and assign values to it from the previous activities or the app properties.</p>
Use certificate for verification	<p>This field is displayed if you enter an absolute path beginning with <code>https://</code> in the URL field. Set this to True to use a certificate for a secure connection to the server.</p> <p>Default: False</p>
Use mTLS	<p>Set to True to enable mutual authentication for a secure connection to the server.</p> <p>Default: False</p>

Field	Description
Client Certificate	<p>This field is displayed only if Use mTLS is set to True.</p> <p>This certificate is used to identify the client by the servers over TLS. The certificate must be PEM encoded. Click Browse and select the client certificate.</p> <p>Alternatively, you can configure the app property by using the Bind an Application Property toggle <input type="radio"/>. Set the Base64-encoded value of the client certificate to the corresponding app property.</p>
Client Key	<p>This field is displayed only if Use mTLS is set to True.</p> <p>Click Browse and select the client key. The key file must be PEM encoded.</p> <p>Alternatively, you can configure the app property by using the Bind an Application Property toggle <input type="radio"/>. Set the Base64-encoded value of the client key to the corresponding app property.</p>
CA/Server Certificate	<p>Click Browse and select a certificate authority (CA) certificate that verified the client's certificate or the server certificate to establish a secure connection during the TLS handshake. The certificate must be PEM encoded.</p> <p>Alternatively, you can configure the app property by using the Bind an Application Property toggle <input type="radio"/>. Set the Base64-encoded value of the CA/Server certificate to the corresponding app property.</p>
Close Connection	<p>Select True if you want to terminate the connection to the server after the response is processed. This affects the performance as the connection is no longer cached.</p> <p>Default: False</p>
Timeout	<p>Specify the timeout period (in milliseconds) for invoking a service. If a timeout value is specified, the Activity waits for the specified time. If the response is not received by the specified time, the request expires with an error.</p> <p>Default: 0 milliseconds (that is, there is no timeout for invoking a service)</p>

Field	Description
-------	-------------

Request Type

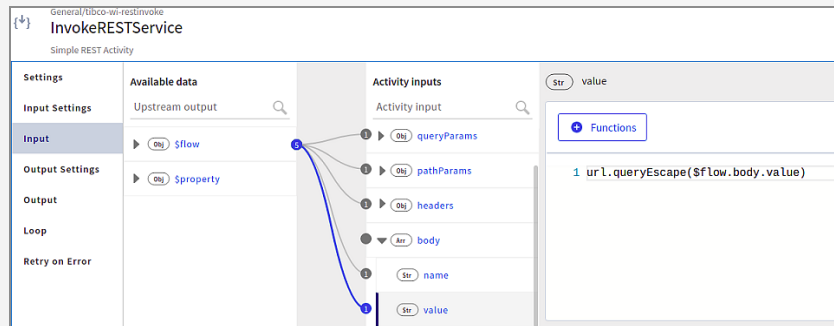
Note: This field is displayed only for the POST, PUT, and PATCH methods.

The Request content type of the REST service. The following content-type are supported:

- text/plain
- application/json
- application/x-www-form-urlencoded
- multipart/form-data

Note:



- If you select `application/x-www-form-urlencoded`, the default schema is set in the **Request Schema** field of the **Input Settings** tab. You can edit the default schema or specify your schema. If you specify your schema, it must be a name-value string pair.
- For the `application/x-www-form-urlencoded` request type, use the `url.queryEscape` function to get the expected input at the server.




- If you select the multipart or form-data as a request type, then you can pass the content-type for the file and files options in the Input.
- If you do not set the content-type, then the application/octet-stream is set as the default content-type for file and files input.

Field	Description
Proxy	<p>Specify the URL to the HTTP proxy server. If a proxy URL is specified, the request to the REST service (specified in the URL field) is routed via this proxy URL.</p> <p>Note: A secure connection to the proxy server is not supported.</p> <p>Default: Proxy URL is disabled.</p>

Input Settings

Field	Description
Query Params	<p>Query parameters to be appended to the path. To add the query parameters, click  and press Enter to save your changes.</p> <ul style="list-style-type: none"> • parameterName: Name of the query parameter. • type: The data type of the query parameter. Supported types are string, number, boolean, and object. <p>Note: Only a simple JSON object type is supported.</p> <ul style="list-style-type: none"> • required: Set to True if the query parameter is a required configuration. The trigger reports an error if no values are provided to the required query parameter.
Path Params	<p>Path parameters that are appended to the path. This is a non-editable field.</p> <ul style="list-style-type: none"> • parameterName: Name of the path parameter. This is the parameter specified in between { } in the Resource Path field or the URL field in Settings. • type: The data type of the path parameter. The Supported type is string.
Request Headers	<p>Header values for the InvokeRESTService Activity. To add the header parameters, click  and press Enter to save your changes.</p>

Field	Description
	<ul style="list-style-type: none"> • parameterName: Name of the header parameter. • type: The data type of the header parameter. Supported types are string, number, and boolean. • required: Set to True if the header parameter is a required configuration. The trigger reports an error if no values are provided to the required header parameter.
Request Schema	<p>Enter a request schema here. This field is visible only if you selected the POST, PUT, PATCH, or DELETE method on the Settings tab.</p> <div data-bbox="394 716 1414 1056" style="background-color: #f0f0f0; padding: 10px;"> <p>Note:</p> <ul style="list-style-type: none"> • For the DELETE method, specifying the request schema here is supported for manual configuration only and not when configured with Tibco Cloud Mesh and API specification. • If you selected <code>application/x-www-form-urlencoded</code> as the Request Type on the Settings tab, the default schema is set here. You can edit the default schema or specify your schema. If you specify your schema, it must be a name-value string pair. </div>
Multipart Data	<p>This field is displayed in place of the Request Schema field if you select <code>multipart/form-data</code> as the Request Type on the Settings tab. Click  to add the parameters.</p> <ul style="list-style-type: none"> • Name: Name of the parameter. • Type: The Supported types are string, object, and filecontent, file, and files. <ul style="list-style-type: none"> ◦ For file types such as images and PDF files, convert your file to Base64 format and enter the encoded value as the Input. ◦ To send the file name and the file content in the same request, use <code>file</code> or <code>files</code> types. With <code>file</code> type, you can send one file and file name. With the type <code>files</code>, you can send multiple file content for one file name. • Required: Check the box if the parameter is a required configuration.

Field	Description
	<ul style="list-style-type: none"> • Schema: Enter the JSON schema in this field if the Type is an object. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note: If you want to pass dynamic data to multipartFormData field, no data must be defined in MultipartData table. Otherwise, the dynamic data is not honored and only table field values are considered.</p> </div>

Input

i **Note:** If you upload a JSON file in the **API Spec** field, the **Input** fields are automatically populated according to the **Resource Path** and **Method** you select.

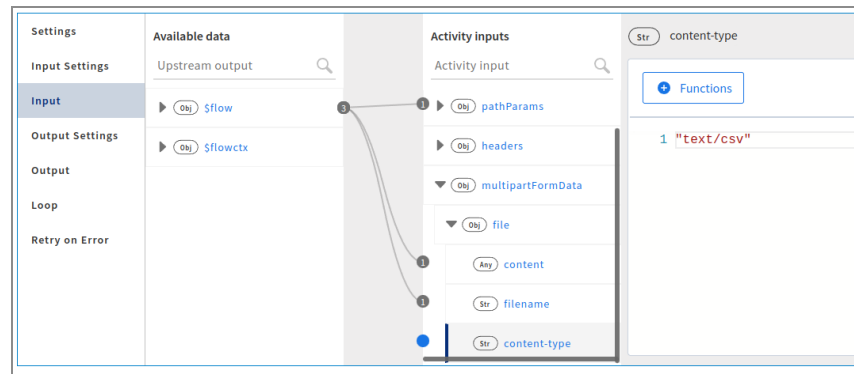
Field	Description
host	Specify the value that must override the hostname:port value specified in the URL at runtime with the value specified in this configuration. Enter a value in the form hostname[:port] where [:port] is optional.
queryParams	Provide a value to the query parameters configured in the Input Settings section.
pathParams	Provide a value to the path parameters defined as part of the URL on the Settings tab.
headers	Header values for the Activity. These values can be manually entered or mapped to the output of the trigger or any preceding Activity.
body	Request Schema values for the Activity. These values can be manually entered or mapped to the output of the trigger or any preceding Activity. This field is visible only if you selected the POST, PUT, PATCH, or DELETE method on the Settings tab.

Field	Description
	<p>Note: For the DELETE method, specifying the request schema here is supported for manual configuration only and not when configured with Tibco Cloud Mesh and API specification.</p>

multipartFormData


This field is visible if you select `file` or `files` type in the **Multipart Data** field on the **Input Settings** tab.

Each file that you enter in the **Multipart Data** table is displayed here. `content`, `filename` and `content-type` fields are displayed for each file.



Output Settings

Note: If you upload a JSON file in the **API Spec** field, the fields in **Output Settings** are automatically populated according to the **Resource Path** you select.

Field	Description
Configure Response Codes	<p>Use this field to configure response codes.</p> <p>Default: False (See "Response Schema" and "Response Type" in this table.)</p> <p>To specify a response code, select True and click . Enter the following details:</p>

Field	Description
	<ul style="list-style-type: none"> • Code: Enter a specific response code or configure a single schema for a category of response codes. For example, if all the status codes are similar (such as 501, 502, 503), you can define a single schema (as 5xx) for them. Defining a single schema saves you time and effort as you do not need to configure each status code separately in the Activity. <div data-bbox="500 527 1382 667" style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>Note: If the status code is provided as a range (5xx in the above example) and also in an absolute format (501 in the above example), the status code in the absolute format is given priority. In the above example, status code 501 is given priority over 5xx at runtime.</p> </div> <ul style="list-style-type: none"> • Type: Select the type of response expected for the Code. Supported types are String and Object. • Response Body: If the Object is selected as the Type, enter the JSON schema in the Response Body column. For String, you need not enter anything in the Response Body column. • Response Headers: Specify the response header corresponding to the response code. • Actions: The actions displayed change based on the type of the response code. <ul style="list-style-type: none"> ◦ Edit, Delete: For an Object type of response, you can edit the details or cancel it. ◦ Save, Cancel: For a String type of response, you can save or cancel the changes. <p>The response codes are displayed on the Output tab.</p>
Response Schema	<div data-bbox="418 1446 1390 1514" style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>Note: This field is displayed only when Configure Response Codes is set to False.</p> </div> <p>The schema for the reply that the server sends.</p>
Response Type	<div data-bbox="418 1635 1390 1703" style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>Note: This field is displayed only when Configure Response Codes is set to False.</p> </div>

Field	Description
	<p>The content type of the REST service. The following content types are supported:</p> <ul style="list-style-type: none"> • application/json • application/xml • text/plain • other <p>Default: application/json</p> <p>JSON to XML conversion is not supported by the REST Activity. Any service that requires data in XML format cannot be invoked after providing data in the JSON format using REST Activity.</p> <p>If the content type is other than the types application/json, application/xml, or text/plain, it is converted into Base64 encoded values. You can use the <code>utils.decodeBase64</code> function to get the actual values. This is applicable when Configure Response Codes is set to True or False.</p>
Output Format	<div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note: This field is displayed only when the Response Type is set as application/xml.</p> </div> <p>The format of the requested content for the application/xml response type. The following formats are supported:</p> <ul style="list-style-type: none"> • JSON Object • XML String
Response Schema	<p>Sample JSON schema for the response that the REST service returns.</p> <p>The JSON schema in this field is editable for the following settings only:</p> <ul style="list-style-type: none"> • When the Response Type is set as application/json • When the Response Type is set as application/xml and the Output Format is set as JSON Object
Response	<p>The header parameters for the reply.</p>

Field	Description
Headers	<p>Tip: If you want to fetch a cookie coming with a response, add a new row with <code>Set-Cookie</code> as the parameter. You can also map this parameter to subsequent activities in the flow.</p>

Output

The Output tab displays the headers and responsebody configured for the response in a tree format. The `responseTimeInMillis` parameter specifies the time taken to receive the response in milliseconds.

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

Retry on Error

Refer to the section on "Using the Retry on Error Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Retry on Error** tab.

i Note: To update all these settings for an Activity configured from Swagger 2.0 or OpenAPI 3.0 specification, make changes in the API specification file and upload it to **Settings**. Do not update the Activity settings as manual updates are removed.

Using SSL

If you choose to set up SSL authentication for the **InvokeRESTService** Activity, you must have a self-signed server certificate that you must upload when setting up the Activity.

i Note: Use a Self-signed PEM certificate for a secure connection.

To set up SSL authentication:

Before you begin

You must have the self-signed server certificate handy on your machine.

Procedure

1. On the flow page, click the **Invoke REST Service** tile to open its properties.
2. On the **Settings** tab, under **Use certificate for verification**, select **True**.
This displays **Browse**. The SSL verification is turned off when the **Use certificate for verification** is set to **False**.
3. Use **Browse** to navigate to the location of the server certificate.
Once the server certificate is uploaded successfully, the connection uses the certificate to authenticate.

LogMessage

LogMessage is an Activity that writes a message to the log. For each app, there is a log file. You can view the logs on the **Log** tab. The Log Messages generated are independent of the engine log level. It is also independent of log level set by PAPI logger or overridden using the environment variables.

Settings

The **Settings** tab has the following fields.

Field	Description
Log Level	Select one of the following log levels: <ul style="list-style-type: none">• Info: logs informational messages highlighting the app progress.• Warning: is the warning message of an unexpected error when running the flow.• Error: logs error conditions and messages.• Debug: can be used for debug-level messages.

Field	Description
Add Flow Details	<p>Appends Flow Instance ID, Flow Name, and Activity Name to the Log Message.</p> <p>By default, this field is set to False.</p>

Input

Provide the following input for this Activity.

Input Item	Description
message	<p>The message to be displayed in the log.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> <p>Note: If the message is left blank when you export the app and import it back again, "" are added to the app's json file automatically.</p> </div>
logLevel	<p>The logLevel to be set for an Activity.</p> <p>Valid Values: INFO, DEBUG, ERROR, WARN</p> <p>While mapping,</p> <ul style="list-style-type: none"> You can bind the log level from the App Properties or the App level schema. If you enter any value other than the values that are available in settings level, You get an error as 'Invalid Log Level'. The log level value set in mapper level takes the precedence over the log level value set in settings level.

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

Mapper

Use this Activity to define a schema to get the desired data. This Activity is particularly useful to define a schema for an object of type any. In the flow, you place the Mapper Activity preceding an Activity whose input requires an object of data type any. This allows you to map the object of type any to the output from the Mapper Activity. An advantage of using this Activity is that you can construct the data for any data type within the flow instead of fetching it from outside.

Input Settings

Field	Description
Input Schema	<p>Enter the JSON schema that is used as the input for this Activity. The elements of this schema are available for mapping on the Input tab and are mappable to the output from any preceding Activity, trigger, or flow input.</p> <p>The Mapper Activity outputs the elements from this schema, so they are also displayed on the Output tab in a tree format. This makes them available for mapping in the following activities.</p>

Input

The **Input** tab displays the schema you entered on the **Input Settings** tab in a tree format. You can map these elements to the output from any preceding Activity, trigger, or flow input.

Output

The **Output** tab displays the elements from the schema you entered on the **Input Settings** tab.

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

ParseJSON

This Activity takes stringified JSON data as input and converts it into a JSON object, which can then be accessed by the downstream activities that follow. You provide the input to the Activity either by entering the stringified JSON data manually on the **Input** tab or saving it in a file and entering the file path on the **Input** tab. The Activity supports output validation if you opt to validate the input JSON data against the output schema that you configure on the **Output Settings** tab.

Settings

Field	Description
Output Validation	<p>True: Select True to validate that the JSON data in your input string matches the schema that you configure on the Output Settings tab of the Activity.</p> <p>False: Select False to skip the validation of the JSON data in your input string against the schema you configured.</p> <p>This field can be configured with an app property.</p>

Input

Field	Description
jsonString	The string containing the JSON data that you want to parse. This Activity creates a JSON object with the parsed JSON data. Enter the string manually or map it to an element from the output of the trigger, flow input, or one of the preceding activities.

Output Settings

Field	Description
Schema	The schema that you want to use to create the JSON object. You have the option to validate the stringified JSON (input to the Activity) against this schema.

Output

The output schema is displayed in a read-only tree format.

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

ReplyToTrigger

This Activity is placed under the **General** category. Use this Activity to map output values and to send the reply back to the trigger. The flow execution continues after the reply is sent to the trigger, unlike the **Return** Activity.

When using this Activity, keep the following considerations in mind:

- The flow in which you want to add the **ReplyToTrigger** Activity must have a trigger.
- The input of the Activity must be consistent with the flow output schema.
- Do not use this Activity with subflows. To send a reply from subflows, use the **Return** Activity.
- If the flow output schema is updated, you must open the Activity once so that the update reflects in the Activity.
- You can add multiple **ReplyToTrigger** activities in one flow but only the first Activity sends the reply to the trigger.

ProtobufToJSON

This Activity is placed under the **General** category. Use the ProtobufToJSON Activity to convert protocol buffer messages to JSON format.

For information about .proto files, see the [proto3 Language Guide](#).

Settings

Field	Description
Proto File	Click Browse and select a <code>.proto</code> file. The file must contain only the proto3 syntax.
Message Type Name	All the message types from the <code>.proto</code> file are provided as list options. Select the message type from the list whose corresponding proto3 message you want to convert.
Include Default Values	The protocol buffer message fields that have null values are excluded when converting to another format. Set this field to True to include the message fields with their default values in the JSON format output. Default: False

Input

Enter the **protoMessage** as the Activity input. The **protoMessage** must be a Base64-encoded string.

Output

The **Output** tab displays the names of the fields for the selected message type from the `.proto` file.

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

No-Op

This Activity is placed under the **Default** category. You can use the **No-Op** Activity to implement branching of activities in the **Main flow** and the **Error handler**. The **No-Op** Activity has no input or output fields. It does not affect the values of the previous Activity. You can insert this Activity at any position in the flow and also use it with subflows.

ReplyToHTTPMessage

This activity is supported for backward compatibility only.

This Activity is not supported in Flogo Enterprise 2.5.0 and above. This Activity applies only to flows that were created in previous versions of Flogo Enterprise (before version 2.5.0) that are imported into the current version.

This Activity automatically gets created when you create a flow with a REST trigger. It is used by the server to reply to a request from the REST client.

Configuration

Field	Description
Reply	<p>Reply sent by the server in response to the REST client request. The two supported replies are Success with Data and Error with Message.</p> <p>If you select Success with Data, the reply schema must be configured on the Input Settings tab in the Schema field. If you select Error with Message, you must configure the error message in the message field of the Input tab.</p>

Input Settings

Field	Description
Schema	Enter the reply schema or sample data using a JSON structure.

Input

Field	Description
message	The string that is included in the reply. If you configured the Reply field on the Configuration tab with Error with Message , the error message must be entered in the message text box. If you configured the Reply field with Success with Data , then you must map your data according to the schema specified on the Input Settings tab.

SendMail

Use the **SendMail** Activity to send emails using an SMTP server.

Note:

- To securely configure the **SendMail** Activity using the `smtp.gmail.com` server, use **TLS** on port 587 or **SSL** on port 465.
- The SendMail activity allows you to use the password of third-party apps as mandated by popular mailboxes such as Gmail, Yahoo, Outlook, and AOL. You cannot use your regular mailbox username and password with this activity.

Settings

The **Settings** tab has the following fields.

Field	Description
Server	The hostname or IP address for the mail server.
Port	The port used to connect to the server.
Username	The username to use when authenticating to the mail server.
Password	The password to use when authenticating to the mail server.
Connection Type	The type of connection to be used to communicate with the mail server. Select TLS or SSL depending upon the security configuration of the mail server. In case no security is enabled on the mail server, select NONE .
Server Certificate	(Available only when Connection Type is set to SSL) The server or CA certificate to be used for the secure connection. The certificate must be PEM encoded.

Input

This tab displays the fields that are used as input for the Activity.

Input Item	Description
message_content_type	The type of message content. Valid types are "text/plain" or "text/html".
sender	The email address of the sender.
recipients	The recipient list for the email. You can send the mail to multiple recipients. Provide a list of recipients in a single string by using a comma as the delimiter.
cc_recipients	The CC recipient list for the email. You can send the mail to multiple recipients. Provide a list of recipients in a single string by using a comma as the delimiter.
bcc_recipients	The BCC recipient list for the email. You can send the mail to multiple recipients. Provide a list of recipients in a single string by using a comma as the delimiter.
reply_to	Email address to which the reply message is to be sent.
subject	The subject of the email.
message	The content of the email message.
attachments	File attachments to be sent along with the email message. To map the child elements, add <code>array.forEach()</code> to the <code>attachments</code> field and then specify the child elements as follows: <ul style="list-style-type: none"> <code>file</code>: Specify the path of the file to be attached using <code>file://<path></code> or specify the content of the file by enclosing it in double-quotes.

Input Item	Description
	<p>Note: In Flow Tester, <code>file://<path></code> cannot be specified.</p> <ul style="list-style-type: none">• <code>filename</code>: Specify the name of the file to be attached.• <code>base64EncodedContents</code>: If the file is a Base64 encoded file, set this field to <code>true</code>. The default is blank (or <code>false</code>). <p>To send multiple attachments, use the Loop feature.</p>

Loop

For information on the Loop tab, see the "Using the Loop Feature in an Activity" section in the *TIBCO Flogo® Enterprise User Guide*.

SharedData

The **SharedData** Activity enables sharing of runtime data within a flow or across flows in an app. This Activity simplifies your flow designs. The advantage of using this feature is that you can set data anywhere in the main flow, subflow, or error handler and the data can be shared across the entire flow or app.

This Activity involves the following operations:

- **Get**: Retrieves data from the selected scope (either flow or app) based on a key.
- **Set**: Sets data for the selected scope based on a key.
- **Delete**: Deletes data from the app data.

For example, you can set the values in a parent flow. In a subflow, you can add the SharedData Activity and use the Get operation to access the values set in the parent flow.

Settings

The **Settings** tab has the following fields.

Field	Description
Scope	<p>Options are:</p> <ul style="list-style-type: none"> • Flow: Data can be shared within the flow instance and its subflow instances only. • Application: Data can be shared across flow instances within an app. <p>Default: Flow</p>
Operation	<p>The operation to be performed. Options are:</p> <ul style="list-style-type: none"> • Get: Retrieve the data from the selected scope by key. • Set: Set the data for the selected scope by key. • Delete: (Available only if Scope is selected as Application) Optionally, you can delete app data based on the input key. For example, if you need data for one-time use only, you can delete the data to avoid storing it in memory unnecessarily. Otherwise, the data is deleted when you scale down or stop the app. <p>Default: Get</p>
Data Type	The data type of the shared data. Supported types are string, integer, number, and object.
Object Schema	(Only if you select Data Type as object) Specify the object's JSON data or schema in Object Schema . You can also specify an app-level schema by using the Use app-level schema option.

Input

This tab displays the fields that are used as input for the Activity.

Input Item	Description
key	<p>Available operations are:</p> <ul style="list-style-type: none"> • Set: Specify any value that you want to use while setting the data. • Get: If you want to retrieve data using the Get operation, you must use

Input Item	Description
	<p>the same value that was specified while using the Set operation for setting the data for a scope.</p> <p>App properties can also be used to set or get the key.</p>
input/data	(Set operation only) You can provide values of the Data Type selected on the Settings tab.

Output

This tab displays the output of the Activity. Note that the **Set** operation does not have any output.

Input Item	Description
exist	Indicates whether the data for the key specified on the Input tab exists.
data	Output data based on the input specified on the Input tab.

Loop

For information on the **Loop** tab, see the "Using the Loop Feature in an Activity" section in the *TIBCO Flogo® Enterprise User Guide*.

Sleep

The Sleep Activity is an asynchronous Activity that suspends the execution of a flow for a specified time.

Settings

Field	Description
Interval Type	The unit of the time interval for which the execution of the flow must be suspended. Supported types are Millisecond , Second , and Minute . Default: Millisecond
Interval	The time interval for which the execution of the flow must be suspended. Default: 0

Input

i **Note:** The fields on the **Input** tab are required only if you need to pass values from the output of a previous Activity or trigger. Otherwise, you can directly specify the values on the **Settings** tab. Values specified on the **Input** tab take precedence over values specified on the **Settings** tab, if values are configured in both the tabs.

Field	Description
Interval Type	The unit of the time interval for which the execution of the flow must be suspended. Supported types are Millisecond , Second , and Minute . Default: Millisecond
Interval	The time interval for which the execution of the flow must be suspended. Default: 0

Loop

Refer to the section on "Using the Loop Feature in an Activity" in the *TIBCO Flogo® Enterprise User Guide* for information on the **Loop** tab.

JSONtoXML

This Activity takes stringified JSON as input and converts it into a XML String, which can then be accessed by the downstream activities that follow.

JSONtoXML Conversion limitation

The difference between XML and JSON impose the following limitation.

- JSON is a maplike structure with key-value pairs and XML stores data in a tree structure with namespaces for different data categories. So the input JSON not having any a tree structure will not result in a valid XML.

Input

Field	Description
jsonString	This field takes stringified JSON data. We can directly pass stringified JSON data manually or map the textContent coming from previous activity (e.g. GET JSON file from S3).

Output

Display output as read-only jsonObject.

Loop

For information on the **Loop** tab, see the "Using the Loop Feature in an Activity" section in the *TIBCO Flogo® Enterprise User Guide*.

XMLToJSON

This Activity takes stringified XML data as input and converts it into a JSON object, which can then be accessed by the downstream activities that follow.

XMLtoJSON Conversion limitations

The differences between XML and JSON impose some of the following limitations.

- The XML element and attribute names should not contain any delimiter used in JSON
- XML comments (<!-- comment -->) are ignored in the JSON document
- DTD declarations are ignored.
- XML processing instructions are ignored.
- All XML element/attribute values are transformed to a JSON string as the conversion is not schema (xsd) aware. Set typecast to true to convert string data starting with number to integer and string data starting with true/false to boolean. Set ordered field as true to set the key in alphabetical order in JSON.
- Entity references are ignored.
- XML attribute then while converting to json, hyphen (-) used as a prefix in json to indicate attribute and # indicate key.

Input

Field	Description
xmlString	This field takes stringified XML data. We can directly pass stringified xml data manually or map the textContent coming from previous activity (e.g. GET XML file from S3).
Ordered	This field takes boolean value. When set ordered to true, set key in JSON in alphabetical order.
typeCast	This flag controls the conversion of string data. When enabled (True), it performs the following operations: <ul style="list-style-type: none"> • String data starting with a number is converted to an integer. • String data starting with "true" or "false" is converted to a boolean.

Output Settings

Field	Description
Schema	Configure expected object structure using JSON schema or JSON sample.

Output

Display output as read-only xmlString.

Loop

For information on the **Loop** tab, see the "Using the Loop Feature in an Activity" section in the *TIBCO Flogo® Enterprise User Guide*.

Connections

Along with the activities and triggers, Flogo Enterprise also provides connections for general use. Connections contain the parameters that are needed for a client connection to an external data provider or interface. These details are then used by activities or triggers to connect at runtime.

HTTP Client Authorization Configuration

You can set up the **HTTP Client Authorization Configuration** connection from the **Connections** tab to add authentication and authorization to your Flogo apps. To enable the connection you have set up, refer to the [InvokeRESTService](#) Activity.

The connection has the following fields:

Field	Description
Name	Enter a name for the connection.

Field	Description
Description (optional)	Enter a description for the connection.
Authorization Type	<p>Select an authentication type.</p> <p>The connection supports two types of authorization:</p> <ul style="list-style-type: none"> • Basic • OAuth2
If you select Basic as the Authorization Type , the following fields are displayed:	
User Name	Enter a username for the connection.
Password (optional)	<p>Enter a password for the connection.</p> <p>Some services can send authentication data with user name only. In such cases, you need not provide any password.</p>
If you select OAuth2 as the Authorization Type , the following fields are displayed:	
Grant Type	<p>Indicates the method by which an app can obtain an access token.</p> <p>Select one of the following supported types:</p> <ul style="list-style-type: none"> • Authorization Code • Client Credentials
Callback URL	<p>The connection is redirected to this URL after authorization. Your app's callback URL must match this URL.</p> <p>The URL is read-only.</p>
Auth URL	<p>Authorization server API endpoint. For example, the Google authorization URL is:</p> <p><code>https://accounts.google.com/o/oauth2/v2/auth</code></p> <p>This field is an app-property enabled field.</p>

Field	Description
Additional Auth URL Query Parameters (optional)	Additional query parameters to get the refresh token based on the service you request for. For example: <code>access_type=offline&prompt=consent</code> <code>token_access_type=offline</code>
Access Token URL	The token API endpoint used to get access tokens. For example: <ul style="list-style-type: none"> • Google: <code>https://oauth2.googleapis.com/token</code> • Salesforce: <code>https://login.salesforce.com/services/oauth2/token</code> This field is an app-property enabled field.
Client Id	The client id of the OAuth2 app. You can change this value at runtime. This field is an app-property enabled field.
Client Secret	The client secret of the OAuth2 app. You can change this value at runtime. This field is an app-property enabled field.
Scope	Specifies the level of access that the app is requesting. You can specify multiple space-delimited values. For example: <code>Salesforce - chatter_api refresh_token</code>
Audience	The unique identifier of the audience for an issued token. The audience value is an app client ID for an ID token or the API that is being called for an access token. This field is an app-property enabled field.
Client Authentication	The method by which authentication parameters are sent. Based on the service request, you can send authentication parameters in Header , Body , or Query .

Field	Description
	<ul style="list-style-type: none">• Header - Indicates sending authentication parameter through headers.• Body - Indicates sending authentication parameters through the body with <code>application/x-www-form-urlencoded</code>.• Query - Indicates sending authentication parameters through query parameters.
Token	Indicates the token which is Base64 encoded value with app property enabled. This field is read-only.

**Note: For refresh tokens:**

The Flogo authorization connection only supports standard OAuth2 workflow.

The OAuth 2.0 service provider must also return refresh tokens when you obtain an access token from the OAuth flow. For information on obtaining refresh tokens, refer to your OAuth 2.0 provider. You can then add this information in the **Additional Auth URL Query Parameters** field or the **Scope** field. This is necessary for long-running Flogo apps where the access tokens might expire.

The refresh token operation only happens when the server returns HTTP status code 401.

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 following documentation for this product 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

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, and Flogo are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

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.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2016-2025. Cloud Software Group, Inc. All Rights Reserved.