Table of Contents

Setup Your Integration Solution

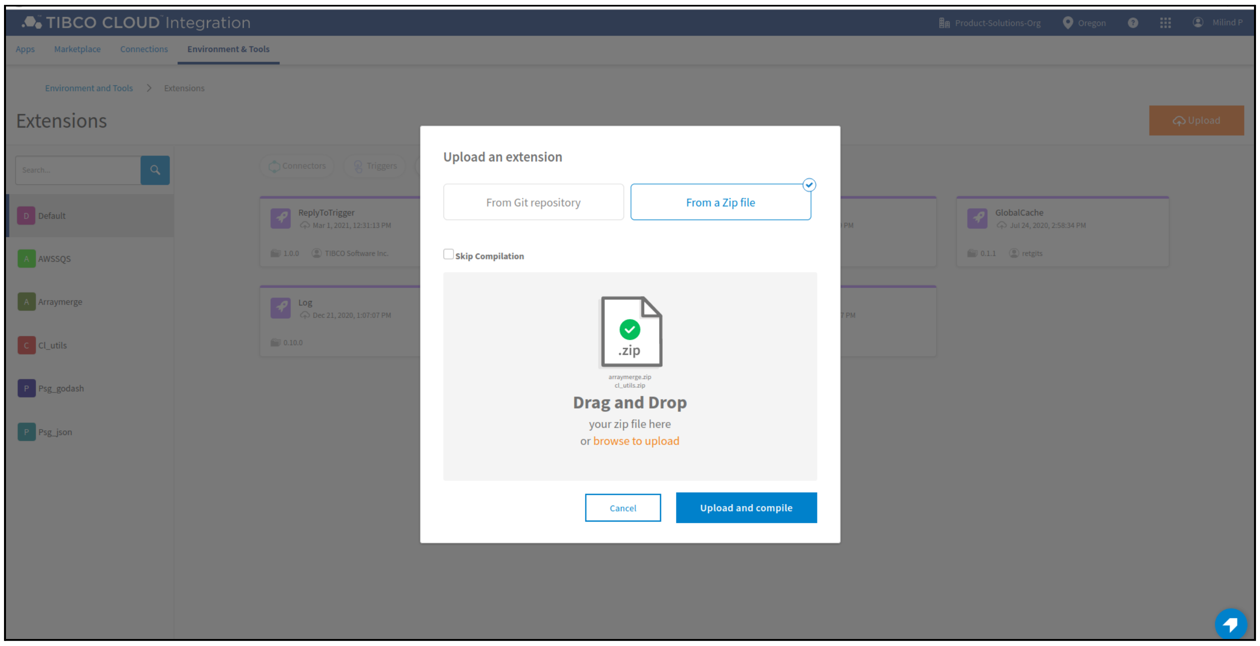
Here, we will go through steps that will demonstrate how quickly users can import and configure the outlined TIBCO Cloud Integration Develop (Flogo) apps in the previous section.

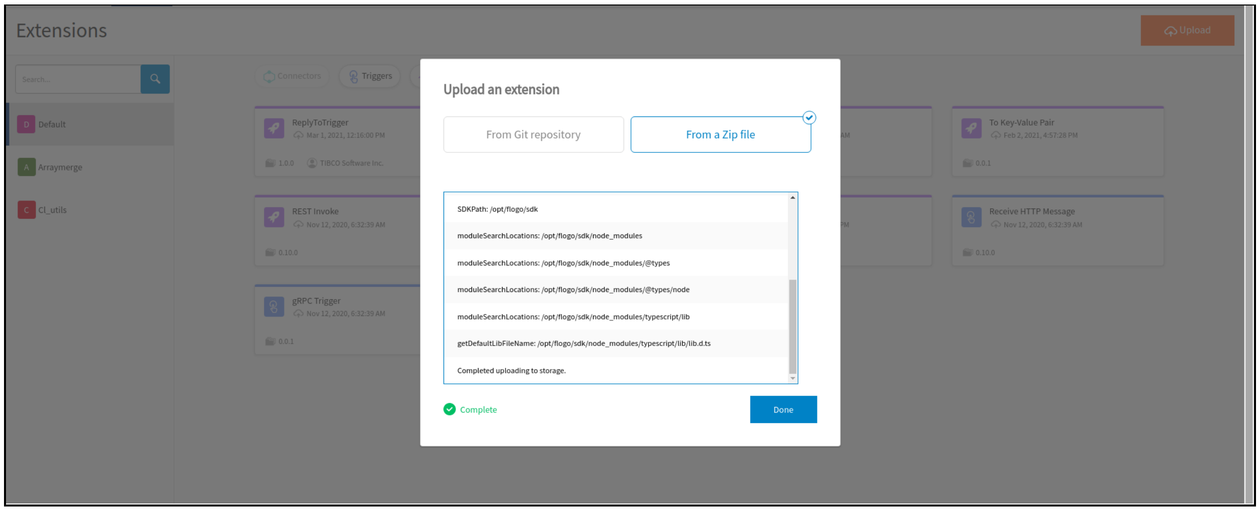
Custom Flogo Extension

All the custom extensions used in the accelerator are available under ~/./src/TCI-Flogo/resources/custom_extensions

- ❏ **CI_utils_2.0**.zip
- ❏ **Arraymerge**.zip
- ❏ **Rest**.zip
- ❏ **Actreply**.zip

Upload above mentioned custom functions to your subscription, select upload and compile option while uploading the extension.





```
SDKPath: /opt/flogo/sdk
moduleSearchLocations: /opt/flogo/sdk/node_modules
moduleSearchLocations: /opt/flogo/sdk/node_modules/@types
moduleSearchLocations: /opt/flogo/sdk/node_modules/@types/node
moduleSearchLocations: /opt/flogo/sdk/node_modules/typescript/lib
getDefaultLibFileName: /opt/flogo/sdk/node_modules/typescript/lib/lib.d.ts
Completed uploading to storage.
```

We have below a set of DB tables created with the sole purpose of external monitoring, liveapps case tracking and keeping the historical incident data for future use (Data Science and ML).
All DB table scripts available under ~/../src/TCI-Flogo/resources/MySQL-DB_Tables/

- **gtfsbusposctrl**.sql
  This will be used for external monitoring purposes where you can look for JOB details and number messages published by that JOB.

- **gtfs_alertcasetracking**.sql
  The purpose of this table is to keep track of live/reported incidents through Liveapps case flow. Also, used for creating Stream of data for visualization.

- **gtfs_incidenthistory**.sql
  This table will hold the closed incident data over a period of time. This table may help in future releases where we'll introduce Data Science and predictive analysis.

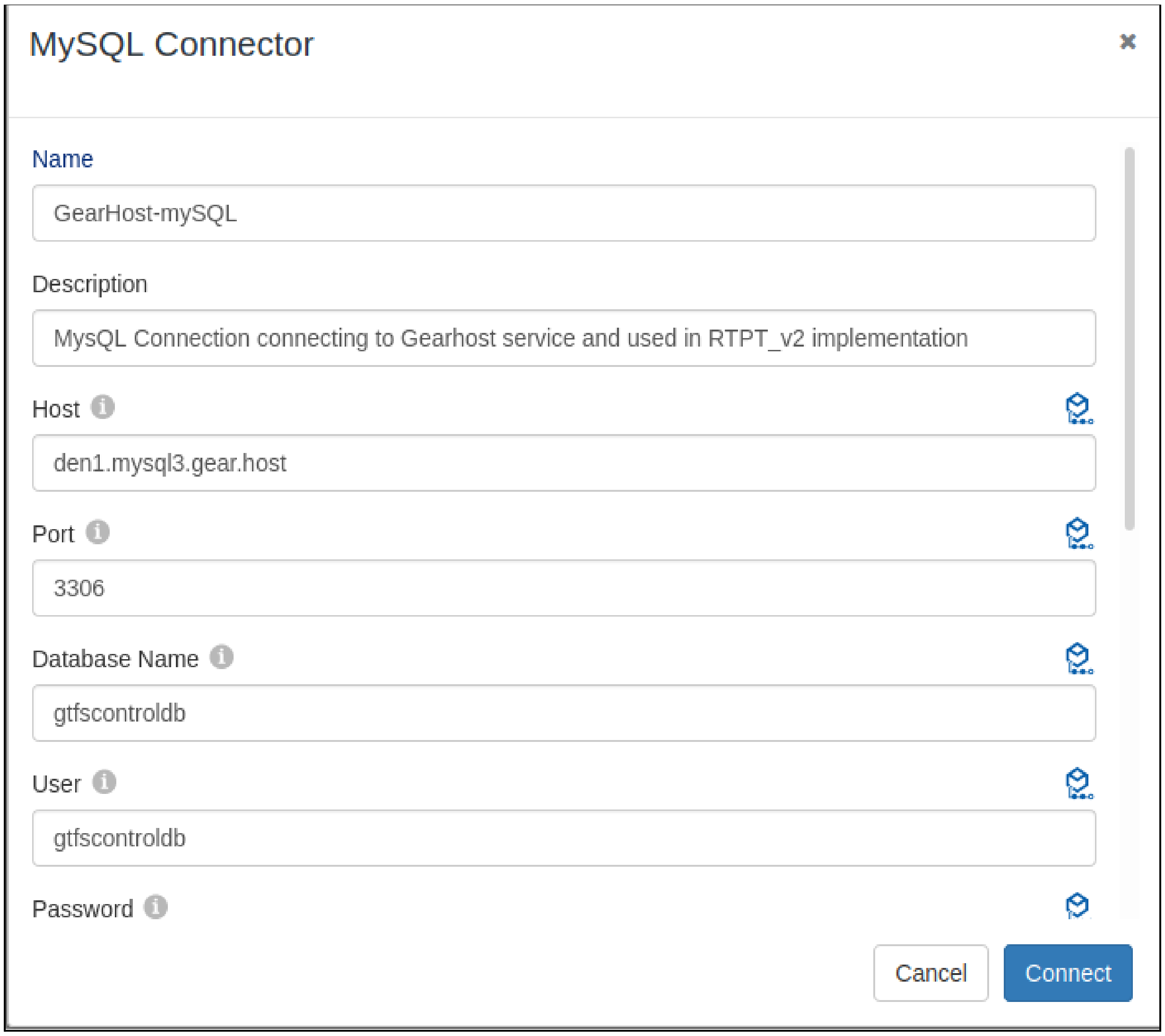
TIBCO Cloud Integration Develop (Flogo) Applications

All the TCI FLOGO applications are available under
~/../src/TCI-Flogo/apps/<xx_xxx_xxx>.json

Shared Connectors

- MySQL Connector
  You need to configure the MySQL server Connection to use in the listed Flogo application.
  We have used the mysql db for multiple purposes (monitoring, tracking, streaming of incidents, etc.).

  You need to configure the Connector with appropriate subscription details.



MySQL Connector

Name: GearHost-mySQL

Description: MySQL Connection connecting to Gearhost service and used in RTPT_v2 implementation

Host: den1.mysql3.gear.host

Port: 3306

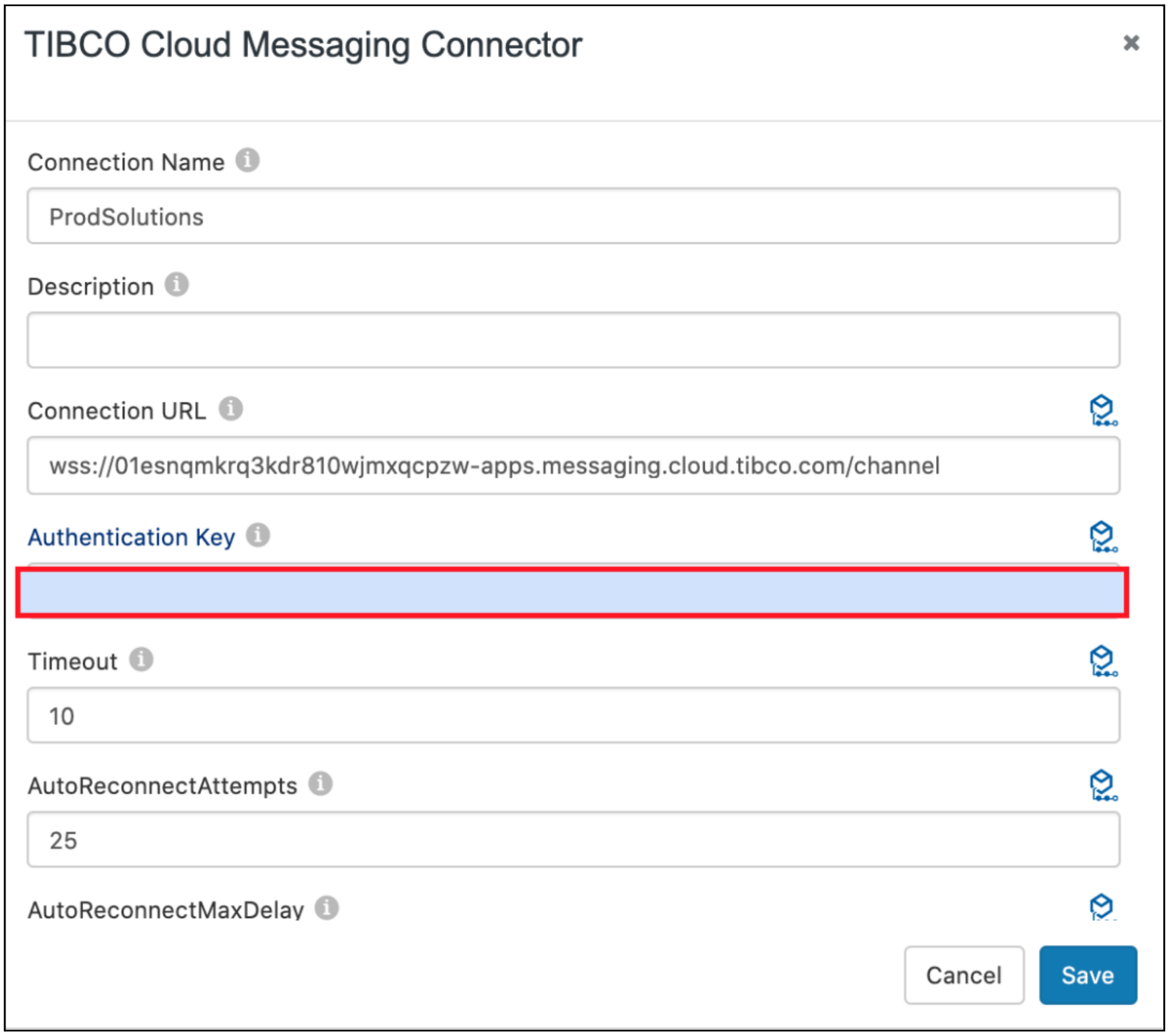
Database Name: gtfscontroldb

User: gtfscontroldb

Password

- TIBCO Cloud Messaging (TCM-eFTL) Connector
  We have used the TCM-eFTL connector to communicate to & from TIBCO Cloud Events with Flogo apps.

  You need to configure the TCM-eFTL connector as per your subscription details.



TIBCO Cloud Messaging Connector

Connection Name: ProdSolutions

Description:

Connection URL: wss://01esnqmkrq3kdr810wjmxqcpzw-apps.messaging.cloud.tibco.com/channel

Authentication Key:

Timeout: 10

AutoReconnectAttempts: 25

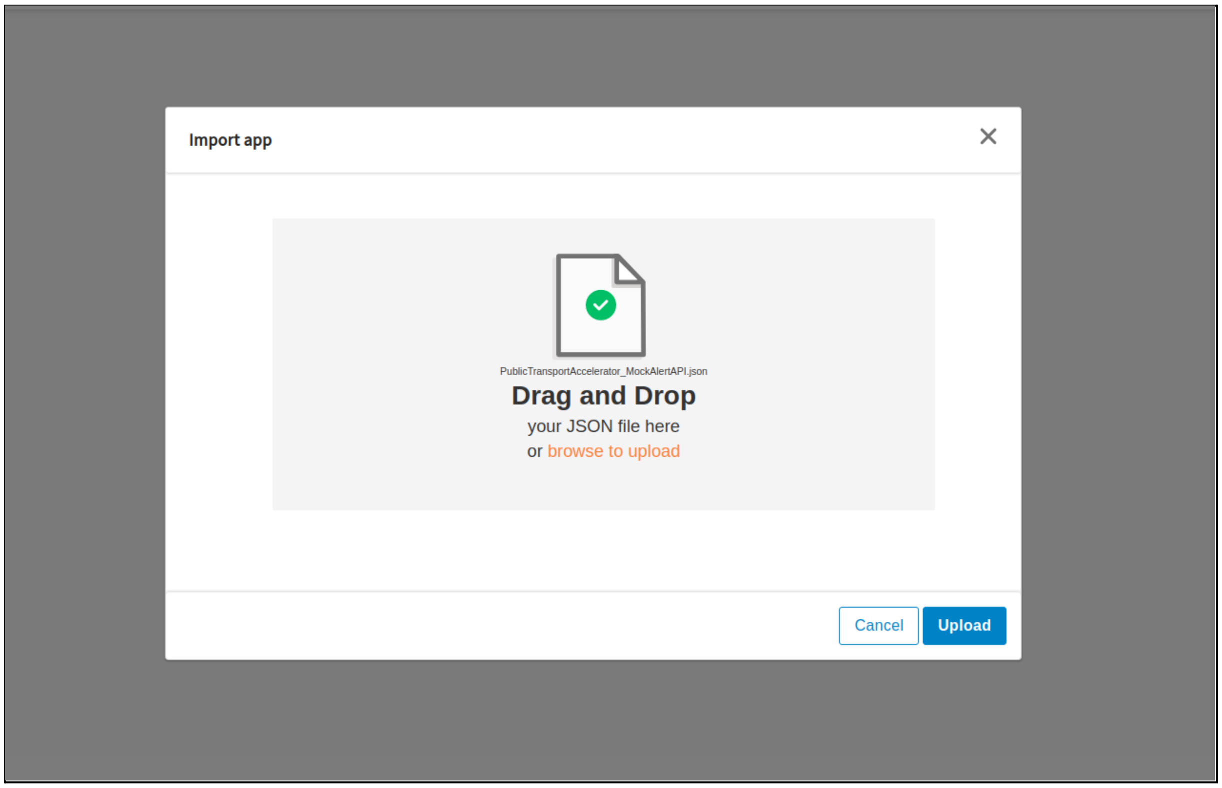
AutoReconnectMaxDelay

MockAlertAPI

Visibility: *Public Endpoints*

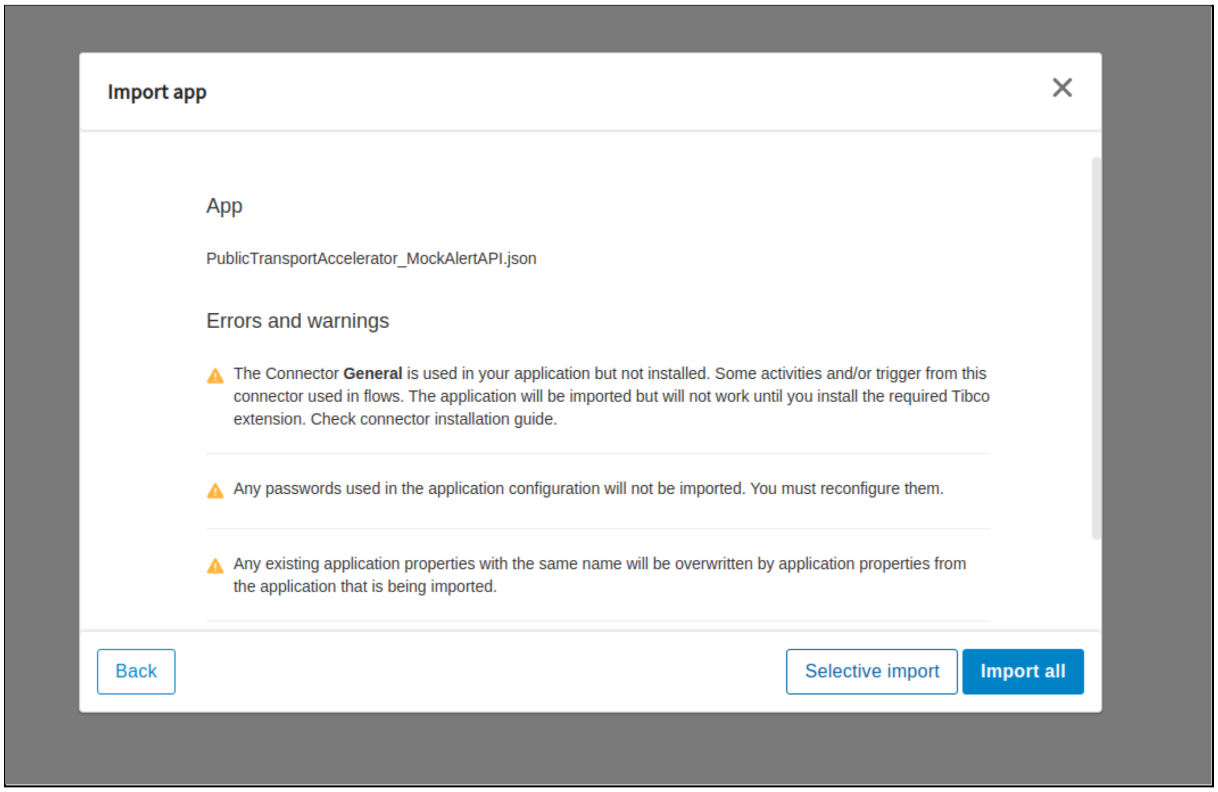
Name: *SmartTransportAccelerator_MockAlertAPI*

- ❏ Skip the following steps if you are already in TIBCO Cloud Integration.
  - ❏ Login to TIBCO Cloud (https://cloud.tibco.com/) (sign up trial option is also available).
  - ❏ In the capabilities page, Select Integration.
  - ❏ Select Integration Apps.
- ❏ Select menu items Apps --> Under Apps, Select the blue Create/Import button --> In the dialog box, select Flogo, select Create New App.



- ❏ Select Import App button --> In the Import App dialog box, select browse to upload --> Import SmartTransportAccelerator_MockAlertAPI
- ❏ In the file dialog box,
- ❏ Select "~/../src/TCI-Flogo/apps/resources/SmartTransportAccelerator_MockAlertAPI.json".
- ❏ Select the Upload button.

- ❑ Ignore the warning message, if there is any and select Done.



App

PublicTransportAccelerator_MockAlertAPI.json

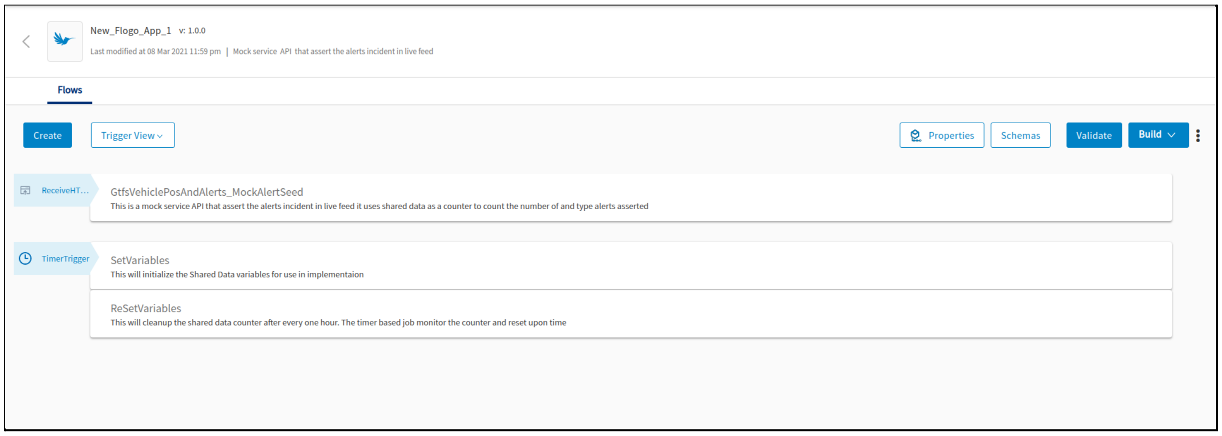
Errors and warnings

The Connector **General** is used in your application but not installed. Some activities and/or trigger from this connector used in flows. The application will be imported but will not work until you install the required Tibco extension. Check connector installation guide.

Any passwords used in the application configuration will not be imported. You must reconfigure them.

Any existing application properties with the same name will be overwritten by application properties from the application that is being imported.

- ❑ Rename the app with the name of your choice or you can always use the above mentioned name.



New_Flogo_App_1 v: 1.0.0

Last modified at 08 Mar 2021 11:59 pm | Mock service API that assert the alerts incident in live feed

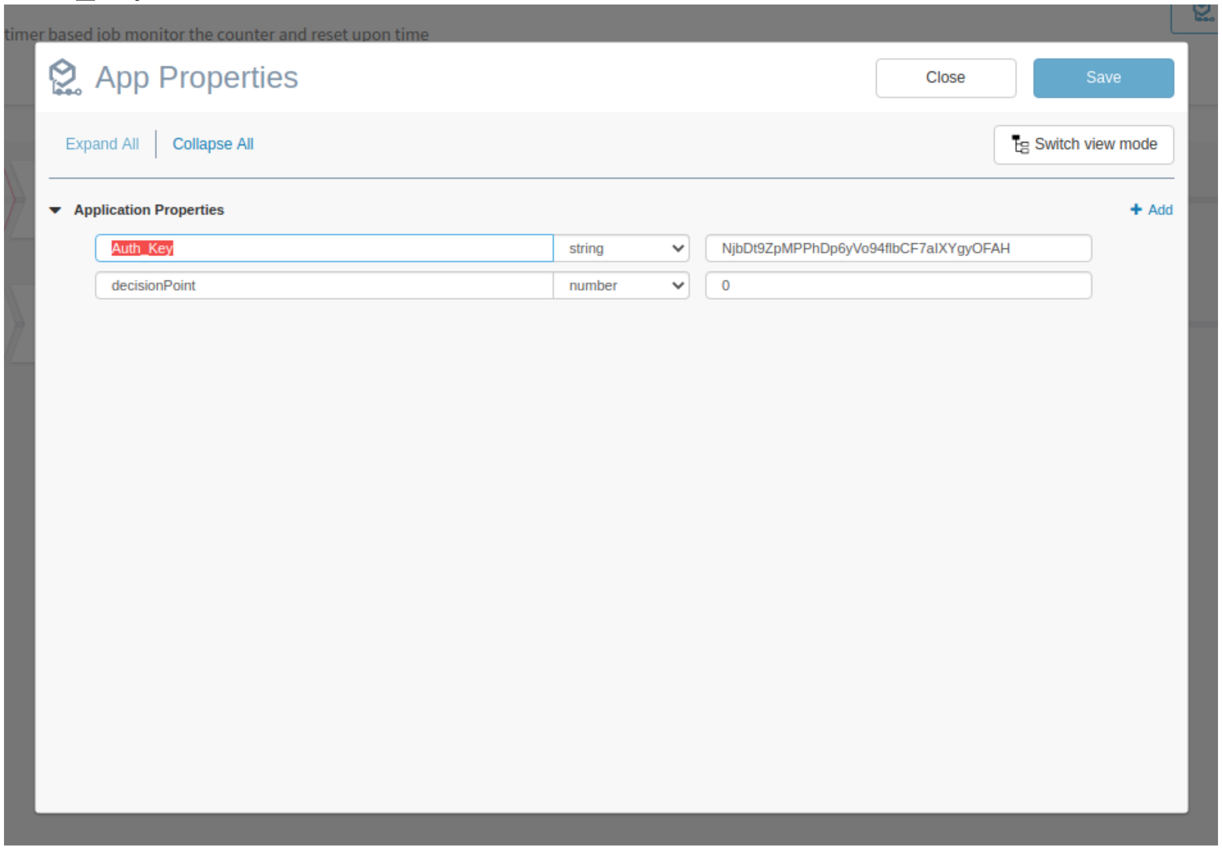
GtfsVehiclePosAndAlerts_MockAlertSeed
This is a mock service API that assert the alerts incident in live feed it uses shared data as a counter to count the number of and type alerts asserted

SetVariables
This will initialize the Shared Data variables for use in implementaion

ReSetVariables
This will cleanup the shared data counter after every one hour. The timer based job monitor the counter and reset upon time

- ❑ You do not need to do any special configuration changes in the application to run it. In case, if you want to change the authentication key for the TfNSW open data hub service then you can change the value of property

“Auth_Key”.


timer based job monitor the counter and reset upon time

App Properties

Application Properties

Auth_Key	string	NjbDt9ZpMPPhDp6yVo94flbCF7aIXYgyOFAH
decisionPoint	number	0

FLOW Details

❑ HTTP Based Trigger Flow:
This flow is responsible to produce the Live GTFS-RT Positions feed in json format for consumers by subscribing it from TfNSW open data hub and convert it into JSON from protobuf using custom extension. Along with that, it will assert the Alert payload into the last bus of the feed with pre-config rules. Every hour only 3 alerts will be asserted.

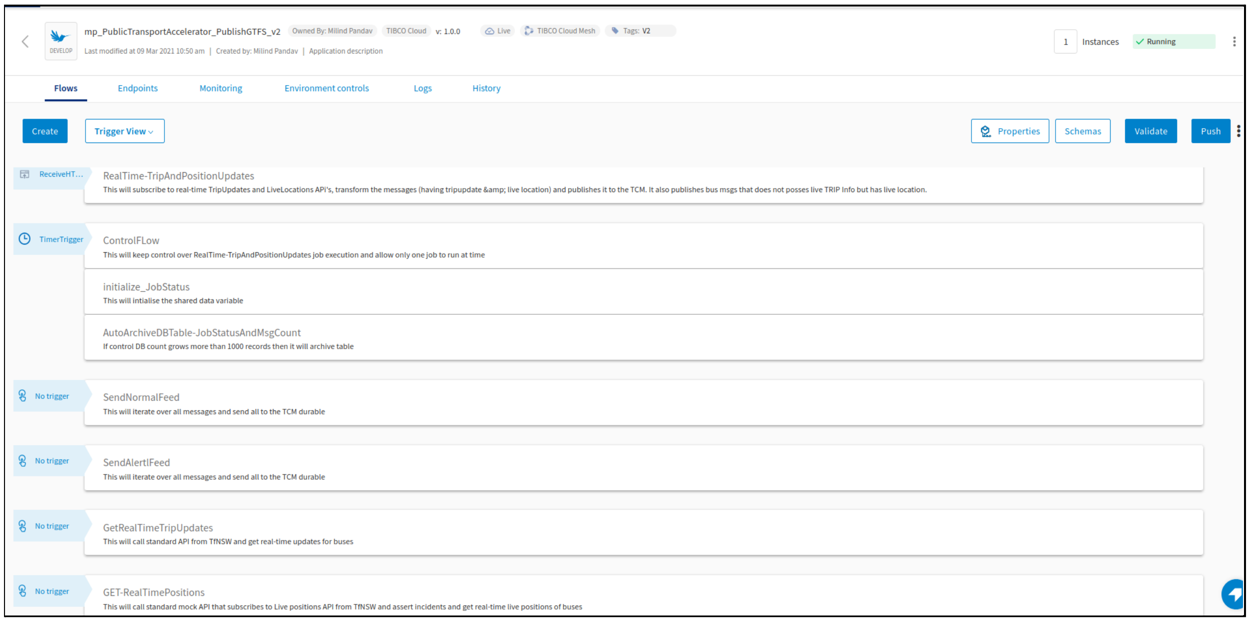
❑ Timer Based Flow:
This will control the generation of alerts through above flow and it will reset the counter every hour. We have used a shared data mechanism to keep track of the number of alerts populated.

PublishGTFS_v2

Visibility: *Private Endpoint(TIBCO Cloud MESH)*

Name: *SmartTransportAccelerator_PublishGTFS_v2*

- ❑ Repeat the steps from a. to h. outlined in the above MOCK API application.



mp_PublicTransportAccelerator_PublishGTFS_v2 v: 1.0.0

Last modified at 09 Mar 2021 10:50 am | Created by: Milind Pandav | Application description

RealTime-TripAndPositionUpdates
This will subscribe to real-time TripUpdates and LiveLocations APi's, transform the messages (having tripupdate &amp; live location) and publishes it to the TCM. It also publishes bus msgs that does not posses live TRIP Info but has live location.

ControlFLow
This will keep control over RealTime-TripAndPositionUpdates job execution and allow only one job to run at time

initialize_JobStatus
This will intialise the shared data variable

AutoArchiveDBTable-JobStatusAndMsgCount
If control DB count grows more than 1000 records then it will archive table

SendNormalFeed
This will iterate over all messages and send all to the TCM durable

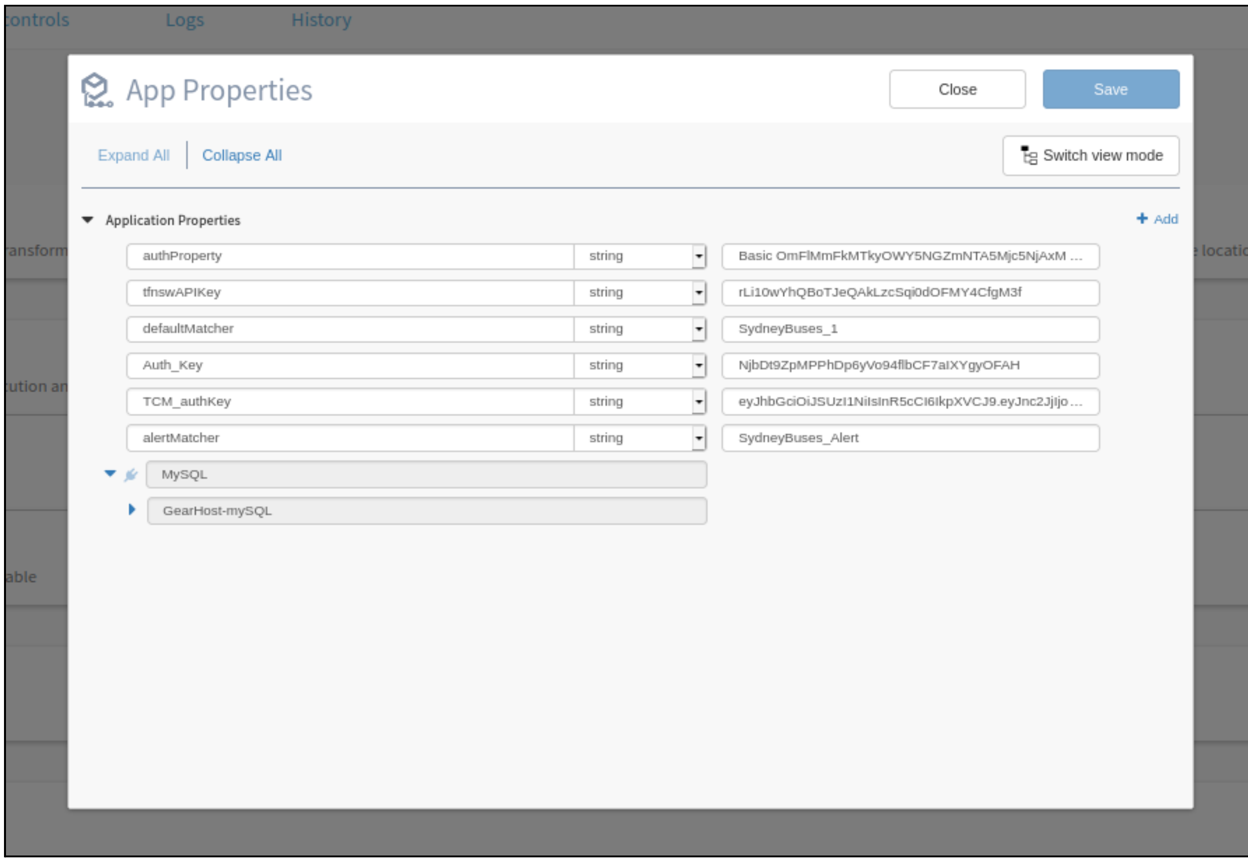
SendAlertlFeed
This will iterate over all messages and send all to the TCM durable

GetRealTimeTripUpdates
This will call standard API from TfNSW and get real-time updates for buses

GET-RealTimePositions
This will call standard mock API that subscribes to Live positions API from TfNSW and assert incidents and get real-time live positions of buses

- ❑ You need to make sure that you have configured the **JDBC Connection** properly and updated the connection details in activities available in the flow.
- ❑ Next, Update the **Authentication key for TfNSW open data hub API** and TCM-eFTL **auth_key** for data publication. You need to use these

authentication keys to connect to respective APIs as depicted below snap.



authProperty	string	Basic OmFlMmFkMTkyOWY5NGZmNTA5Mjc5NjAxM ...
tfnswAPIKey	string	rLi10wYhQBoTJeQAkLzcSqi0dOFMY4CfgM3f
defaultMatcher	string	SydneyBuses_1
Auth_Key	string	NjbDt9ZpMPPhDp6yVo94flbCF7aIXYgyOFAH
TCM_authKey	string	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJnc2Jjljo ...
alertMatcher	string	SydneyBuses_Alert

MySQL

GearHost-mySQL

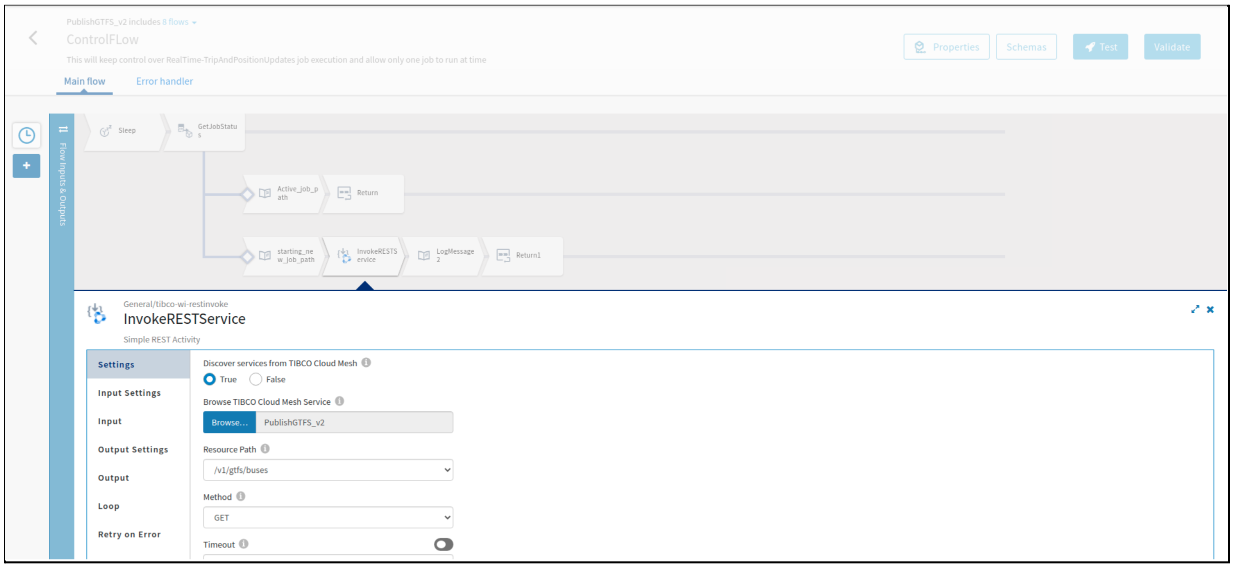
- ❑ Once the above steps are performed, then validate the application for any errors. Now, your application is ready for deployment so click the PUSH button.

FLOW Details

- ❑ HTTP Based Trigger Flow:
  This is responsible for subscribing the live GTFS feed from TfNSW real-time trip updates and live positions/alerts generated through MockAlertAPI described above. The flow will perform tasks of conversion of protobuf to json, transformation and massaging on received payload into required format then pass the produced output to the TCM-eFTL using APIs exposed.

- ❑ Timer Based Trigger Flow:
  This is a flow-control mechanism in place for HTTP trigger to tackle only one request at a time. We are using a shared data mechanism to set and get the JOB status and make sure that no two jobs will run concurrently. Make sure to configure the invoke activity for MESH service to call the

trigger to initiate the JOB.


PublishGTFS_v2 includes 8 flows

ControlFLow

This will keep control over RealTime-TripAndPositionUpdates job execution and allow only one job to run at time

General/tibco-wi-restinvoke

InvokeRESTService

Simple REST Activity

Discover services from TIBCO Cloud Mesh: [✓ True] False

Browse TIBCO Cloud Mesh Service: Browse... PublishGTFS_v2

Resource Path: /v1/gtfs/buses

Method: GET

Timeout

- **SubFlows**:
  - SendNormalFeed

    This subflow will call the private endpoint exposed through TIBCO Cloud MESH to send the set of messages that may have real-time trip updates and locations as well as only live positions without trip information. Normal feed in the sense of NO ALERT. You need to configure the Rest Invoke activity once you set-up the PublisheFTL application with a private endpoint.

  - SendAlertFeed

    This is responsible to send the received alert messages/entities to TCM-eFTL. This flow is configured to send the messages to TCM-eFTL using exposed ReST APIs.

  - GET-RealTimeTripUpdates

    This will subscribe to live GTFS feed of real-time trip updates in protobuf format, subsequently, it will convert that proto message into JSON using a custom function and will pass the generated payload to main flow for processing.

  - GET-RealTimePositions

    This flow is responsible to subscribe the live positions or alerts from MOCKAPI service and pass it to main flow for processing.

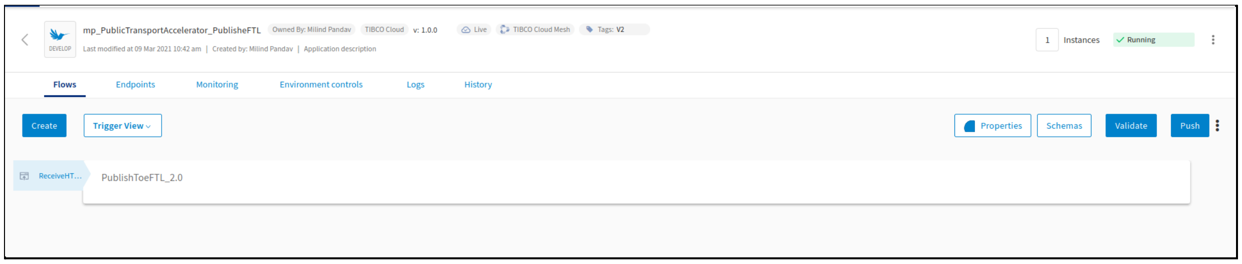
NOTE: *Please follow the performance tuning parameters suggested in the Performance tuning section of Accelerator for this application.*

❑ PublisheFTL

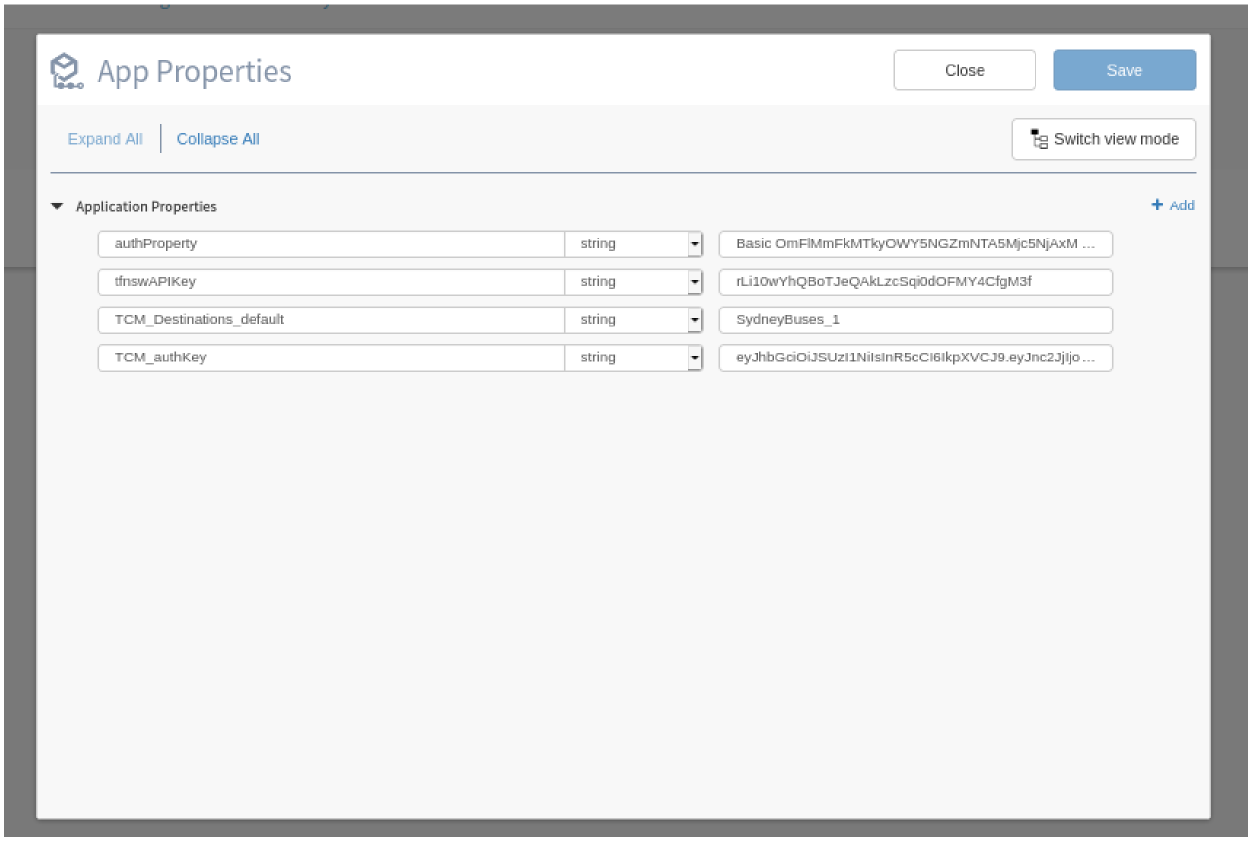
Visibility: *Private Endpoint (TIBCO Cloud MESH)*

Name: *SmartTransportAccelerator_PublisheFTL*

- ❑ Repeat the steps a. to h as outlined in the first application.



- ❑ Next, update the **Authentication** key for TCM-eFTL to publish the data, as shown in below snap.



authProperty	string	Basic OmFlMmFkMTkyOWY5NGZmNTA5Mjc5NjAxM ...
tfnswAPIKey	string	rLi10wYhQBoTJeQAkLzcSqi0dOFMY4CfgM3f
TCM_Destinations_default	string	SydneyBuses_1
TCM_authKey	string	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJnc2JjIjo ...

- ❑ At last, validate the application for any errors. If there is no error reported PUSH the app for deployment.

FLOW Details

- ❏ HTTP Based Trigger Flow:
  This is responsible for receiving the messages from PublishGTFS_v2 application NORMAL FEED and publishing them to the TCM-eFTL environment.

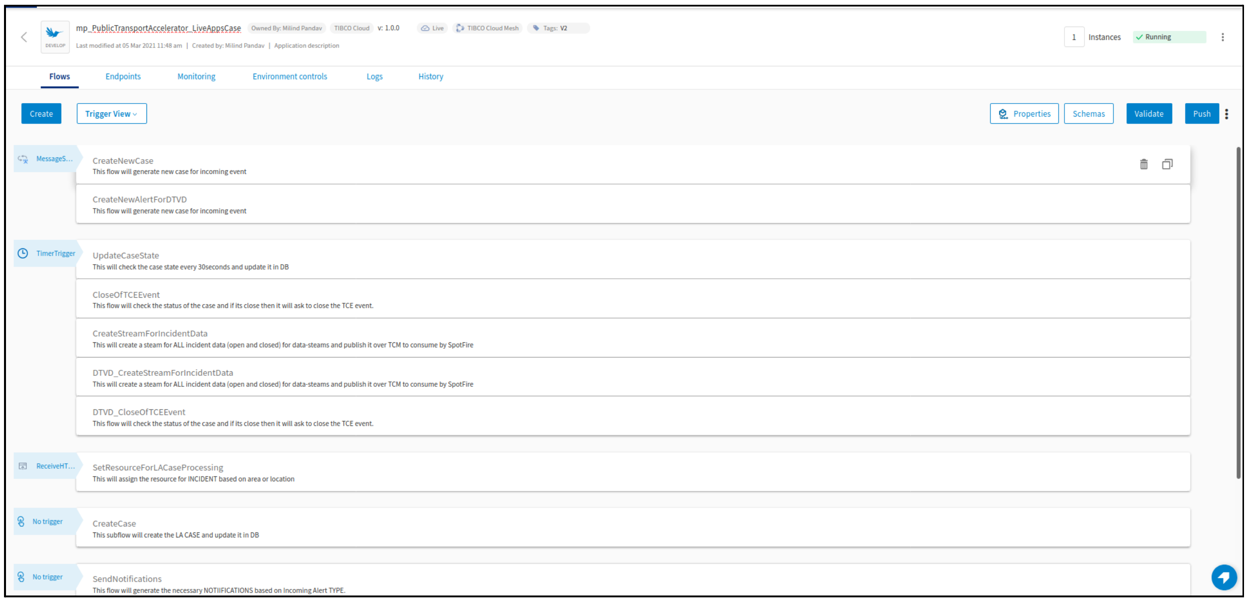
NOTE: *Please follow the performance tuning parameters suggested in the Performance tuning section of Accelerator for this application.*

- ❑ **LiveAppsCase**

  Visibility: *Private Endpoint (TIBCO Cloud MESH)*

  Name: *SmartTransportAccelerator_LiveAppsCase*

  - ❑ Repeat the steps a. to h. as outlined in the first application.



mp_PublicTransportAccelerator_LiveAppsCase Owned By: Milind Pandav TIBCO Cloud v: 1.0.0 Live TIBCO Cloud Mesh Tags: V2

Last modified at 05 Mar 2021 11:48 am | Created by: Milind Pandav | Application description

CreateNewCase
This flow will generate new case for incoming event

CreateNewAlertForDTVD
This flow will generate new case for incoming event

UpdateCaseState
This will check the case state every 30seconds and update it in DB

CloseOfTCEEvent
This flow will check the status of the case and if its close then it will ask to close the TCE event.

CreateStreamForIncidentData
This will create a steam for ALL incident data (open and closed) for data-steams and publish it over TCM to consume by SpotFire

DTVD_CreateStreamForIncidentData
This will create a steam for ALL incident data (open and closed) for data-steams and publish it over TCM to consume by SpotFire

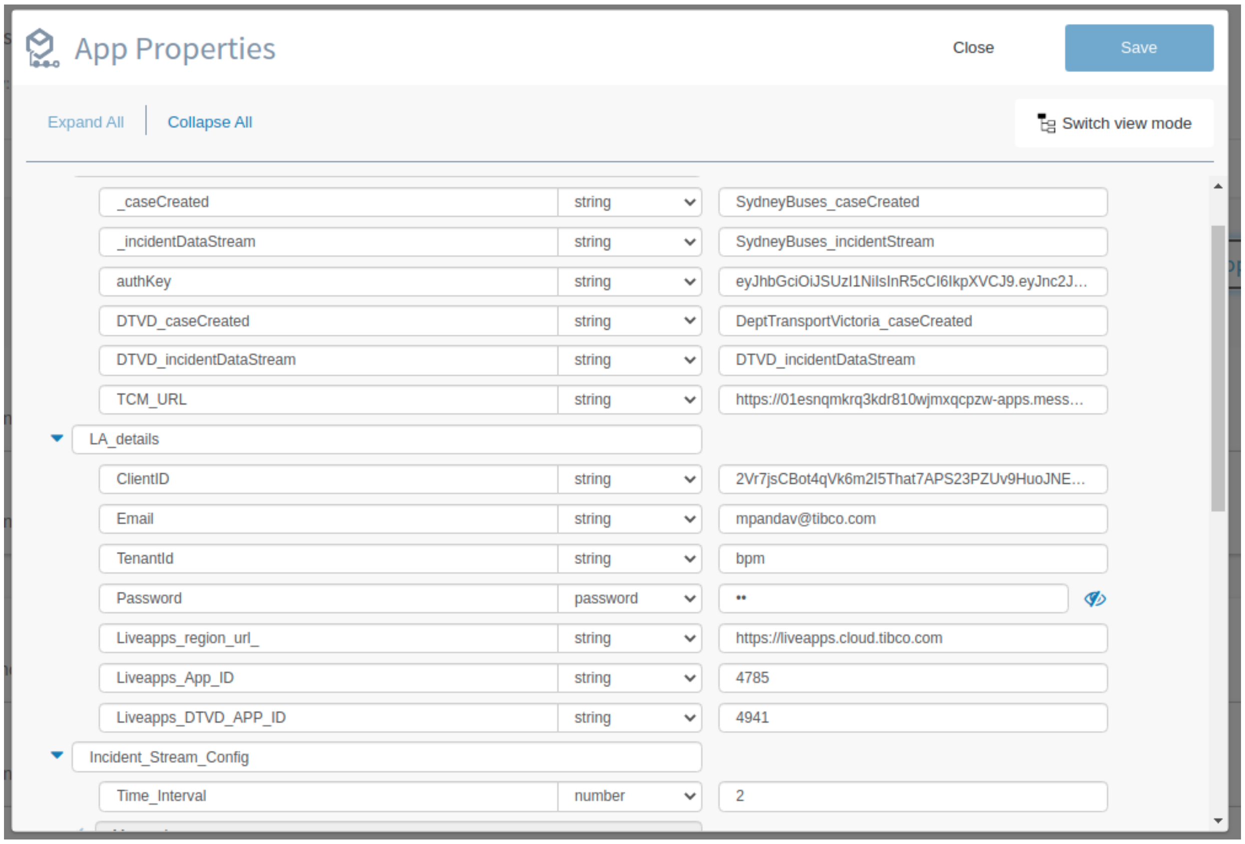
DTVD_CloseOfTCEEvent
This flow will check the status of the case and if its close then it will ask to close the TCE event.

SetResourceForLACaseProcessing
This will assign the resource for INCIDENT based on area or location

CreateCase
This subflow will create the LA CASE and update it in DB

SendNotifications
This flow will generate the necessary NOTIIFICATIONS based on Incoming Alert TYPE.

  - ❑ Next, update the application properties for the **Authentication key for** TCM-eFTL to publish the data. You need to update the authentication key and **TCM_URL** to connect to your TCM-eFTL endpoints as shown in the below snap.



_caseCreated	string	SydneyBuses_caseCreated
_incidentDataStream	string	SydneyBuses_incidentStream
authKey	string	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJnc2J...
DTVD_caseCreated	string	DeptTransportVictoria_caseCreated
DTVD_incidentDataStream	string	DTVD_incidentDataStream
TCM_URL	string	https://01esnqmkrq3kdr810wjmxqcpzw-apps.mess...
LA_details		
ClientID	string	2Vr7jsCBot4qVk6m2I5That7APS23PZUv9HuoJNE...
Email	string	mpandav@tibco.com
TenantId	string	bpm
Password	password	••
Liveapps_region_url_	string	https://liveapps.cloud.tibco.com
Liveapps_App_ID	string	4785
Liveapps_DTVD_APP_ID	string	4941
Incident_Stream_Config		
Time_Interval	number	2

- ❏ The **LA_details group** properties are responsible to interact with your **Liveapps subscription** and will help your app to connect with TIBCO Cloud Live Apps APIs. You need to update the **ClientID**, **Email** and **Password** as per your TIBCO Cloud Live Apps subscription along with **Liveapps_App_ID** and **Liveapps_region_url_**.
- ❏ This app also makes use of DB, so you need to update the **MySQLDB user and password** details according to your instance. The App has dependency over TCM-eFTL so you need to update TCM-eFTL **Messaging Connection details** in the connection section as per your subscription.
- ❏ The **Incident_Stream_Config** is used to configure the time window for incident data to be published to the Data Stream in hours. The default value is 2 hours.
- ❏ At last, validate the application for validations. If no error then publish the app.

FLOW Details

- ❏ TCM-eFTL Message Subscriber Trigger Flow

- ❏ CreateNewCase
  This flow is responsible for creating/calling the appropriate alerts case for each message received from TIBCO Cloud Events through TIBCO Cloud Messaging destination. It will return the CaseId and State of case on successful execution as a response to the received event.

❏ Timer Based Trigger Flow

- ❏ UpdateCaseState
  The timer based flow which keeps track of each case state and updates the state in the DB whenever there is any change in Case State with necessary details.

- ❏ CloseOffTCEEvent
  As soon as Case State changes to ‘Closed’ is detected, the flow will send the closure message to TIBCO Cloud Events to clear the Alert from its working memory. This correlation arrangement avoids the duplication of the alerts and Liveapps cases.

- ❏ CreateStreamForIncidentData
  The flow is responsible for producing a stream of data for DataStream and that subsequently will be visualised in Spotfire Dashboard.

❏ HTTP Based Trigger Flow

- ❏ SetResourceForLACaseProcessing
  The API is specifically designed for TIBCO Cloud Live Apps Case Flow. Whenever a new case is triggered this API will be executed to identify the resource to work on that particular case. This is an Auto Assignment of resources to the case.
  Currently, it’s assignment happens randomly, but one can use some business rules to select a specific set of users from TIBCO Cloud Live Apps subscription.
  The API uses third party API for reverse geocoding to get the location information that will be feed into the TIBCO Cloud Live Apps case

Ex.
Based on area/region or Type of Alert, etc.

❏ **SubFlows**

- ❏ CreateCase
  This flow is actually responsible for handling the case creation through Liveapps connector, add newly created Alert to DB for tracking and send response for TCE through main flow.

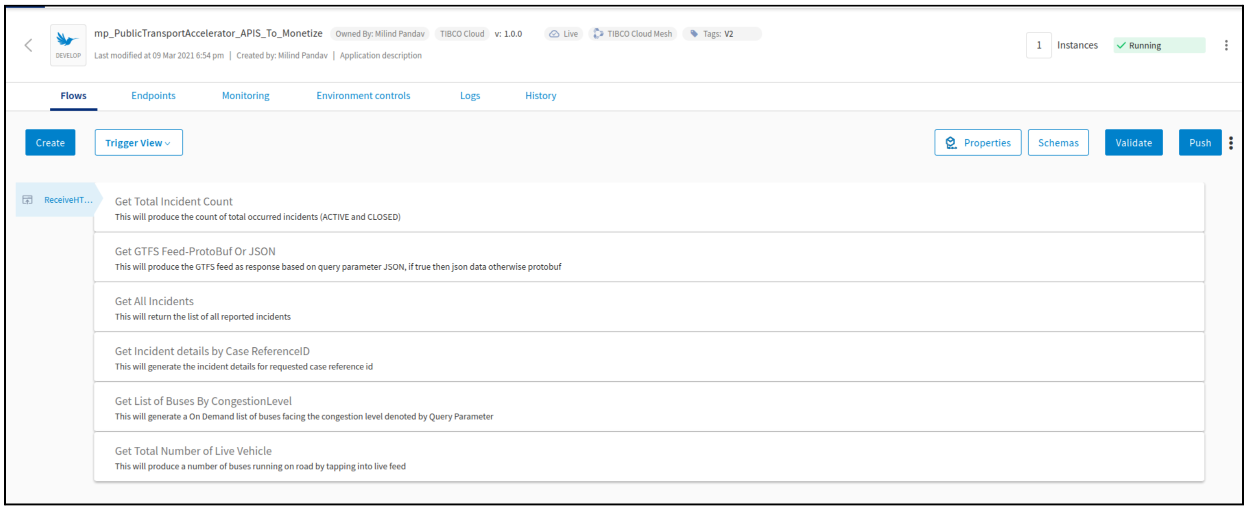
- ❏ SendNotifications
  This is a place holder created for users to add the choice of notification mechanism whenever any new alert is created. The notification could be SMS, E-Mail, Phone Call, etc. We're allowing users to choose their own option.

❑ <u>**APIS_To_Monetize**</u>

Visibility: *Private Endpoint (TIBCO Cloud MESH)*

Name: *SmartTransportAccelerator_API_To_Monetize*

- ❑ Repeat the steps a. to h. As outlined in the first application.



mp_PublicTransportAccelerator_APIS_To_Monetize v: 1.0.0

Last modified at 09 Mar 2021 6:54 pm | Created by: Milind Pandav | Application description

Get Total Incident Count
This will produce the count of total occurred incidents (ACTIVE and CLOSED)

Get GTFS Feed-ProtoBuf Or JSON
This will produce the GTFS feed as response based on query parameter JSON, if true then json data otherwise protobuf

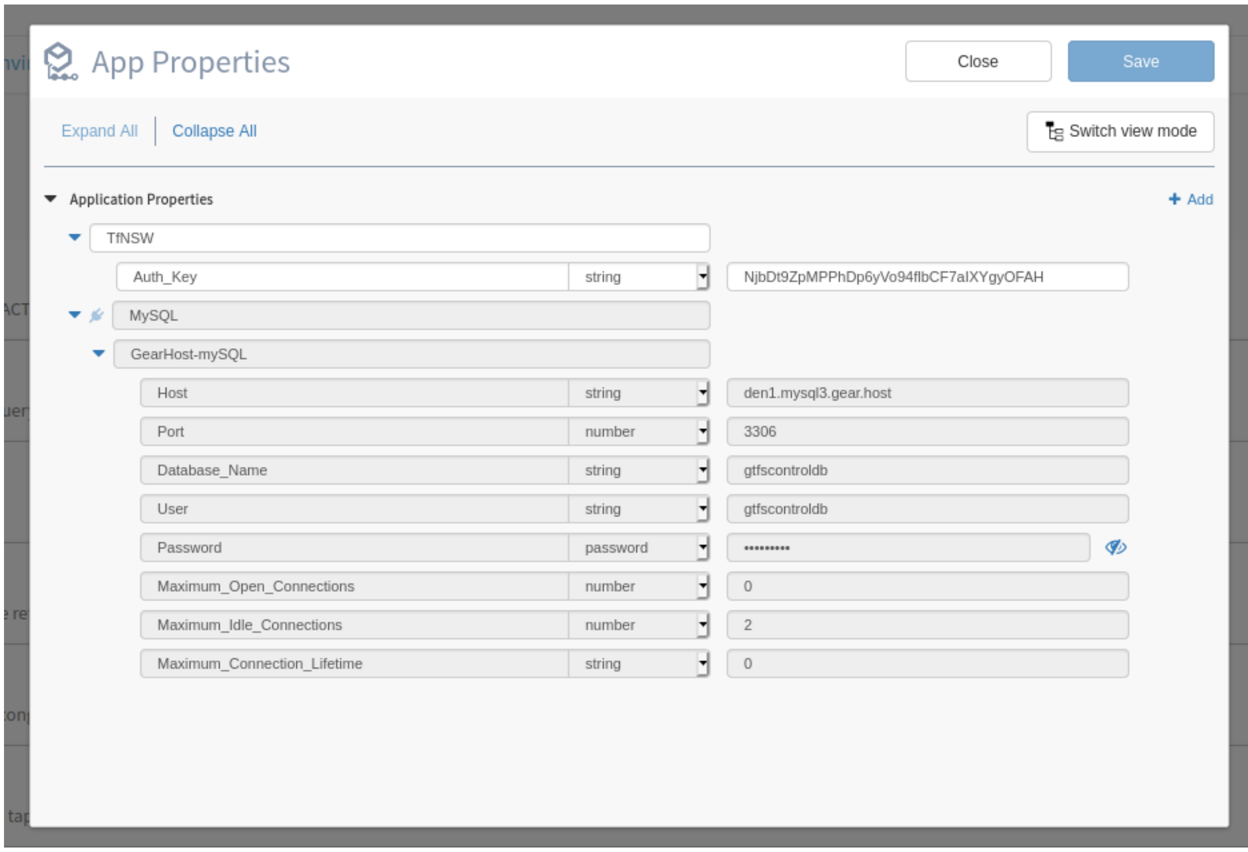
Get All Incidents
This will return the list of all reported incidents

Get Incident details by Case ReferenceID
This will generate the incident details for requested case reference id

Get List of Buses By CongestionLevel
This will generate a On Demand list of buses facing the congestion level denoted by Query Parameter

Get Total Number of Live Vehicle
This will produce a number of buses running on road by tapping into live feed

❑ Users need to update the **API KEY as well as DB user/password** details. As shown in the Application properties editor.



TfNSW		
Auth_Key	string	NjbDt9ZpMPPhDp6yVo94flbCF7alXYgyOFAH
MySQL		
GearHost-mySQL		
Host	string	den1.mysql3.gear.host
Port	number	3306
Database_Name	string	gtfscontroldb
User	string	gtfscontroldb
Password	password	•••••••••
Maximum_Open_Connections	number	0
Maximum_Idle_Connections	number	2
Maximum_Connection_Lifetime	string	0

❑ At last, look for any errors using the validate option and push the application.

FLOW Details

❑ HTTP Based Trigger Flows
We have GET operations implemented here to allow users to tap into real time data.

- ❑ Get Total Incident Count
  This will produce the count of total occurred incidents (ACTIVE and CLOSED)

- ❑ Get GTFS Feed-ProtoBuf Or JSON
  This will produce the GTFS feed as response based on query parameter JSON, if true then json data otherwise protobuf.

- ❑ Get All Incidents

This will return the list of all reported incidents

- ❑ Get Incident details by Case ReferenceID
  This will generate the incident details for requested case reference id.

- ❑ Get List of Buses By CongestionLevel
  This will generate a On Demand list of buses facing the congestion level denoted by Query Parameter

- ❑ Get Bus Info by ID
  This will produce the live details of given vehicle id by tapping into live feed

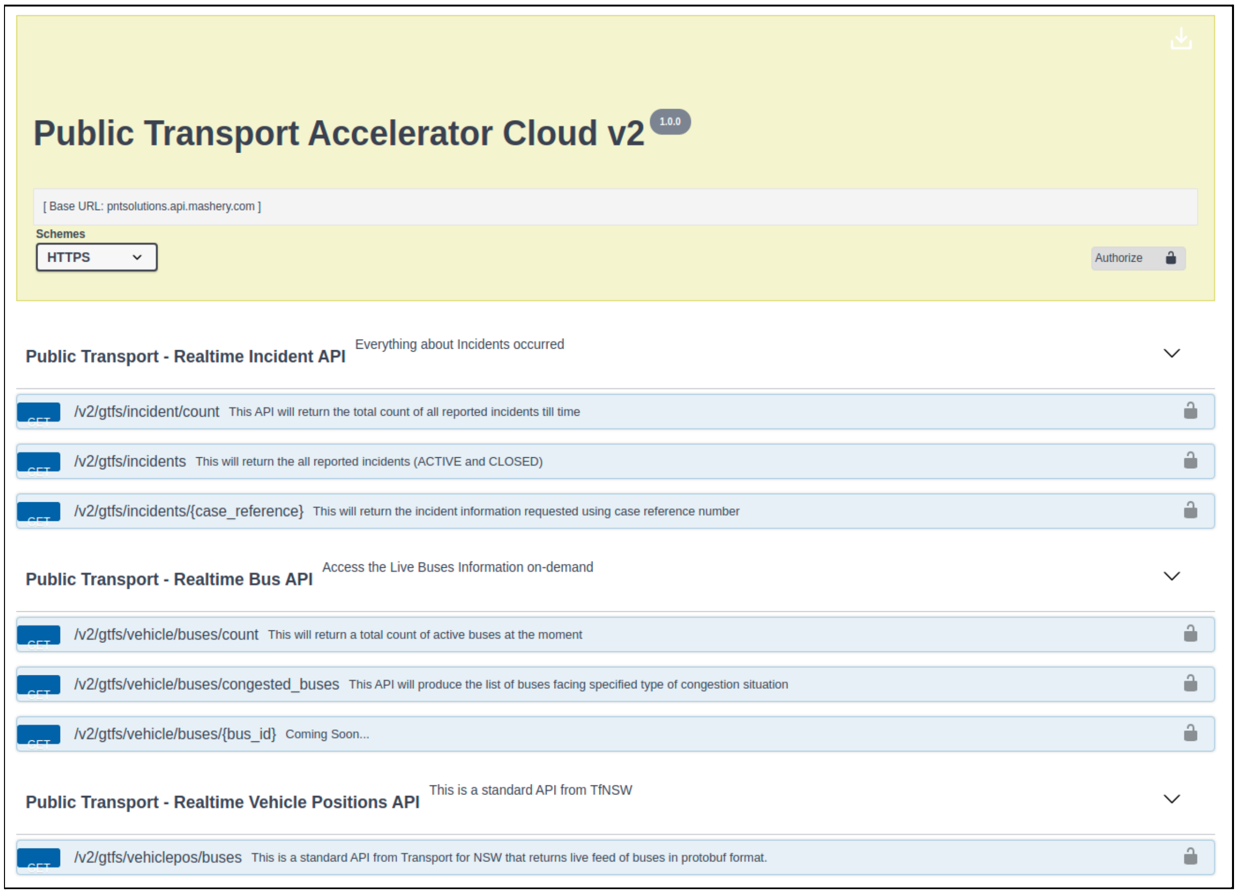
- ❑ Get Total Number of Live Vehicle
  This will produce a number of buses running on road by tapping into live feed

Developer Portal View

The API will be exposed to TIBCO Cloud Mashery using Responsive Application MESH (private endpoints) technology and will be available for end customers with and without authentication (API_KEY) as per the requirement.

The **IO documentation** for these APIs will be available under ~/../src/TC-Mashery/IO_Documentation_SmartTransportAccelerator_v2.json

❏ The exposed API will be available with swagger UI to test it from the browser window as shown below:



Public Transport Accelerator Cloud v2 1.0.0

[Base URL: pntsolutions.api.mashery.com]

Schemes: HTTPS

Authorize

Public Transport - Realtime Incident API Everything about Incidents occurred

- GET /v2/gtfs/incident/count This API will return the total count of all reported incidents till time
- GET /v2/gtfs/incidents This will return the all reported incidents (ACTIVE and CLOSED)
- GET /v2/gtfs/incidents/{case_reference} This will return the incident information requested using case reference number

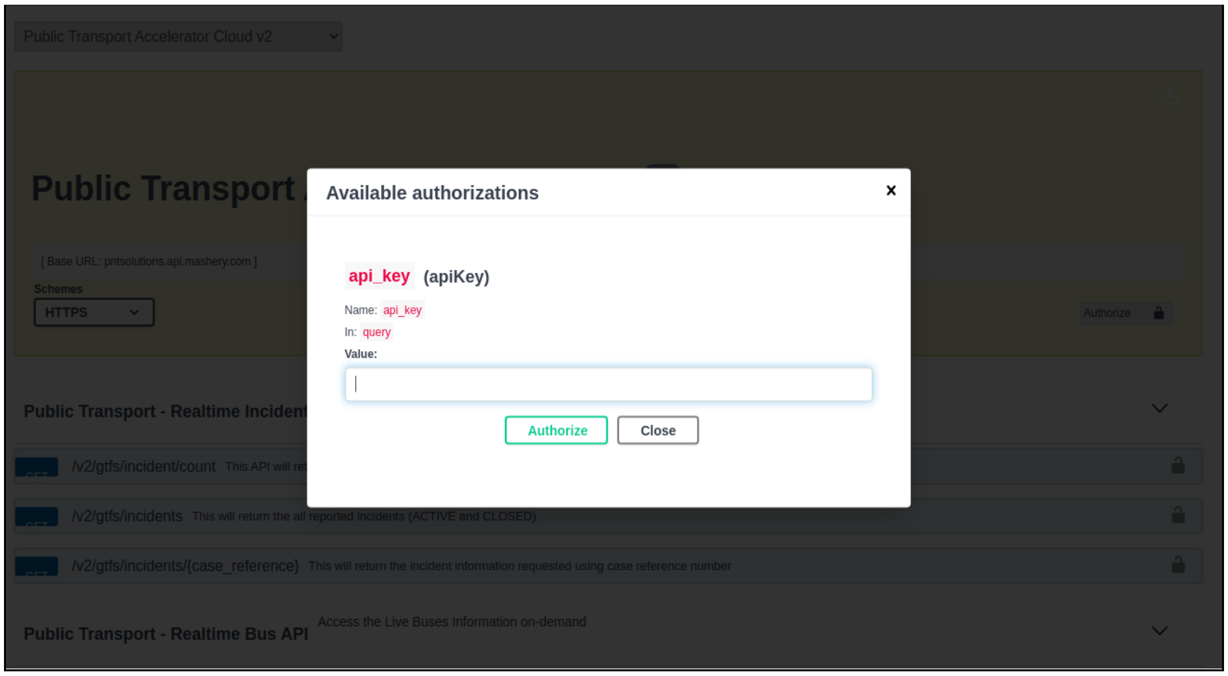
Public Transport - Realtime Bus API Access the Live Buses Information on-demand

- GET /v2/gtfs/vehicle/buses/count This will return a total count of active buses at the moment
- GET /v2/gtfs/vehicle/buses/congested_buses This API will produce the list of buses facing specified type of congestion situation
- GET /v2/gtfs/vehicle/buses/{bus_id} Coming Soon...

Public Transport - Realtime Vehicle Positions API This is a standard API from TfNSW

- GET /v2/gtfs/vehiclepos/buses This is a standard API from Transport for NSW that returns live feed of buses in protobuf format.

❑ Securing the API's with **API_KEY**



Public Transport Accelerator Cloud v2

Public Transport

[Base URL: pntsolutions.api.mashery.com]

Schemes: HTTPS

Available authorizations

api_key (apiKey)

Name: api_key

In: query

Value:

Public Transport - Realtime Incident

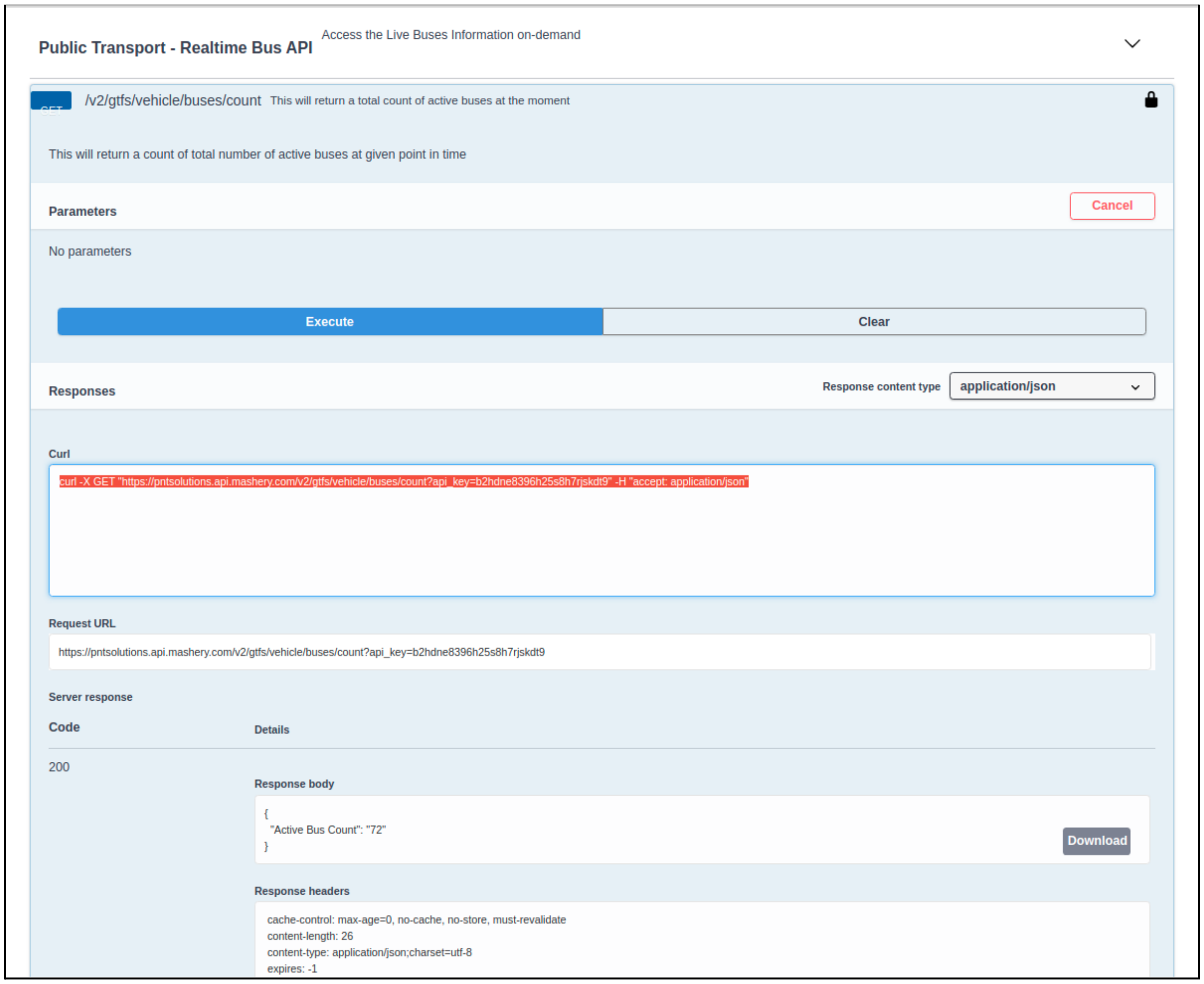
GET /v2/gtfs/incident/count This API will ret

GET /v2/gtfs/incidents This will return the all reported incidents (ACTIVE and CLOSED)

GET /v2/gtfs/incidents/{case_reference} This will return the incident information requested using case reference number

Public Transport - Realtime Bus API Access the Live Buses Information on-demand

❑ Accessing API using swagger UI from browser.


Public Transport - Realtime Bus API

Access the Live Buses Information on-demand

GET /v2/gtfs/vehicle/buses/count This will return a total count of active buses at the moment

This will return a count of total number of active buses at given point in time

Parameters

No parameters

Responses

Response content type: application/json

Curl

```
curl -X GET "https://pntsolutions.api.mashery.com/v2/gtfs/vehicle/buses/count?api_key=b2hdne8396h25s8h7rjskdt9" -H "accept: application/json"
```

Request URL

```
https://pntsolutions.api.mashery.com/v2/gtfs/vehicle/buses/count?api_key=b2hdne8396h25s8h7rjskdt9
```

Server response

Code	Details
200	Response body

```
{
 "Active Bus Count": "72"
}
```

Response headers

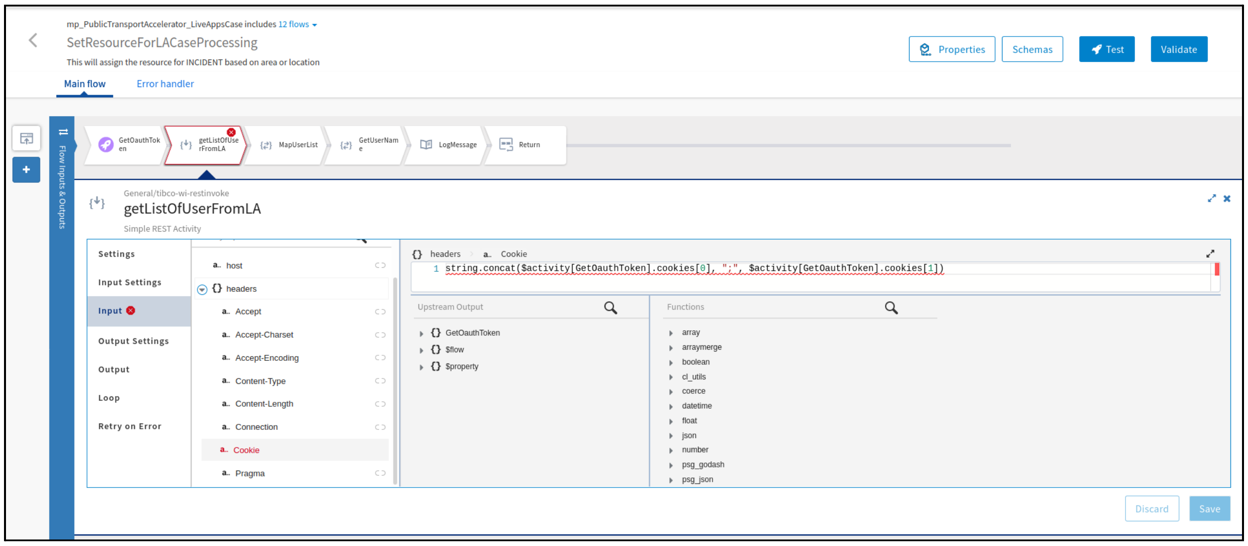
```
cache-control: max-age=0, no-cache, no-store, must-revalidate
content-length: 26
content-type: application/json;charset=utf-8
expires: -1
```

Known Issues

When you import the above applications into your subscription, you'll see some of the apps showing validation errors for input mappings that will lead you to think that application is a failure.
This is a known issue (FLOGO-6815) and will be addressed in future releases.

Below application has design-time validation (harmless) errors reported in it and that can be ignored as the app worked perfectly at runtime.

1. **SmartTransportAccelerator_LiveAppsCase**
   a. Set Resource For Case Processing



mp_PublicTransportAccelerator_LiveAppsCase includes 12 flows

SetResourceForLACaseProcessing

This will assign the resource for INCIDENT based on area or location

General/tibco-wi-restinvoke

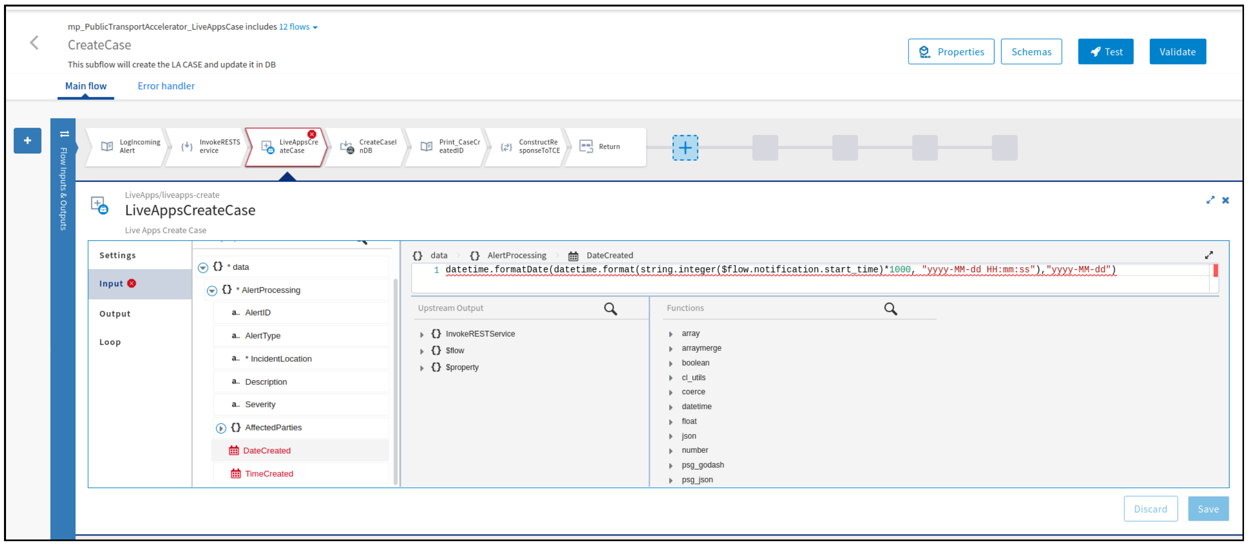
getListOfUserFromLA

Simple REST Activity

headers › Cookie

```
string.concat($activity[GetOauthToken].cookies[0], ";", $activity[GetOauthToken].cookies[1])
```

   b. Create Case



mp_PublicTransportAccelerator_LiveAppsCase includes 12 flows

CreateCase

This subflow will create the LA CASE and update it in DB

LiveApps/liveapps-create

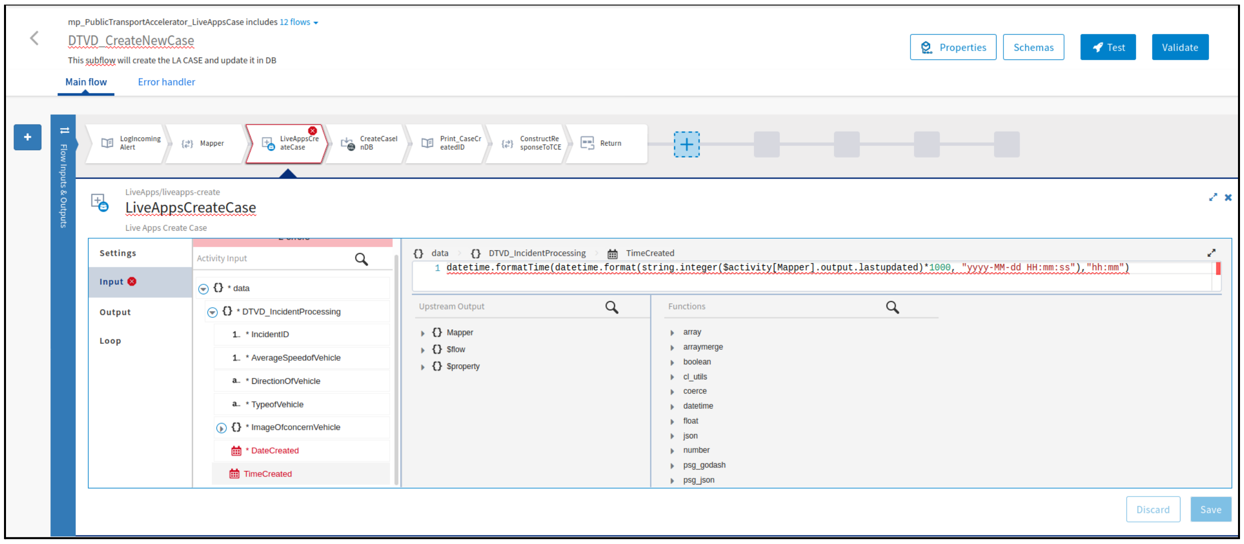
LiveAppsCreateCase

Live Apps Create Case

data › AlertProcessing › DateCreated

```
datetime.formatDate(datetime.format(string.integer($flow.notification.start_time)*1000, "yyyy-MM-dd HH:mm:ss"),"yyyy-MM-dd")
```

c. DTVD_CreateNewCase



mp_PublicTransportAccelerator_LiveAppsCase includes 12 flows

DTVD_CreateNewCase

This subflow will create the LA CASE and update it in DB

LiveApps/liveapps-create

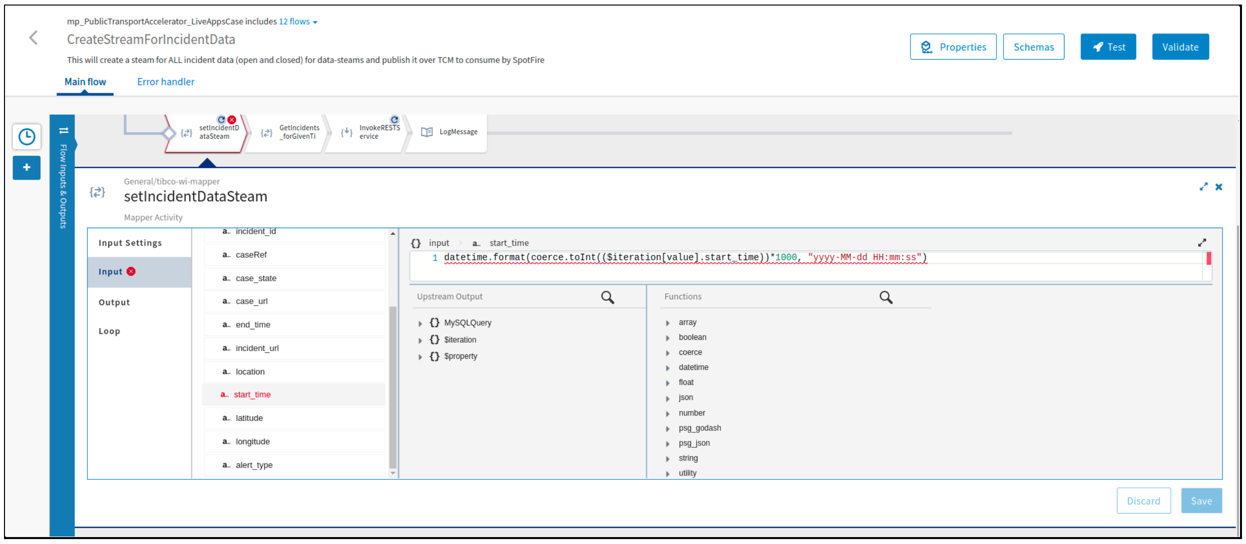
LiveAppsCreateCase

Live Apps Create Case

data DTVD_IncidentProcessing TimeCreated

```
datetime.formatTime(datetime.format(string.integer($activity[Mapper].output.lastupdated)*1000, "yyyy-MM-dd HH:mm:ss"),"hh:mm")
```

d. Create Stream For Incident Data



mp_PublicTransportAccelerator_LiveAppsCase includes 12 flows

CreateStreamForIncidentData

This will create a steam for ALL incident data (open and closed) for data-steams and publish it over TCM to consume by SpotFire

General/tibco-wi-mapper

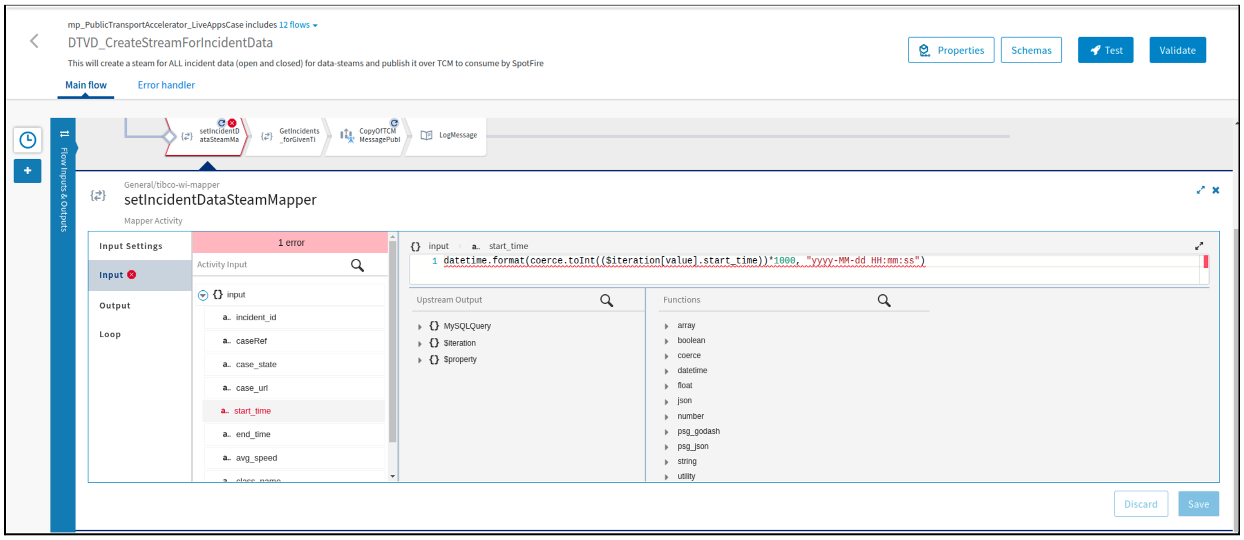
setIncidentDataStream

Mapper Activity

input a.. start_time

```
datetime.format(coerce.toInt(($iteration[value].start_time))*1000, "yyyy-MM-dd HH:mm:ss")
```

e. DTVD_Create Stream For IncidentData



mp_PublicTransportAccelerator_LiveAppsCase includes 12 flows

DTVD_CreateStreamForIncidentData

This will create a steam for ALL incident data (open and closed) for data-steams and publish it over TCM to consume by SpotFire

General/tibco-wi-mapper

setIncidentDataSteamMapper

Mapper Activity

input › start_time

```
datetime.format(coerce.toInt(($iteration[value].start_time))*1000, "yyyy-MM-dd HH:mm:ss")
```