**NAME: Suneetha REKAPALLI**

**EMPID: 802665**

**Real-Time Project Exercise – MuleSoft Transport Management**

# Objective:

TripEasy is a startup company providing a platform to perform booking services for land, air and sea transport across the world.

This platform will enable customers to query location routes and the departure schedule from multiple transport providers, and make booking directly to the platform. The platform will enable the customer to make payments, and issues tickets to the customers. Customer also will be able

This platform will connect to the respective transport providers back end systems, to query for the available routes and schedules and make purchases, track their booking and make changes to their bookings. This platform will also provide the capability for the regulatory agencies to manage their bookings, and all updates will be reflected to the transport provider back-end system.

# API Landscape

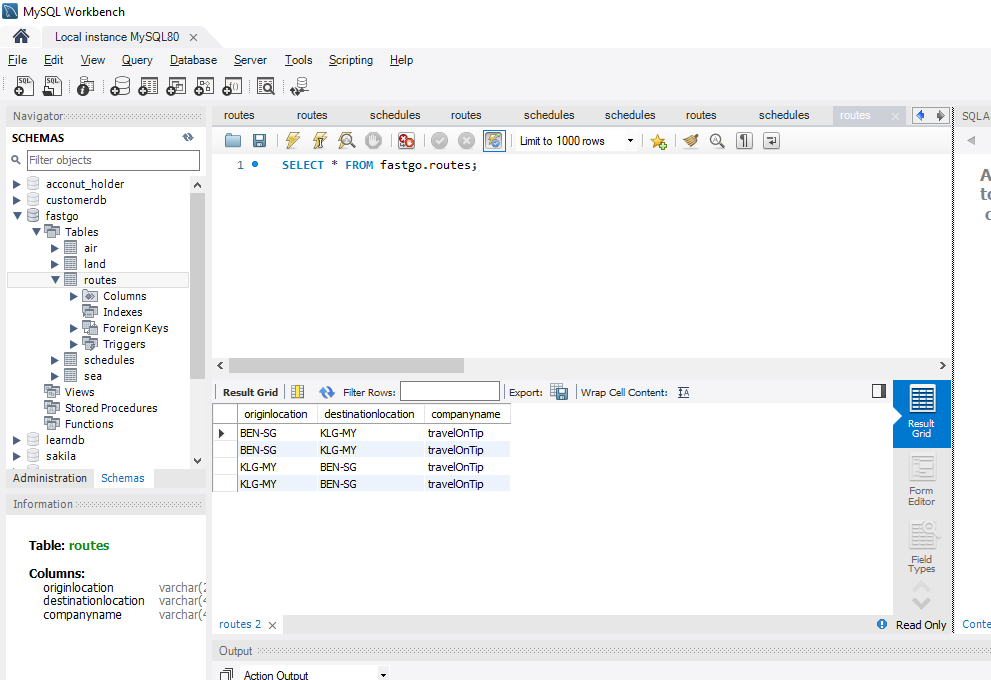
Diagram

Description automatically generated

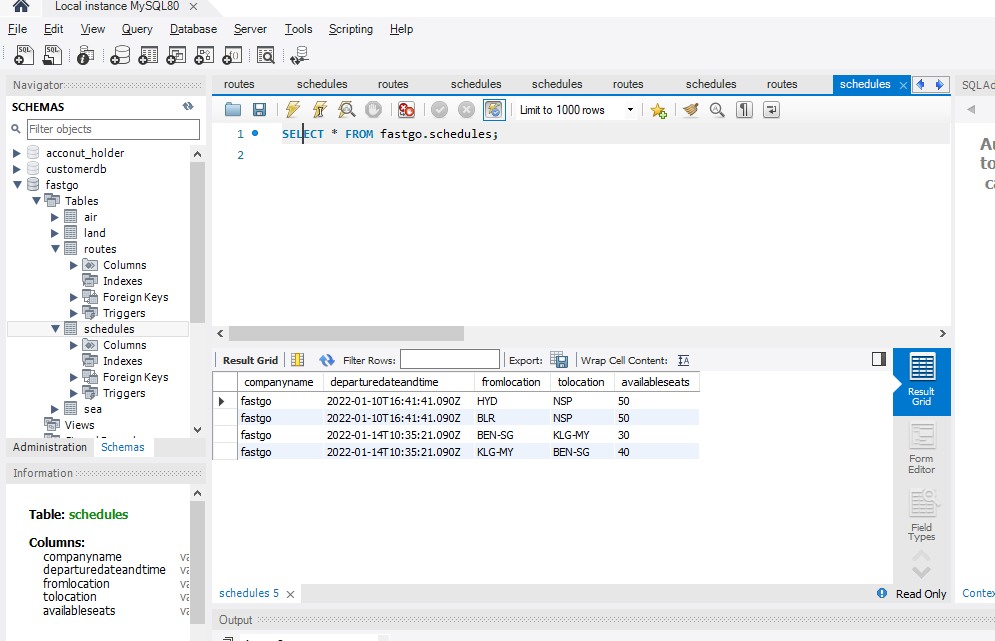
Process of Implementing the Project:

Step 1: Create databases Travelontip and Fastgo and create tables Routes and schedules in the respective databases.

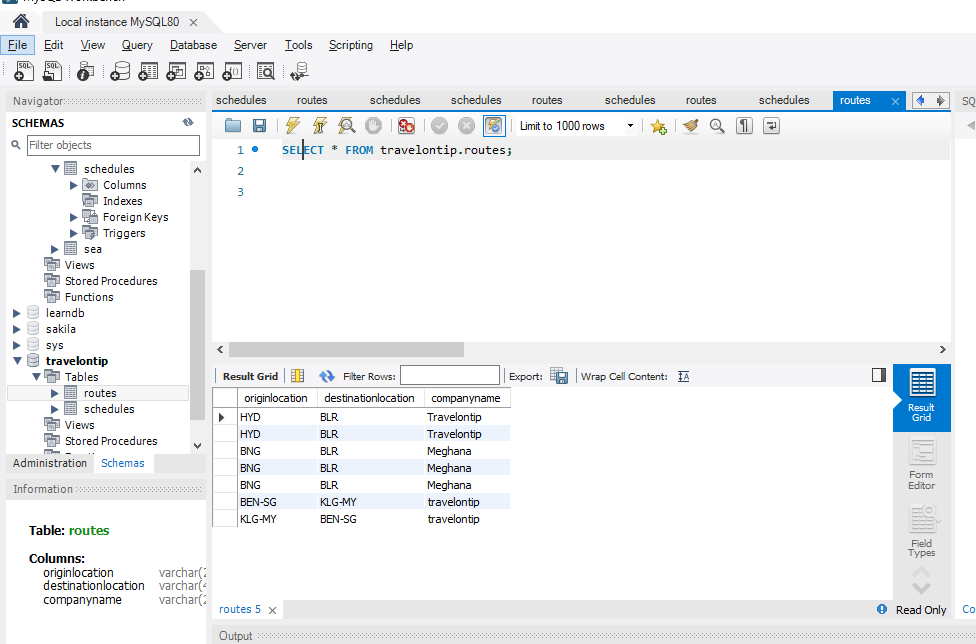
1. Fastgo database-routes table.



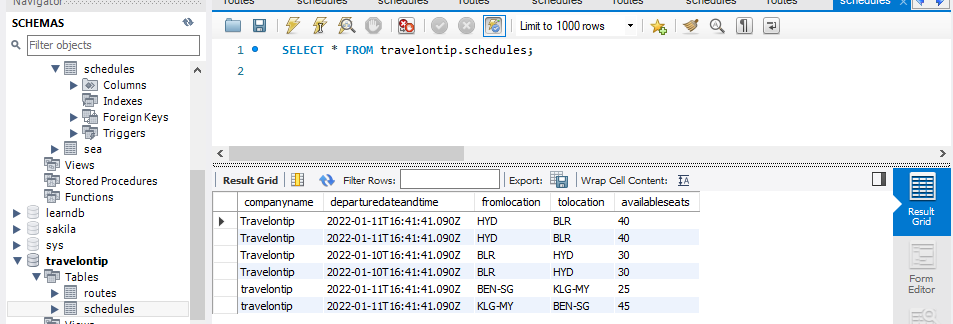
1. Fastgo database-Schedules table:



1. Travelontip database-Routes table:



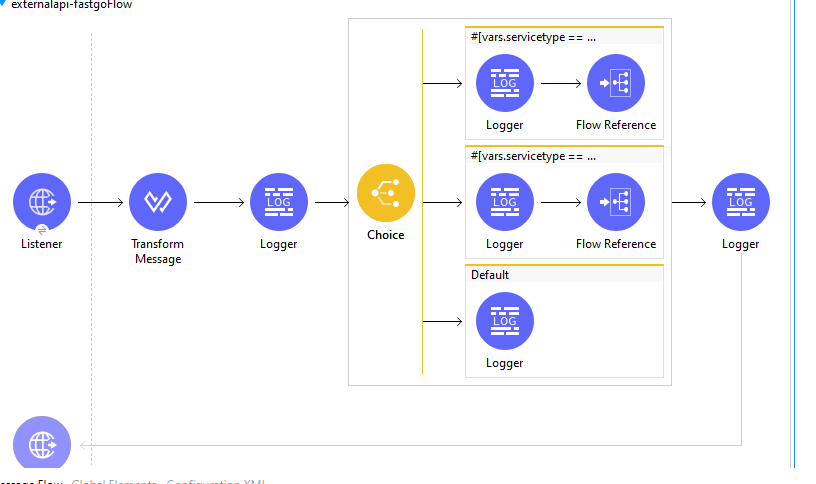
1. Travelontip database-Schedules table:



Step 2:

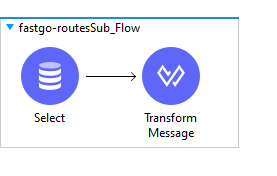
Create two External Systems Travelontip Booking System and Fastgo Booking system to read data from Travelontip and Fastgo databases.

**Fastgo external API:**

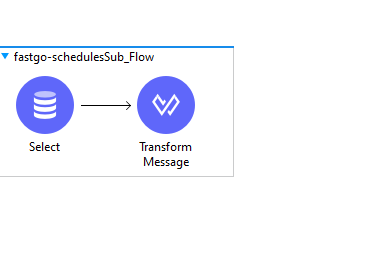


The flow represents the process of External API for Fastgo database:

* we are creating a variable “service type”, based on the service type the choice router component will executed.
* If servicetype == “routes” then corresponding subflow will called from flow referenece.

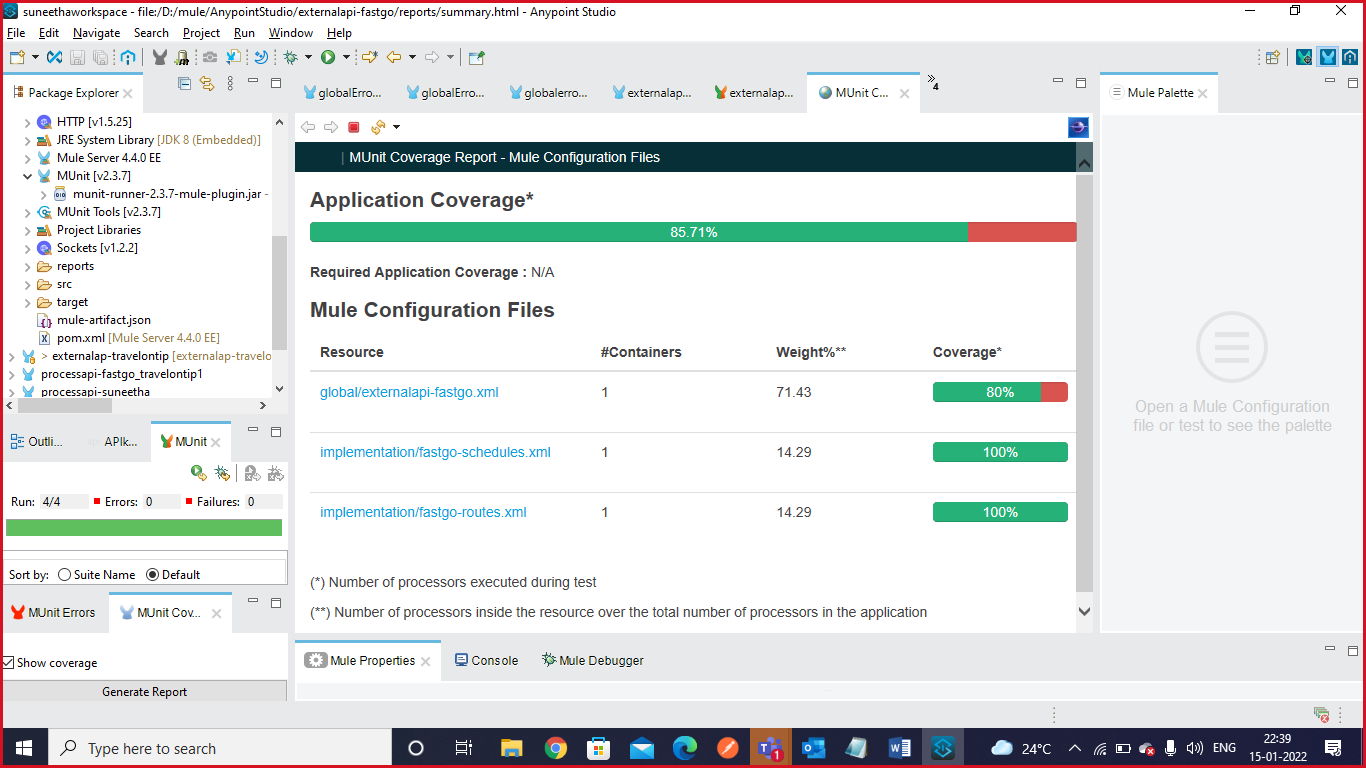


* If servicetype == “schedules” then corresponding subflow will called from flow referenece.

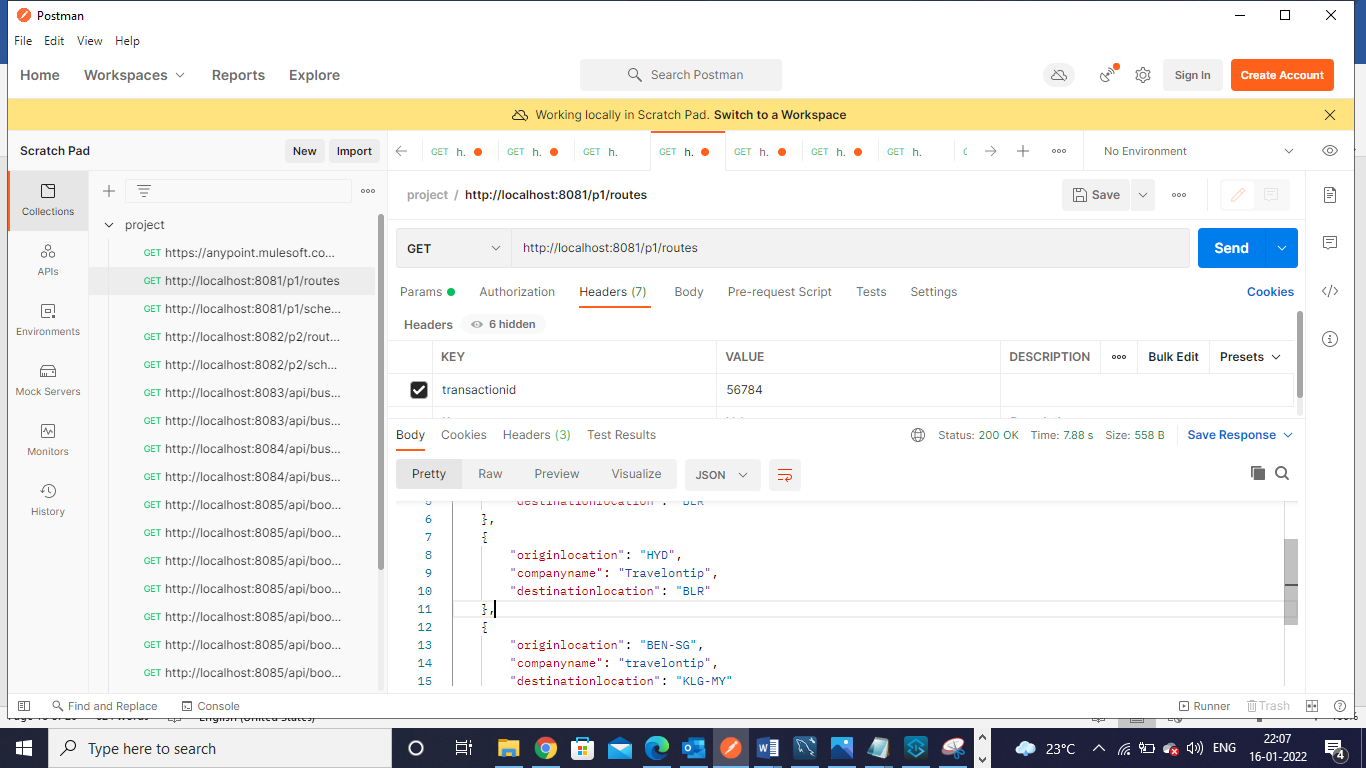


* In select we write a query for retrieving data from database. And in payload we get the details of the selected query.

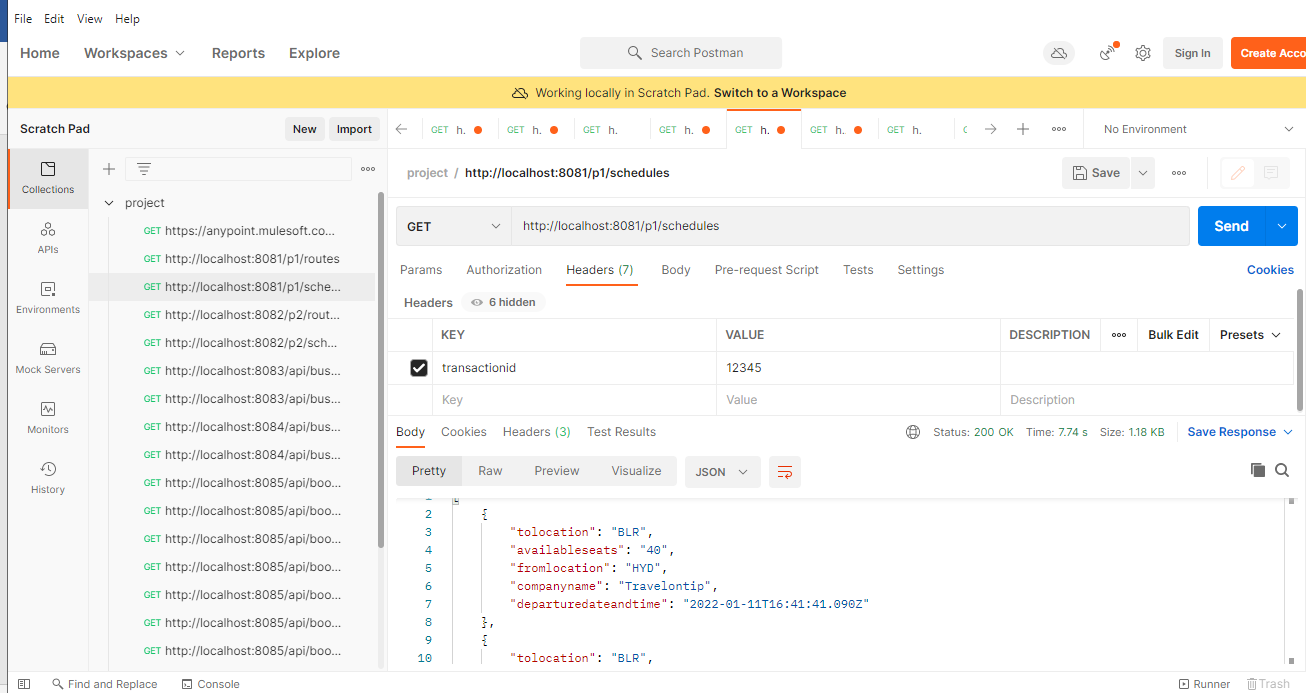
Test Coverage after Munit test:



Output at Postman:

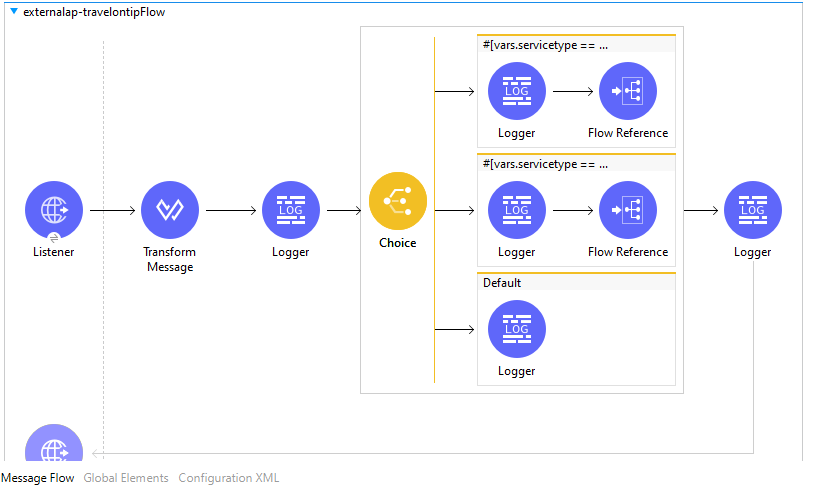


URL: <http://localhost:8081/p1/routes>



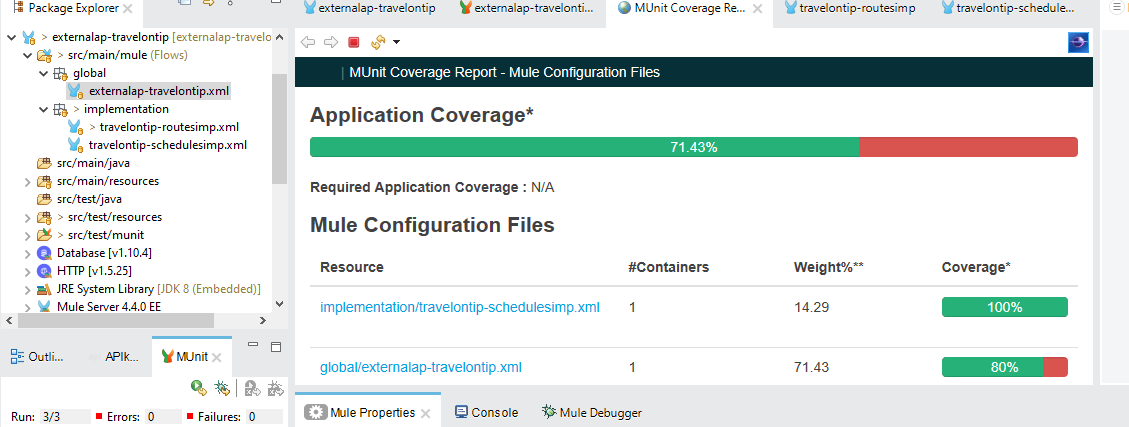
URL: <http://localhost:8081/p1/schedules>

**Travelontip external API:**

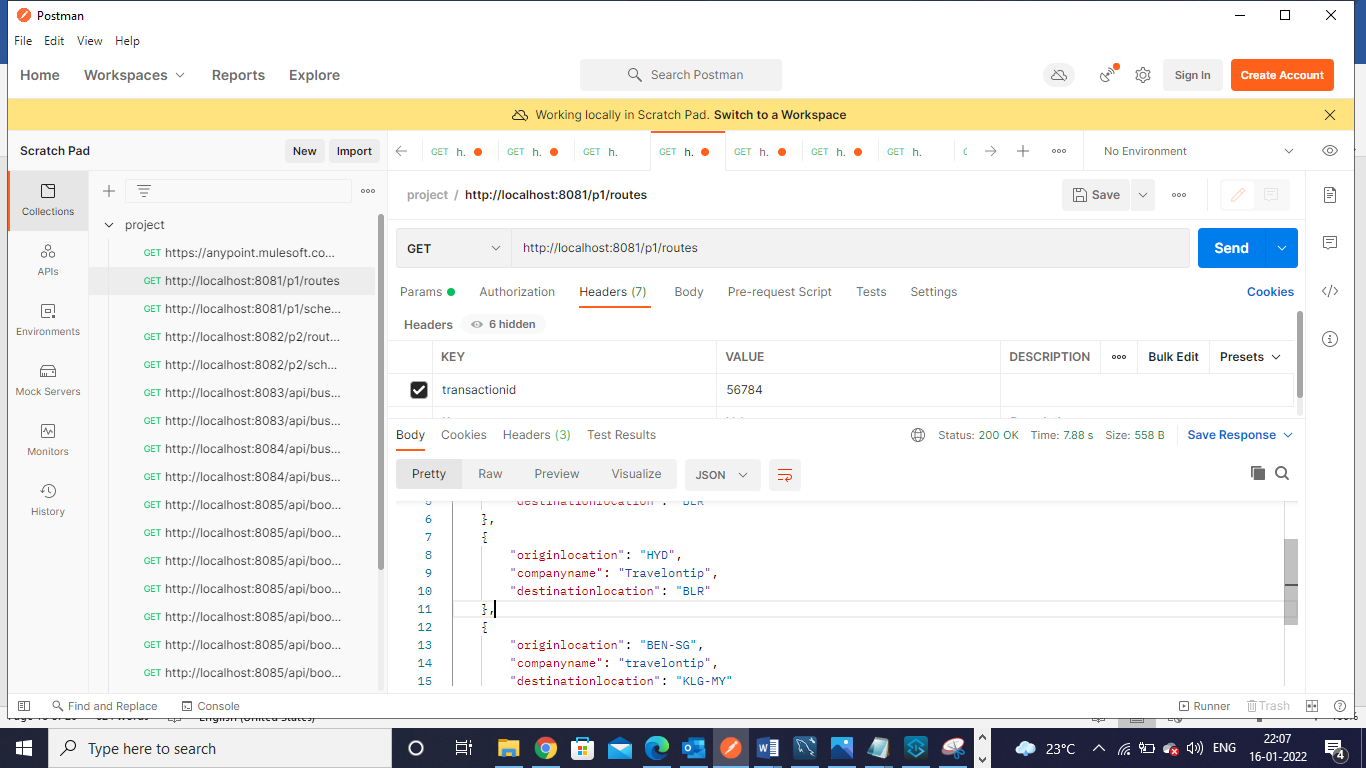


* As like Fastgo we can implement this API also.

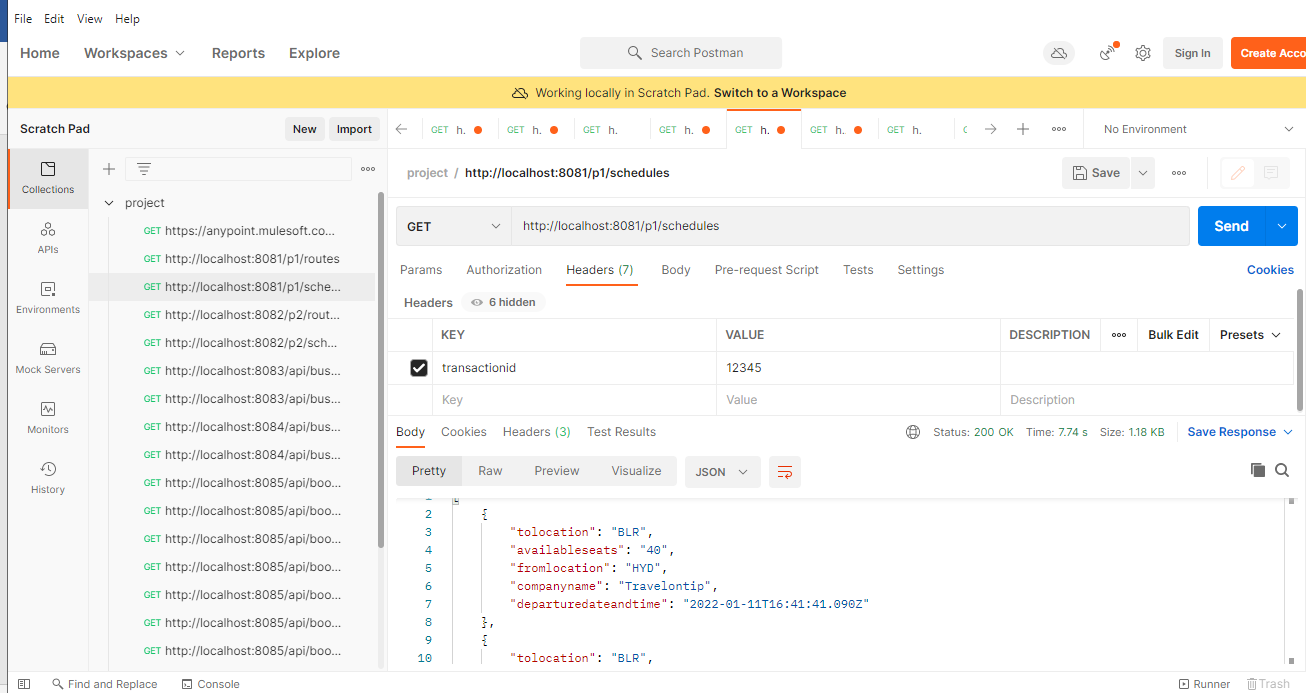
Test Coverage after Munit:



Output at Postman:



URL: <http://localhost:8081/p1/routes>

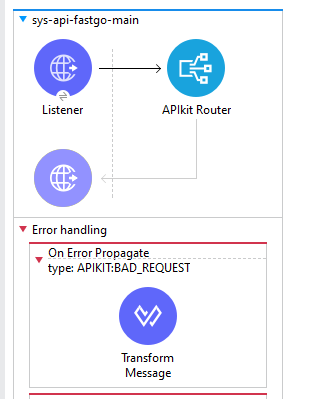


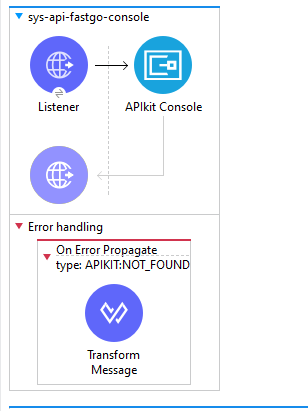
URL: <http://localhost:8081/p1/schedules>

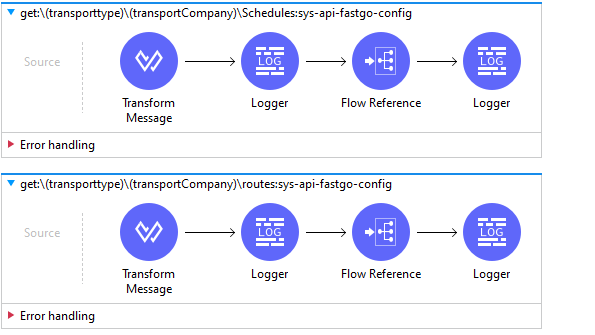
Step 3:

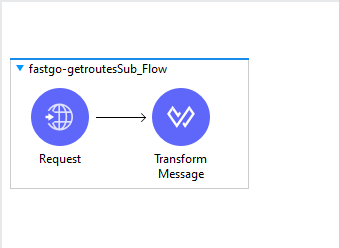
Create a System API for Fastgo and Travelontip based on RAML created on AnyPoint Platform.

**Systemapi-fastgo:**

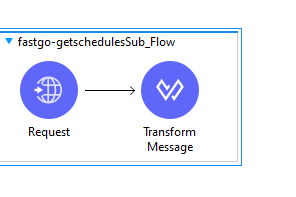




Use flow-reference component to call subflows. 

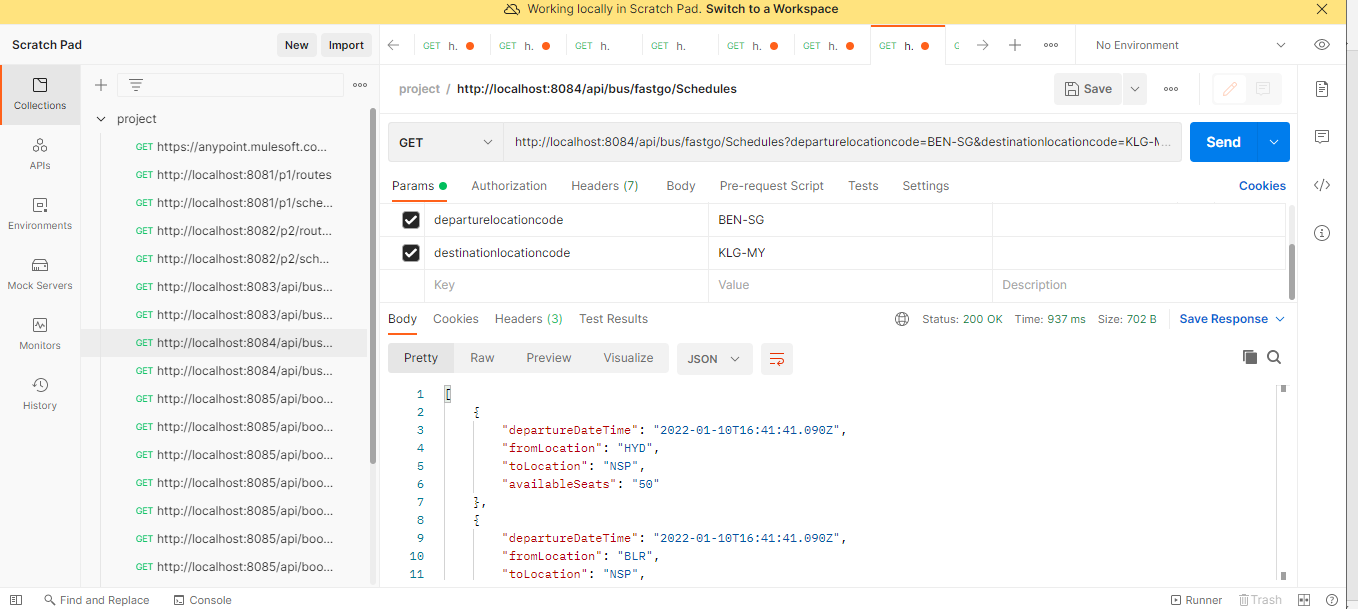


Sub flow for schedules

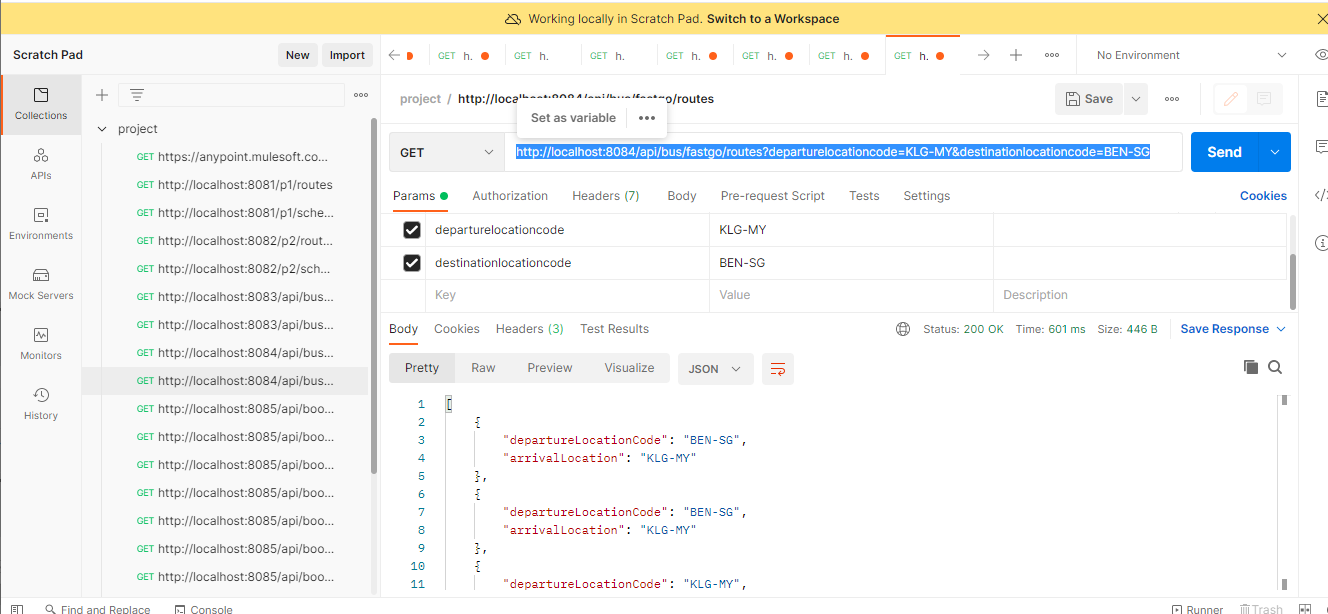


* In both subflows use Request Connector Component to consume external http connection of external-Fastgo API.
* Test Coverage after Munit:

Output at Postman:

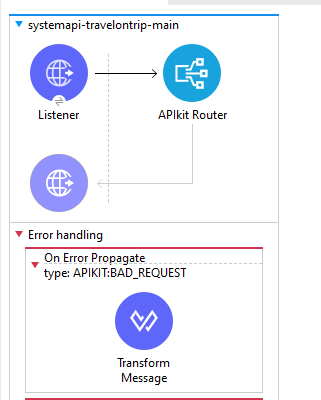


**URL:** <http://localhost:8084/api/bus/fastgo/Schedules?departurelocationcode=BEN-SG&destinationlocationcode=KLG-MY>



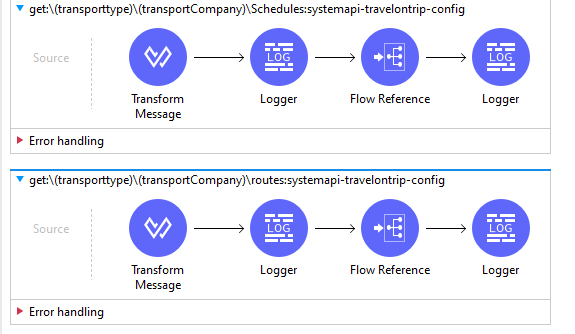
**URL:** <http://localhost:8084/api/bus/fastgo/routes?departurelocationcode=KLG-MY&destinationlocationcode=BEN-SG>

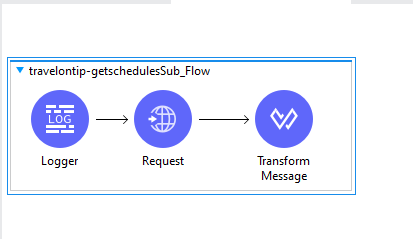
**Systemapi-travelontip:**

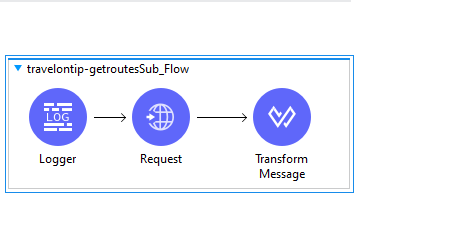


Use flow-reference component to call subflows. Subflows are implemented with

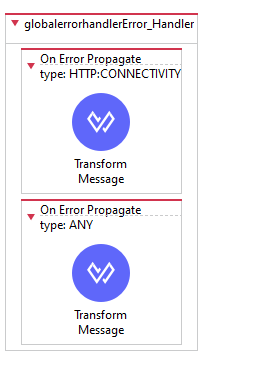
Request connectors used to consume external http connection of External API.



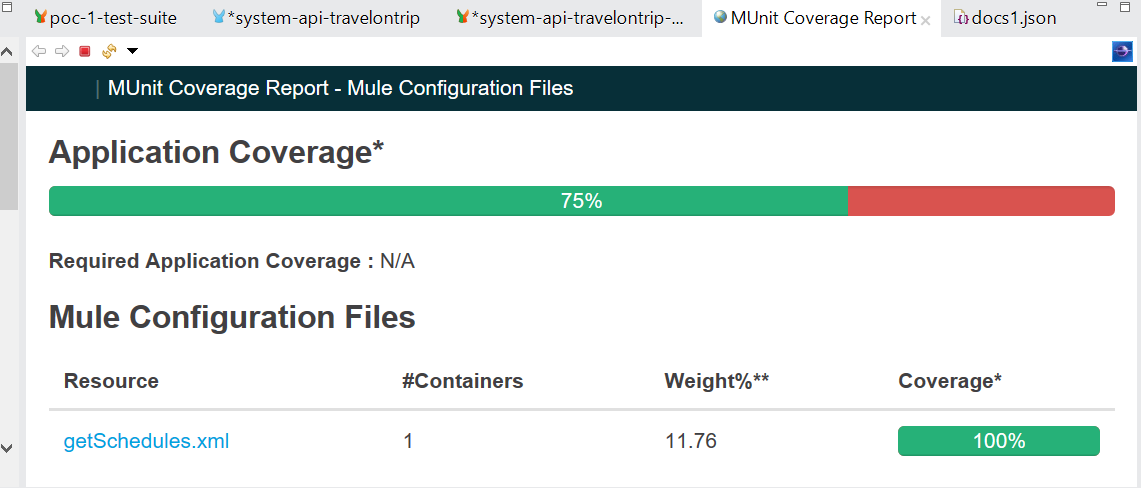
****

****

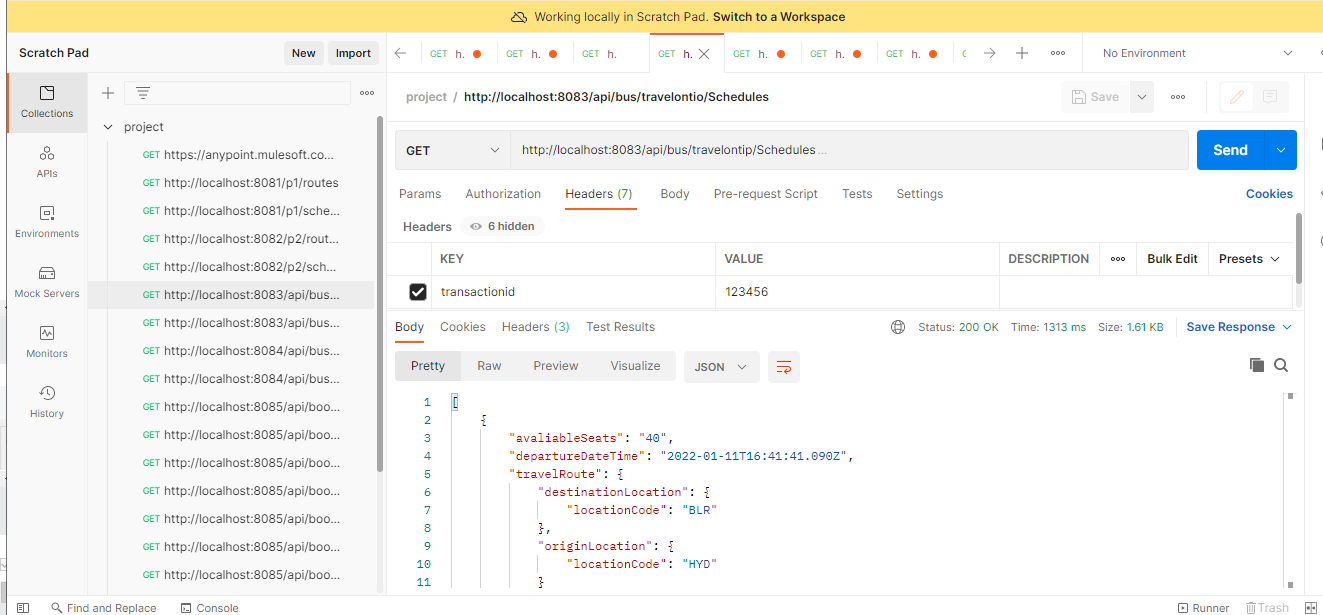
**U**se Error handler component to handle the error Globally.



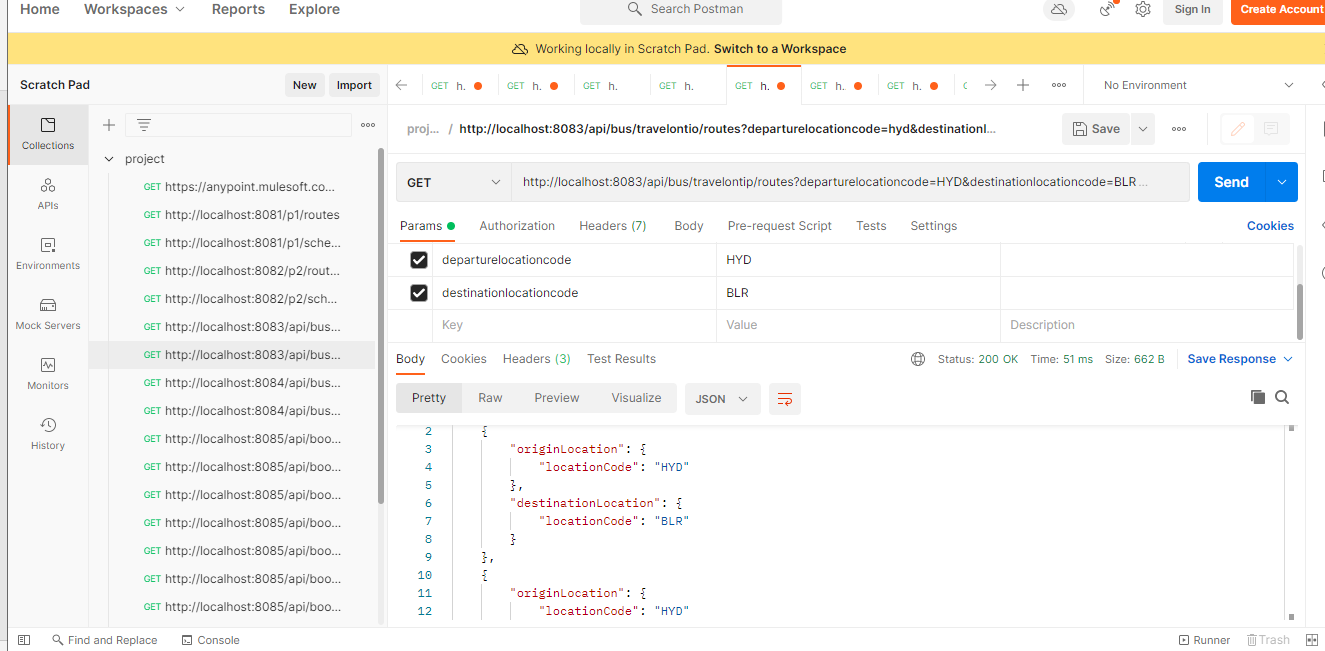
* Test Coverage after Munit:



Output at Postman:



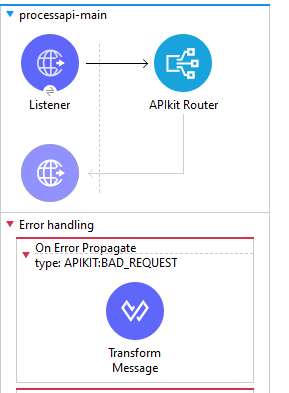
http://localhost:8083/api/bus/travelontip/Schedules



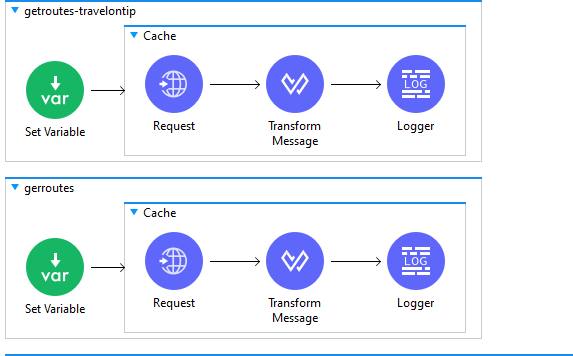
http://localhost:8083/api/bus/travelontip/routes?departurelocationcode=HYD&destinationlocationcode=BLR

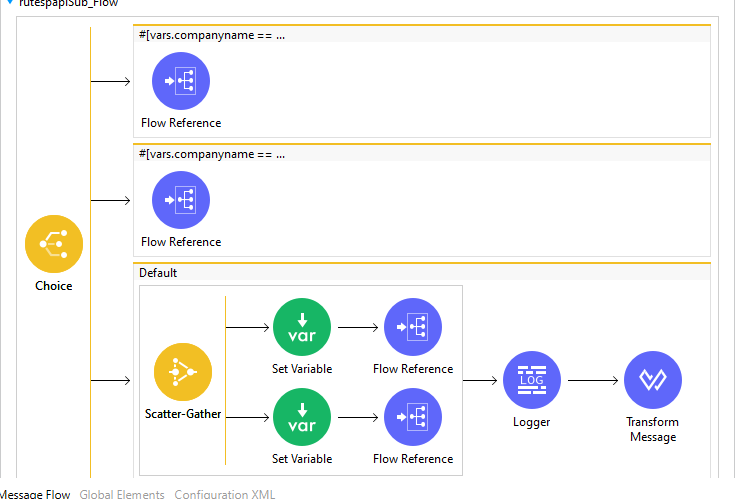
Step 4:

* Create a Process API for Fastgo and Travelontip as a consolidated API based on RAML created on AnyPoint Platform.
* Consume the both System Travelontip and Fastgo API’s to get consolidated Process API.

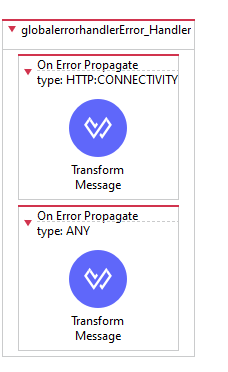


* Here we using choice router component to implement the flow.
* Based on companyname corresponding block is executed.
* If companyname is Fastgo, use request component to consume Systemapi-fastgo.
* If companyname is Travelontip, use request component to consume Systemapi-travelontip.
* If we don’t pass any companyname then default block executed with both routes and schedules, use request component to consume Systemapi-fastgo and Systemapi-travelontip.

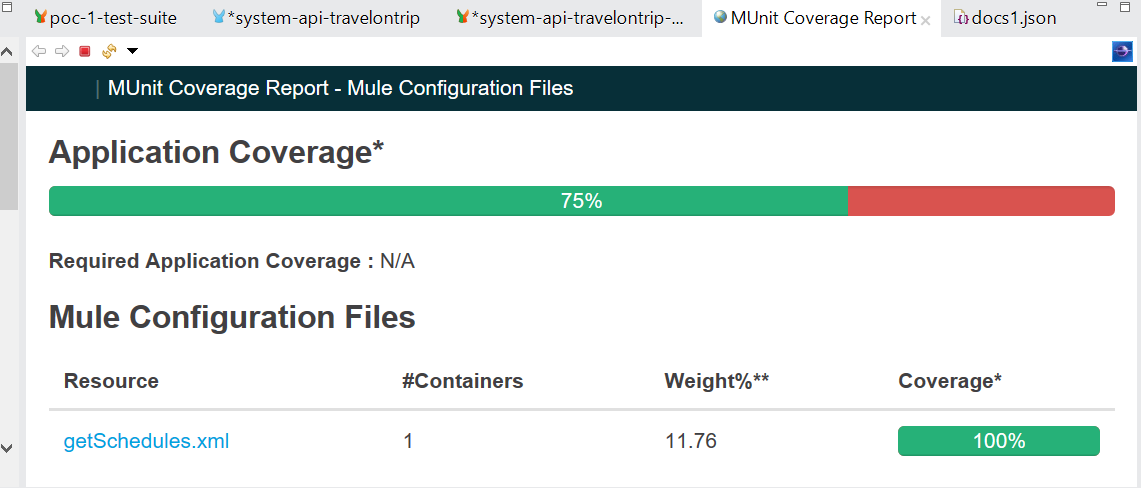




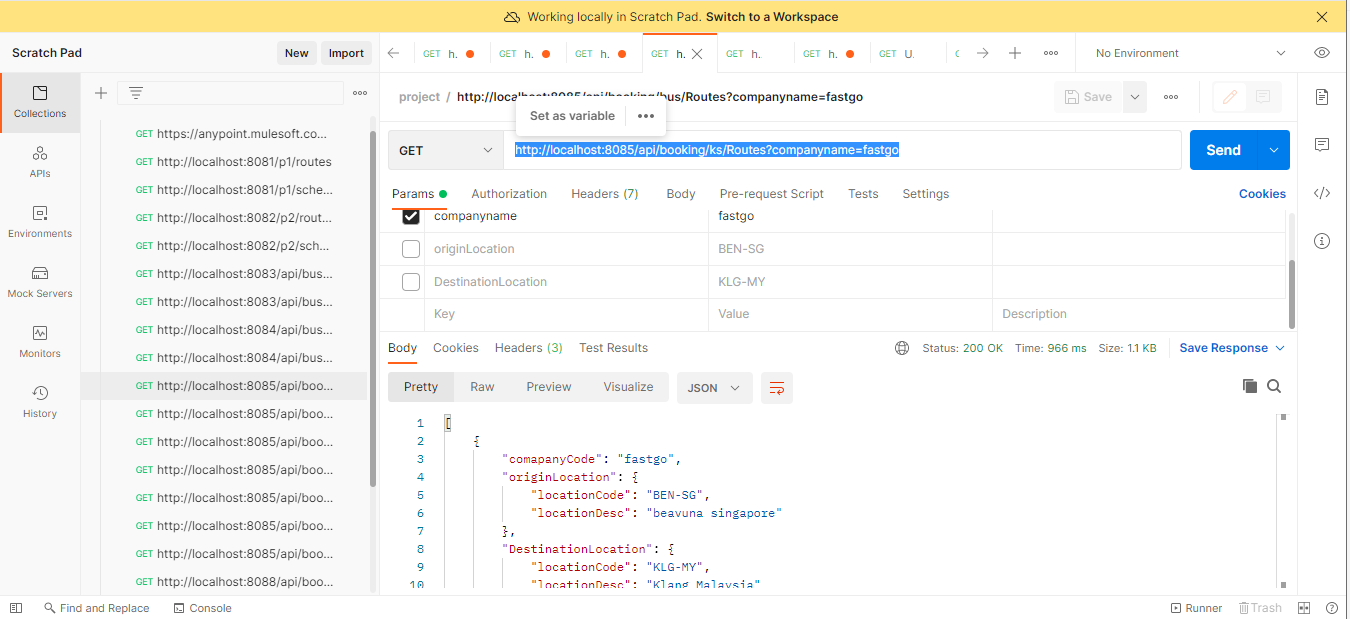
* Implement Error Component to handle errors Globally. If any error is raised which is not handled component and flow level then it is handled globally.



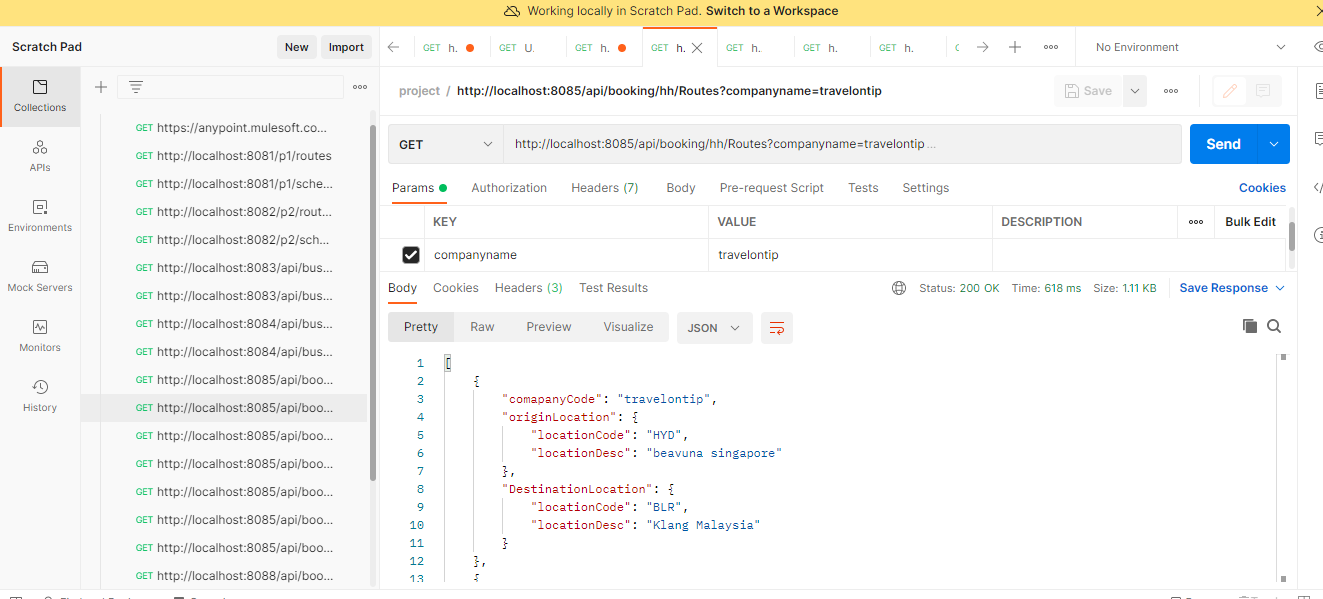
* Test Coverage after Munit:



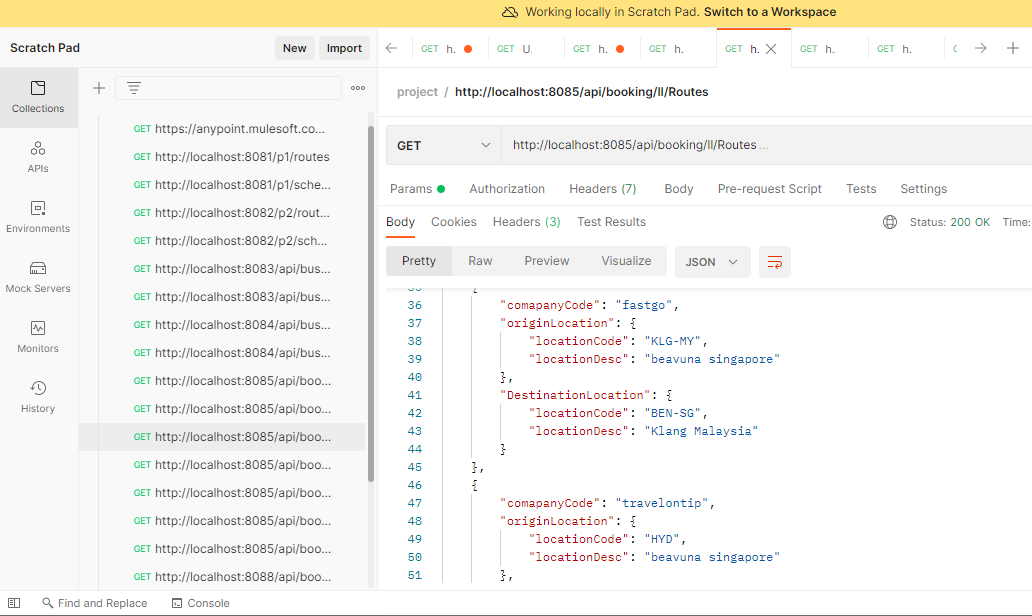
Output at Postman:



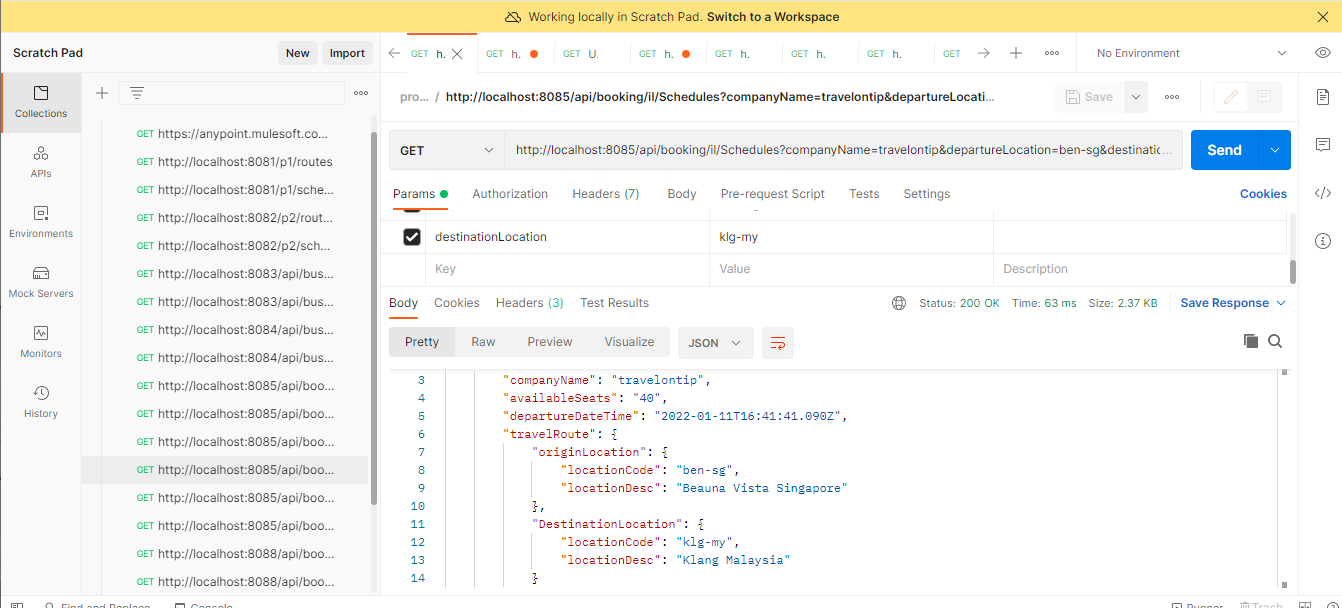
URL: <http://localhost:8085/api/booking/ks/Routes?companyname=fastgo>



URL: <http://localhost:8085/api/booking/hh/Routes?companyname=travelontip>



URL: <http://localhost:8085/api/booking/ll/Routes>

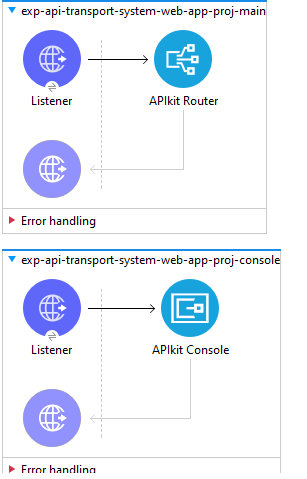


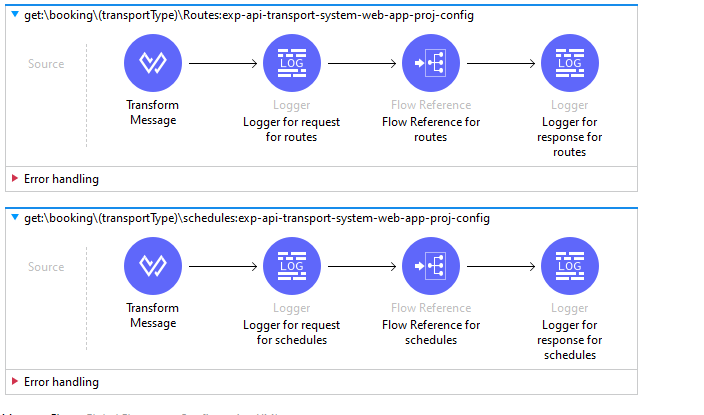
URL: <http://localhost:8085/api/booking/il/Schedules?companyName=travelontip&departureLocation=ben-sg&destinationLocation=klg-my>

Step 5: Create Experience API for both Web Application and also foe Mobile application.

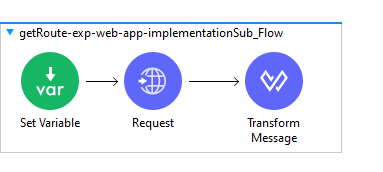
Experience API for Web Application:

Import RAML for experience API for Web Application from AnyPoint Platform design center.

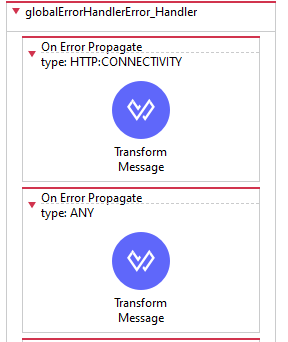


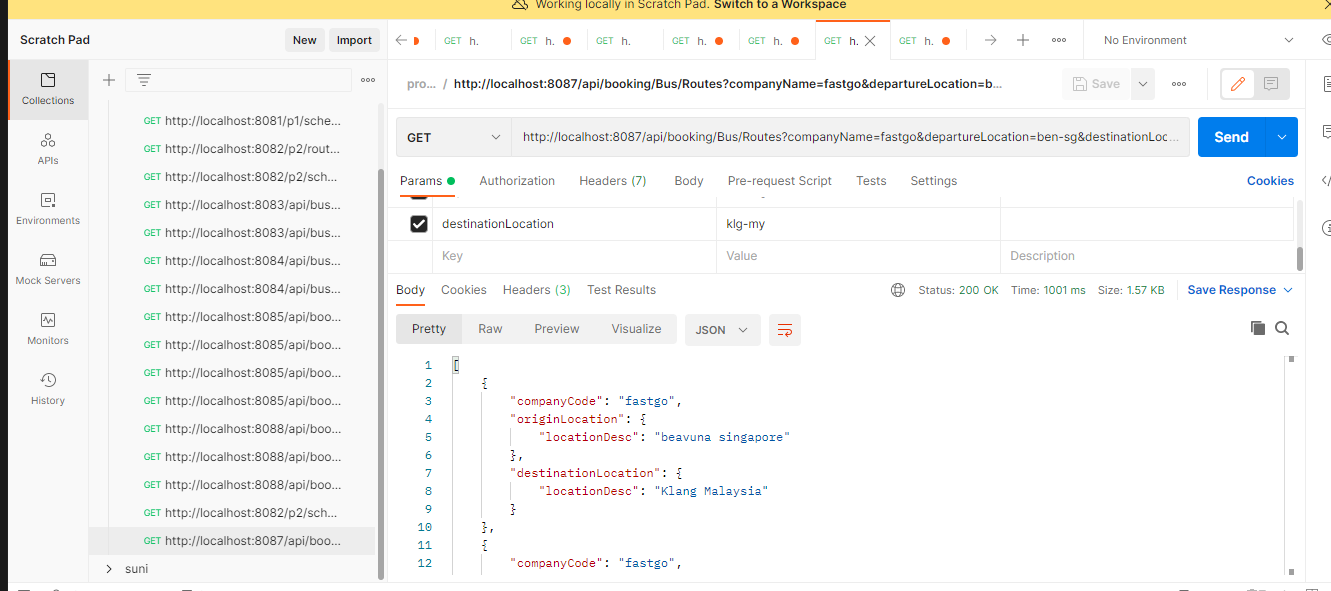


* Use request connector component to consume ProcessAPI http Connector.

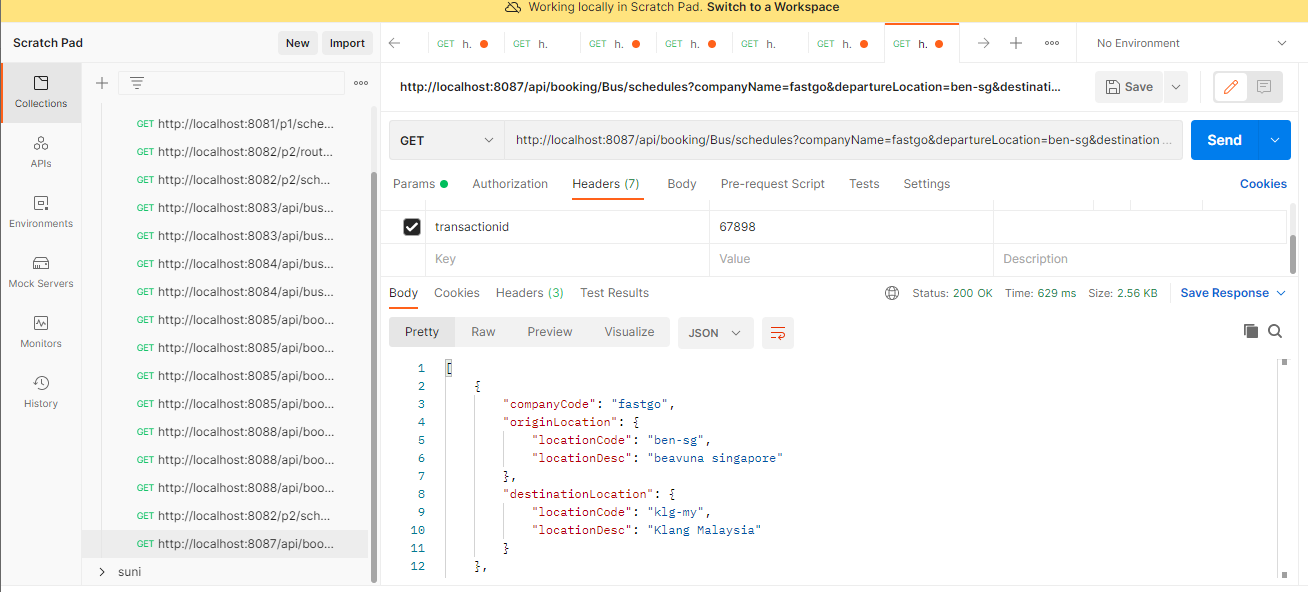


* Implement Error Component to handle errors Globally. If any error is raised which is not handled component and flow level then it is handled globally.





URL: <http://localhost:8087/api/booking/Bus/Routes?companyName=fastgo&departureLocation=ben-sg&destinationLocation=klg-my>

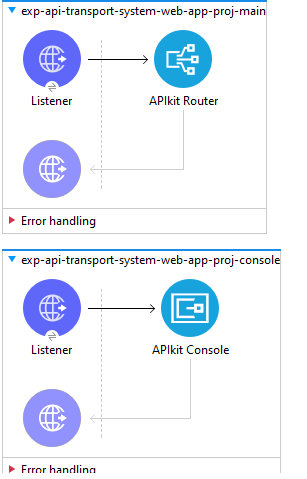


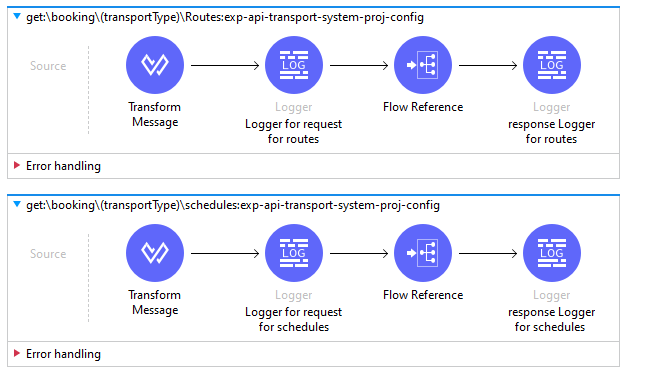
URL: <http://localhost:8087/api/booking/Bus/schedules?companyName=fastgo&departureLocation=ben-sg&destinationLocation=klg-my>

Step 6:

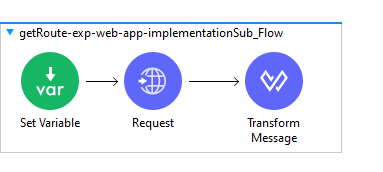
Experience API for Mobile Application:

Import RAML for experience API for Mobile Application from AnyPoint Platform design center.

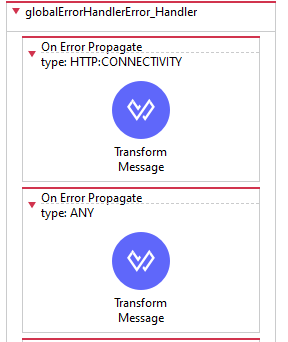




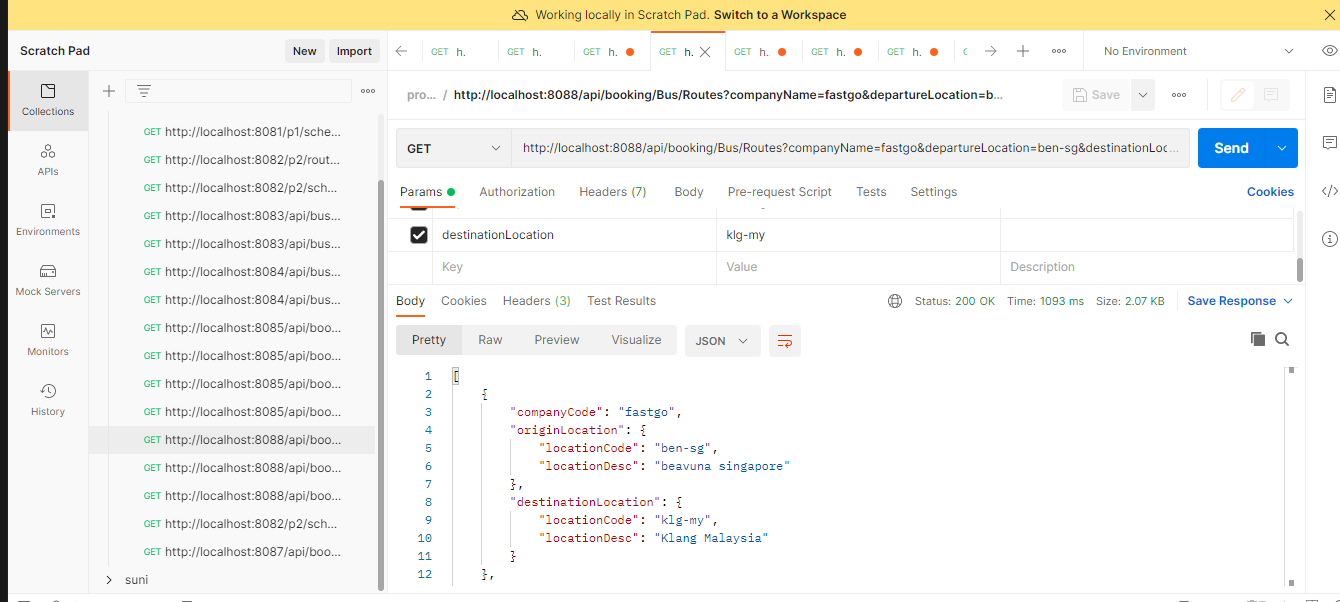
* Use request connector component to consume ProcessAPI http Connector.



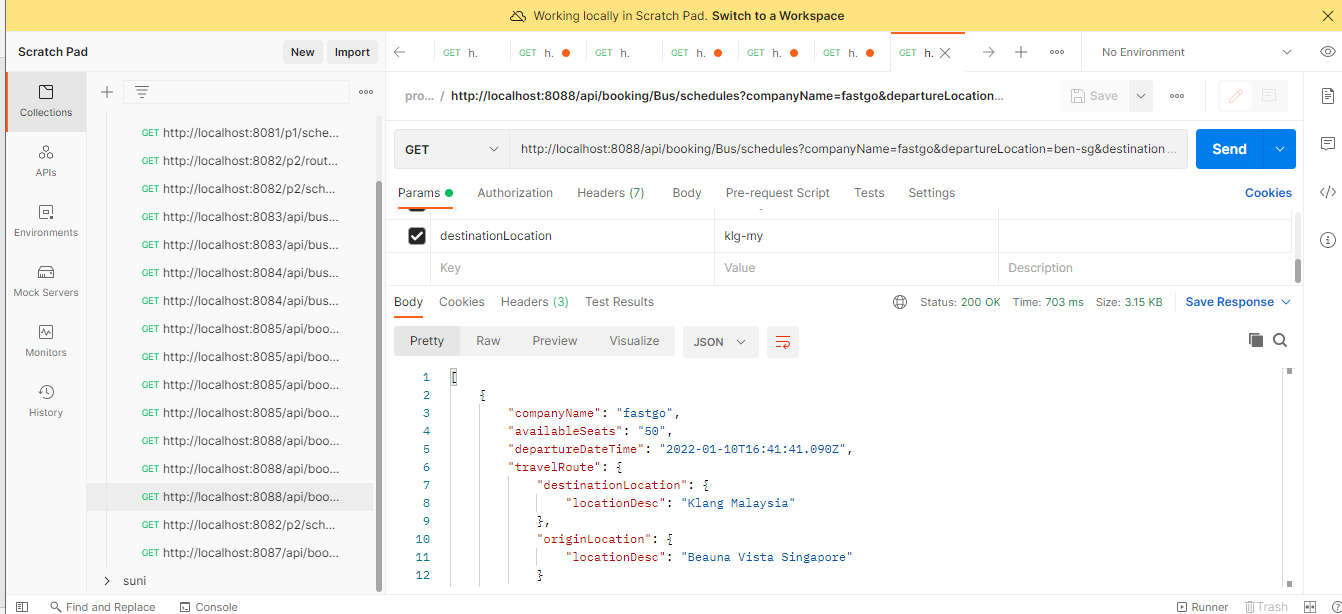
* Implement Error Component to handle errors Globally. If any error is raised which is not handled component and flow level then it is handled globally.



Ouput at Postman:



URL: <http://localhost:8088/api/booking/Bus/Routes?companyName=fastgo&departureLocation=ben-sg&destinationLocation=klg-my>



URL: <http://localhost:8088/api/booking/Bus/schedules?companyName=fastgo&departureLocation=ben-sg&destinationLocation=klg-my>

GIT HUB URL: https://github.com/REKAPALLISUNEETHA/project-repo-suneetha