***TNT Digital - Back-End Software Developer Assessment***

This is to build an aggregation service for consolidating the interface with 3 external APIs into a single endpoint that can handle multiple logical requests in one network requests. Each of the APIs accepts a query parameter that can accept multiple queries. Expect all APIs to always return 200 OK response with well formed input or 503 unavailable when it is unavailable.

**IDE USED: IntelliJ**

**TECHNOLOGIES USED: SpringBoot, MySql.**

**DEPENDENCIES ADDED: MySql Driver, Spring web, Spring Data JPA, Lombok**

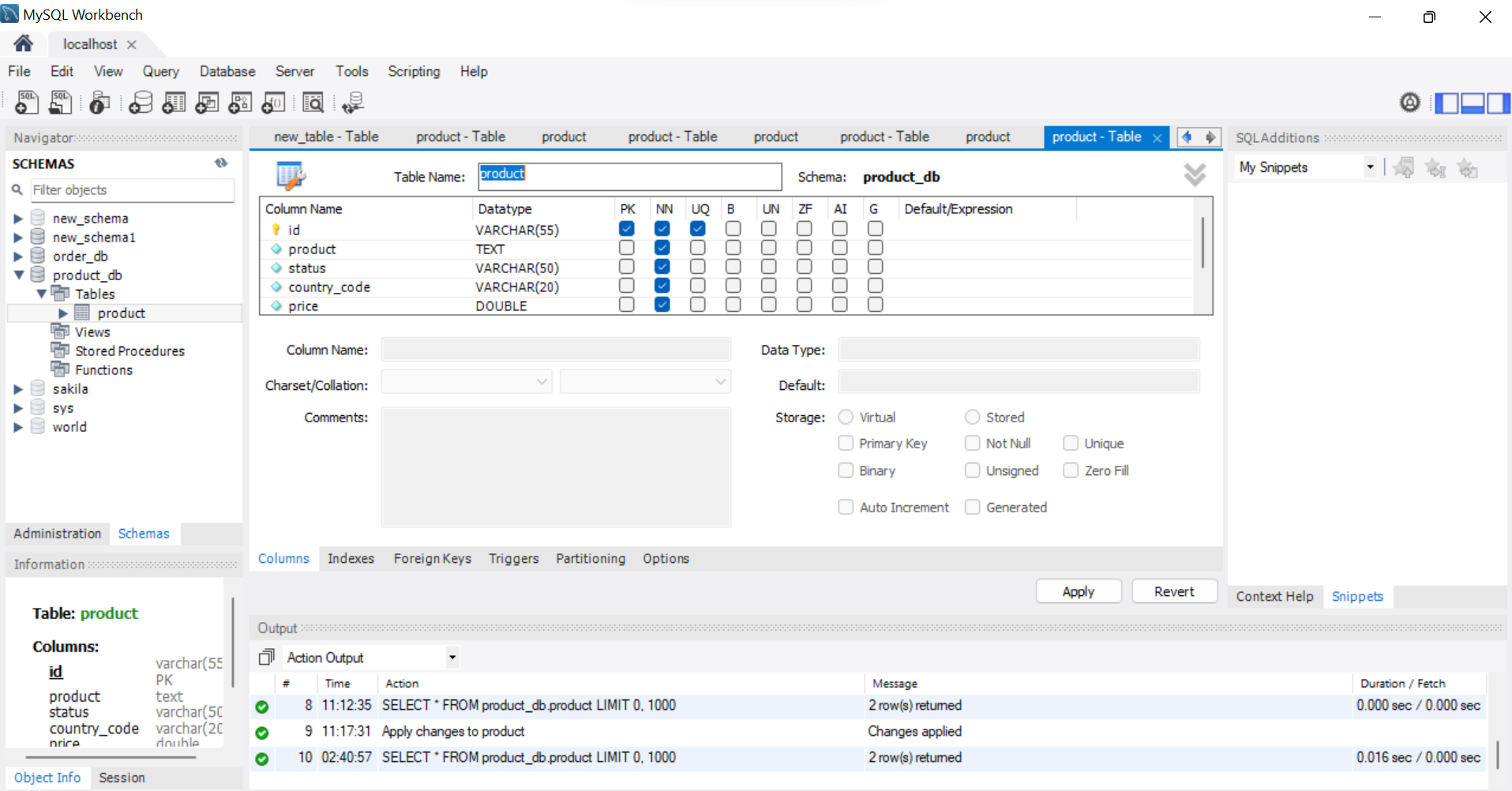
***HOW IT DID***

1.Create the project file using Spring initializr by giving java version project name and appropriate dependencies. Then create entity, controller, service and repository classes.

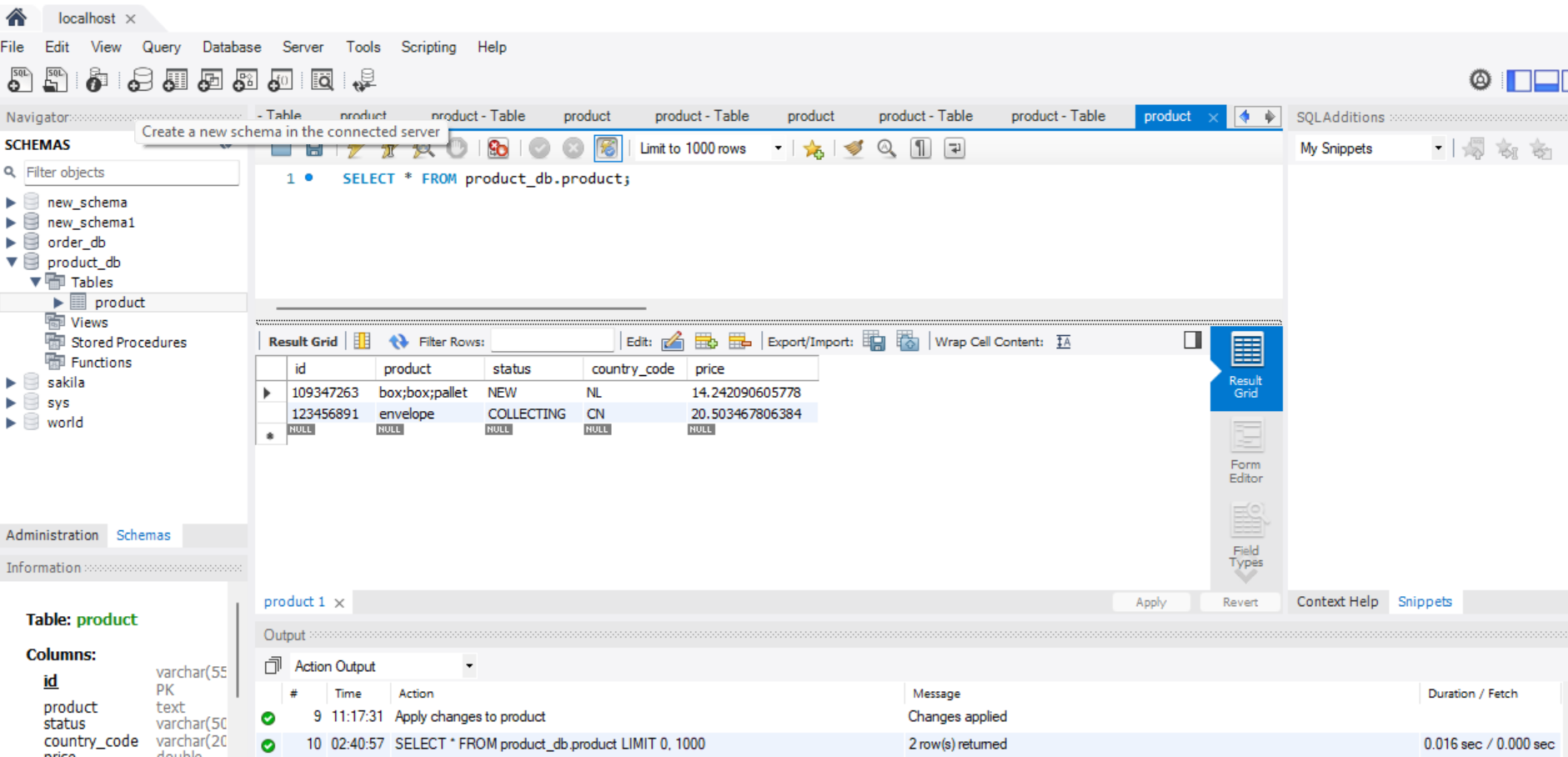
**Graphical user interface, text, application

Description automatically generated**

2.Create a database named product\_db using MySql workbench. Attributes of the product\_db are id, product, status, country\_code price.



Corresponding values of the attribute is given in the below picture.

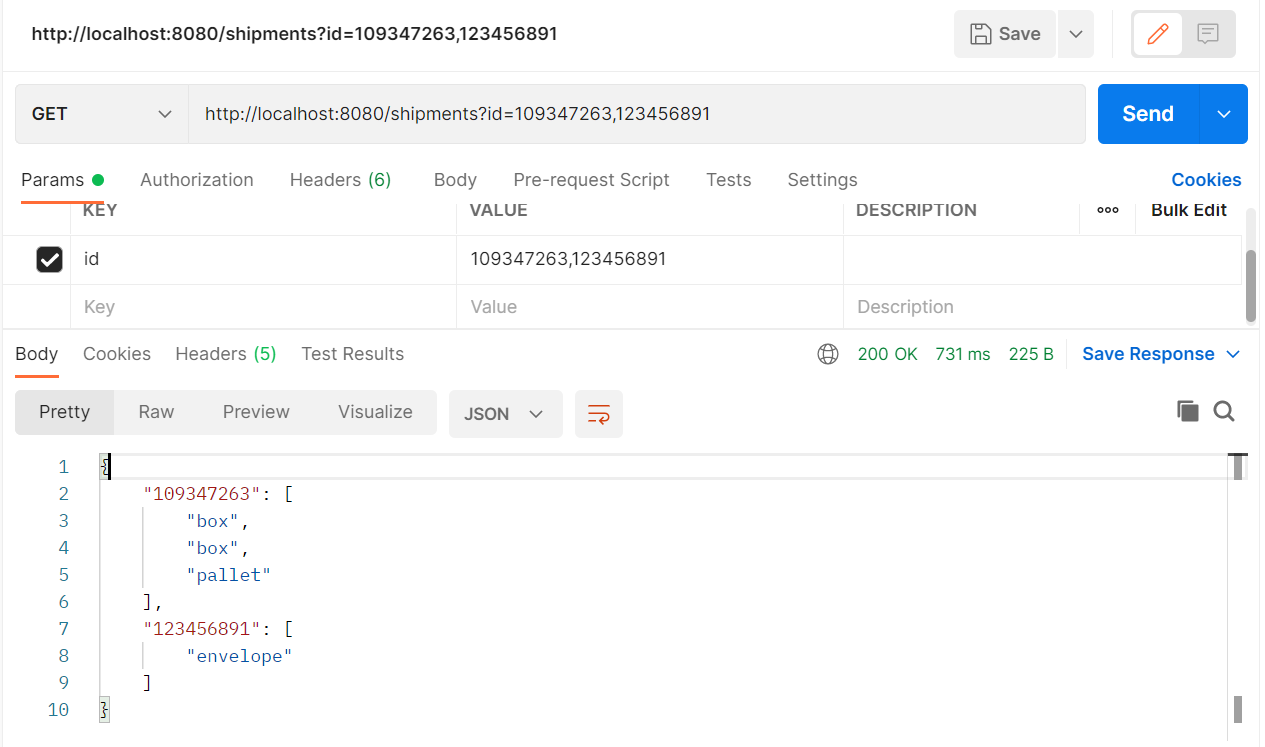


3.The application run in localhost 8080 port. The 4 API request and corresponding output are shown below.

1. **Shipment API:**

This will return a list od product corresponding to the given id.

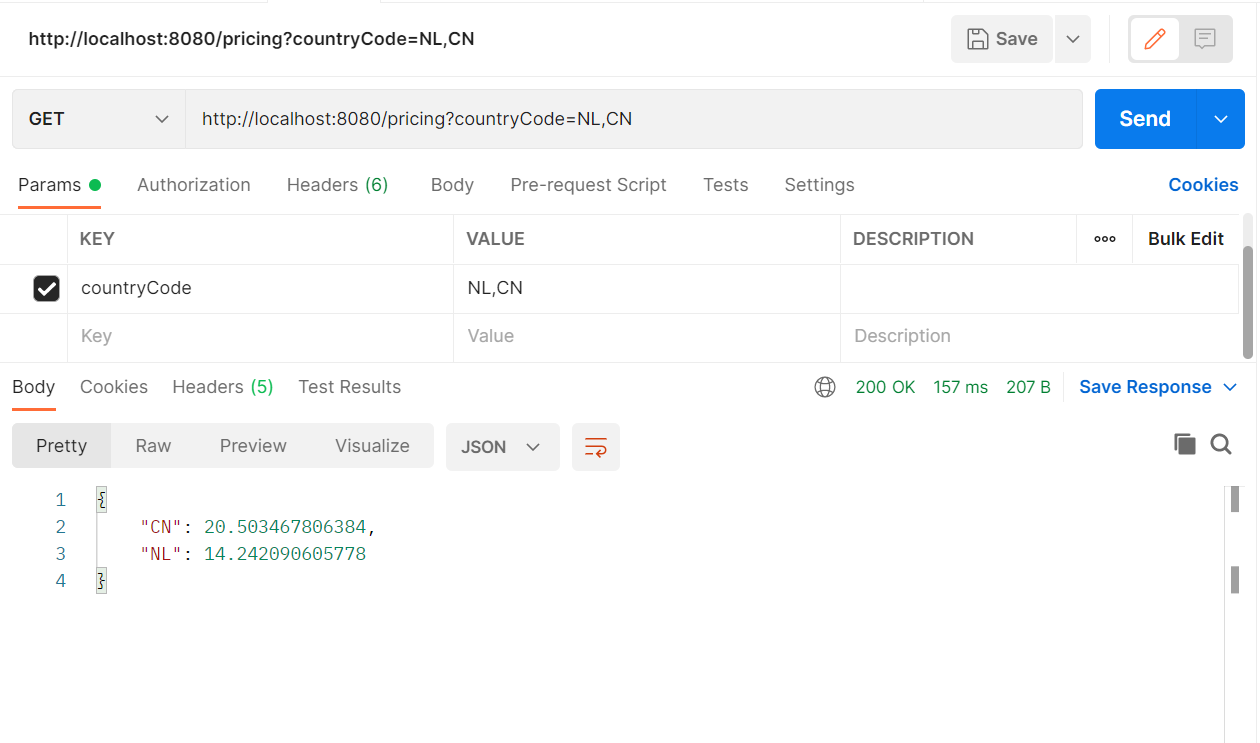
API Request: GET-<http://localhost:8080/shipments?id=109347263,123456891>



**2.Track API:**

This will return the status of the products when the id is sent in the url.

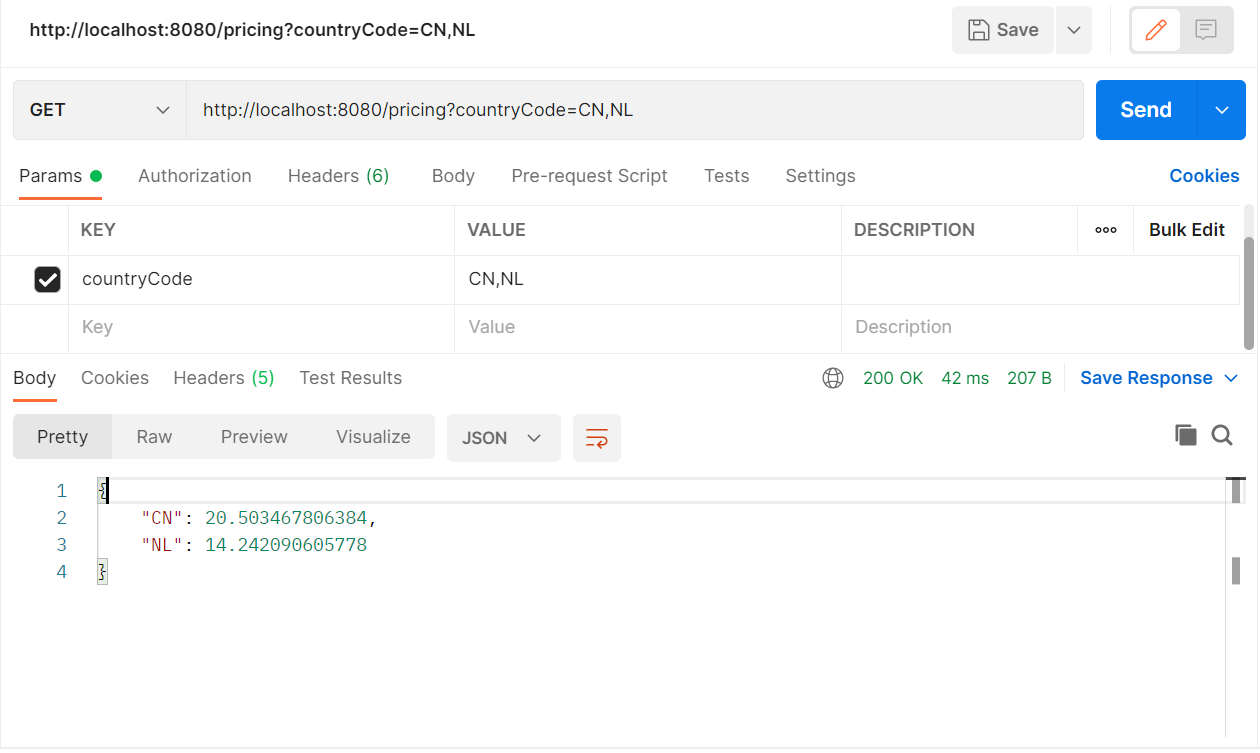
API Request: GET-<http://localhost:8080/track?id=109347263,123456891>



**3.Pricing API:**

This will return the price of a product when we send country code in the url.

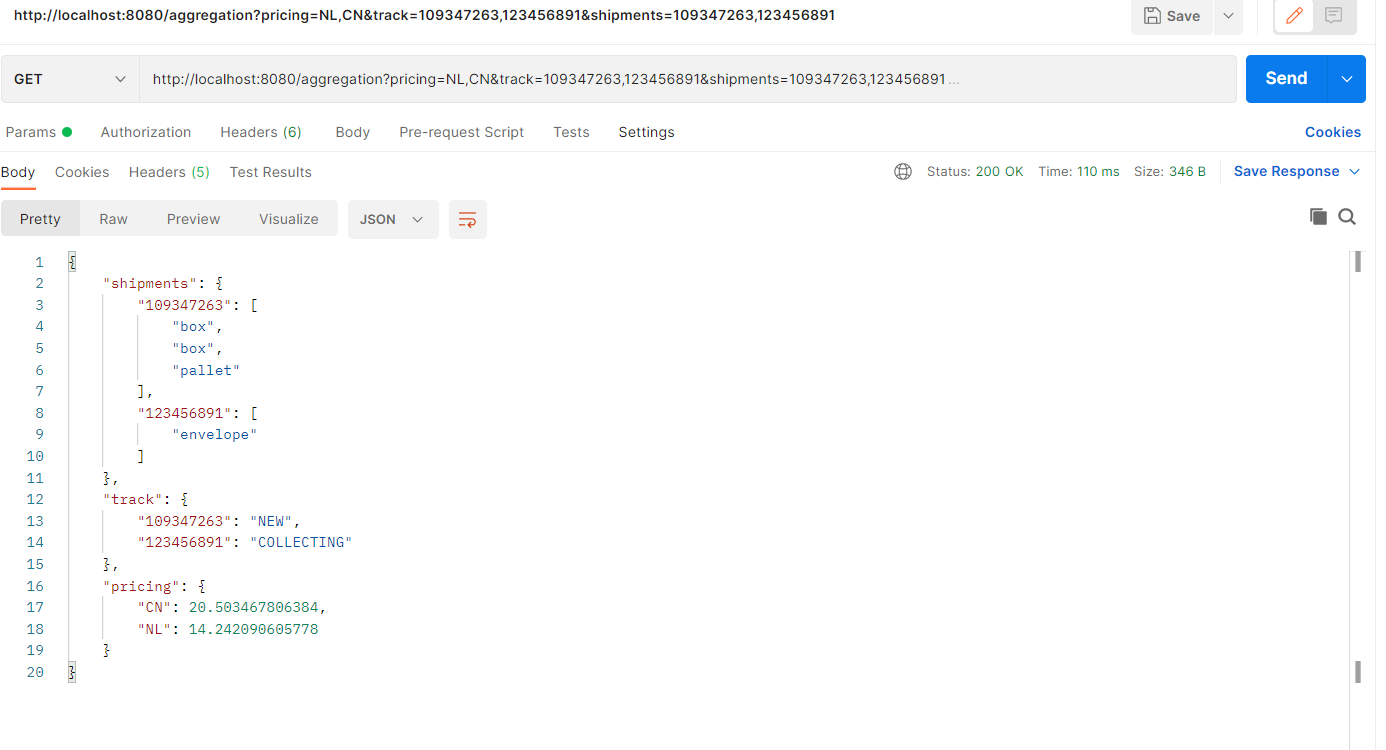
API Request: GET-http://localhost:8080/pricing?countryCode=NL,CN



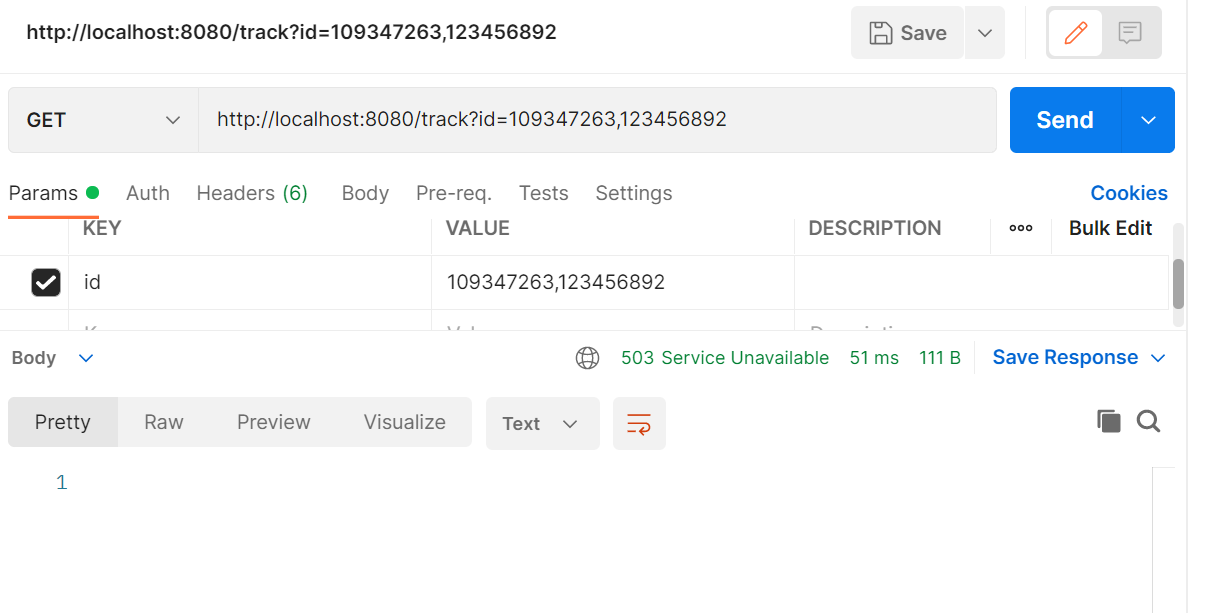
**4.Aggregation API:**

This is the consolidation of above three APIs.

API Request:  GET- <http://localhost:8081/aggregation?pricing=NL,CN&track=109347263,123456891&shipments=109347263,123456891>



It shows 503 unavailable when they are unavailable.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*