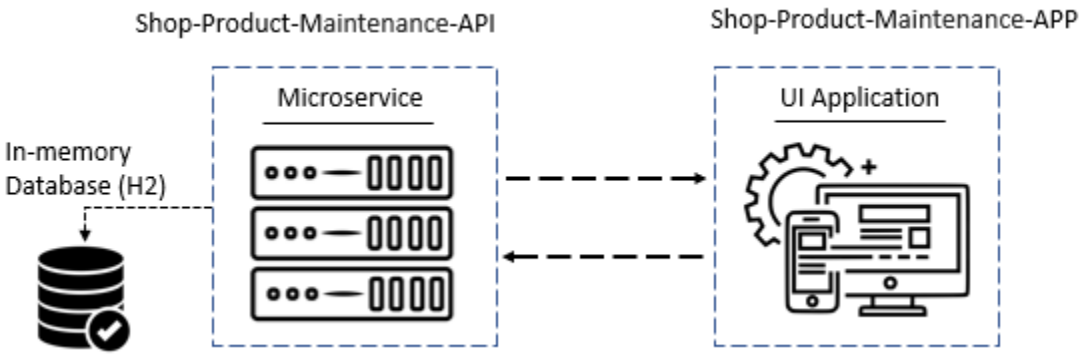
SHOP-PRODUCT-MAINTAINANCE-APPLICATION

Technologies Used:

1. Programming Language: Java8, Typescript
2. Backend Framework: SpringBoot2
3. Frontend Framework: Angular
4. Database: H2 (In-Memory Database)
5. Styling and Styling Framework: Bootstrap, CSS
6. Testing Framework: Junit, Mockito

Application Flow:



Module Description:

1. Shop-Product-Maintenance-API:

API Description	API to update the product details by providing the id of the record and the updated product details.
Http Method	Put
URL	http://localhost:8080//shop-product-maintainance/v1/product/{id}

Sample Request Body	{
	"productName": "Updated Product Name",
	"productDesc": "Updated Desc",
	"productPrice": 0.0,
	"productQuantity": 0,
	"productPriceCurrency": "INR"
	}

Sample Response Body	{ "productName": "Updated Product Name", "productDesc": "Updated Desc", "productPrice": 0.0, "productQuantity": 0, "productPriceCurrency": "INR"

	}

API Description	API to Delete the product details by providing the id of the record.
Http Method	Delete
URL	http://localhost:8080//shop-product-maintainance/v1/product/{id}
Sample Request Body	NA

Sample Response Body	{
	"deleted": true
	}

API Description	API to get all the product details present in the H2 database.
Http Method	Get
URL	http://localhost:8080//shop-product-maintainance/v1/product
Sample Request Body	NA
Sample Response Body	[{ "id": 1, "productName": "Updated Product Name", "productDesc": "Updated Desc", "productCategory": null, "productPrice": 0.0, "productQuantity": 0, "productPriceCurrency": "INR" }, { "id": 2, "productName": "Updated Product Name", "productDesc": "Updated Desc", "productCategory": null,

	"productPrice": 0.0, "productQuantity": 0, "productPriceCurrency": "INR" }]

API Description	API to get the product details of a particular product by providing the id of the record.
Http Method	Get
URL	http://localhost:8080//shop-product-maintainance/v1/product/{id}
Sample Request Body	NA

Sample Response Body	{
	"id": 2,
	"productName": "Updated Product Name",
	"productDesc": "Updated Desc",
	"productCategory": null,
	"productPrice": 0.0,
	"productQuantity": 0,
	"productPriceCurrency": "INR"
	}

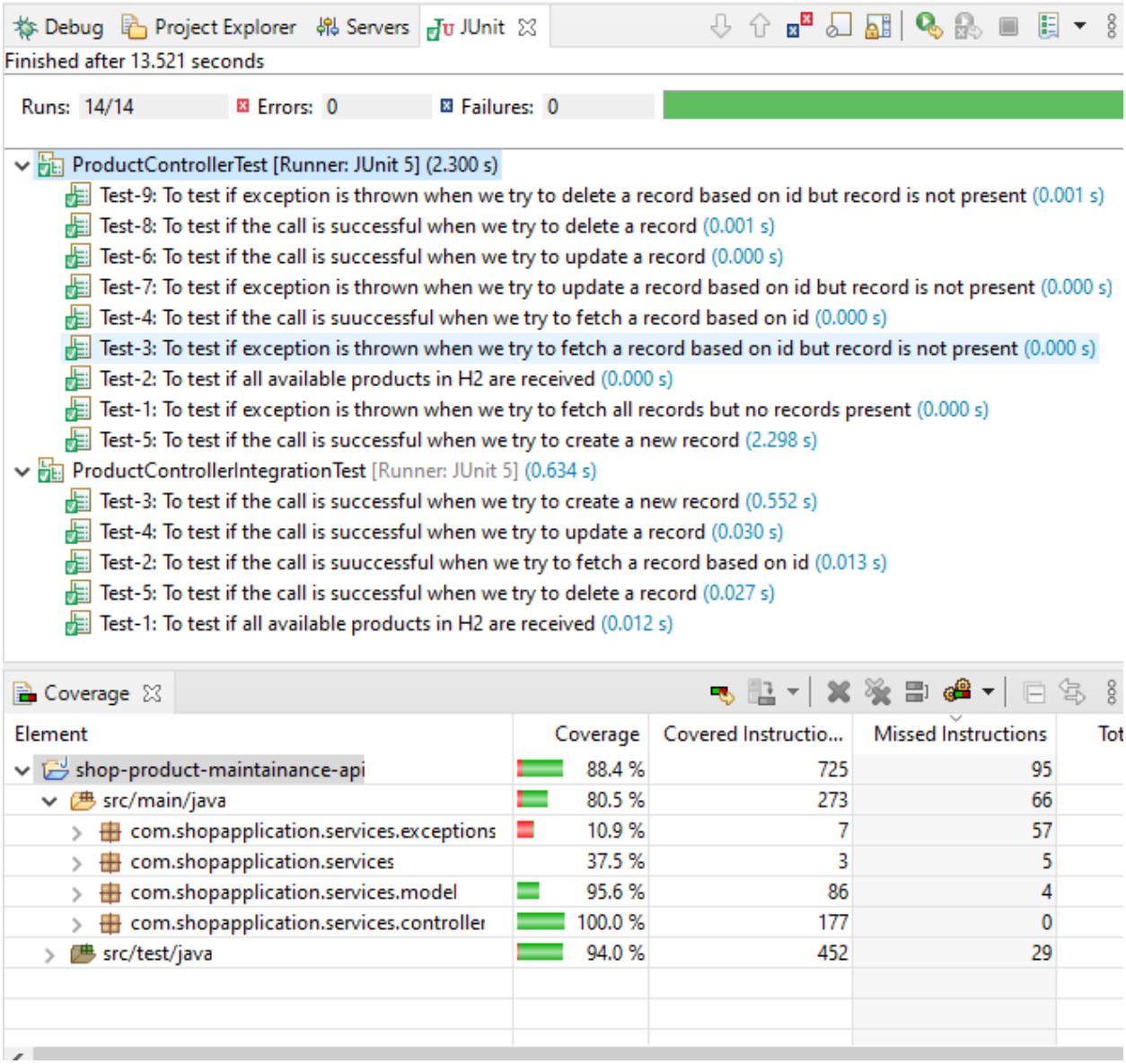
Error Response Body	{
	"timestamp": "2021-03-18T15:36:54.861+00:00",
	"message": "Product not found for this id :: 1",
	"details": "uri=//shop-product-maintainance/v1/product/1"
	}

API Description	Event-Producer-Service produces and commits scheduled events at every 2 mins to Kafka broker. As of now for demo purpose, it is configured to Mumbai as cityName.
Http Method	POST
URL	http://localhost:8080//shop-product-maintainance/v1/product
Sample Request Body	{ "productName": "Updated Product Name", "productDesc": "Updated Desc",

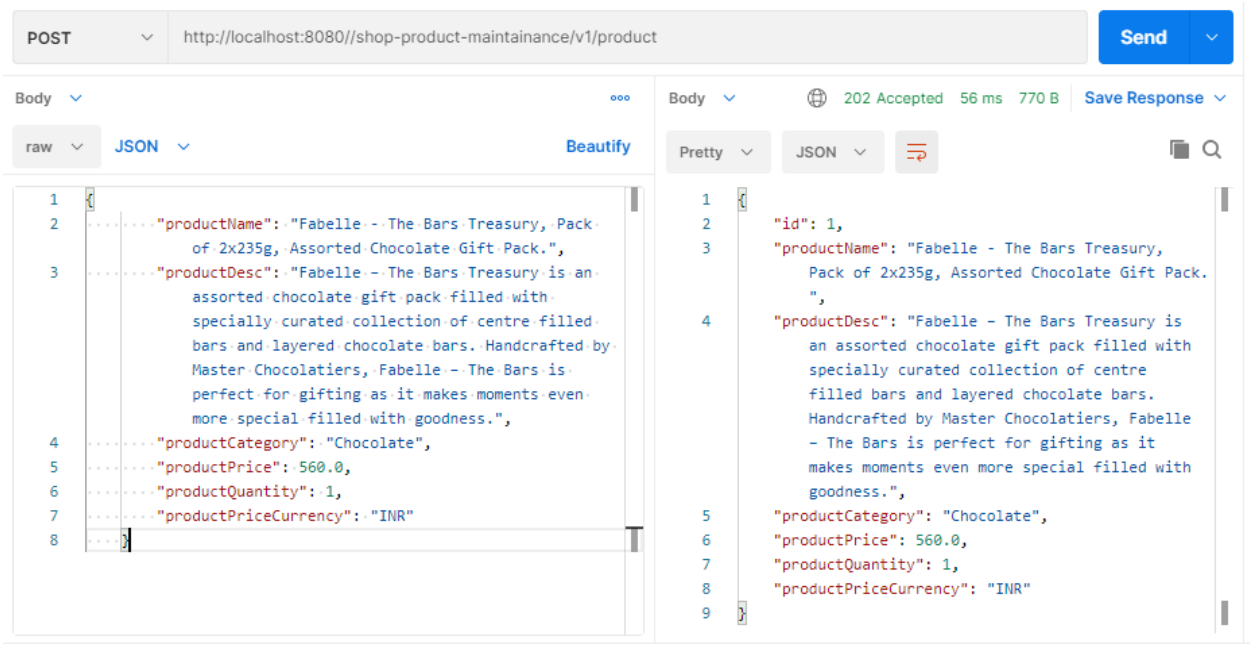
	"productPrice": 0.0, "productQuantity": 0, "productPriceCurrency": "INR" }

Sample Response Body	{
	"id": 1,
	"productName": "Updated Product Name",
	"productDesc": "Updated Desc",
	"productCategory": null,
	"productPrice": 0.0,
	"productQuantity": 0,
	"productPriceCurrency": "INR"
	}

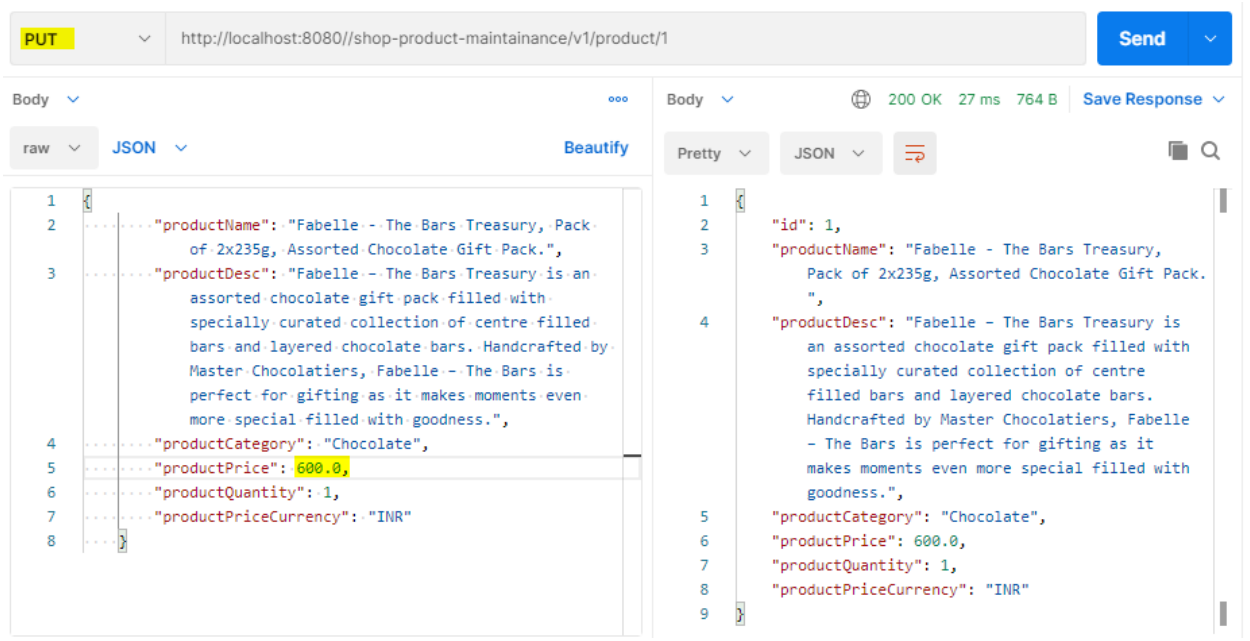
Junit Test Cases and Code Coverage Report:



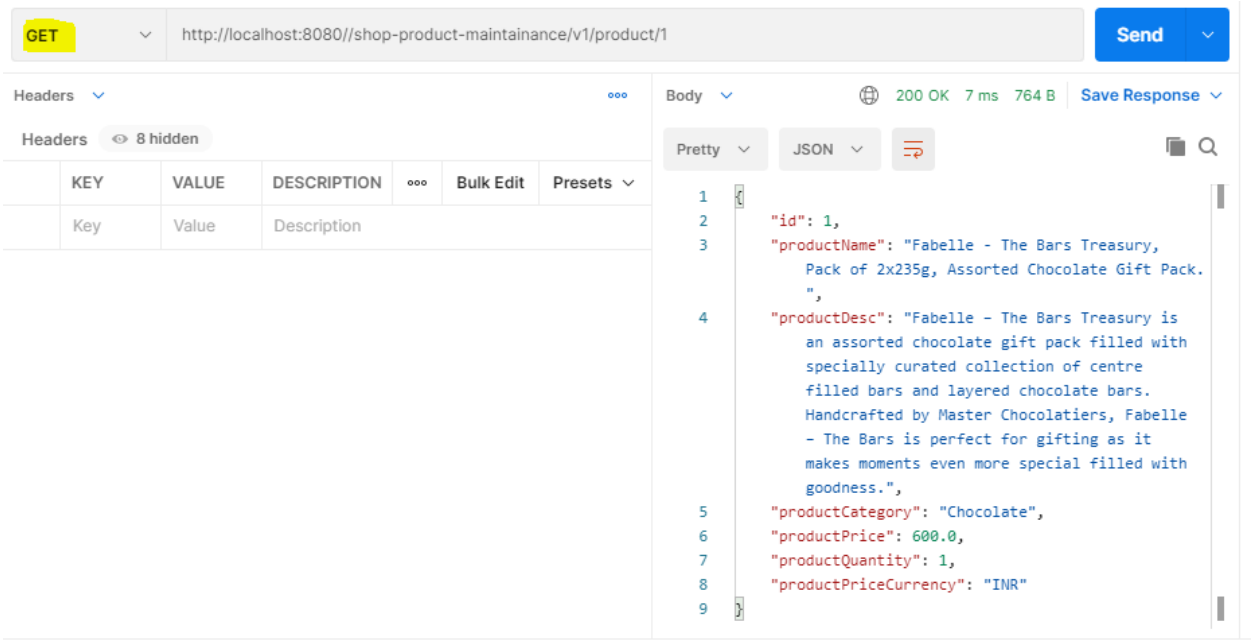
CASE-1: Adding a product, Getting HTTP-Status <202> and new product is added.



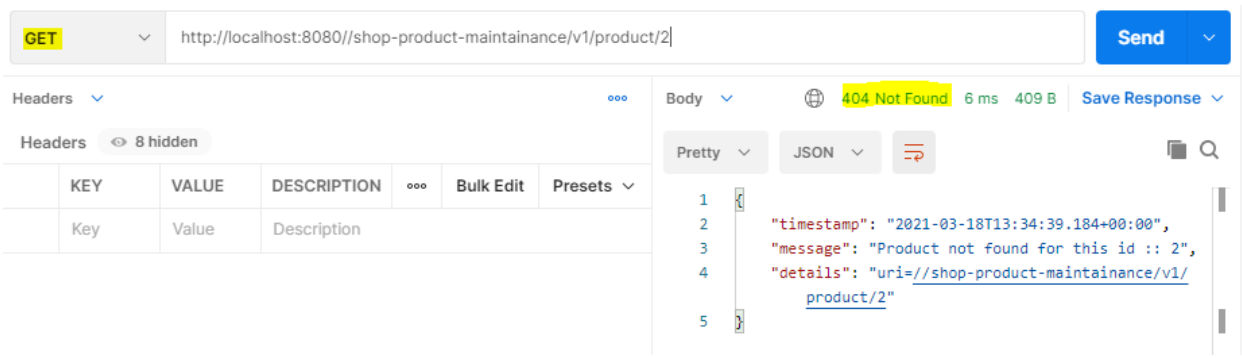
CASE-2: Editing a product details, Getting HTTP-Status <200> and user details are updated successfully.



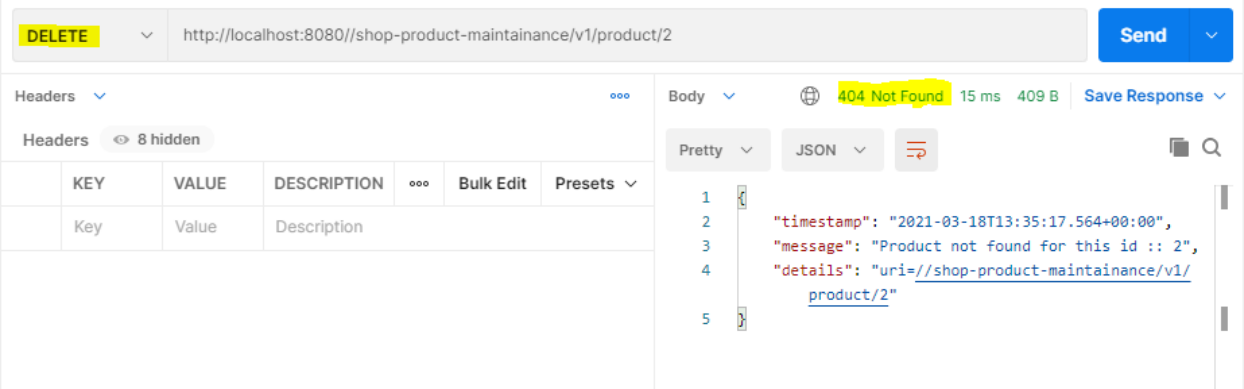
CASE-3: Getting a product details based on product id, Getting HTTP-Status <200> and product details received successfully.



CASE-4: Getting a product details based on product id, Getting HTTP-Status <404> Not Found as product details were not found in database for the given id.



CASE-5: Deleting product details based on product id, Getting HTTP-Status <404> Not Found as product details were not found in database for the given id.



CASE-6: Deleting product details based on product id, Getting HTTP-Status <200> and the given record is deleted successfully from the H2 Database.