|  | | | | |
| --- | --- | --- | --- | --- |
|  | RELEASE NOTES | | | |
|  |  | [Type the document title] | | |
|  |  |  | | |
|  | | |  | [Pick the date] |

**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver#** | **Date** | **Author** | **Description** | **Reviewed by** |
| 0.1 | 28-Feb-2021 | Srividhya Srinivasan | Hulk Store – REST API Services |  |

Table of Contents

[1. Requirement Details 4](#_Toc65420724)

[2. Approach Followed 4](#_Toc65420725)

[2.1 Scope Definition: 4](#_Toc65420726)

[2.2 Workflow Diagram: 4](#_Toc65420727)

[2.3 Tools & Techniques Used 5](#_Toc65420728)

[3. Sources 5](#_Toc65420729)

[3.1 Java Files 5](#_Toc65420730)

[3.2 Database Configuration 6](#_Toc65420731)

[3.3 Git Hub Details 7](#_Toc65420732)

[4. Test Cases 7](#_Toc65420733)

[4.1 Unit Test Plan 7](#_Toc65420734)

[4.2 Screenshots 7](#_Toc65420735)

# Requirement Details

Hulkstore Application provides you the possibility of accessing the Marvel and DC superheroes related products like T-Shirts, Cups, Books etc.

There are 3 APIs implemented in the Hulkstore Application. We have the possibility of creating the two categories say Marvel and DC in the CategoryAPI service which has endpoints for creation of Category, Updation of Category, Inquiry all the categories and Inquiry by Category Id.

Next we have the ProductAPI, where it allows you to create products associated with the superheroes where you could create a Product, Update the attributes of a product, inquire all the products, inquire product by a particular id.

The InventoryAPI has been created with endpoints to add a new inventory and update the inventory details after its been sold to a customer.

# Approach Followed

## Scope Definition:

To Automate the Hulk Store Inventory, the following API services has been created to do the respective functions.

**Category Services:**

* Creation of Category ID and Category name using POST Method
* Update the category ID and Name using the PUT Method
* Inquire by Category ID using GET Method
* Inquire by ALL using GET Method

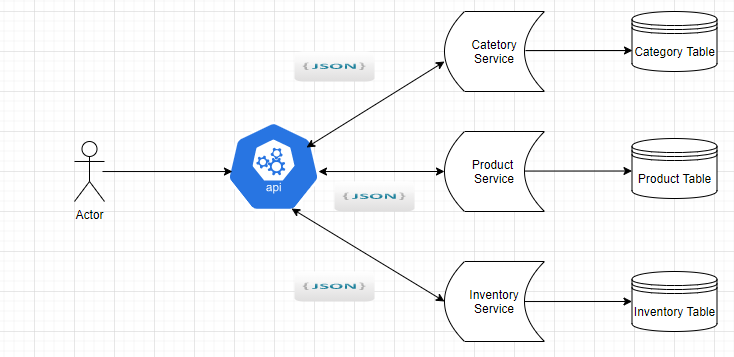
**Product Services:**

* Creation of Product using Product ID/Name/Category ID/ Product Count
* Update of Product Attributes

**Inventory Services:**

* Increment of Product Count when the product is bought
* Decrement of Product Count when the product is sold

## Workflow Diagram:



## Tools & Techniques Used

* + Programming Language: JAVA
  + Database: Lombok with H2 in memory database
  + Framework: Spring boot + JPA
  + Architecture: REST API
  + IDE Used: IntelliJ IDEA
  + JSON Message – Request/Response
  + API Testing Tool: Postman
  + GIT hub for Source repository

# Sources

This section has list of source files added or customized for this module. The source files are provided as part of the release.

## Java Files

| Sr. No. | **Service Name** | **Package** |
| --- | --- | --- |
| 1 | CategoryApi.java | com.srividhya.hulkstore.api |
| 2 | CategoryController.java | com.srividhya.hulkstore.controller |
| 3 | Category.java | com.srividhya.hulkstore.entity |
| 4 | CategoryRepository.java | com.srividhya.hulkstore.repository |
| 5 | CategoryService.java | com.srividhya.hulkstore.service |
| 6 | ProductApi.java | com.srividhya.hulkstore.api |
| 7 | ProductController.java | com.srividhya.hulkstore.controller |
| 8 | Product.java | com.srividhya.hulkstore.entity |
| 9 | ProductRepository.java | com.srividhya.hulkstore.repository |
| 10 | ProductService.java | com.srividhya.hulkstore.service |
| 11 | InventoryApi.java | com.srividhya.hulkstore.api |
| 12 | OrderRequest.java | com.srividhya.hulkstore.dto |
| 13 | OrderResponse.java | com.srividhya.hulkstore.dto |
| 14 | GenericError.java | com.srividhya.hulkstore.errors |
| 15 | ItemExistException | com.srividhya.hulkstore.exceptions |
| 16 | ItemNotExistException | com.srividhya.hulkstore.exceptions |
| 17 | OutOfStockException | com.srividhya.hulkstore.exceptions |
| 18 | OrderRequest | com.srividhya.hulkstore.dto |
| 19 | OrderResponse | com.srividhya.hulkstore.dto |
| 20 | ResponseUtils | com.srividhya.hulkstore.utils |

## Database Configuration

| Sr. No. | **Table Name** | **Table Columns** |
| --- | --- | --- |
| 1 | CATEGORY | Long categoryId, String categoryName |
| 2 | PRODUCT | Long productid, String productName, Integer productCount, Long categoryId |

## Git Hub Details

The Sources & artifacts are available in the following link

<https://github.com/SrividhyaSrinivasan/HulkStore-Application>

# Test Cases

## Unit Test Plan

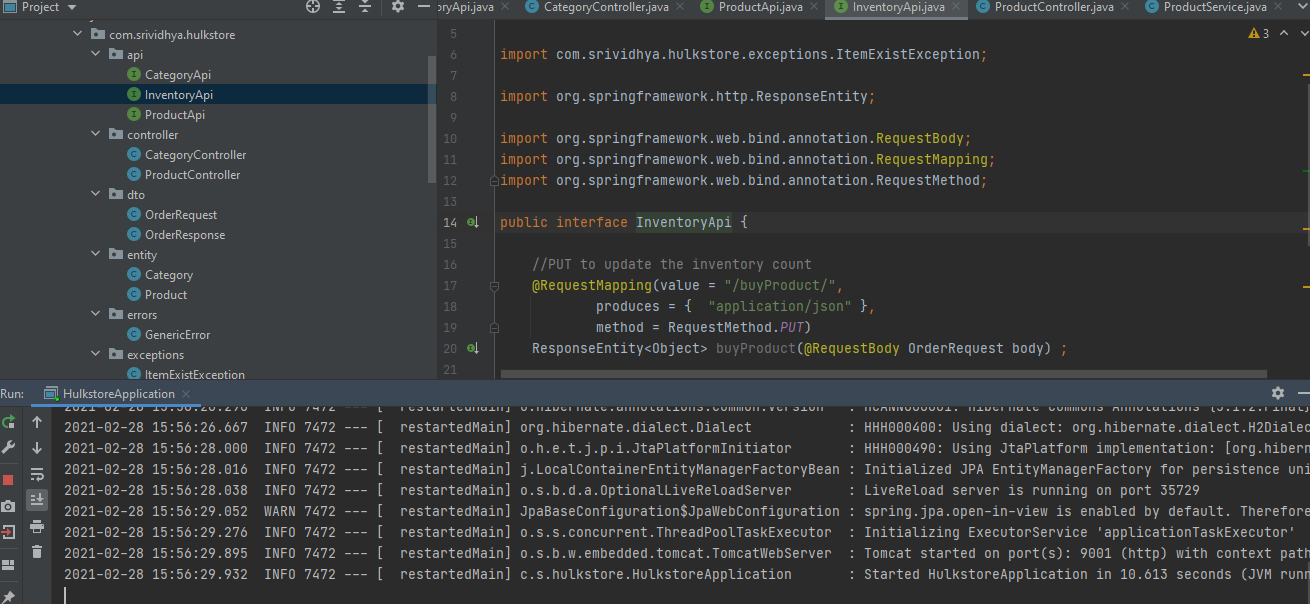
Although there is various scenarios which was testing during this Drop 1 phase of the project. There are number of validations which can be enhanced for completeness of the project.



## Screenshots with Evidence

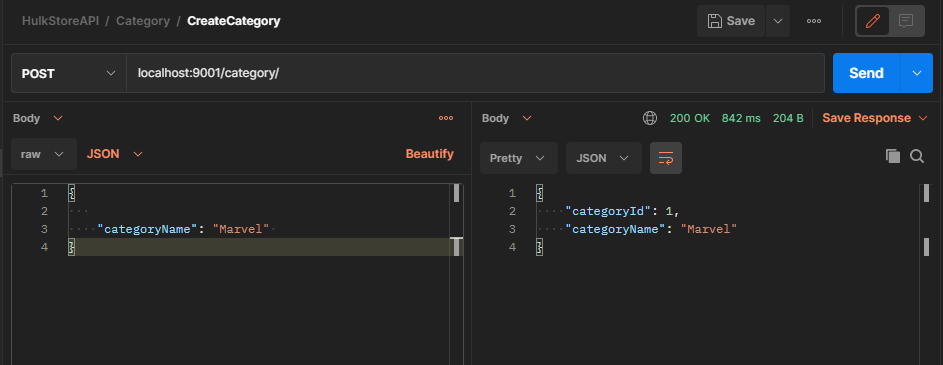
**HULKSTORE APPLICATION**

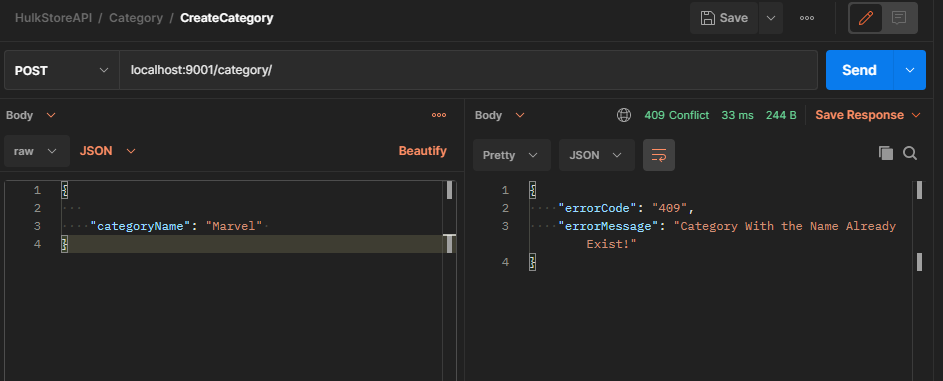
The application has been started using IntelliJ IDEA.



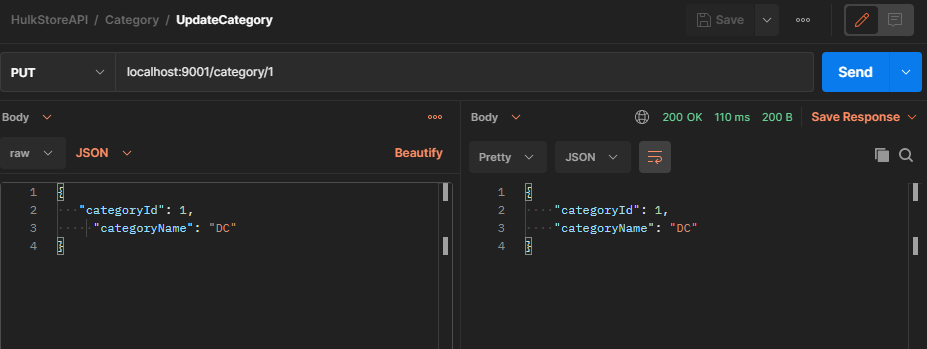
Testing my application in PostMan API tool.

TC\_ID\_01 : Category Creation – POST Method – SUCCES scenario

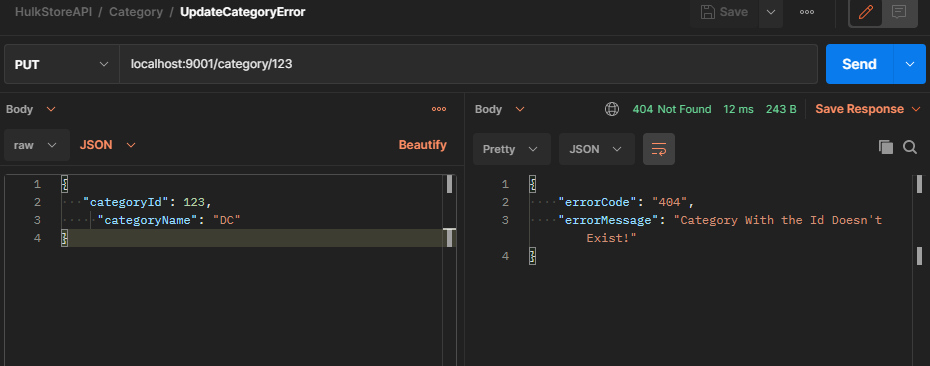
  
TC\_ID\_02 : Category Creation – POST – FAILURE – 409 CONFLICT



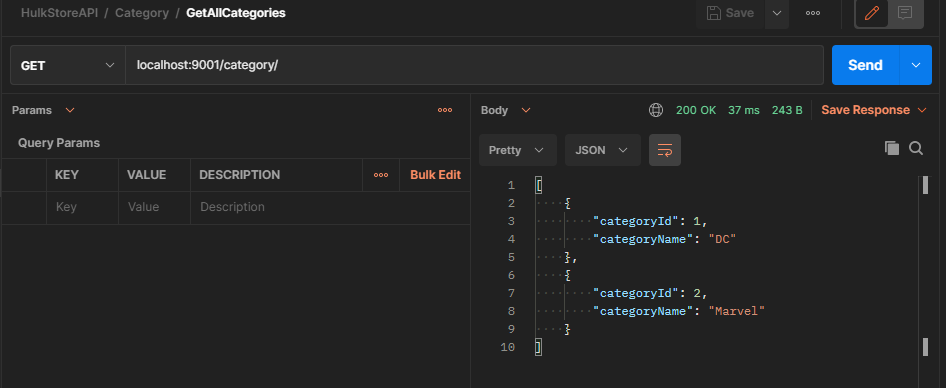
TC\_ID\_03 : Update Category – PUT method - SUCCESS – 200 OK



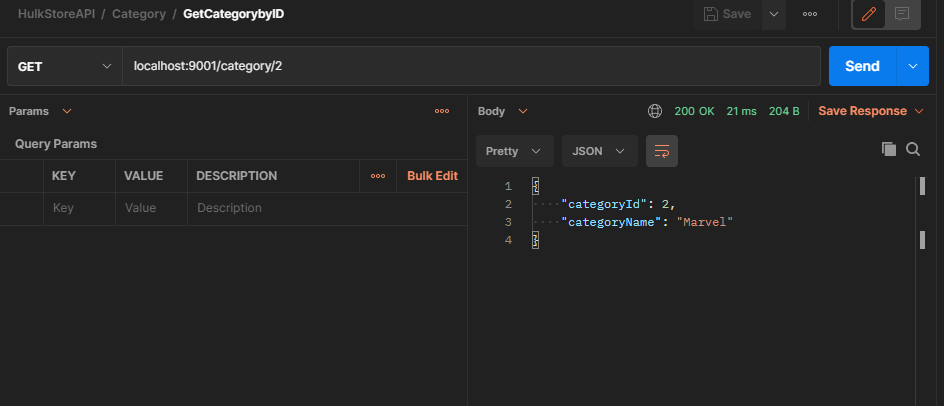
TC\_ID\_04 : PUT – FAILURE – 404 NOT FOUND



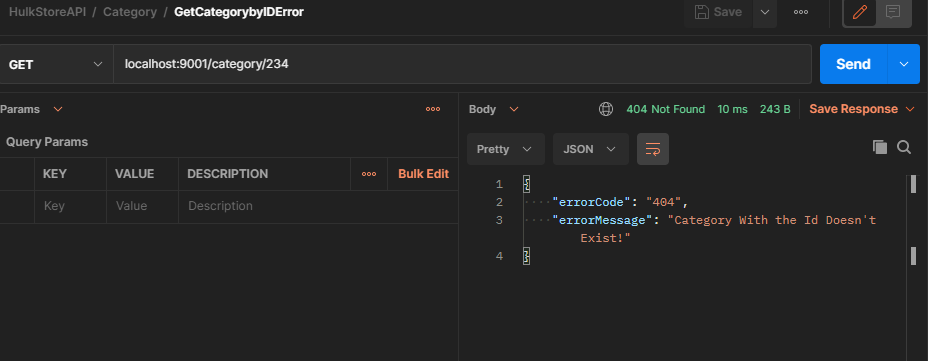
TC\_ID\_05 : Get All Category – SUCCESS – 200 OK



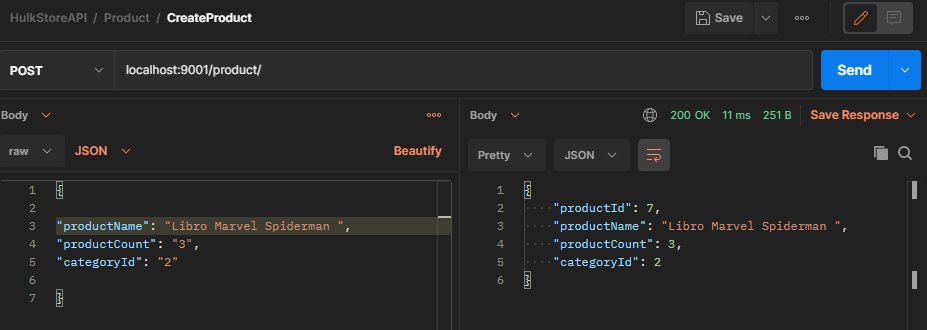
TC\_ID\_06 : Get category by Id – SUCCESS – 200 OK



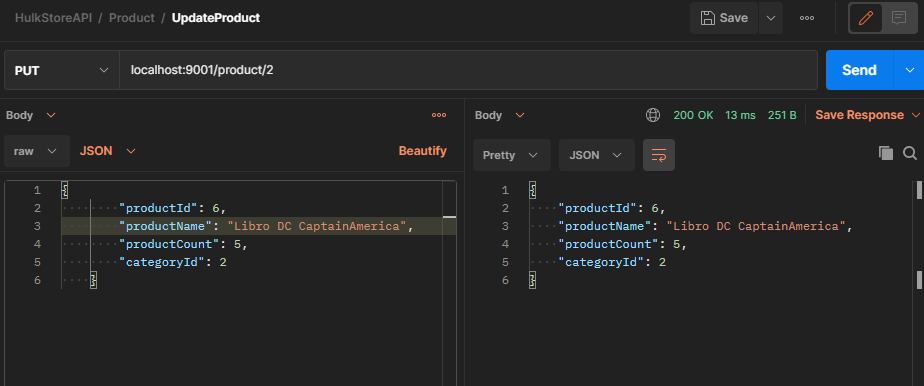
TC\_ID\_07 : Get Category by ID – FAILURE – 404 NOT FOUND



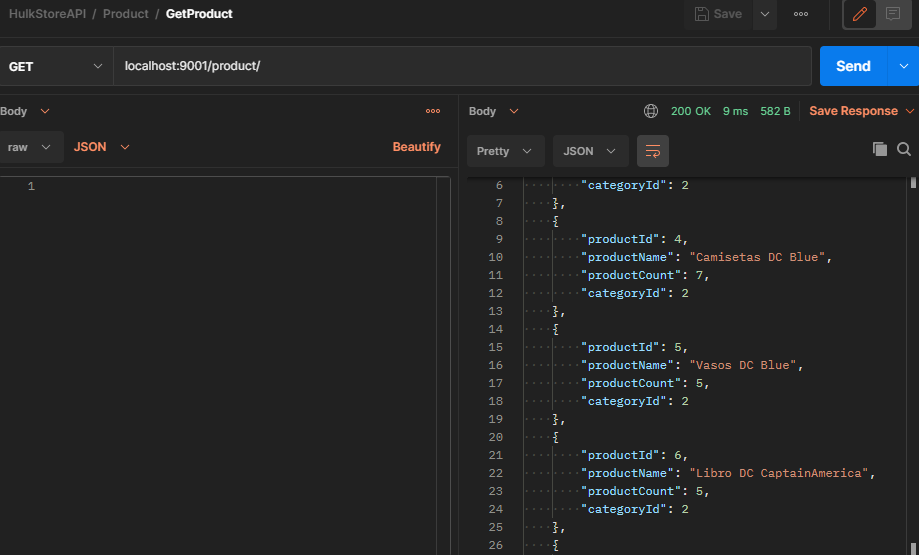
TC\_ID\_08 : Create Product – POST - SUCCESS – 200 OK



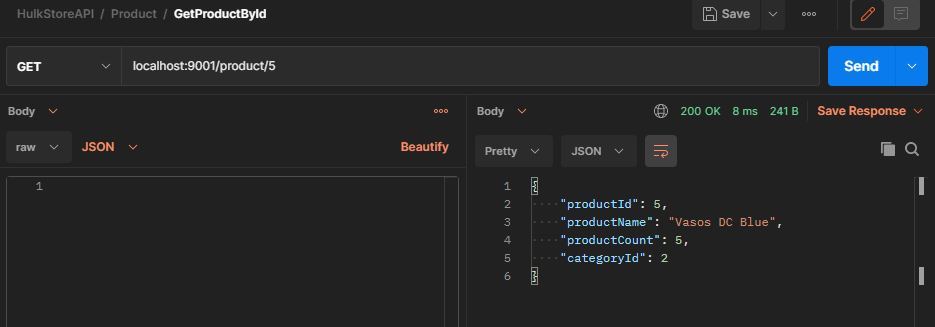
TC\_ID\_09 : Update Product – PUT – SUCCESS – 200 OK



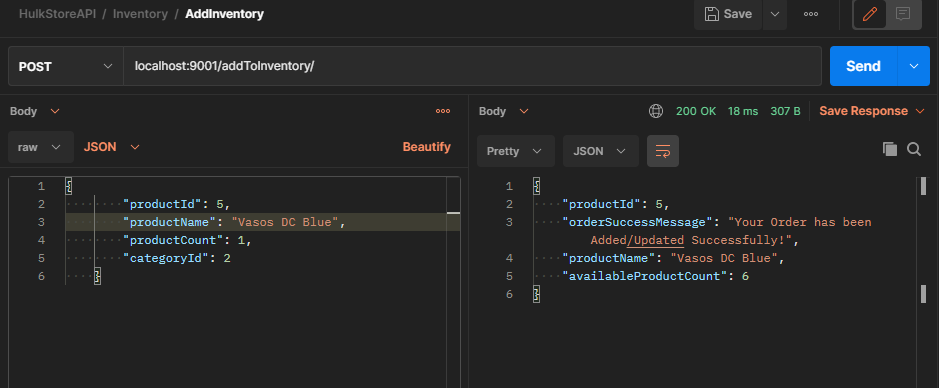
TC\_ID\_10 : Get All Product – GET – SUCCESS -200 OK



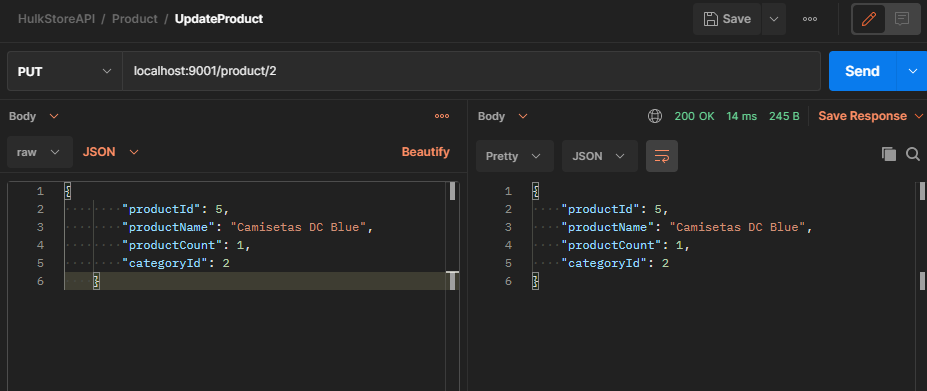
TC\_ID\_11 : Get Product by ID – GET – SUCCESS – 200 OK



TC\_ID\_12 : Increase inventory count on adition of a Product – 200 OK - SUCCESS



TC\_ID\_13 : Descrease in inventory count on reduction(selling) of a Product – 200 OK – SUCCESS



TC\_ID\_14 : Trying to order a producto which is not in stock – 404 NOT FOUND - FAILURE

