**MyRetail RESTful Service**

MyRetail RESTful service retrieves product details from an external API and get the price details from the database and provide integrated results to the users. It also provides API to update the price of the product.

**Technology Stack**

* Spring Boot - <https://projects.spring.io/spring-boot/>
* Swagger - <http://swagger.io/>
* Gradle 4.7 - [https://gradle.org](https://gradle.org/)
* MySQL - <https://www.mysql.com/>
* JDK 1.8

**Instructions to Setup**

1. Install MySQL in your system - Host: localhost Port: 3306
2. Create User

create user 'myretail\_user'@'%' identified by 'password';

grant create, drop, alter, index, insert, select, delete, update, execute ON myretail.\* TO 'myretail\_user';

1. Create Schema myretail

create schema myretail;

1. Create table product in myretail schema

create table Product (productId VARCHAR(255) NOT NULL, prodName VARCHAR(255),currentPrice VARCHAR(255),currencyCode VARCHAR(255));

1. Install Gradle 4.7 - https://gradle.org/gradle-download/
2. Clone the code from git repository - https://github.com/Mannu2116/RedskyMyRetail.git
3. Make sure you are in the RedskyMyRetail directory
4. Run the following command to start ./gradlew bootRun
5. Open browser and visit Swagger. http://localhost:8080/swagger-ui.html

**Get Product Information:**

MyRetail API when receives the request, it sends a request to "redsky.target.com" and retrieves the product information. Pricing information is retrieved from a data store and combines it with the information from the HTTP request into a single response. The GET request is done at the path "/products/{id}".

**Input:**

Performing GET request on http://localhost:8080/products/13860428 will return json object with product information and pricing information.

GET http://localhost:8080/products/13860428

**Output**: -

{"productId":"13860428","prodName":"The Big Lebowski (Blu-ray)","currentPrice":"58","currencyCode":"USD"}

**Update Product Price in the datastore:**

To perform PUT operation, send JSON object with updated price in request body, it will return JSON object with updated pricing information. The request is done at the path "/products/{id}".

**Input:**

{" productId ":13860428," prodName ":"The Big Lebowski (Blu-ray) (Widescreen)"," currentPrice ":{"value": 256," currencyCode ":"USD"}}

**Output**: -

Success message is returned if it is updated successfully.

**Testing:**

The testcases are present in the folder 'src\test\java\myRetail\controller'.

The test cases can be executed by running the command './gradlew test'

**API Requests and Responses**

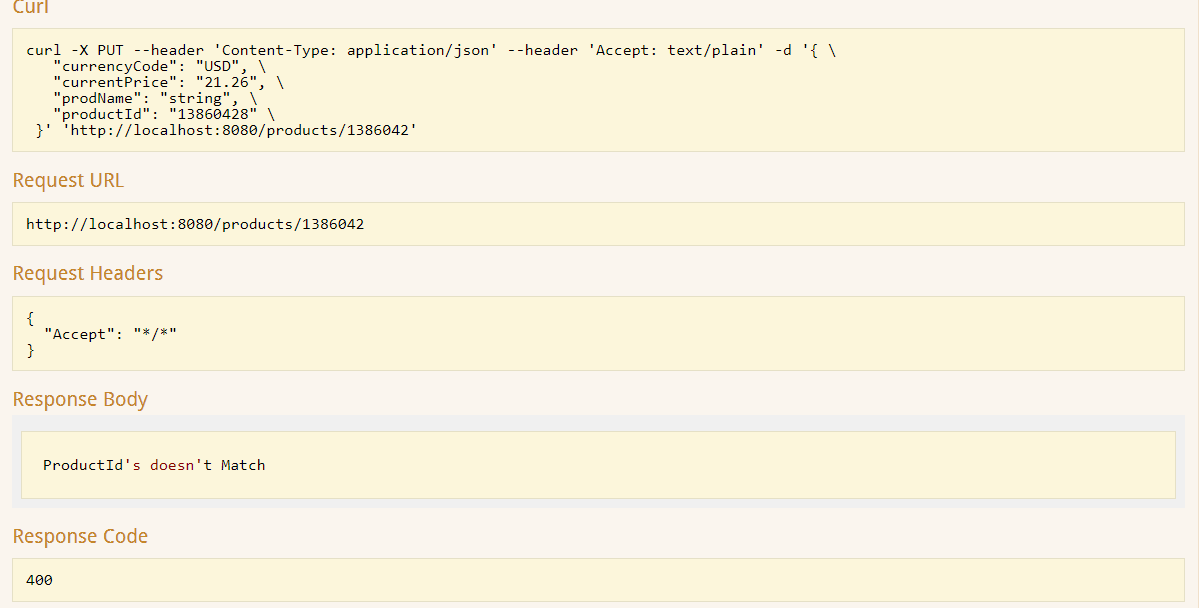
**PUT Request**

Following PUT request will store information of productID:13860428 in MySQL database.

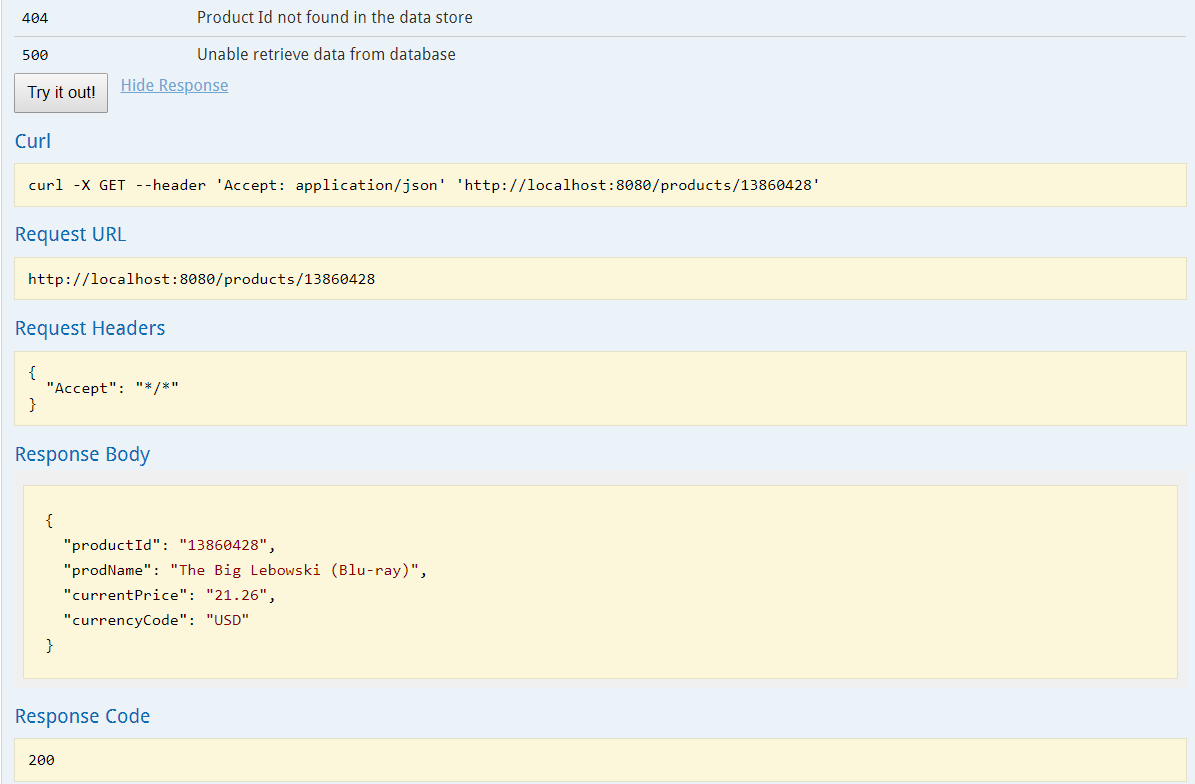
**Request# When product details updated successfully**



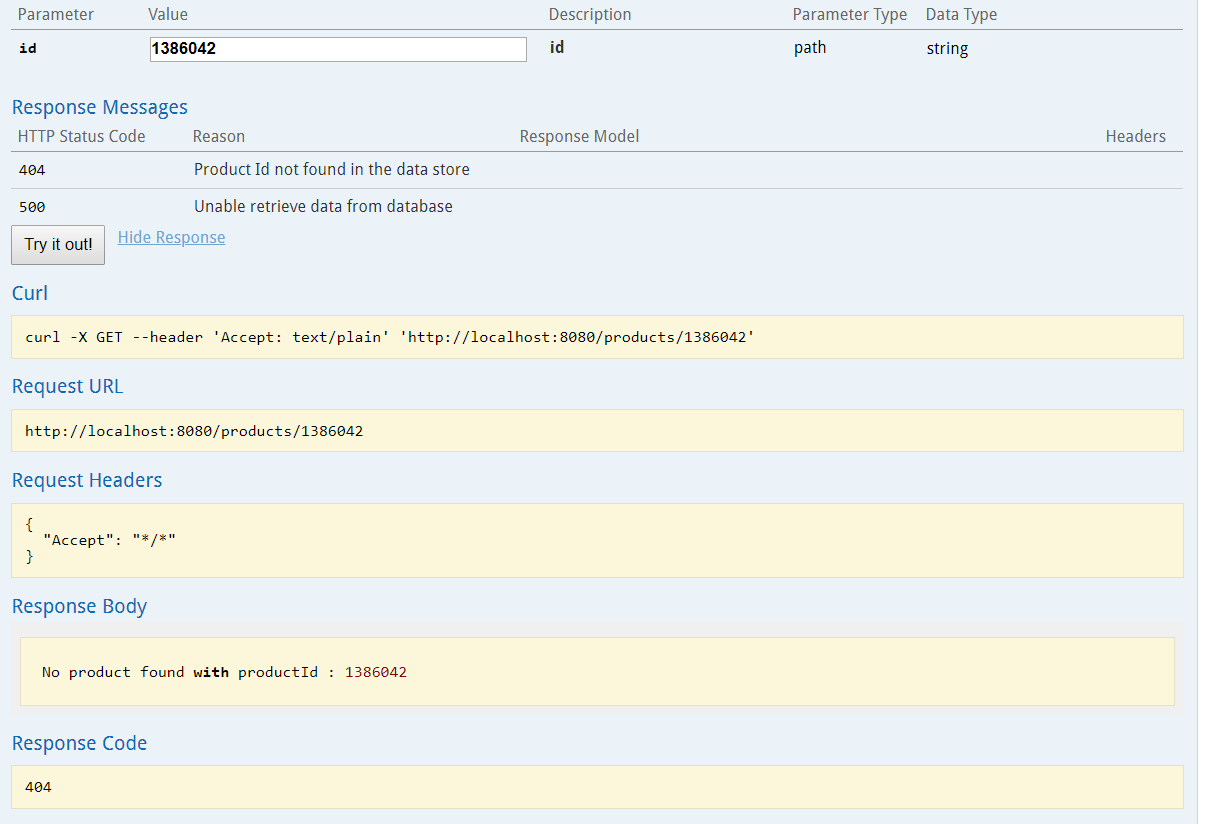
**Request# When ProductId in request header and body doesn't match**

**GET Request**

**Request# Product Details retrieved successfully.**



**Request# When given productid doesn’t exist**



**POST request:**

**Request# New product added to MySQL**

