myRetail RESTful service case study

The goal for this exercise is to create an end-to-end Proof-of-Concept for a products API, which will aggregate product data from multiple sources and return it as JSON to the caller.

The application is build using spring boot framework and NoSQL MongoDB. The POC is running on my personal laptop (<http://redskyretail.ddns.net/api/v1/products/15117729> ) using No-IP.

Note: It is also hosted on the heroku.com (<https://redskyretail.herokuapp.com/api/v1/products/4242>) but currently there is a issue with the mongodb hosted on their site.

This document provides the details on how to access the case study code base from GitHub and the external URL to access the REST APIs for the product.

* **Link to codebase:**

<https://github.com/SR-Alchimie/myretail/>

|  |  |  |  |
| --- | --- | --- | --- |
| **RestAPI URls Details** | | | |
| **Sr.No** | **Method** | **API path** | **Payload** |
| 1 | GET | /api/vi/products/{id} | N/A |
| 2 | PUT | /api/vi/products/{id} | {"id":2000999,"name":"Find Name Doctor Who: Power Of The Daleks"," current\_price":{ "currency\_code":"USD","value":24.98}} |
| 3 | POST | /api/vi/products | {"id":2000999,"name":"Find Name Doctor Who: Power Of The Daleks", "current\_price":{ "currency\_code":"USD","value":24.98}} |

Below are the links to access the RESTful service that can retrieve product and price details by ID. Implementation is done only for GET and POST product details. For the price update the PUT request

**GET Service Request service links scenarios**

1. Success – <http://redskyretail.ddns.net/api/v1/products/15117729>
2. Product not found - <http://redskyretail.ddns.net/api/v1/products/43422442>
3. Passing non numeric value - - <http://redskyretail.ddns.net/api/v1/products/1511AAA7729>
4. Passing negative value - <http://redskyretail.ddns.net/api/v1/products/1511AAA7729>
5. Passing null - <http://redskyretail.ddns.net/api/v1/products/null>
6. Not passing any value - <http://redskyretail.ddns.net/api/v1/products/>

**PUT Service Request service link to update the price**

**URL**:- <http://redskyretail.ddns.net/api/v1/products/2000999>

**Set content-type**:- Content-Type: application/json

**pay load**:-

{

"id": 2000999,

"name": "Find Name Doctor Who: Power Of The Daleks",

"current\_price": {

"currency\_code": "USD",

"value": **24.98**

}

}

**Post methods –**

1. Adding a new product successfully

[http://redskyretail.ddns.net/api/v1/products](http://localhost:8080/api/v1/products)

Content-Type: application/json

pay load - {

"id": 2000999,

"name": "Find Name Doctor Who: Power Of The Daleks",

"current\_price": {

"currency\_code": "USD",

"value": 24.98

}

}

1. Product id already exists base on product id

[http://redskyretail.ddns.net/api/v1/products](%20http://redskyretail.ddns.net/api/v1/products)

Content-Type: application/json

pay load - {

"id": 2000999,

"name": "Find Name Doctor Who: Power Of The Daleks **– updated Name**",

"current\_price": {

"currency\_code": "USD",

"value": 24.98

}

}

1. Product id already exists base on product id

[http://redskyretail.ddns.net/api/v1/products](http://localhost:8080/api/v1/products)

Content-Type: application/json

pay load - {

"id": 2000999,

"name": "Find Name Doctor Who: Power Of The Daleks",

"current\_price": {

"currency\_code": "USD",

"value": 24.98

}

}