Case Study myRetail RESTful service

Technologies/Tools Used for Case study

1. Spring Boot 1.4.2.RELEASE
2. Spring Security (Basic authentication)
3. Eh Cache
4. Swagger 2.6.1 using Springfox API
5. Cassandra 2.1.8
6. Spring Data
7. slf4j logger
8. jUnit
9. Mockito
10. Maven 3
11. JDK 1.8
12. STS 3.9.0

How to run application:

The application can be started by running the below maven command

mvn clean install spring-boot:run

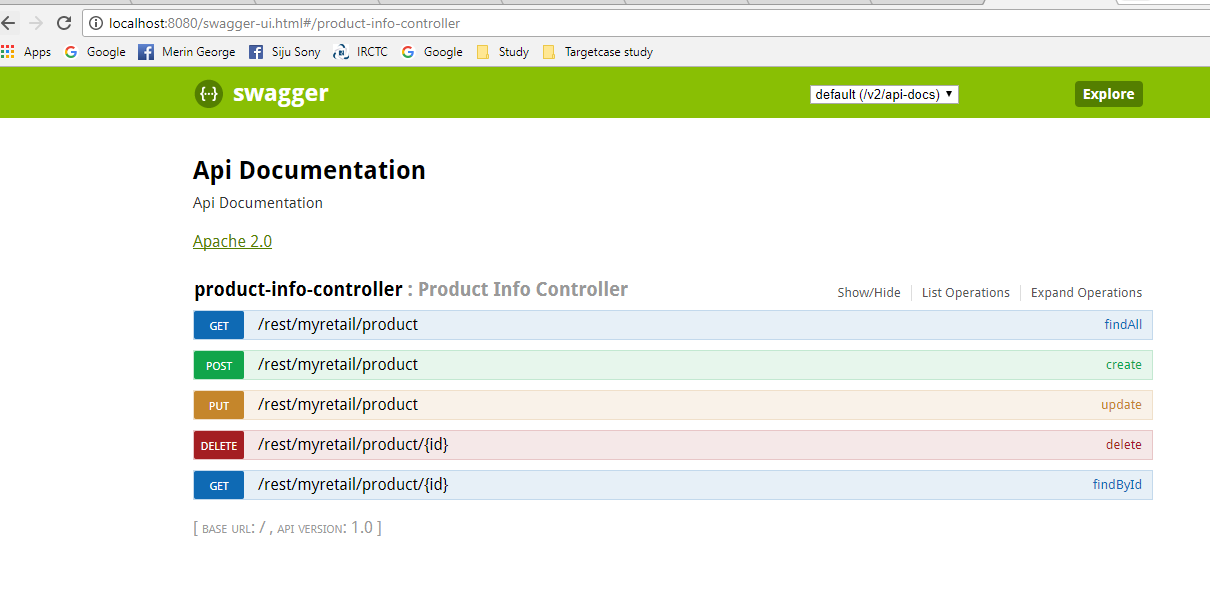
API Information Document

API Information is available under swagger UI below

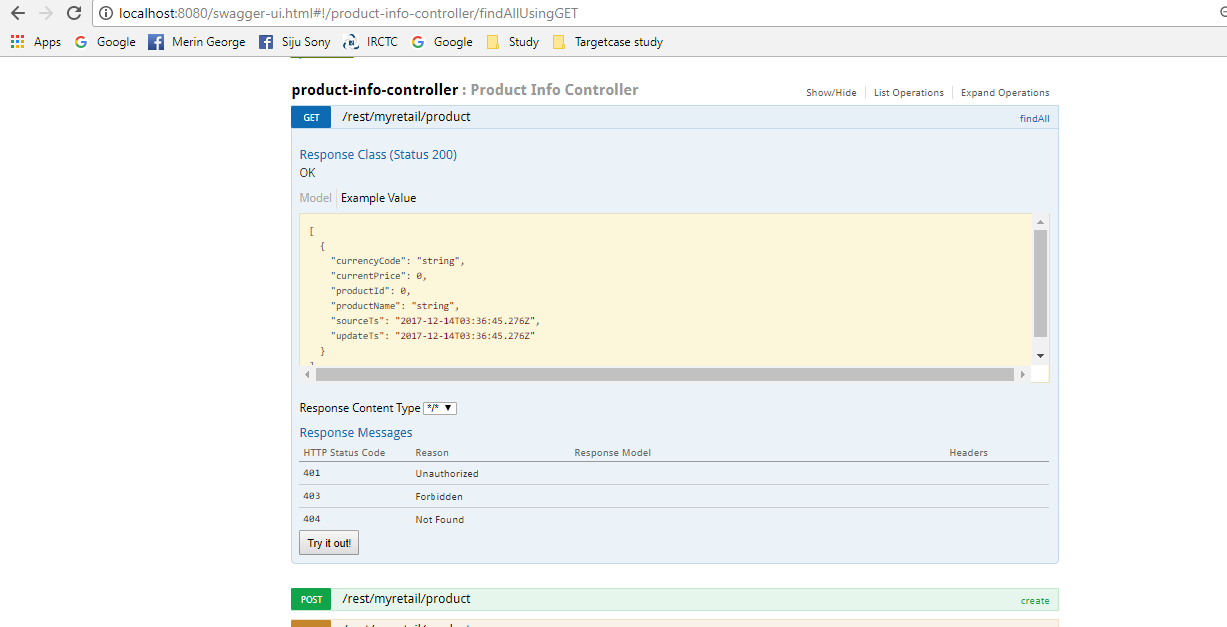
<http://localhost:8080/swagger-ui.html#/>

username : myretail

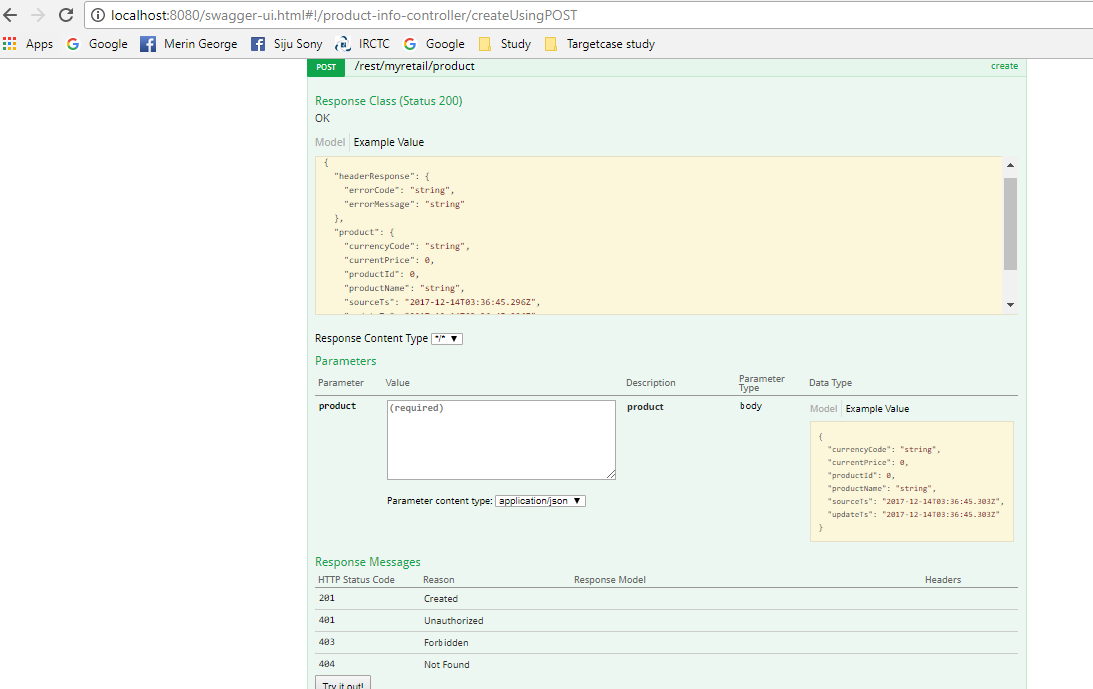
password : myretail



Get Request Info



POST Request Info



Testing Application:

Testing for the application was performed using Postman and junit test cases.

1. Get Request :

Authentication: usr/pswd : myretail/myretail

Request: <http://localhost:8080/rest/myretail/product/13860428>

Response:

{

"product": {

"productId": 13860428,

"sourceTs": 1512994332000,

"updateTs": 1513167132000,

"productName": "The Big Lebowski (Blu-ray)",

"currencyCode": "USD",

"currentPrice": 123.33

},

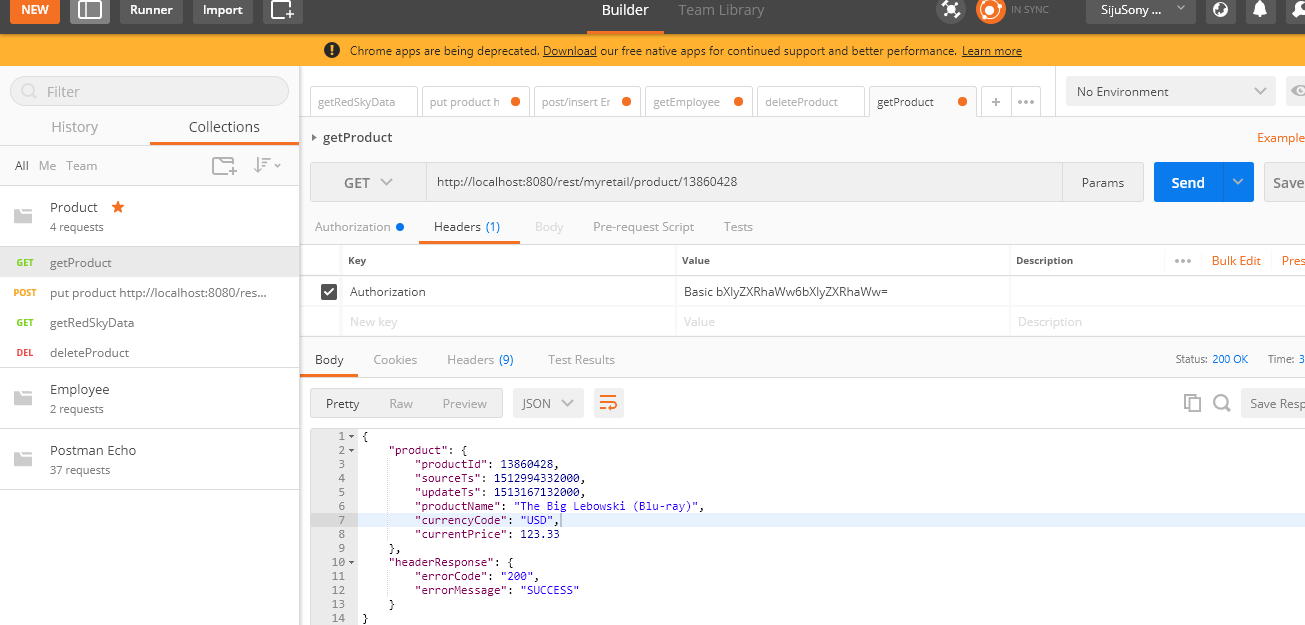
"headerResponse": {

"errorCode": "200",

"errorMessage": "SUCCESS"

}

}



1. POST Request :

Authentication: usr/pswd : myretail/myretail

Request URL : http://localhost:8080/rest/myretail/product

Request:

{"productId":13860428,

"sourceTs":"2017-12-11T12:12:12",

"updateTs":"2017-12-13T12:12:12",

"productName":"Test",

"currencyCode":"USD",

"currentPrice":123.33

}

Response:

{

"product": {

"productId": 13860428,

"sourceTs": 1512994332000,

"updateTs": 1513167132000,

"productName": "Test",

"currencyCode": "USD",

"currentPrice": 123.33

},

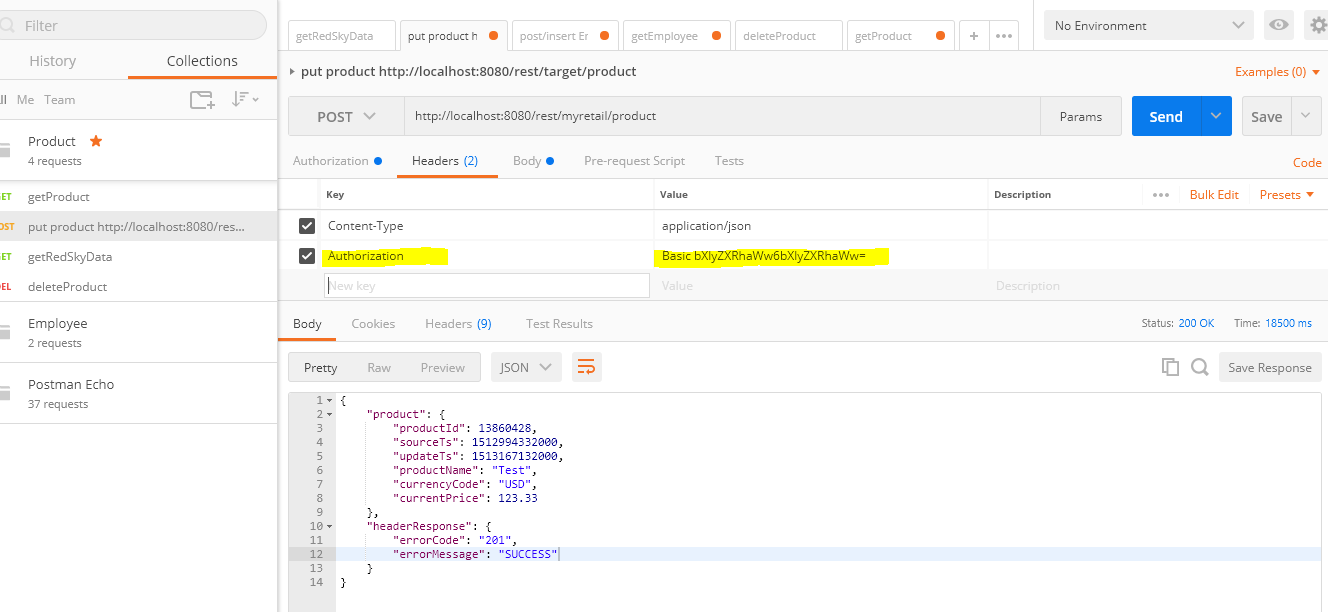
"headerResponse": {

"errorCode": "201",

"errorMessage": "SUCCESS"

}

}



Delete Request:

URL : <http://localhost:8080/rest/myretail/product/13860428>

GET ALL

URL : <http://localhost:8080/rest/myretail/product/>

Cassandra table Format

Table is designed in such a way that the latest record for a particular product is always available as first record

CREATE TABLE price\_info (

product\_id bigint,

source\_ts TIMESTAMP,

update\_ts TIMESTAMP,

product\_name TEXT,

current\_price DOUBLE,

currency\_code TEXT,

PRIMARY KEY ((product\_id), source\_ts, update\_ts)

) WITH CLUSTERING ORDER BY (source\_ts DESC, update\_ts DESC);

