

ADIDAS BACKEND DEVELOPMENT CHALLENGE-P1

User Manual

Author: Anushka Sharma

Email: anushkasharma0809@gmail.com

Contact: +91-8966055832

INDEX:

1. Initial setup.
 - 1.1. JDK 1.8 installation (Skip if already present).
 - 1.2. DB setup
 - 1.3. Port opening (Skip if 8080,8088 are available)
 - 1.4. Postman desktop client (Skip if already present).
2. Starting the APIs.
 - 2.1. Starting 'Write REST API'
 - 2.2. Starting 'Product API'
3. API usage and verification.
 - 3.1. Import request collection in postman UI.
 - 3.2. Execute the requests for verification
4. Test cases.
 - 4.1. Positive test cases
 - 4.2. Negative test cases
5. Miscellaneous.

1. INITIAL SETUP

1.1 JDK installation:

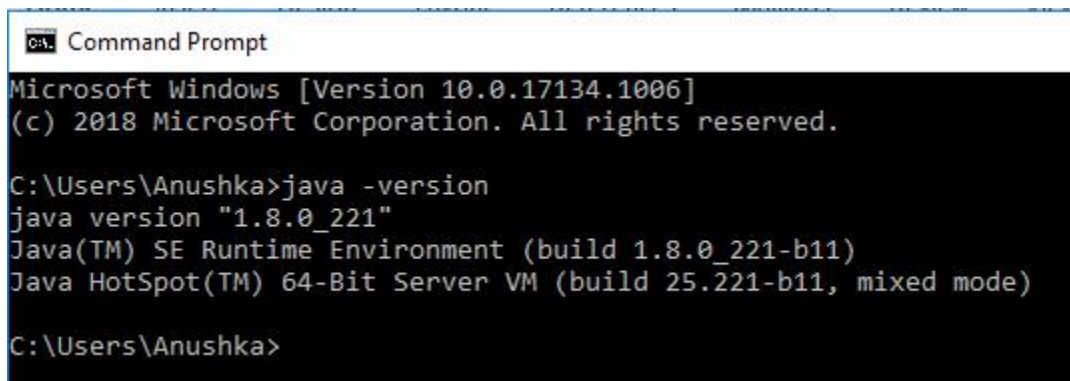
JDK 1.8 is required to run the APIs. Please make sure JDK is installed in the system where the APIs will be running.

To download JDK from the web, please go through the below link.

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

To verify the installation, run below command on the command prompt:

Java -version



```
Command Prompt
Microsoft Windows [Version 10.0.17134.1006]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Anushka>java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)

C:\Users\Anushka>
```

1.2 DB Setup

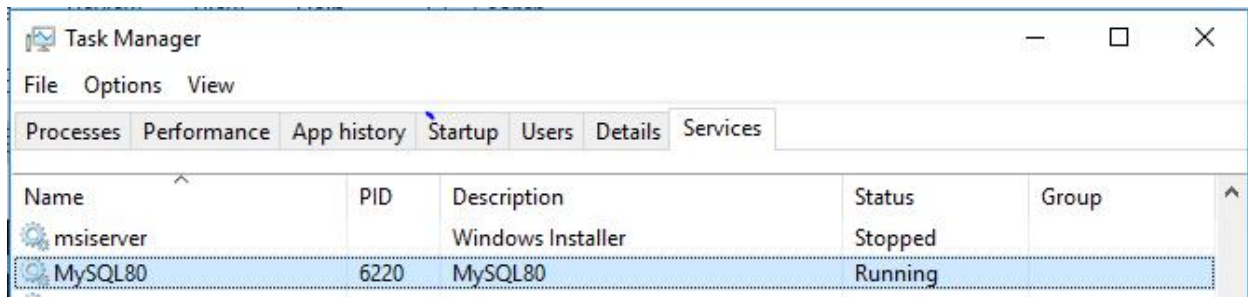
1.2.1 MySQL DB installation:

Please install MySQL DB v8 or below. This will act as the backend DB of the persistence service.

To download the installer please go to the below URL:

<https://dev.mysql.com/downloads/mysql/>

To verify the installation and server startup, check the running services:



1.2.2 Schema creation:

Once the DB is installed, user needs to import the SQL file into the database through MySQL Workbench.

SQL file name: *'adidas_product'*

Location: *..\..\ADIDAS_BACKEND_CODING_CHALLENGE_ANUSHKA\test*

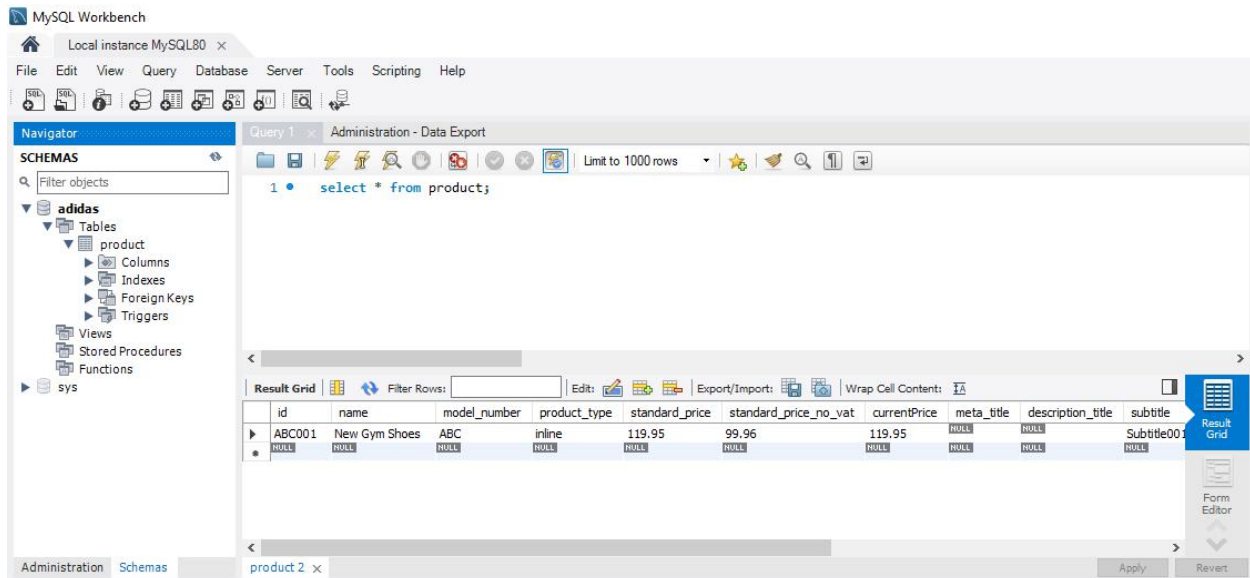
Importing this file will do following operations:

- A. Create a DB with name *'adidas'*.
- B. Create a Table with name *'product'*.
- C. Create a Primary Key on ID column.
- D. Insert one row of data into the table.

Please note: The DB name can be changed by editing the SQL file and manually running the three scripts through MySQL workbench.

Kindly do not change the meta data structure other than the schema name.

To verify the initial DB setup, go to the MySQL workbench.



1.3 Port Opening:

Port 8080 and 8089 should be listening on the system where the APIs will be running.

To verify the ports, please execute below command on the command prompt:
netstat -an

```

Microsoft Windows [Version 10.0.17134.1006]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Anushka>netstat -an

Active Connections

Proto Local Address           Foreign Address         State
TCP 0.0.0.0:135              0.0.0.0:0               LISTENING
TCP 0.0.0.0:445              0.0.0.0:0               LISTENING
TCP 0.0.0.0:3306             0.0.0.0:0               LISTENING
TCP 0.0.0.0:5040             0.0.0.0:0               LISTENING
TCP 0.0.0.0:7680             0.0.0.0:0               LISTENING
TCP 0.0.0.0:33060            0.0.0.0:0               LISTENING
TCP 0.0.0.0:49664            0.0.0.0:0               LISTENING
TCP 0.0.0.0:49665            0.0.0.0:0               LISTENING
TCP 0.0.0.0:49666            0.0.0.0:0               LISTENING

```

1.4 Postman application:

To test the API responses, download and install postman in one click from the below URL:

<https://www.getpostman.com/downloads/>

To understand the usage of the application, please refer Index number-3, API usage and verification.

2 STARTING THE APIs

2.1 Starting the 'Write REST API': Runs on port 8080

Open a command prompt from the folder

`../..ADIDAS_BACKEND_CODING_CHALLENGE_ANUSHKA/jar`

Run below command to start the API. Please ignore the SLF4J warnings.

`java -jar Write_REST_API.jar <connectionURL> <userName> <password>`

```
A:\Anushka Sharma>java -jar Write_REST_API.jar jdbc:mysql://127.0.0.1:3306/adidas root adidas
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

Here,

<connectionURL> = It is the connection URL to connect to the DB created in the step 1.2. – DB Setup.

Typically the connection URL contains of below parameters,

`jdbc:mysql://hostname:port number/database name`

<userName>= username of the MySQL DB having the admin privileges on the database name mentioned in the connection URL.

<password>= password of the specific user to connect to the database. The user should have read and write permission on the database name given in the URL

2.2 Starting the product API.: Runs on port 8088.

Follow the same instructions as in 2.1. Run the below command:

`java -jar Product_API.jar <connectionURL> <userName> <password>`

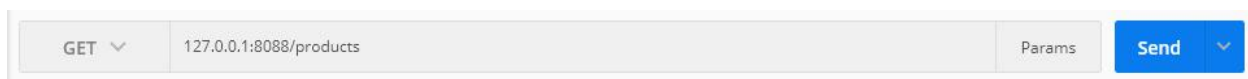
3 API USAGE AND VERIFICATION

Postman is an application that is used to test the REST API requests.

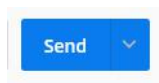
Collection in postman is a logical group of requests.

One can edit the requests and execute it as per the requirement dynamically.

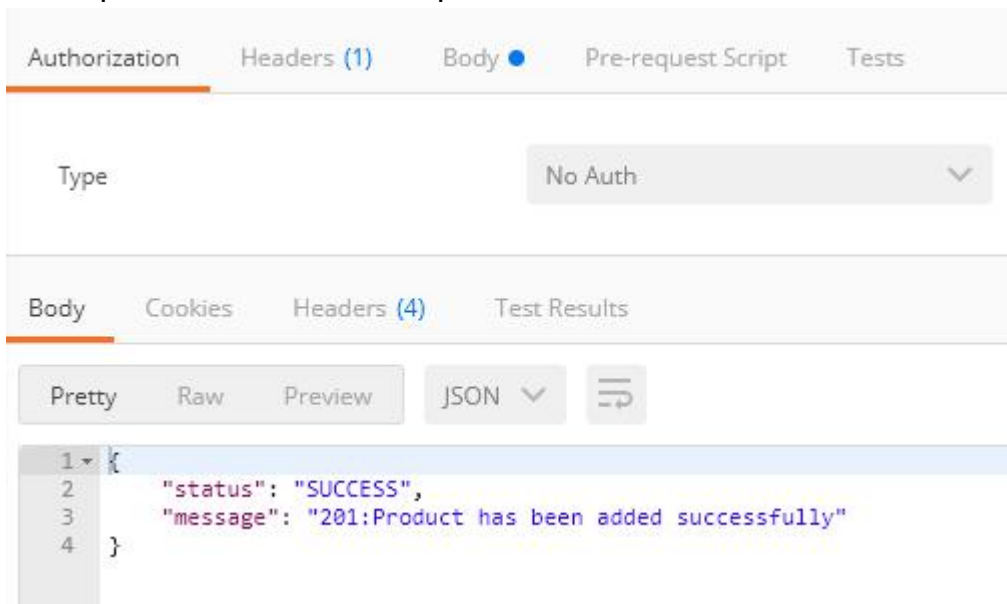
To test any request, provide the URL in the address bar.



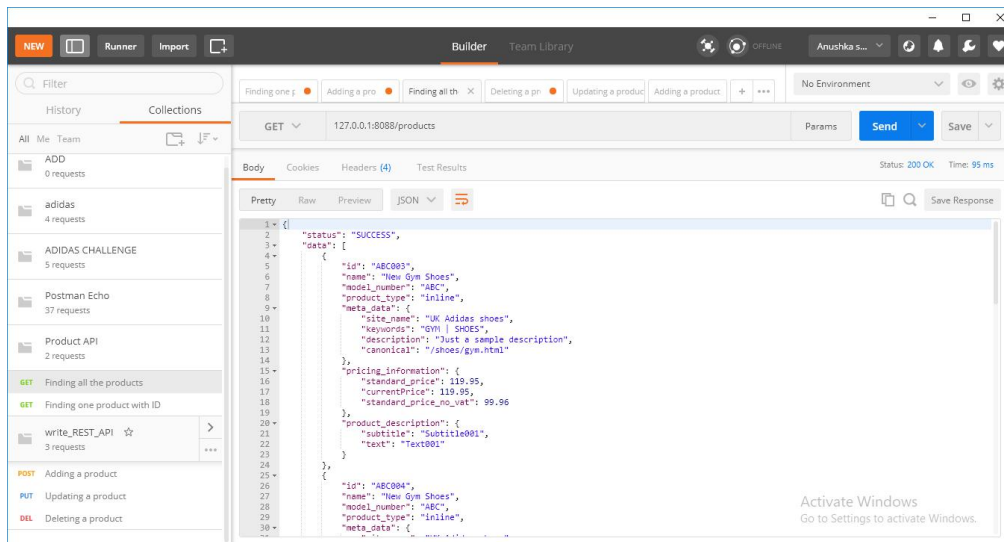
Click on 'send' button to execute the API request.



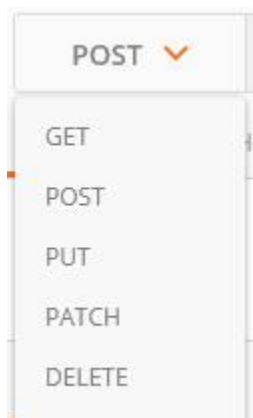
The responses to each API request can be seen in the 'Authorization'



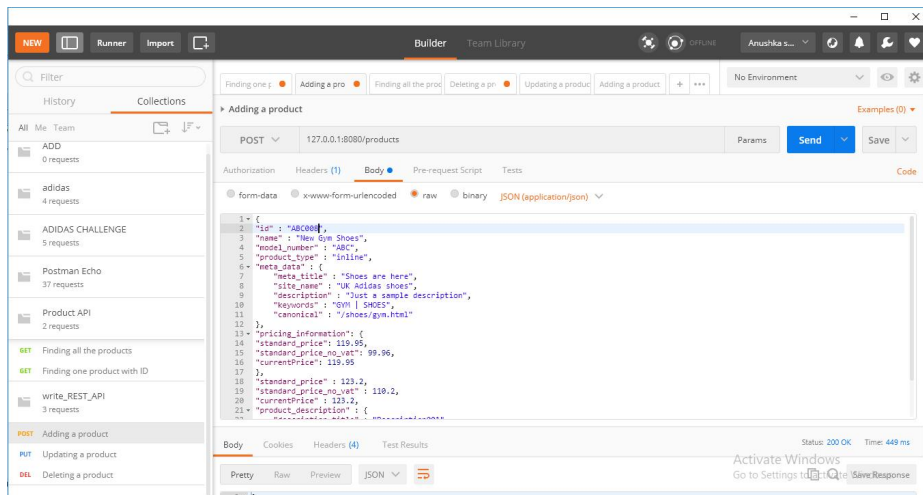
The results of operation like 'GET' will be displayed in the 'body tab':



To test different API operations/commands, change the command type from below drop down:



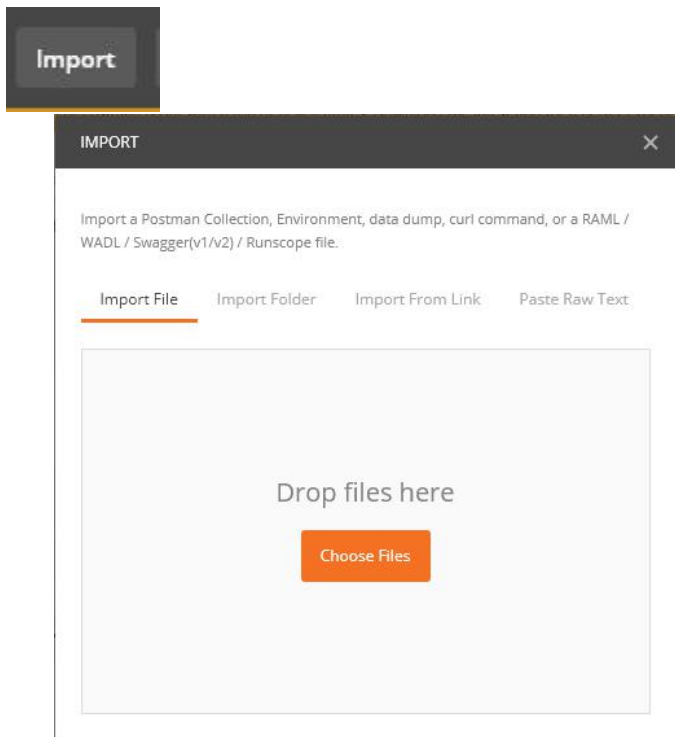
In case of POST and PUT operations, body of the product to be added/updated should be given in the 'request body' tab.



3.1 Import request collection in postman UI:

Import the collection of requests present in the artifacts by the name *'write_REST_API.postman_collection'* in the postman UI.

`\\ADIDAS_BACKEND_CODING_CHALLENGE_ANUSHKA\test`



Click on Choose Files and select the file `write_REST_API.postman_collection`

Repeat the same steps for importing Product API.postman_collection.

3.2 Execute the requests for verification

Once both the collections are imported, send the requests and validate the results.

Write REST API: Supports CREATE,UPDATE AND DELETE

Product API: Supports READ(all products), READ (Specific product ID)

4 TEST CASES

4.1 Positive Test cases:

- ✓ CREATE(POST): Add a valid product in the request body.
- ✓ UPDATE(PUT): Add a valid product and product id in the request body and URL
- ✓ DELETE: Add a valid product id in the request URL.
- ✓ READ(GET/ID): Add a valid product id in the request URL

4.2 Negative Test cases:

- ✓ CREATE(POST) : Add an invalid product in the request body.
- ✓ CREATE(POST): Provide a duplicate product id in the request body.
- ✓ CREATE(POST): Provide an empty product with only id.
- ✓ CREATE(POST): Provide an empty product with empty id.

- ✓ UPDATE(PUT): Provide an invalid request body for the product.

- ✓ READ(GET/ID): Provide an id that is not present in the database.
- ✓ READ(GET): Execute this request when no data is present in the database.

- ✓ DELETE: Delete an id that is not present in the DB.

5 MISCELLANEOUS

- a.** Kindly do not change the meta data structure present in the SQL file.
- b.** The username used to connect to the MYSQL database should have read and write permissions on the schema created.
- c.** The port on which the APIs run are 8080 and 8088.
- d.** The APIs have been developed having MYSQL database as the repository.