# ADIDAS BACKEND DEVELOPMENT CHALLENGE-P1

User Manual

Author: Anushka Sharma

Email: [anushkasharma0809@gmail.com](mailto:anushkasharma0809@gmail.com)

Contact: +91-8966055832

INDEX:

1. Initial setup.
   1. JDK 1.8 installation (Skip if already present).
   2. DB setup
   3. Port opening (Skip if 8080,8088 are available)
   4. Chrome extension- postman (Skip if already present).
2. Starting the application.
   1. Starting ‘Write REST API’
   2. Starting ‘Product API’
3. Application usage and verification.
   1. Import request collection in postman UI.
   2. Execute the requests for verification
4. Test cases.
   1. Positive test cases
   2. Negative test cases
5. List of Artifacts
6. **INITIAL SETUP**
   1. **JDK installation:**

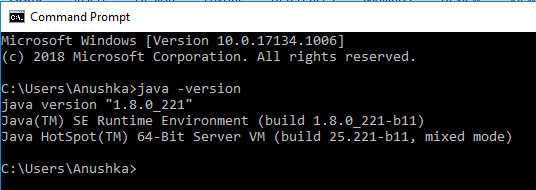
JDK 1.8 is required to run the application. Please make sure JDK is installed in the system where the application will be running.

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



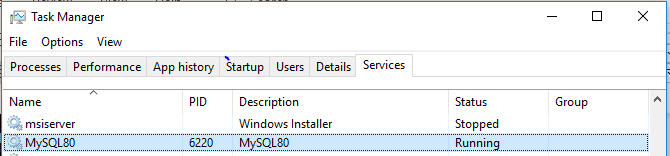
* 1. **DB Setup**
     1. MySQL DB installation:

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

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. 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’- present along with this document.

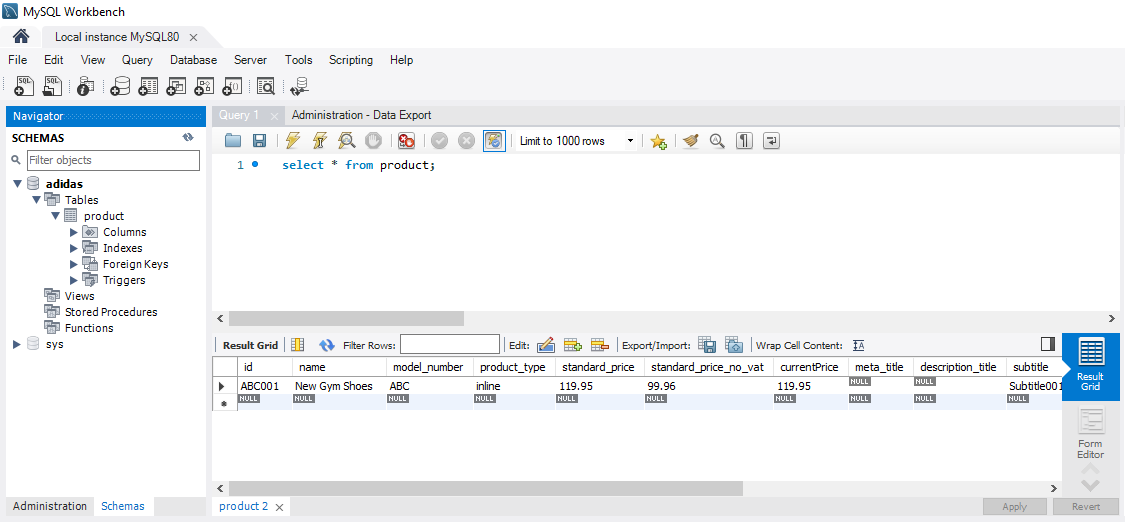
Importing this file will do following operations:

1. Create a DB with name ‘adidas’.
2. Create a Table with name ‘product’.
3. Create a Primary Key on ID column.
4. 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.

**Kindly do not change the table name**

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

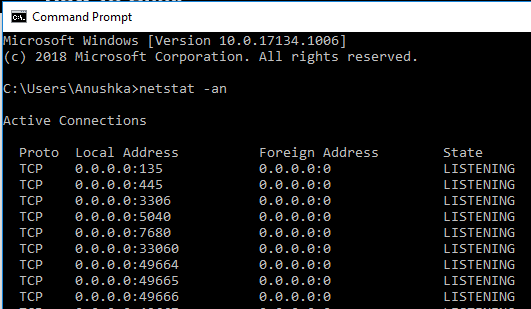


* 1. Port Opening:

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

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

netstat -an

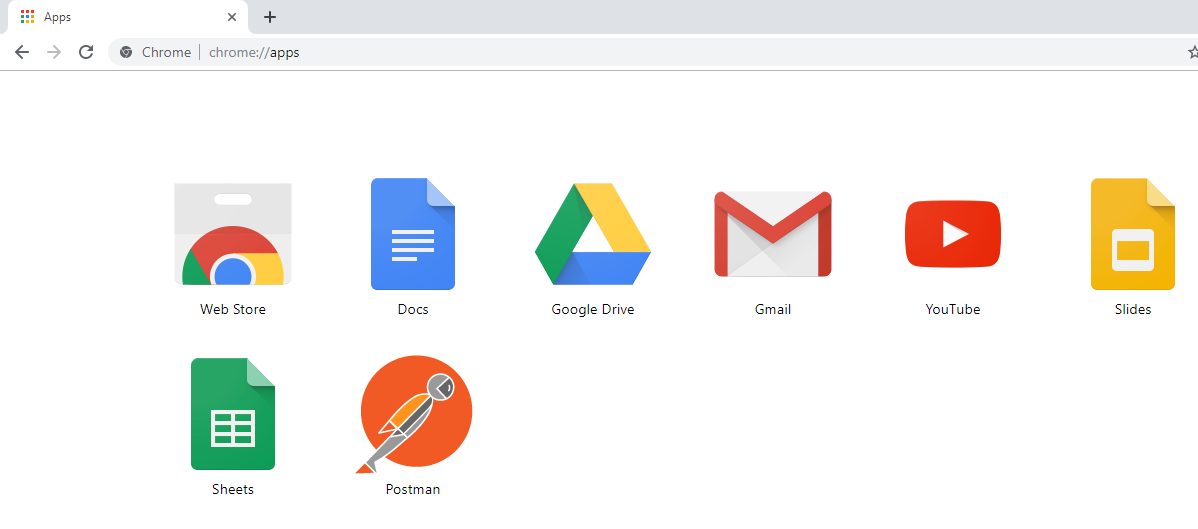


* 1. Chrome extension-postman:

To test the API responses, download and install postman by following very simple steps on the below URL:

[https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop//%40](https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop/%40)

Once installed, postman can be launched from the chrome applications:

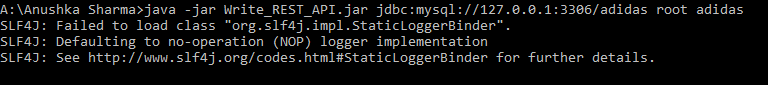


1. **STARTING THE APPLICATION**
   1. Starting the ‘Write REST API’: Runs on port 8080

Open a command prompt from the folder where all the artifacts are present.

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

Java -jar Write\_REST\_API.jar <connectionURL> <userName> <password>



Here,

<connectionURL> = 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.

* 1. Starting the product API.: Runs on port 8088

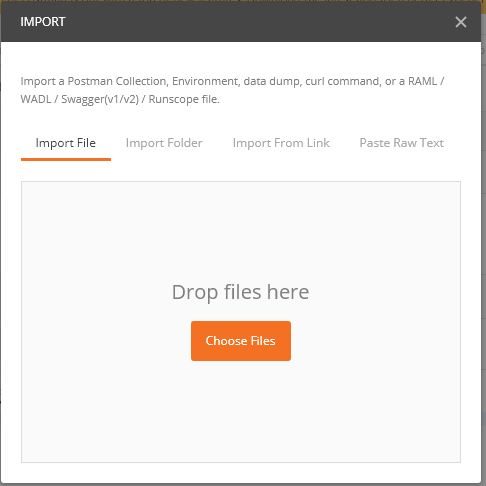
Follow the same instructions as in 2.1. Run the below command:

Java -jar Product\_API.jar <connectionURL> <userName> <password>

1. **APPLICATION USAGE AND VERIFICATION:**
   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.





Click on Choose Files and select the file write\_REST\_API.postman\_collection

**Repeat the same steps for importing Product API.postman\_collection.**

* 1. 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, READ (SPECIFIC ID)

1. **TEST CASES**
   1. **Positive Test cases:**

* CREATE(POST): Add a valid product in the request body.
* UPDATE(PUT): Add a valid product in the request body.
* UPDATE(PUT): Add a valid product id in the request URL.
* DELETE: Add a valid product id in the request URL.
* READ(GET): Add a valid product id in the request URL
  1. **Negative Test cases:**
* CREATE(POST) : Add an invalid product in the request body.