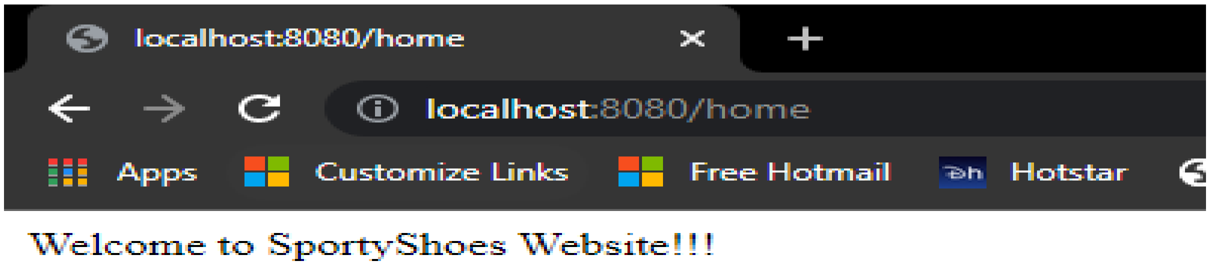
Screenshots of Project Working-

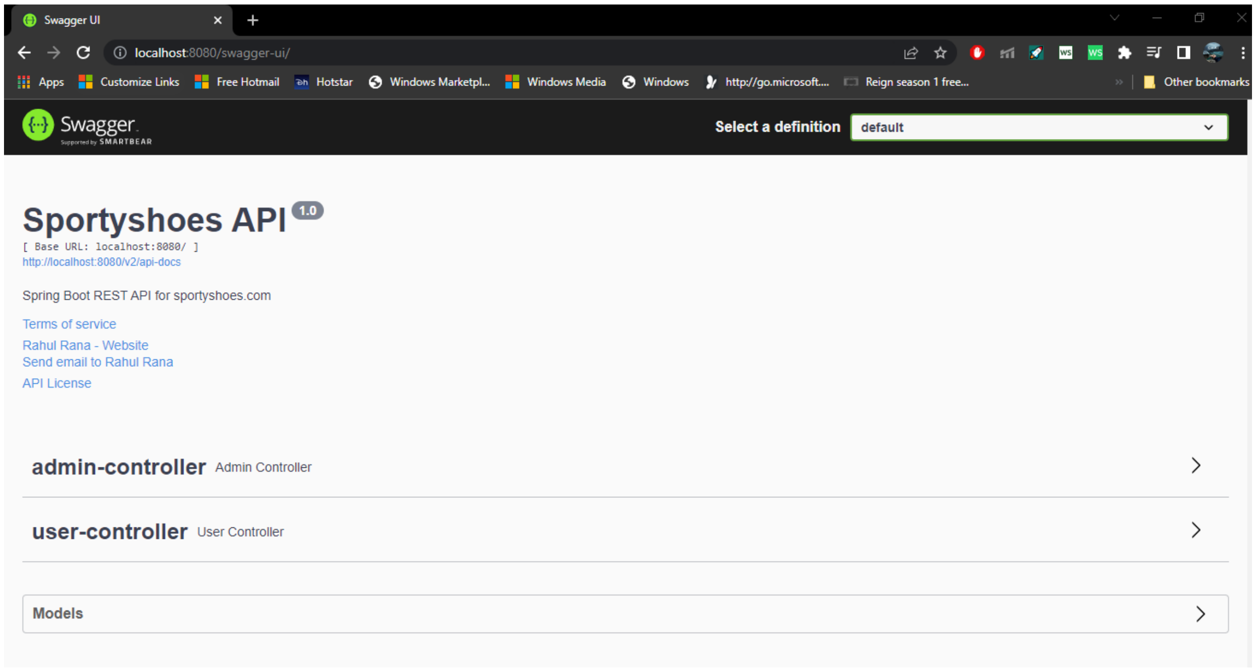
On running the **SportyshoesApplication.java** file using **Eclipse IDE** , it will start working and reflect **welcome to sportshoes** on the console. On entering the url : http://localhost:8080/home on the chrome or other browser it will reflect the Welcome message on the server.



Welcome to SportyShoes Website!!!

Picture 1- Welcome screen

On running the **Swagger 2** using url: http://localhost:8080/swagger-ui/



Sportyshoes API 1.0

[Base URL: localhost:8080/]

http://localhost:8080/v2/api-docs

Spring Boot REST API for sportyshoes.com

Terms of service

Rahul Rana - Website

Send email to Rahul Rana

API License

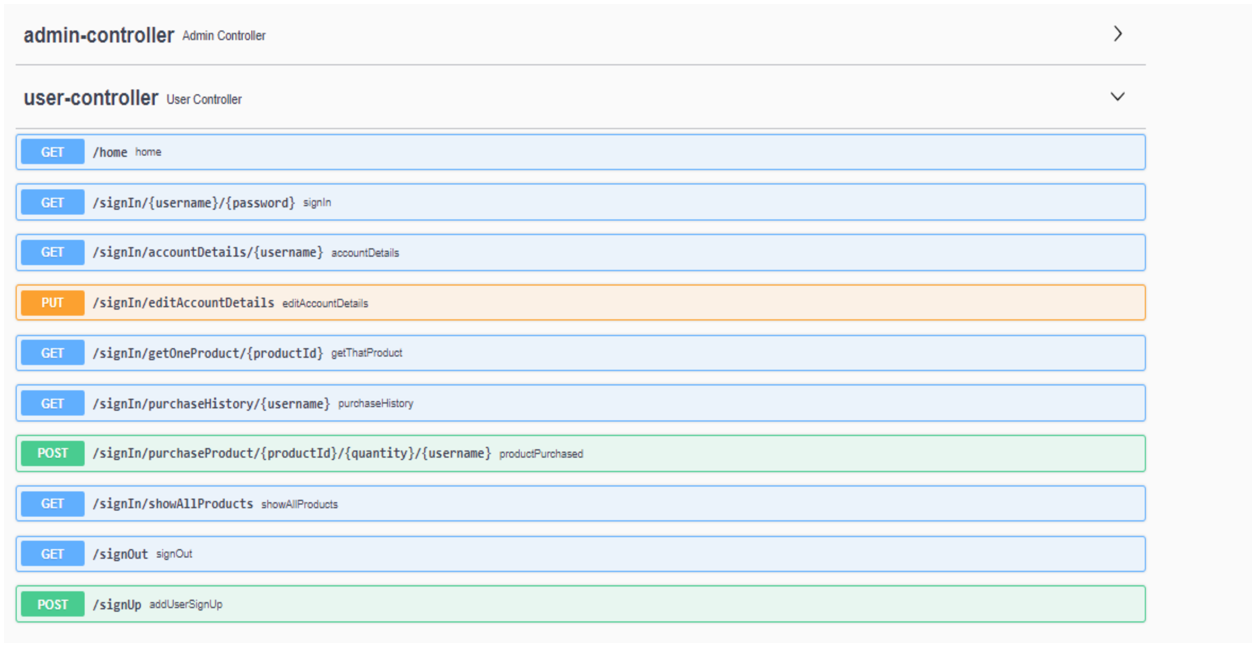
admin-controller Admin Controller

user-controller User Controller

Models

Picture 2- swagger url is entered it will show both admin & user controller all functions

Now all user controls are shown in below image.

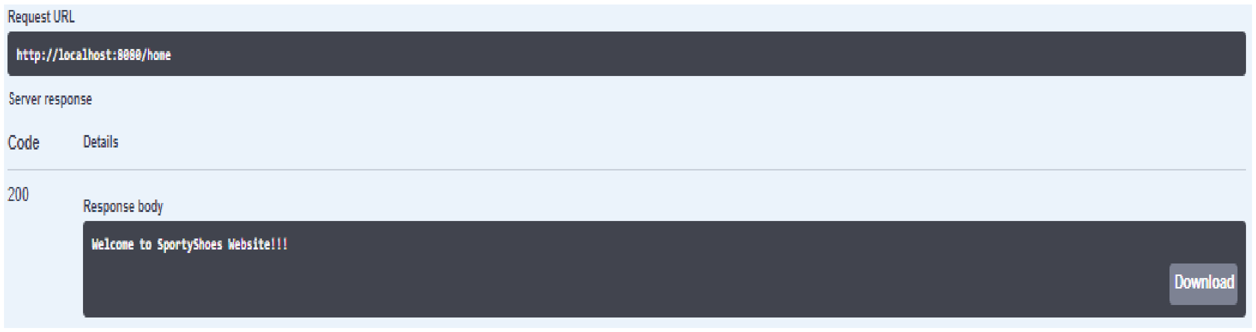


admin-controller Admin Controller
user-controller User Controller

```
GET  /home  home
GET  /signIn/{username}/{password}  signIn
GET  /signIn/accountDetails/{username}  accountDetails
PUT  /signIn/editAccountDetails  editAccountDetails
GET  /signIn/getOneProduct/{productId}  getThatProduct
GET  /signIn/purchaseHistory/{username}  purchaseHistory
POST /signIn/purchaseProduct/{productId}/{quantity}/{username}  productPurchased
GET  /signIn/showAllProducts  showAllProducts
GET  /signOut  signOut
POST /signUp  addUserSignUp
```

Picture 3- all user functions

Lets say we call /home using swagger or via Url: http://localhost:8080/home

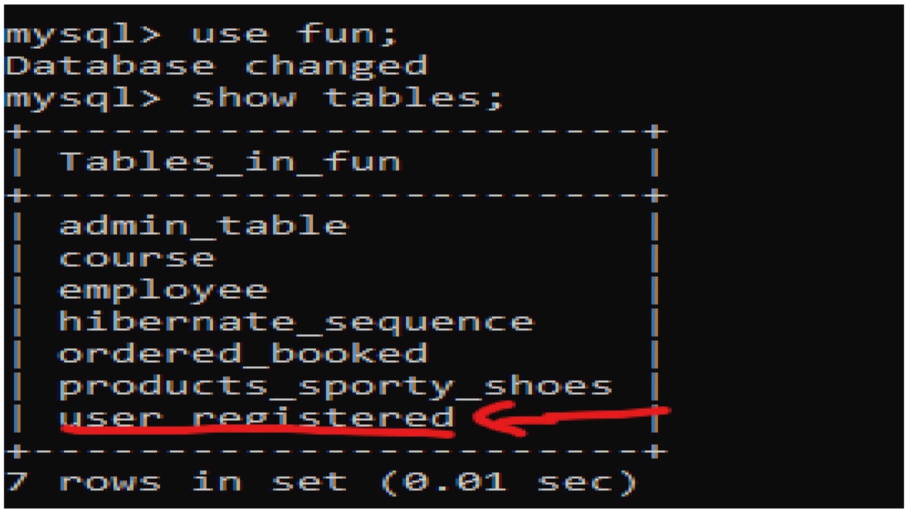


```
http://localhost:8080/home
```

```
Welcome to SportyShoes Website!!!
```

Picture 4- /home GetMapping function for home function

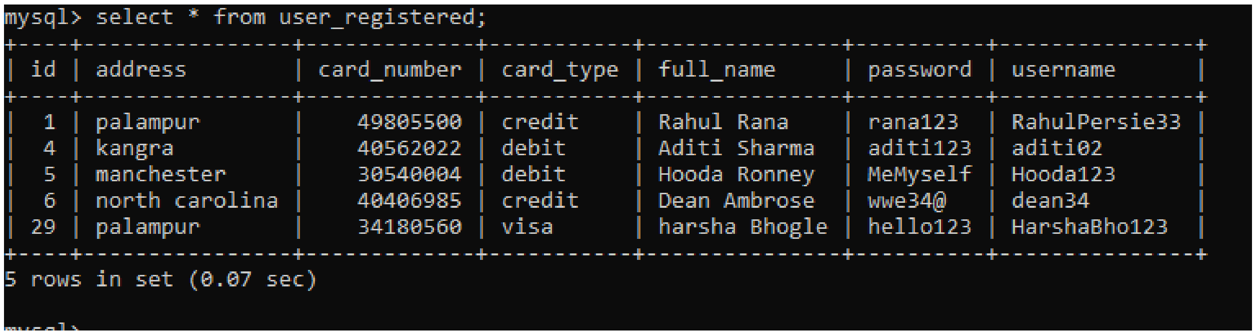
If User want to signUp for the SportyShoes, then his/her details should be saved to **user_registered** table.



```
mysql> use fun;
Database changed
mysql> show tables;
+-------------------------+
| Tables_in_fun           |
+-------------------------+
| admin_table             |
| course                  |
| employee                |
| hibernate_sequence      |
| ordered_booked          |
| products_sporty_shoes   |
| user_registered         |
+-------------------------+
7 rows in set (0.01 sec)
```

Picture 5- all tables in DB – fun

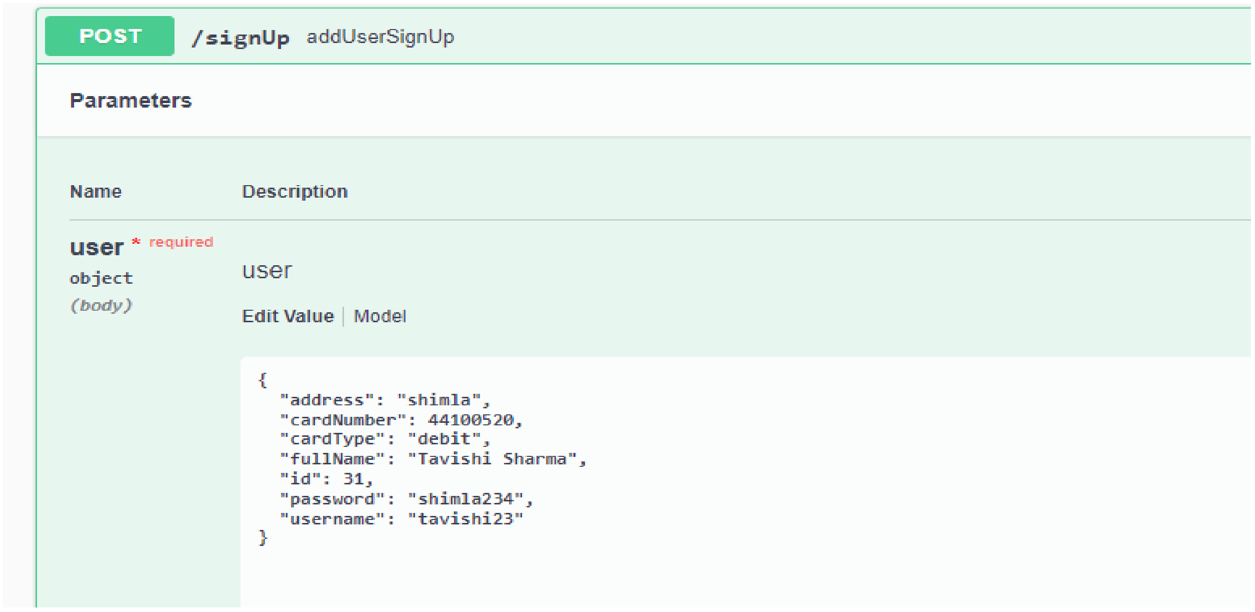
User_Registered table before adding or signing up the new user.



```
mysql> select * from user_registered;
+----+----------------+-------------+-----------+--------------+----------+---------------+
| id | address        | card_number | card_type | full_name    | password | username      |
+----+----------------+-------------+-----------+--------------+----------+---------------+
|  1 | palampur       |    49805500 | credit    | Rahul Rana   | rana123  | RahulPersie33 |
|  4 | kangra         |    40562022 | debit     | Aditi Sharma | aditi123 | aditi02       |
|  5 | manchester     |    30540004 | debit     | Hooda Ronney | MeMyself | Hooda123      |
|  6 | north carolina |    40406985 | credit    | Dean Ambrose | wwe34@   | dean34        |
| 29 | palampur       |    34180560 | visa      | harsha Bhogle | hello123 | HarshaBho123 |
+----+----------------+-------------+-----------+--------------+----------+---------------+
5 rows in set (0.07 sec)
```

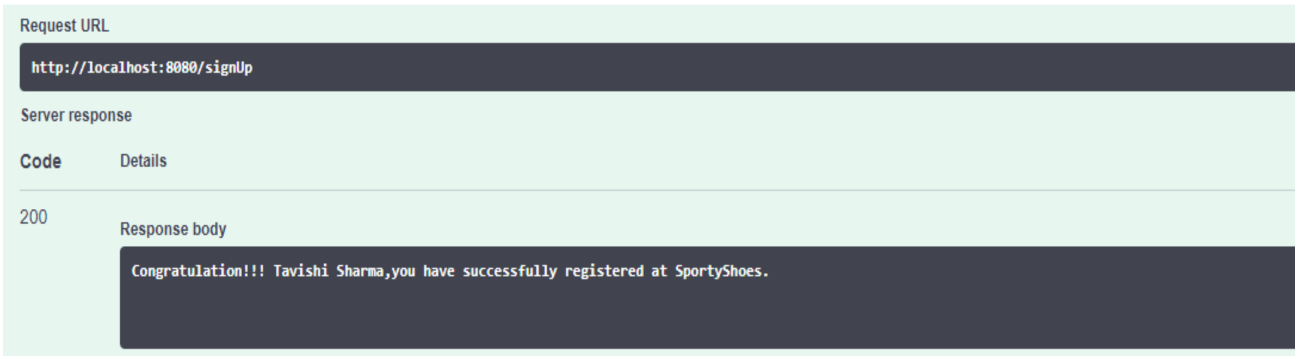
Picture 6- User_registered table before signing up the new user

When we click on **/signUp** in swagger2 then it will add the new user to **User_registered** table.



```
{
  "address": "shimla",
  "cardNumber": 44100520,
  "cardType": "debit",
  "fullName": "Tavishi Sharma",
  "id": 31,
  "password": "shimla234",
  "username": "tavishi23"
}
```

Picture 7- Enter the signUp details in the json format in the swager2



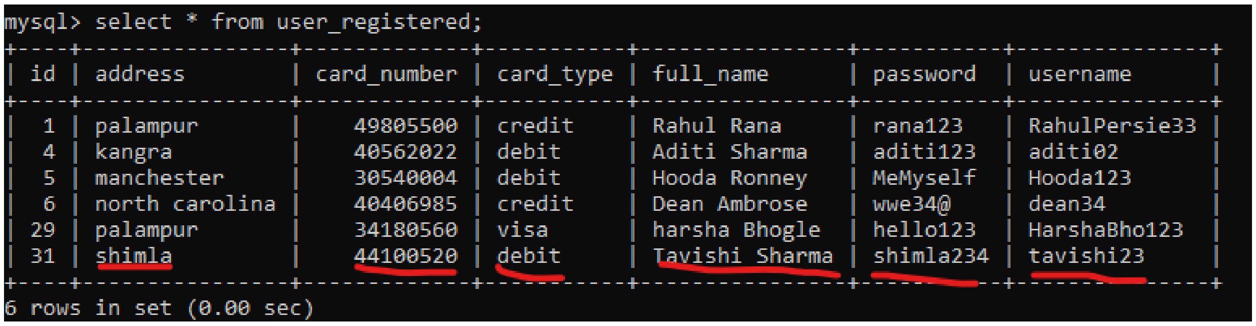
```
http://localhost:8080/signUp
```

200

```
Congratulation!!! Tavishi Sharma,you have successfully registered at SportyShoes.
```

Picture 8- showing result of signUp command in swagger2

After sign up the data will be save in the table **user_registered**.

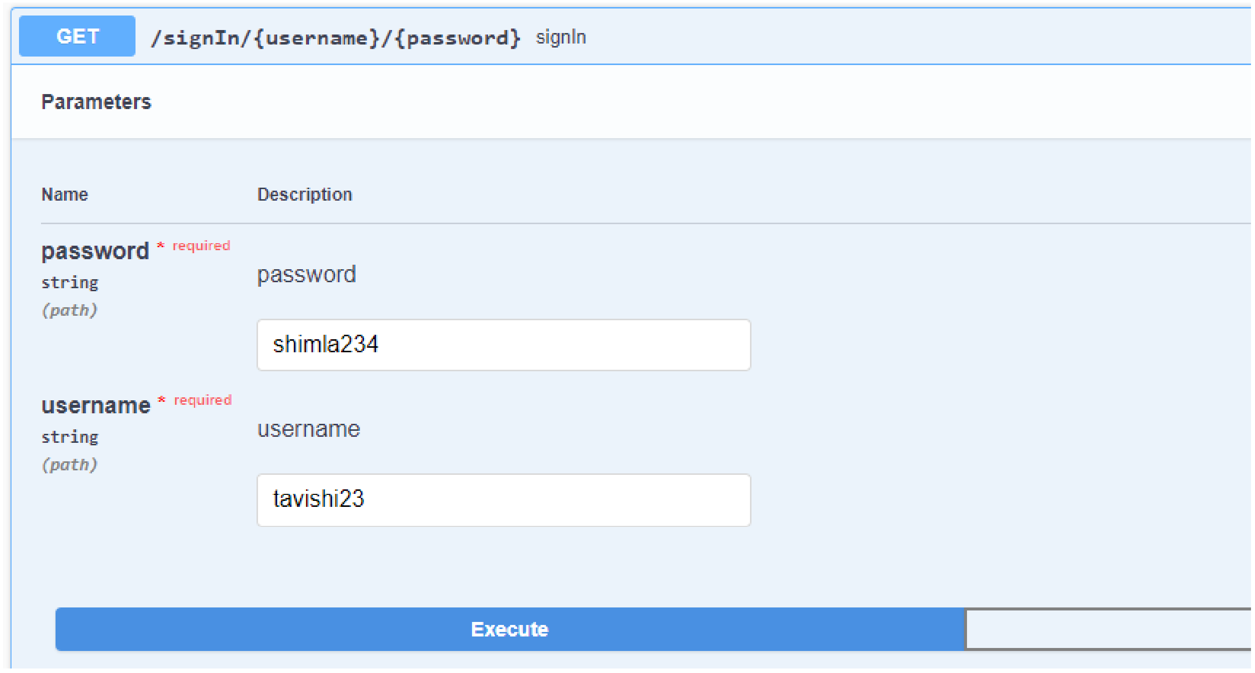


```
mysql> select * from user_registered;
+----+-----------------+-------------+-----------+----------------+-----------+----------------+
| id | address         | card_number | card_type | full_name      | password  | username       |
+----+-----------------+-------------+-----------+----------------+-----------+----------------+
|  1 | palampur        |    49805500 | credit    | Rahul Rana     | rana123   | RahulPersie33  |
|  4 | kangra          |    40562022 | debit     | Aditi Sharma   | aditi123  | aditi02        |
|  5 | manchester      |    30540004 | debit     | Hooda Ronney   | MeMyself  | Hooda123       |
|  6 | north carolina  |    40406985 | credit    | Dean Ambrose   | wwe34@    | dean34         |
| 29 | palampur        |    34180560 | visa      | harsha Bhogle  | hello123  | HarshaBho123   |
| 31 | shimla          |    44100520 | debit     | Tavishi Sharma | shimla234 | tavishi23      |
+----+-----------------+-------------+-----------+----------------+-----------+----------------+
6 rows in set (0.00 sec)
```

Picture 9- data save in the table

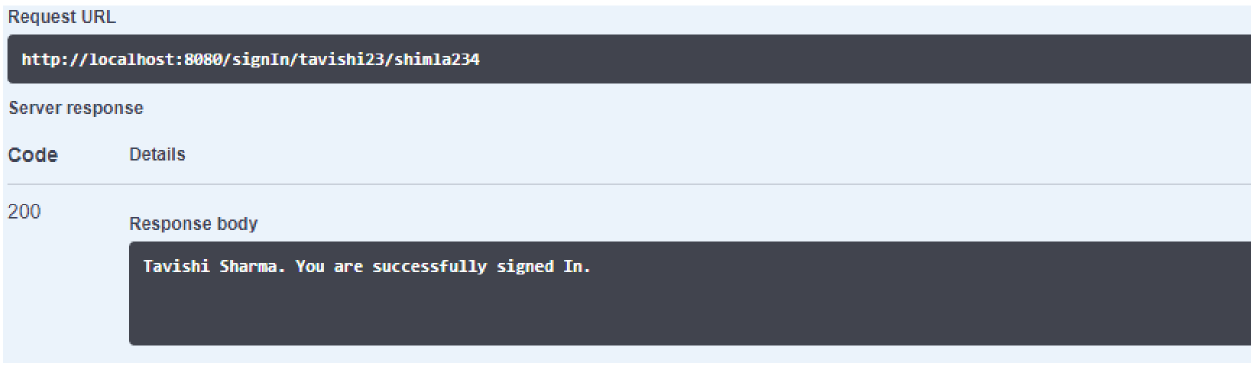
Now if user signIn using password and username at swagger2 option of `/signIn/{username}/{password}`

It will check if user valid then return hello user ,if invalid then ask for you register yourself.



password: shimla234
username: tavishi23

Picture 10- signIn user is valid or not



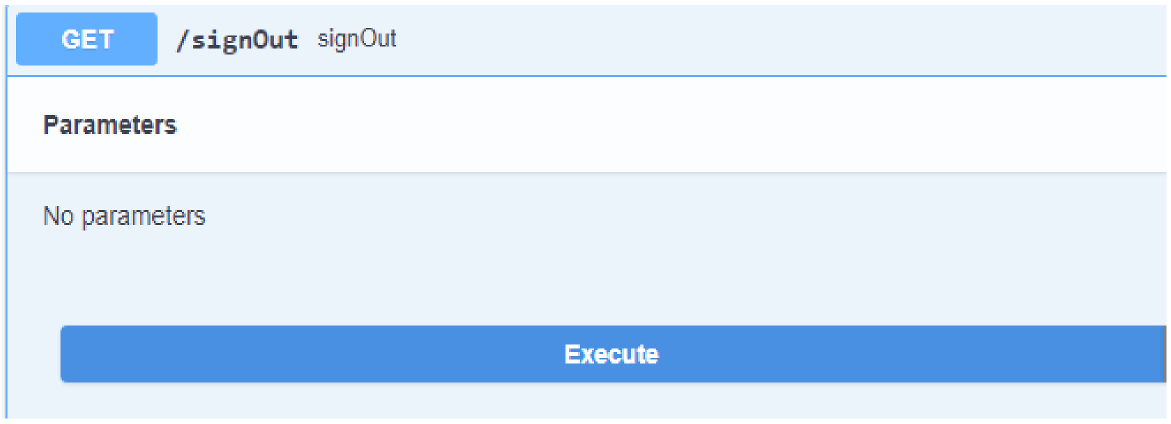
```
http://localhost:8080/signIn/tavishi23/shimla234
```

200

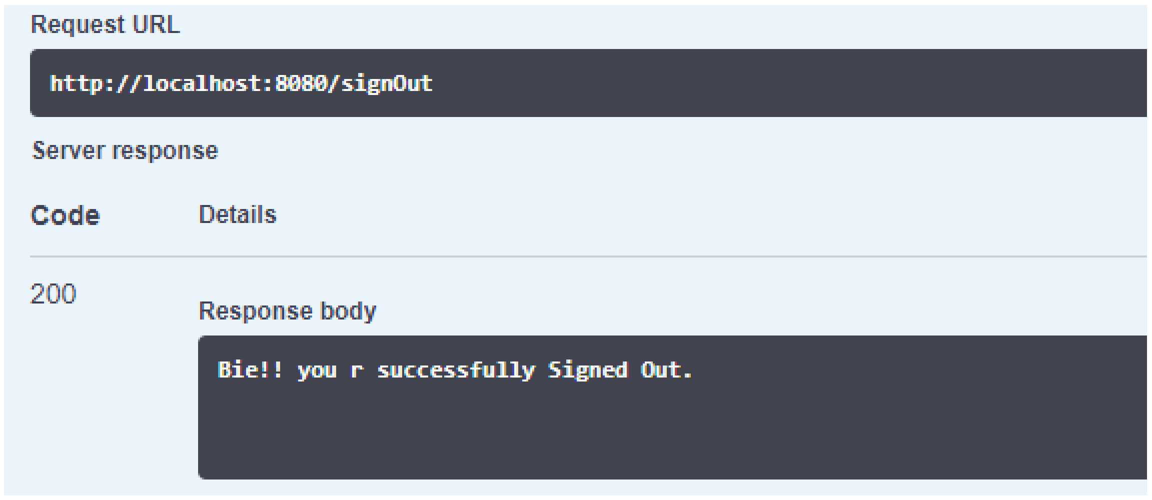
```
Tavishi Sharma. You are successfully signed In.
```

Picture 11- signIn user found in DB then output

If user select **/signOut** from swagger2 then it will reflect the string message bie, signOut successfully.



Picture 12- signOut function of swagger2



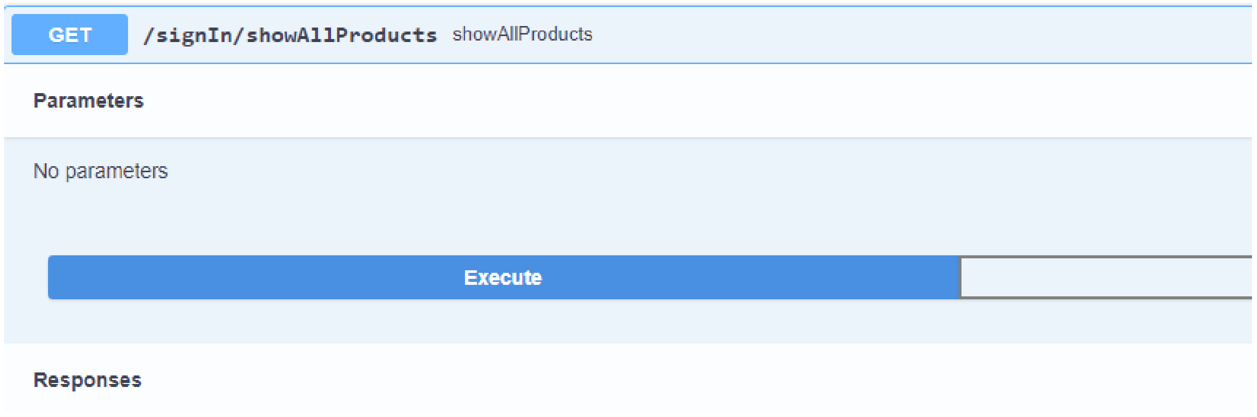
```
http://localhost:8080/signOut
```

200

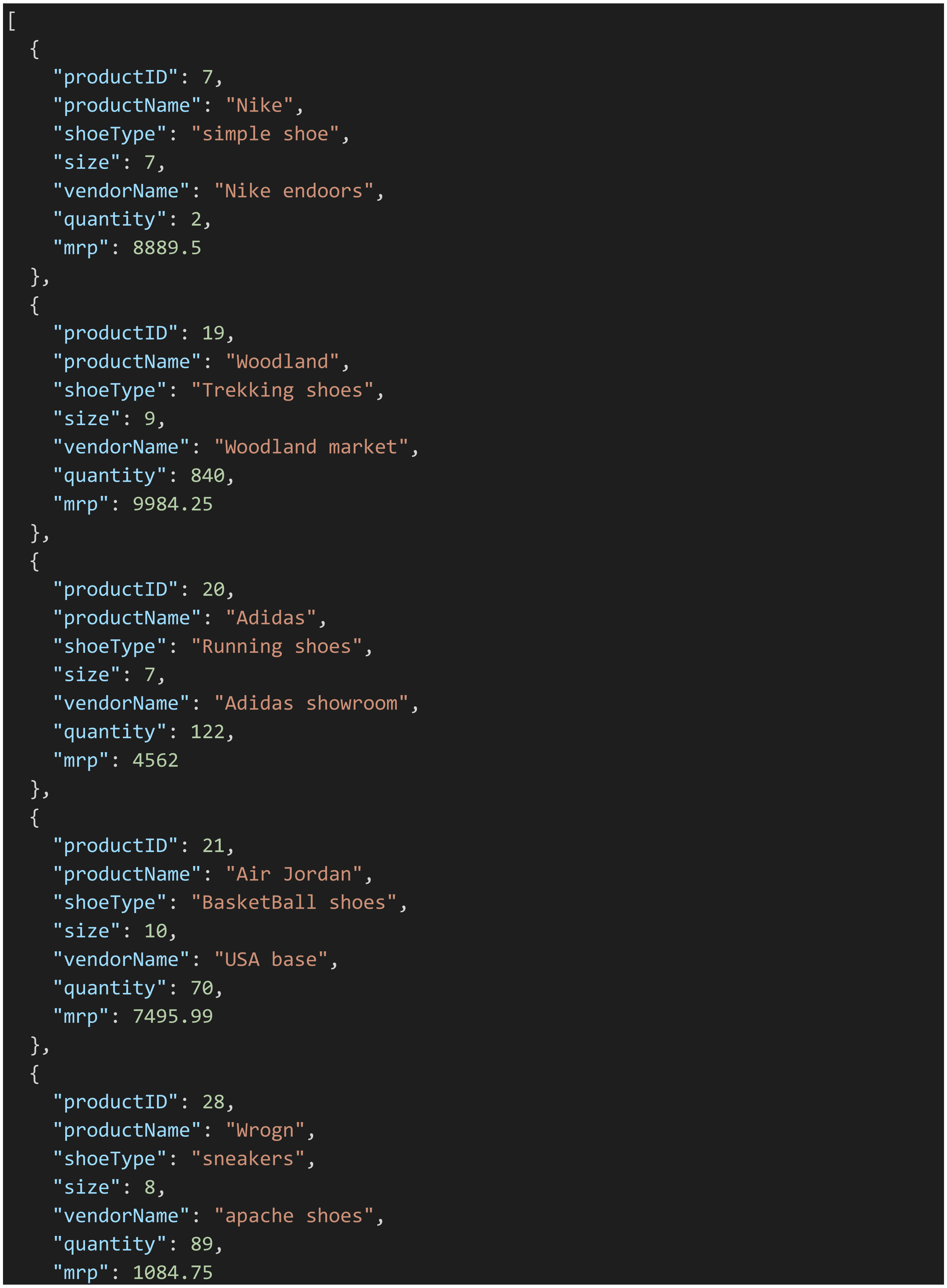
```
Bie!! you r successfully Signed Out.
```

Picture 13-signOut result

If user select show all Products option of swagger 2 i.e/signIn/showAllProducts



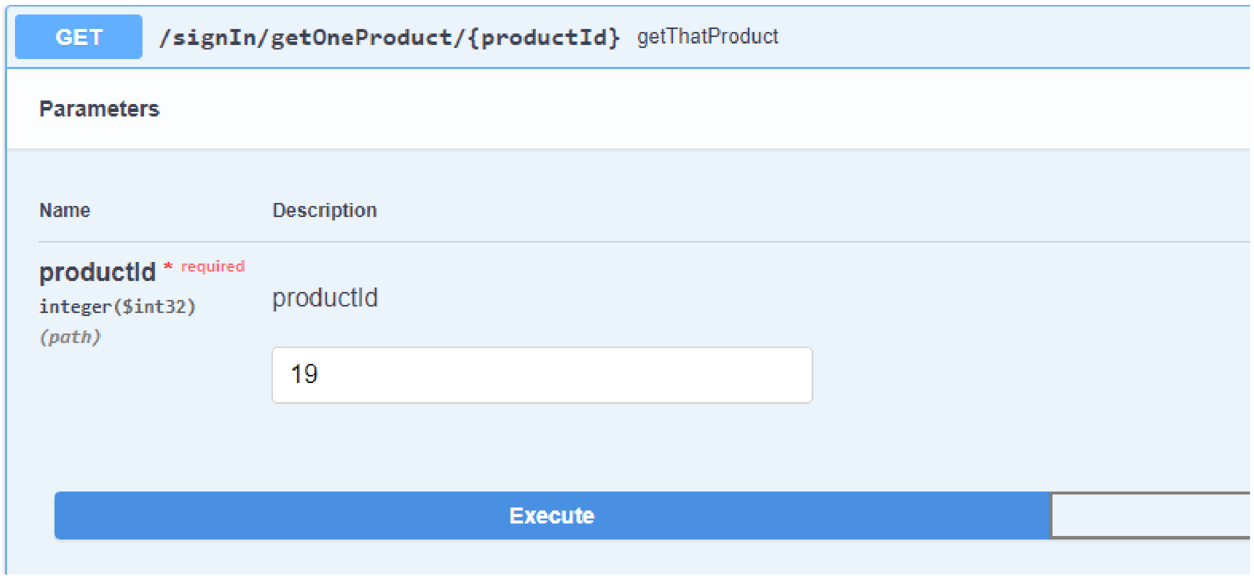
Picture 14-on selecting show all products of sporty shoes



```
[
  {
    "productID": 7,
    "productName": "Nike",
    "shoeType": "simple shoe",
    "size": 7,
    "vendorName": "Nike endoors",
    "quantity": 2,
    "mrp": 8889.5
  },
  {
    "productID": 19,
    "productName": "Woodland",
    "shoeType": "Trekking shoes",
    "size": 9,
    "vendorName": "Woodland market",
    "quantity": 840,
    "mrp": 9984.25
  },
  {
    "productID": 20,
    "productName": "Adidas",
    "shoeType": "Running shoes",
    "size": 7,
    "vendorName": "Adidas showroom",
    "quantity": 122,
    "mrp": 4562
  },
  {
    "productID": 21,
    "productName": "Air Jordan",
    "shoeType": "BasketBall shoes",
    "size": 10,
    "vendorName": "USA base",
    "quantity": 70,
    "mrp": 7495.99
  },
  {
    "productID": 28,
    "productName": "Wrogn",
    "shoeType": "sneakers",
    "size": 8,
    "vendorName": "apache shoes",
    "quantity": 89,
    "mrp": 1084.75
  }
]
```

Picture 15- output of /signIn/showAllProduts in json format

Suppose if user want to obtain particular product details by using productId.

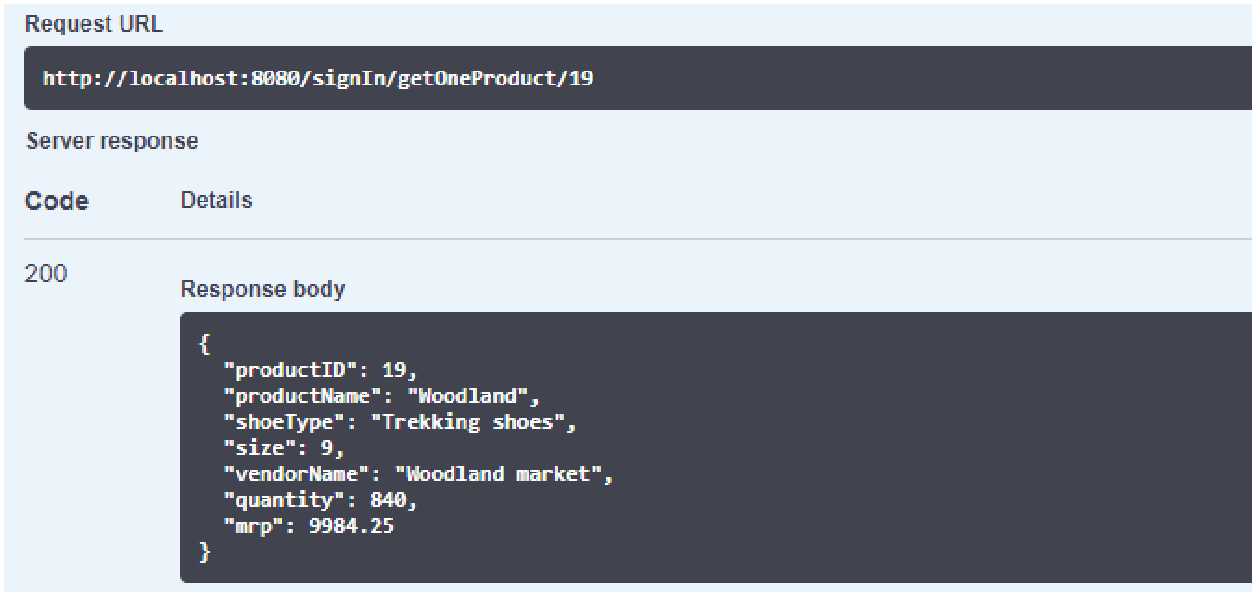


GET /signIn/getOneProduct/{productId} getThatProduct

Parameters

Name	Description
productId * required integer($int32) (path)	productId 19

Picture 16-entering product Id for obtain particular product details



Request URL

```
http://localhost:8080/signIn/getOneProduct/19
```

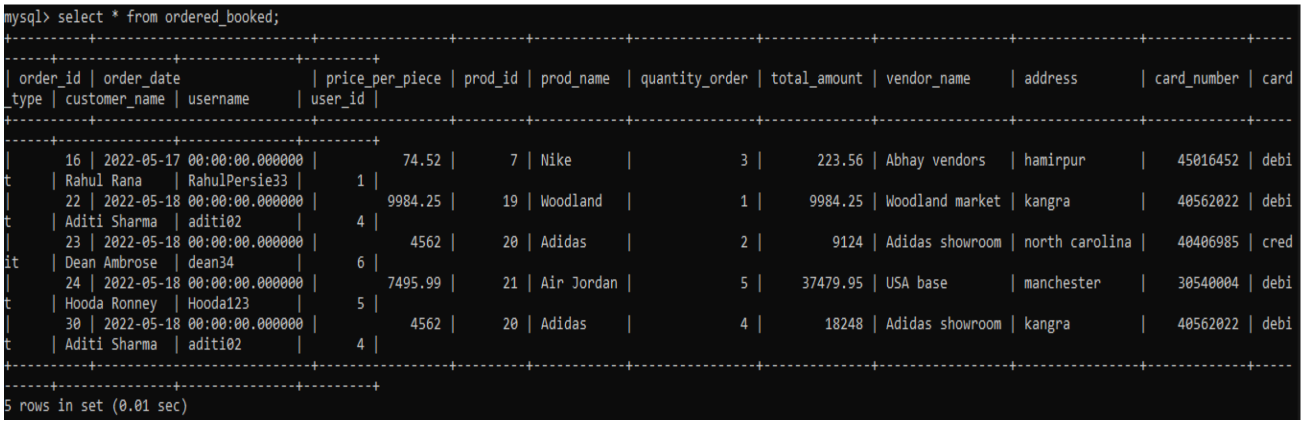
Server response

Code	Details
200	Response body

```
{
  "productID": 19,
  "productName": "Woodland",
  "shoeType": "Trekking shoes",
  "size": 9,
  "vendorName": "Woodland market",
  "quantity": 840,
  "mrp": 9984.25
}
```

Picture 17- output of /signIn/getOneProduct/{productId}

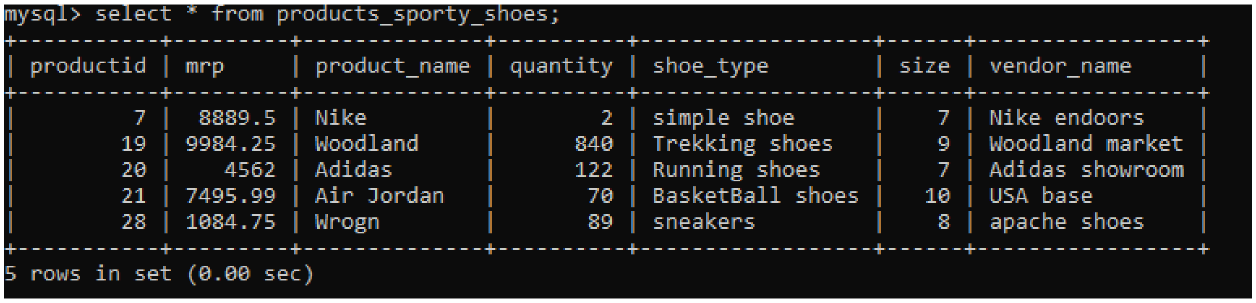
Suppose user want to purchase the product by taking productId in swagger then later on it will reflect changes in **Ordered_Booked** table.



```
mysql> select * from ordered_booked;
+----------+----------------------------+-----------------+---------+------------+----------------+--------------+-----------------+----------------+-------------+-----
------+---------------+---------------+---------+
| order_id | order_date                 | price_per_piece | prod_id | prod_name  | quantity_order | total_amount | vendor_name     | address        | card_number | card
_type | customer_name | username      | user_id |
+----------+----------------------------+-----------------+---------+------------+----------------+--------------+-----------------+----------------+-------------+-----
------+---------------+---------------+---------+
|       16 | 2022-05-17 00:00:00.000000 |           74.52 |       7 | Nike       |              3 |       223.56 | Abhay vendors   | hamirpur       |    45016452 | debi
t     | Rahul Rana    | RahulPersie33 |       1 |
|       22 | 2022-05-18 00:00:00.000000 |         9984.25 |      19 | Woodland   |              1 |      9984.25 | Woodland market | kangra         |    40562022 | debi
t     | Aditi Sharma  | aditi02       |       4 |
|       23 | 2022-05-18 00:00:00.000000 |            4562 |      20 | Adidas     |              2 |         9124 | Adidas showroom | north carolina |    40406985 | cred
it    | Dean Ambrose  | dean34        |       6 |
|       24 | 2022-05-18 00:00:00.000000 |         7495.99 |      21 | Air Jordan |              5 |     37479.95 | USA base        | manchester     |    30540004 | debi
t     | Hooda Ronney  | Hooda123      |       5 |
|       30 | 2022-05-18 00:00:00.000000 |            4562 |      20 | Adidas     |              4 |        18248 | Adidas showroom | kangra         |    40562022 | debi
t     | Aditi Sharma  | aditi02       |       4 |
+----------+----------------------------+-----------------+---------+------------+----------------+--------------+-----------------+----------------+-------------+-----
------+---------------+---------------+---------+
5 rows in set (0.01 sec)
```

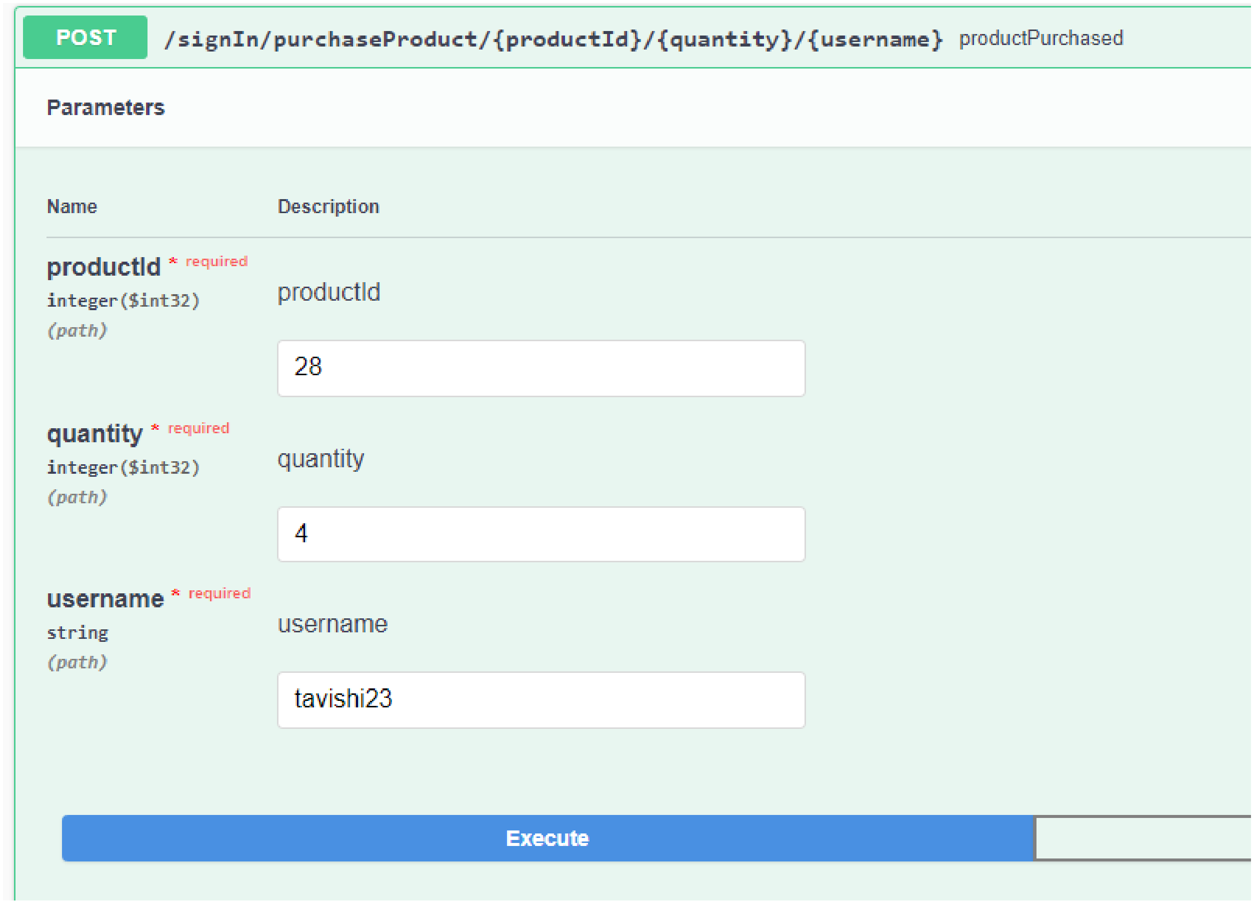
Picture 18- ordered_booked before purchase by username: tavishi23

Also **Product_sporty_shoes** table quantity changes after purchase.



```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       89 | sneakers        |    8 | apache shoes    |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
5 rows in set (0.00 sec)
```

Picture 19- Product_sporty_shoes quantity before purchase by user

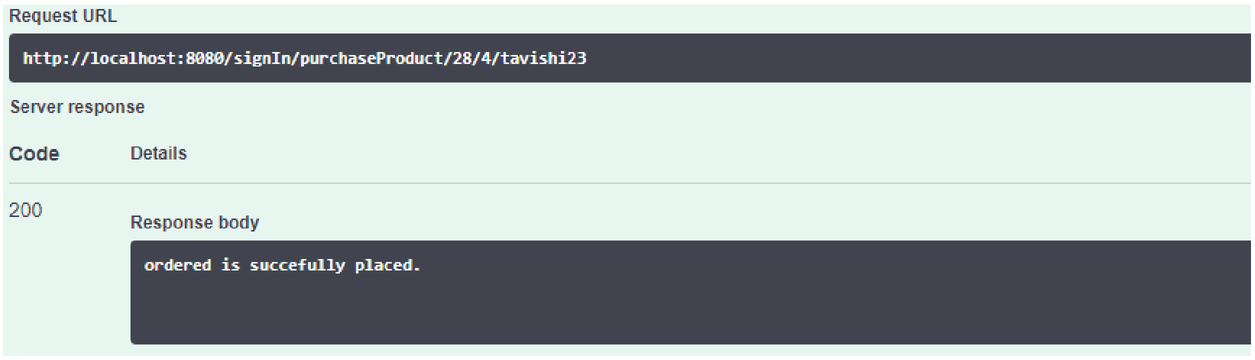

POST /signIn/purchaseProduct/{productId}/{quantity}/{username} productPurchased

Parameters

Name	Description
productId * required integer($int32) (path)	productId 28
quantity * required integer($int32) (path)	quantity 4
username * required string (path)	username tavishi23

Execute

Picture 20- when user:tavishi23 purchase product productId:28 quantity:4


Request URL

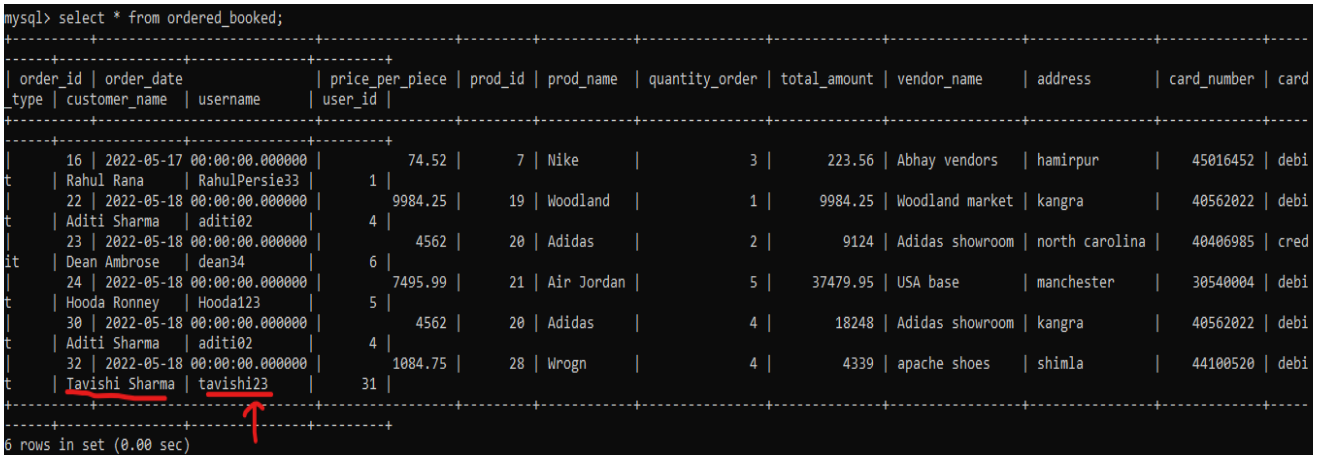
```
http://localhost:8080/signIn/purchaseProduct/28/4/tavishi23
```

Server response

Code	Details
200	Response body

```
ordered is succefully placed.
```

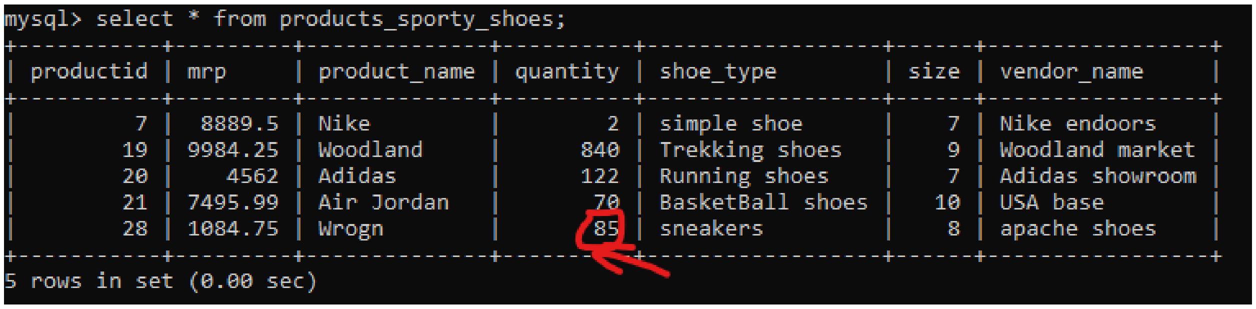
Picture 21- output of /signIn/purchaseProduct/{productId}/{quantity}/{username}



```
mysql> select * from ordered_booked;
+----------+----------------------------+-----------------+---------+------------+----------------+--------------+-----------------+------------------+-------------+-----
-----+---------------+---------------+---------+
| order_id | order_date                 | price_per_piece | prod_id | prod_name  | quantity_order | total_amount | vendor_name     | address          | card_number | card
_type | customer_name | username      | user_id |
+----------+----------------------------+-----------------+---------+------------+----------------+--------------+-----------------+------------------+-------------+-----
-----+---------------+---------------+---------+
|       16 | 2022-05-17 00:00:00.000000 |           74.52 |       7 | Nike       |              3 |       223.56 | Abhay vendors   | hamirpur         |    45016452 | debi
t     | Rahul Rana    | RahulPersie33 |       1 |
|       22 | 2022-05-18 00:00:00.000000 |         9984.25 |      19 | Woodland   |              1 |      9984.25 | Woodland market | kangra           |    40562022 | debi
t     | Aditi Sharma  | aditi02       |       4 |
|       23 | 2022-05-18 00:00:00.000000 |            4562 |      20 | Adidas     |              2 |         9124 | Adidas showroom | north carolina   |    40406985 | cred
it    | Dean Ambrose  | dean34        |       6 |
|       24 | 2022-05-18 00:00:00.000000 |         7495.99 |      21 | Air Jordan |              5 |     37479.95 | USA base        | manchester       |    30540004 | debi
t     | Hooda Ronney  | Hooda123      |       5 |
|       30 | 2022-05-18 00:00:00.000000 |            4562 |      20 | Adidas     |              4 |        18248 | Adidas showroom | kangra           |    40562022 | debi
t     | Aditi Sharma  | aditi02       |       4 |
|       32 | 2022-05-18 00:00:00.000000 |         1084.75 |      28 | Wrogn      |              4 |         4339 | apache shoes    | shimla           |    44100520 | debi
t     | Tavishi Sharma | tavishi23    |      31 |
+----------+----------------------------+-----------------+---------+------------+----------------+--------------+-----------------+------------------+-------------+-----
-----+---------------+---------------+---------+
6 rows in set (0.00 sec)
```

Picture 22- after purchase done Ordered_booked table changes

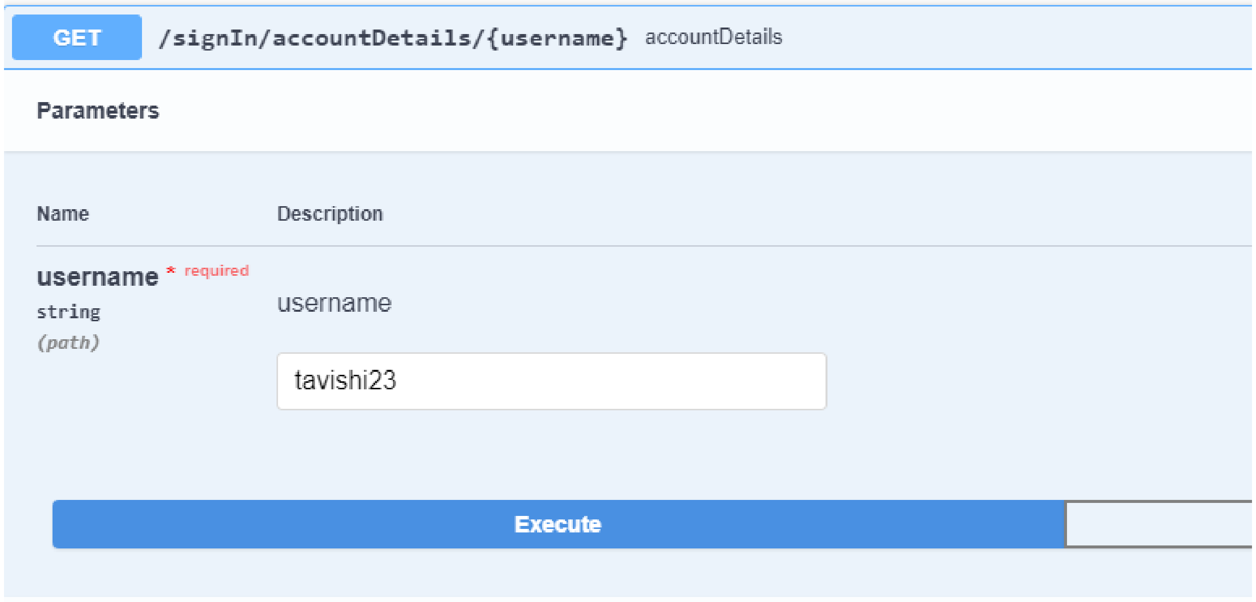
Here quantity of productId:28 reduces from 89 to 84. You can see in the picture.



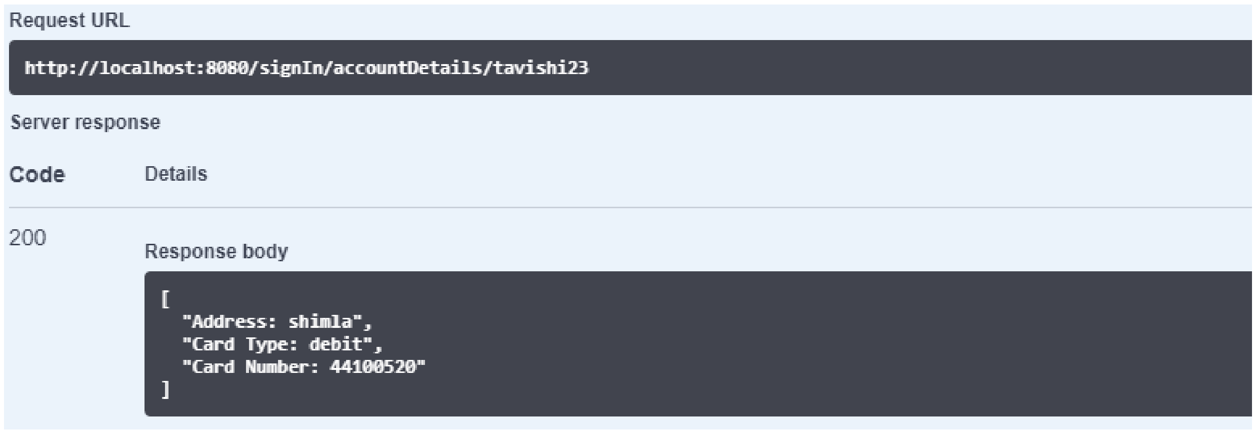
```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes |  10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers        |    8 | apache shoes    |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
5 rows in set (0.00 sec)
```

Picture 23- quantity of productId:28 reduces by 4 after purchase

If user selected display account details it will display the card number , card type and address of the user in browser using swagger2.



Picture 24- Account Details of user:tavishi23

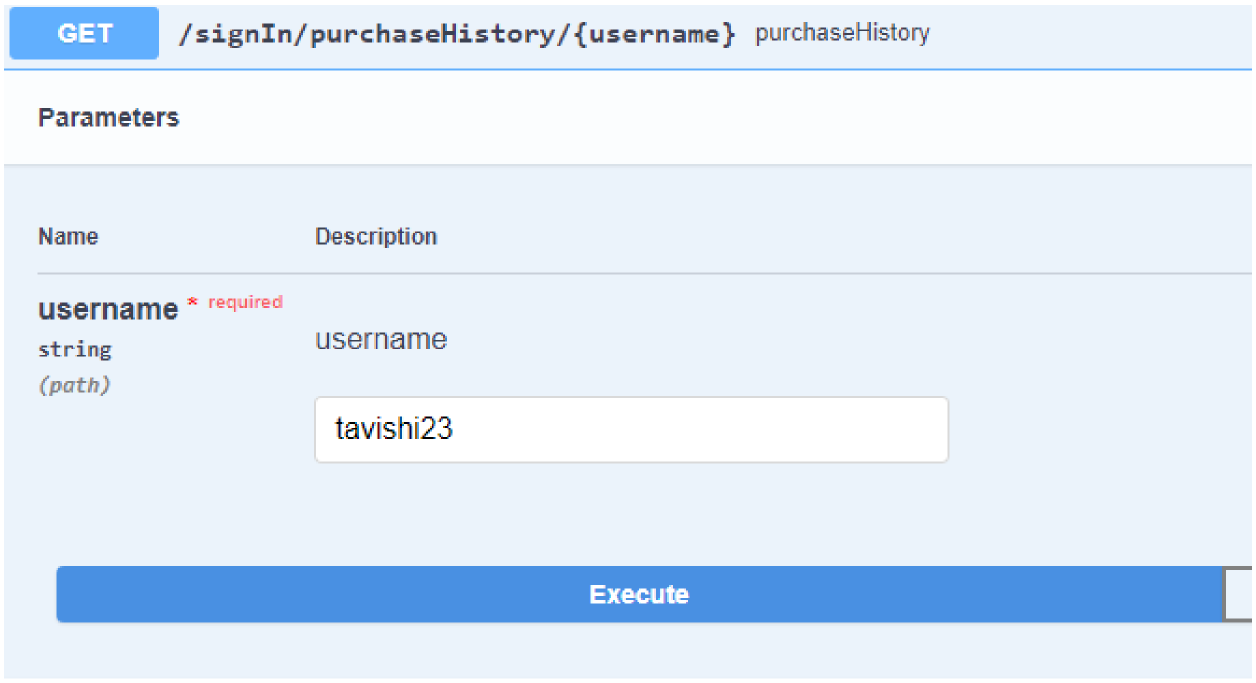


```
http://localhost:8080/signIn/accountDetails/tavishi23
```

```
[
  "Address: shimla",
  "Card Type: debit",
  "Card Number: 44100520"
]
```

Picture 25- output of /signIn/accountDetails/{username}

If user want to display history of purchase using username.

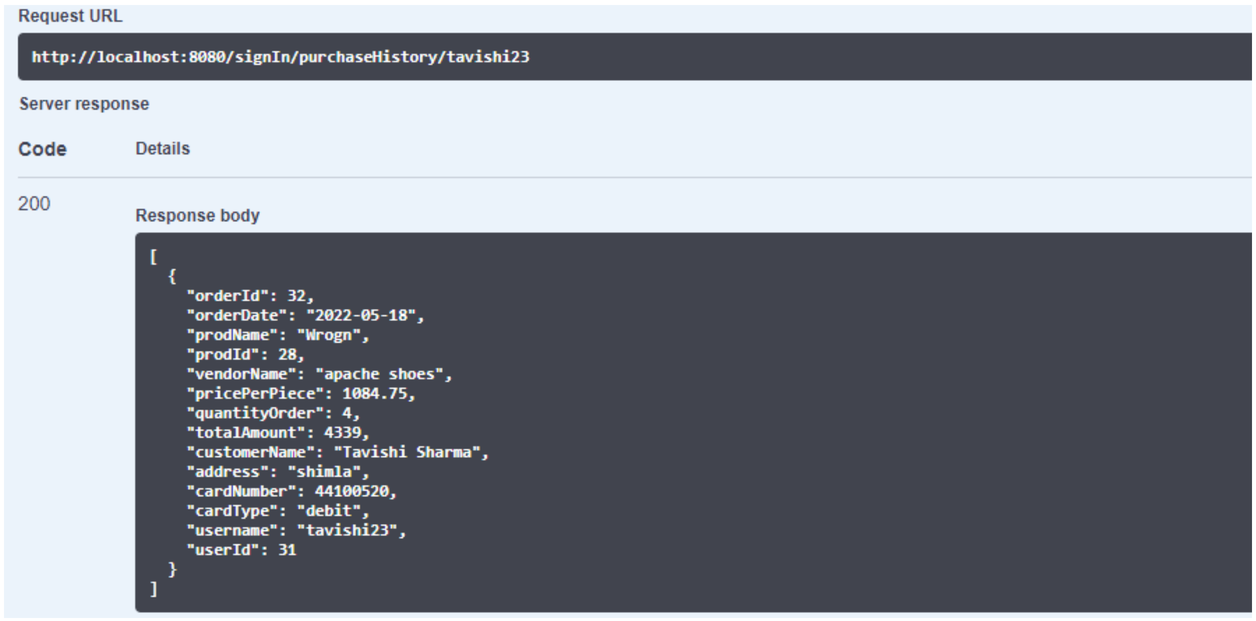

GET /signIn/purchaseHistory/{username} purchaseHistory

Parameters

Name	Description
username * required string (path)	username tavishi23

Execute

Picture 26- Exceute the /signIn/purchaseHistory/{username}


Request URL

```
http://localhost:8080/signIn/purchaseHistory/tavishi23
```

Server response

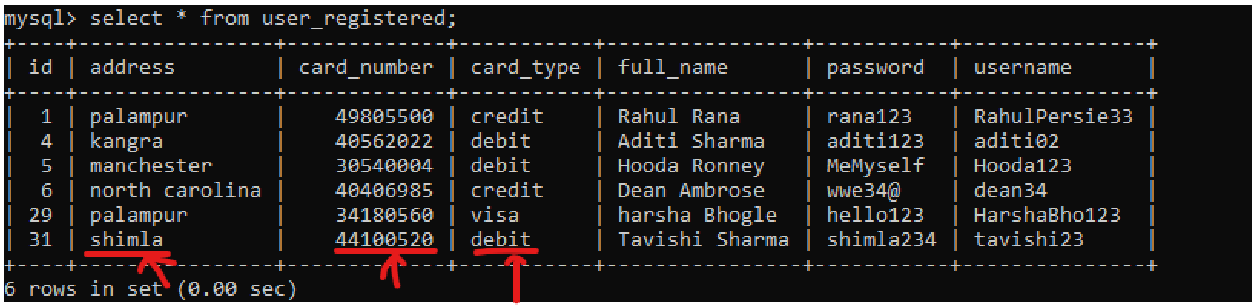
Code Details

200 Response body

```
[
  {
    "orderId": 32,
    "orderDate": "2022-05-18",
    "prodName": "Wrogn",
    "prodId": 28,
    "vendorName": "apache shoes",
    "pricePerPiece": 1084.75,
    "quantityOrder": 4,
    "totalAmount": 4339,
    "customerName": "Tavishi Sharma",
    "address": "shimla",
    "cardNumber": 44100520,
    "cardType": "debit",
    "username": "tavishi23",
    "userId": 31
  }
]
```

Picture 27- output of /signIn/purchaseHistory/tavishi23 reflect all purchase done by tavishi23 user

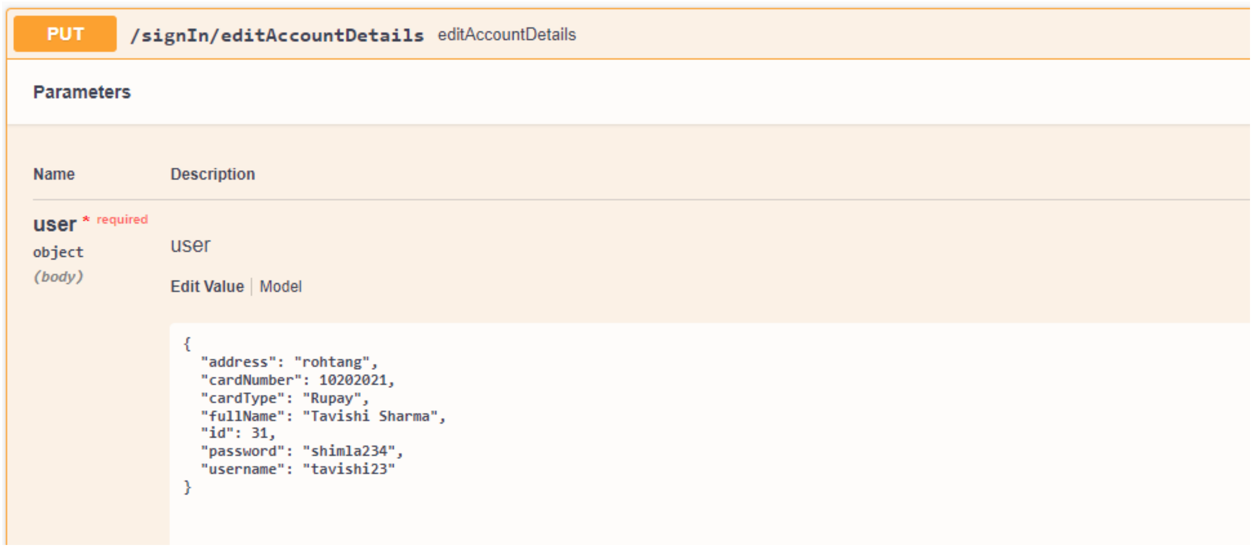
Suppose if user want to edit his/her account details, he/she can hover over the swagger2 function **/signIn/editAccountDetails.**



```
mysql> select * from user_registered;
+----+----------------+-------------+-----------+----------------+-----------+----------------+
| id | address        | card_number | card_type | full_name      | password  | username       |
+----+----------------+-------------+-----------+----------------+-----------+----------------+
|  1 | palampur       |    49805500 | credit    | Rahul Rana     | rana123   | RahulPersie33  |
|  4 | kangra         |    40562022 | debit     | Aditi Sharma   | aditi123  | aditi02        |
|  5 | manchester     |    30540004 | debit     | Hooda Ronney   | MeMyself  | Hooda123       |
|  6 | north carolina |    40406985 | credit    | Dean Ambrose   | wwe34@    | dean34         |
| 29 | palampur       |    34180560 | visa      | harsha Bhogle  | hello123  | HarshaBho123   |
| 31 | shimla         |    44100520 | debit     | Tavishi Sharma | shimla234 | tavishi23      |
+----+----------------+-------------+-----------+----------------+-----------+----------------+
6 rows in set (0.00 sec)
```

Picture 28- table user_registered before update account details

The address , crad number and card type will be change after the executing the account details for user:tavishi23.



PUT /signIn/editAccountDetails editAccountDetails

Parameters

Name Description

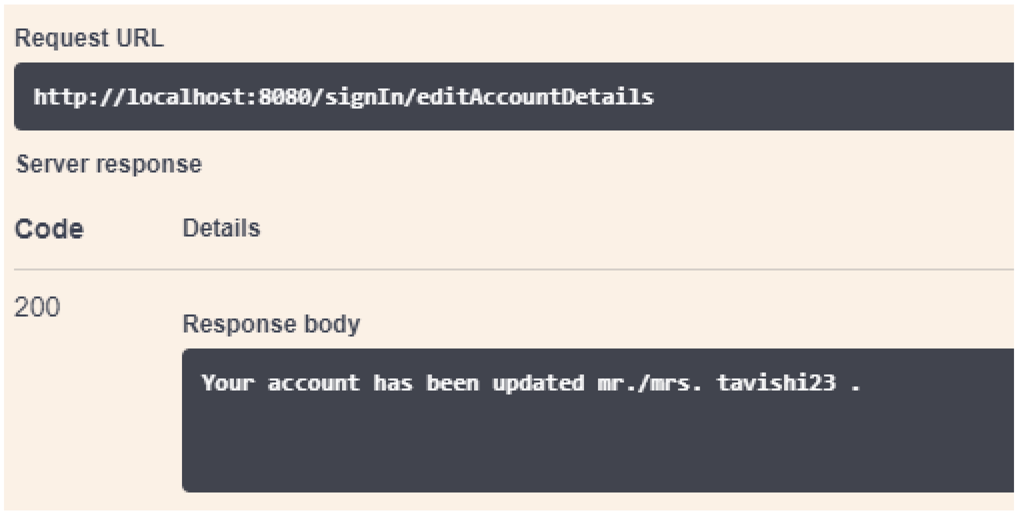
user * required
object
(body)

user

Edit Value | Model

```
{
  "address": "rohtang",
  "cardNumber": 10202021,
  "cardType": "Rupay",
  "fullName": "Tavishi Sharma",
  "id": 31,
  "password": "shimla234",
  "username": "tavishi23"
}
```

Picture 29-Executing the /signIn/editaccountDetails



Request URL

```
http://localhost:8080/signIn/editAccountDetails
```

Server response

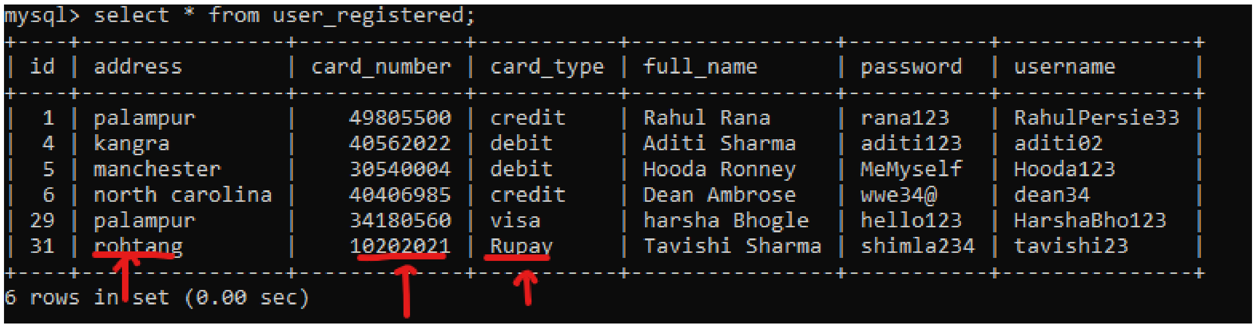
Code Details

200

Response body

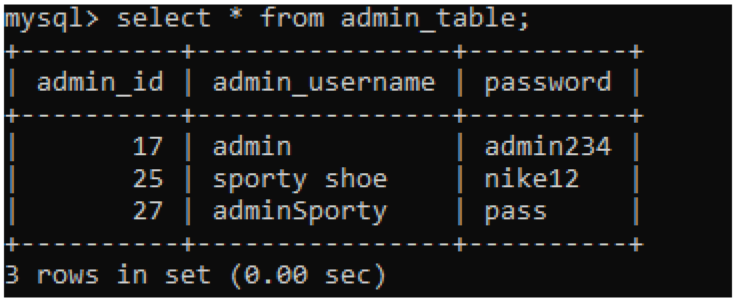
```
Your account has been updated mr./mrs. tavishi23 .
```

Picture 30- after update the details of user:tavishi23 it will reflect msg – successful updation


```
mysql> select * from user_registered;
+----+----------------+-------------+-----------+----------------+-----------+---------------+
| id | address        | card_number | card_type | full_name      | password  | username      |
+----+----------------+-------------+-----------+----------------+-----------+---------------+
|  1 | palampur       |    49805500 | credit    | Rahul Rana     | rana123   | RahulPersie33 |
|  4 | kangra         |    40562022 | debit     | Aditi Sharma   | aditi123  | aditi02       |
|  5 | manchester     |    30540004 | debit     | Hooda Ronney   | MeMyself  | Hooda123      |
|  6 | north carolina |    40406985 | credit    | Dean Ambrose   | wwe34@    | dean34        |
| 29 | palampur       |    34180560 | visa      | harsha Bhogle  | hello123  | HarshaBho123  |
| 31 | rohtang        |    10202021 | Rupay     | Tavishi Sharma | shimla234 | tavishi23     |
+----+----------------+-------------+-----------+----------------+-----------+---------------+
6 rows in set (0.00 sec)
```

Picture 31- user_registered table address, card number and card type for tavishi23 change

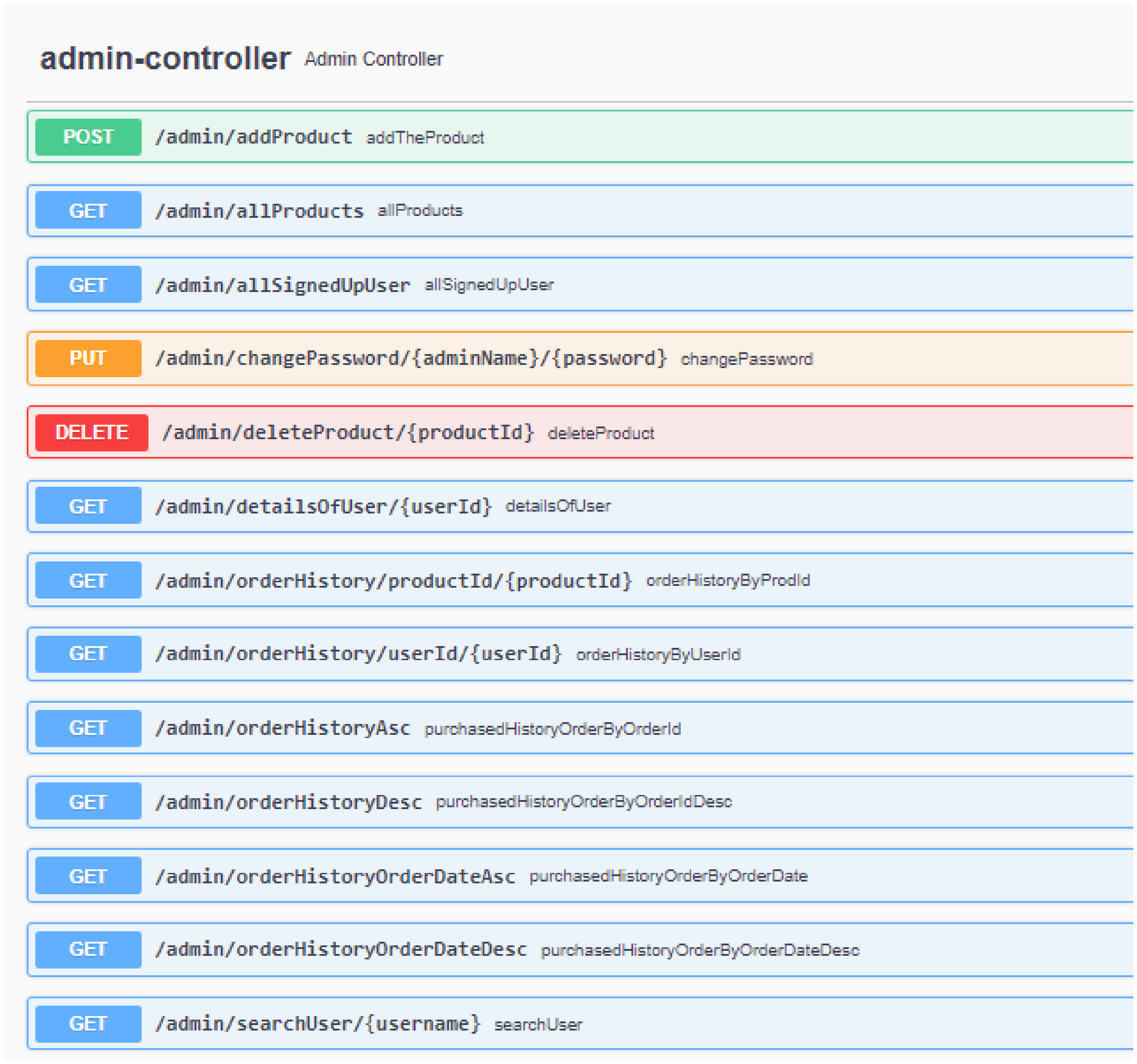
Now if user is admin and have username and password belonging to **admin_table.** He/she can use admin function from adminController.


```
mysql> select * from admin_table;
+----------+----------------+----------+
| admin_id | admin_username | password |
+----------+----------------+----------+
|       17 | admin          | admin234 |
|       25 | sporty shoe    | nike12   |
|       27 | adminSporty    | pass     |
+----------+----------------+----------+
3 rows in set (0.00 sec)
```

Picture 32- admin_table all admin names and their password

If admin name and password is from these 3 the user can be admin either cannot use admin functions.

All admin functions shown by swagger2 in the browser.


admin-controller Admin Controller

POST /admin/addProduct addTheProduct

GET /admin/allProducts allProducts

GET /admin/allSignedUpUser allSignedUpUser

PUT /admin/changePassword/{adminName}/{password} changePassword

DELETE /admin/deleteProduct/{productId} deleteProduct

GET /admin/detailsOfUser/{userId} detailsOfUser

GET /admin/orderHistory/productId/{productId} orderHistoryByProdId

GET /admin/orderHistory/userId/{userId} orderHistoryByUserId

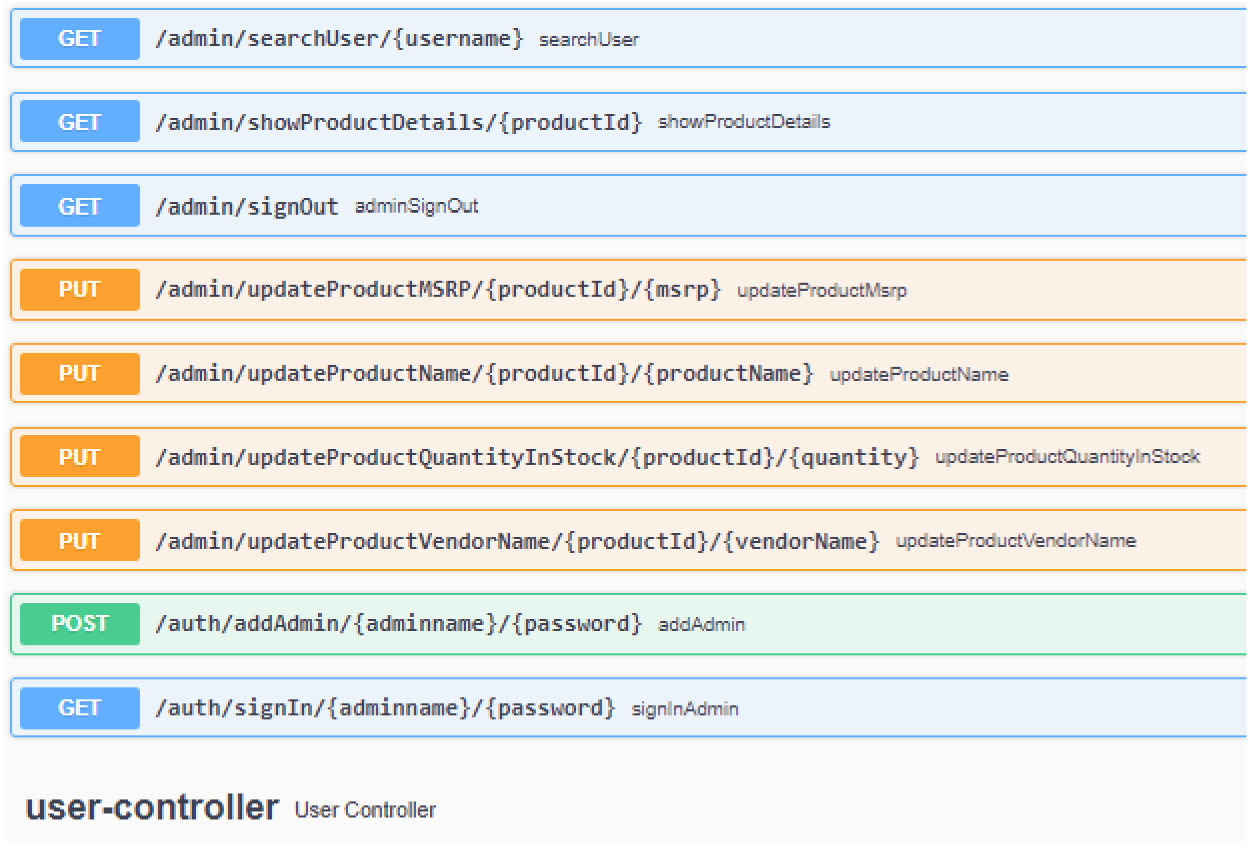
GET /admin/orderHistoryAsc purchasedHistoryOrderByOrderId

GET /admin/orderHistoryDesc purchasedHistoryOrderByOrderIdDesc

GET /admin/orderHistoryOrderDateAsc purchasedHistoryOrderByOrderDate

GET /admin/orderHistoryOrderDateDesc purchasedHistoryOrderByOrderDateDesc

GET /admin/searchUser/{username} searchUser



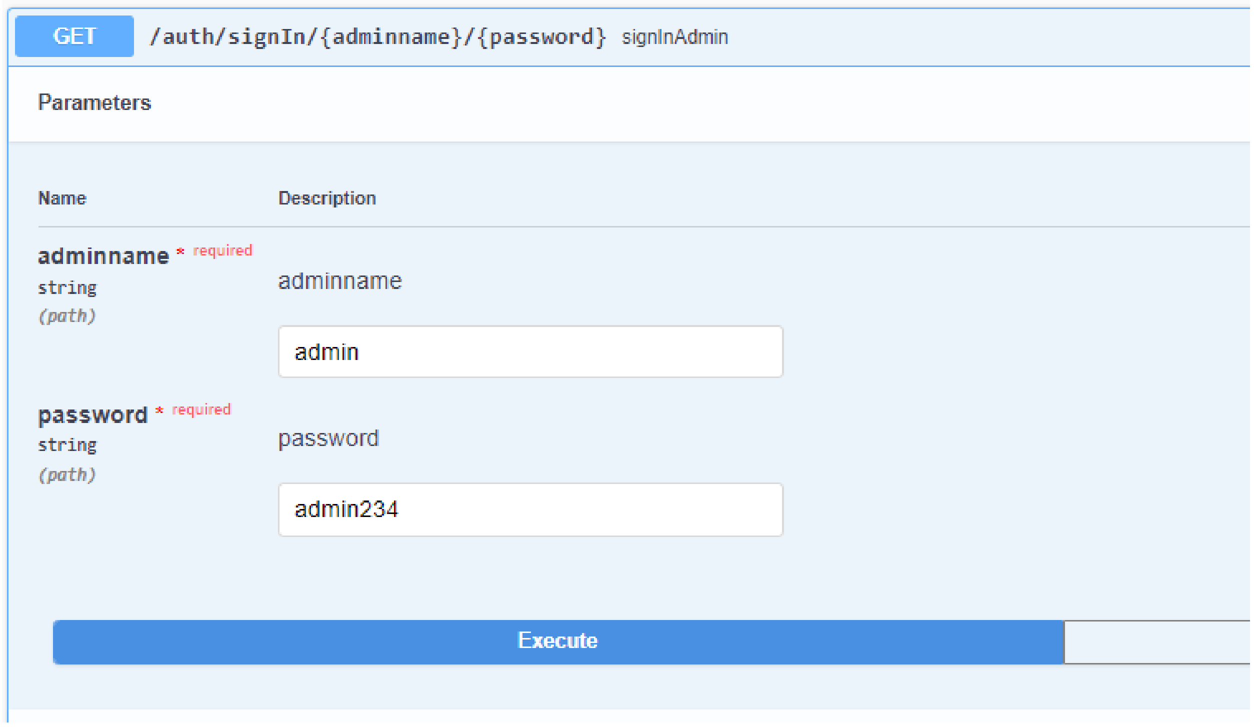
```
GET    /admin/searchUser/{username}  searchUser
GET    /admin/showProductDetails/{productId}  showProductDetails
GET    /admin/signOut  adminSignOut
PUT    /admin/updateProductMSRP/{productId}/{msrp}  updateProductMsrp
PUT    /admin/updateProductName/{productId}/{productName}  updateProductName
PUT    /admin/updateProductQuantityInStock/{productId}/{quantity}  updateProductQuantityInStock
PUT    /admin/updateProductVendorName/{productId}/{vendorName}  updateProductVendorName
POST   /auth/addAdmin/{adminname}/{password}  addAdmin
GET    /auth/signIn/{adminname}/{password}  signInAdmin
```

user-controller User Controller

Picture 33- swagger2 adminController all options

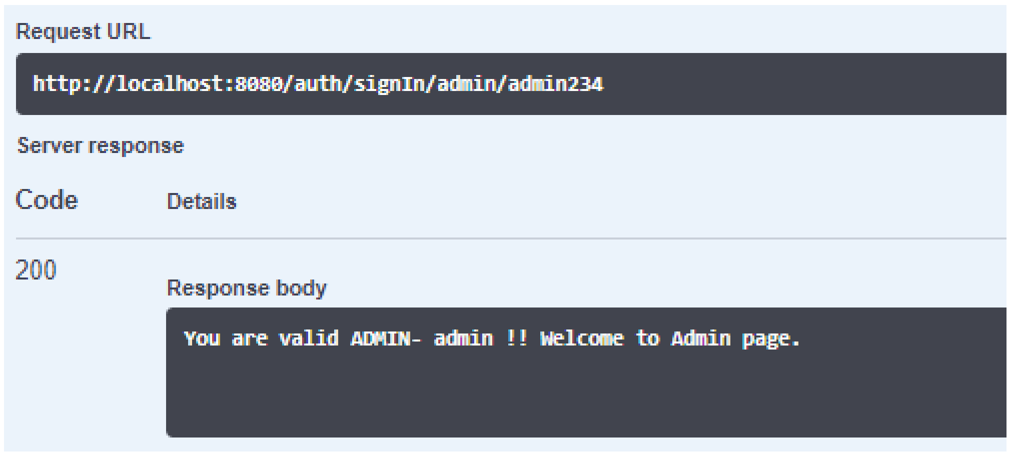
Suppose if user is admin then he/she will select **/auth/signIn/{adminname}/{password}** option from swagger 2 options.

And then the function will match the username and password from the database if valid then return message you are valid admin.



Picture 34- On executing the /auth/signIn/{adminname}/{password}

Since adminname:**admin** password:**admin234** is present in **admin_table,** it will reflect message, welcome admin.



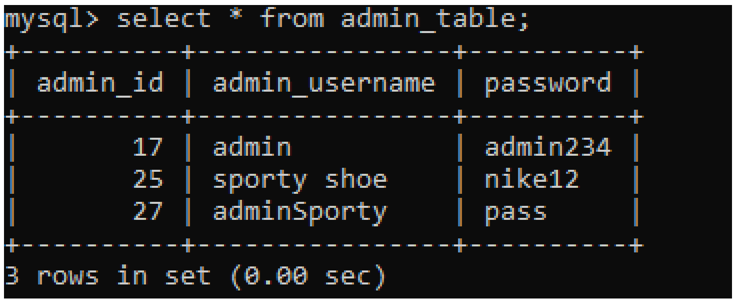
```
http://localhost:8080/auth/signIn/admin/admin234
```

```
You are valid ADMIN- admin !! Welcome to Admin page.
```

Picture 35- result of /auth/signIn/admin/admin234

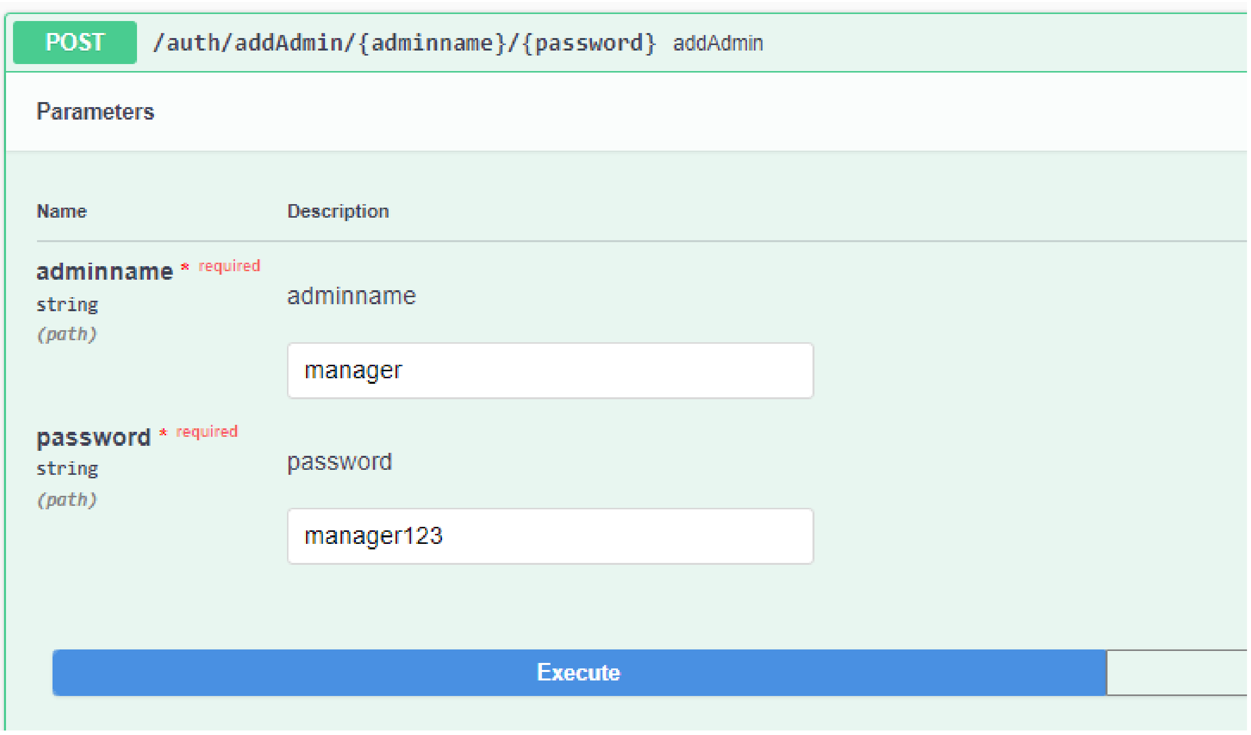
If admin want to add other admin into **admin_table,** then can use option **/auth/addAdmin/{username}/{password}** it will add the other user in the table **admin_table.**

Databse table **admin_table** before admin added to table.



```
mysql> select * from admin_table;
+----------+----------------+----------+
| admin_id | admin_username | password |
+----------+----------------+----------+
|       17 | admin          | admin234 |
|       25 | sporty shoe    | nike12   |
|       27 | adminSporty    | pass     |
+----------+----------------+----------+
3 rows in set (0.00 sec)
```

Picture 35- Before adding the admin in the admin_table



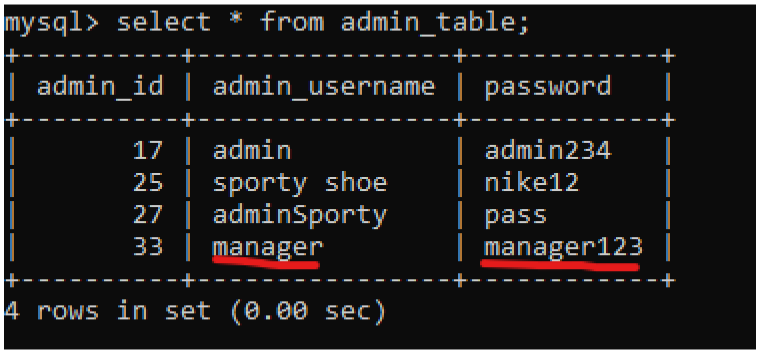
POST /auth/addAdmin/{adminname}/{password} addAdmin

Parameters

Name	Description
adminname * required string (path)	adminname: manager
password * required string (path)	password: manager123

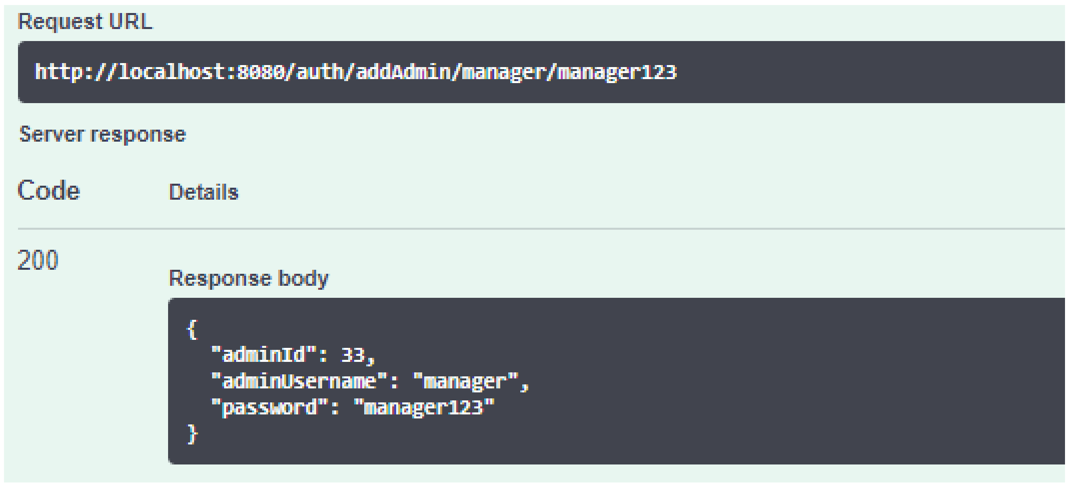
Picture 36- select addAdmin option in Swagger2

After adding admin:**manager** password:**manager123**, a new admin will be added in the **admin_table.**



```
mysql> select * from admin_table;
+----------+----------------+------------+
| admin_id | admin_username | password   |
+----------+----------------+------------+
|       17 | admin          | admin234   |
|       25 | sporty shoe    | nike12     |
|       27 | adminSporty    | pass       |
|       33 | manager        | manager123 |
+----------+----------------+------------+
4 rows in set (0.00 sec)
```

Picture 37- After adding the admin to admin_table

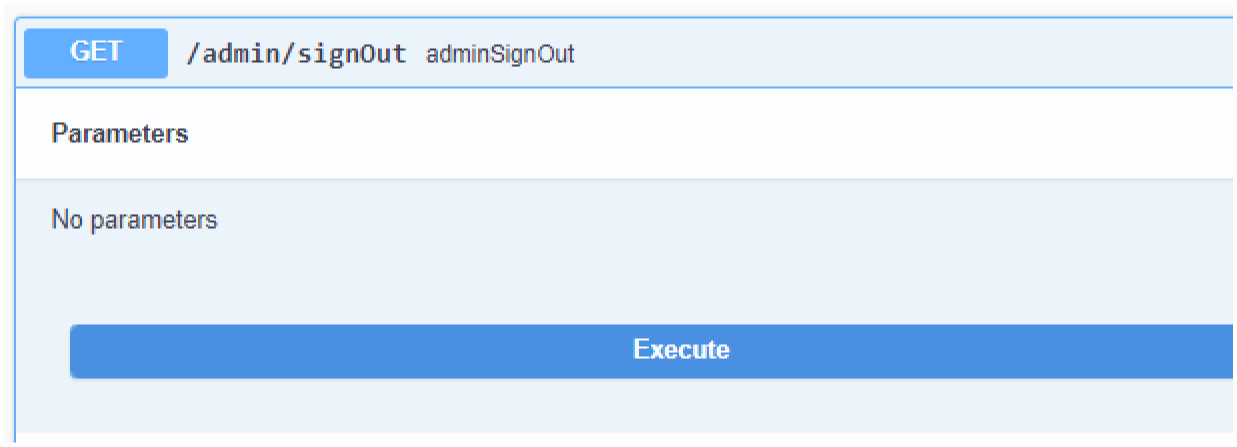


```
http://localhost:8080/auth/addAdmin/manager/manager123
```

```
{
  "adminId": 33,
  "adminUsername": "manager",
  "password": "manager123"
}
```

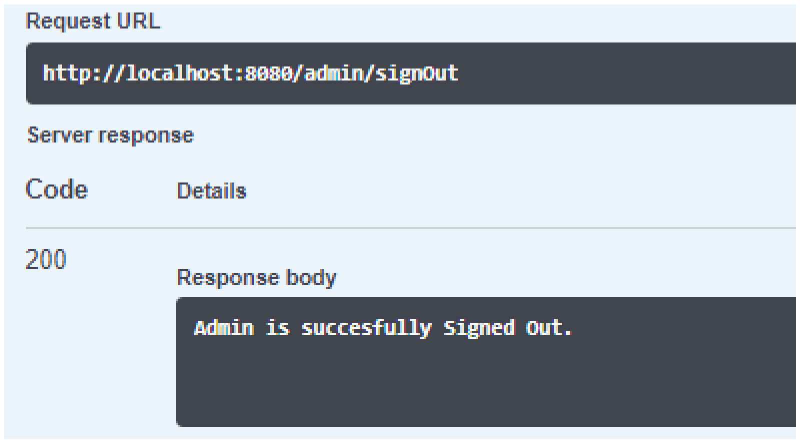
Picture 38- /auth/addAdmin/manager/manager123 i.e addAdmin result in swagger2

If admin want to signOut, he/she have to just select the signOut option in swagger2.



Picture 39- Execute /admin/signOut in the swagger2

After successfully admin signOut, it will reflect message of logout in the browser.

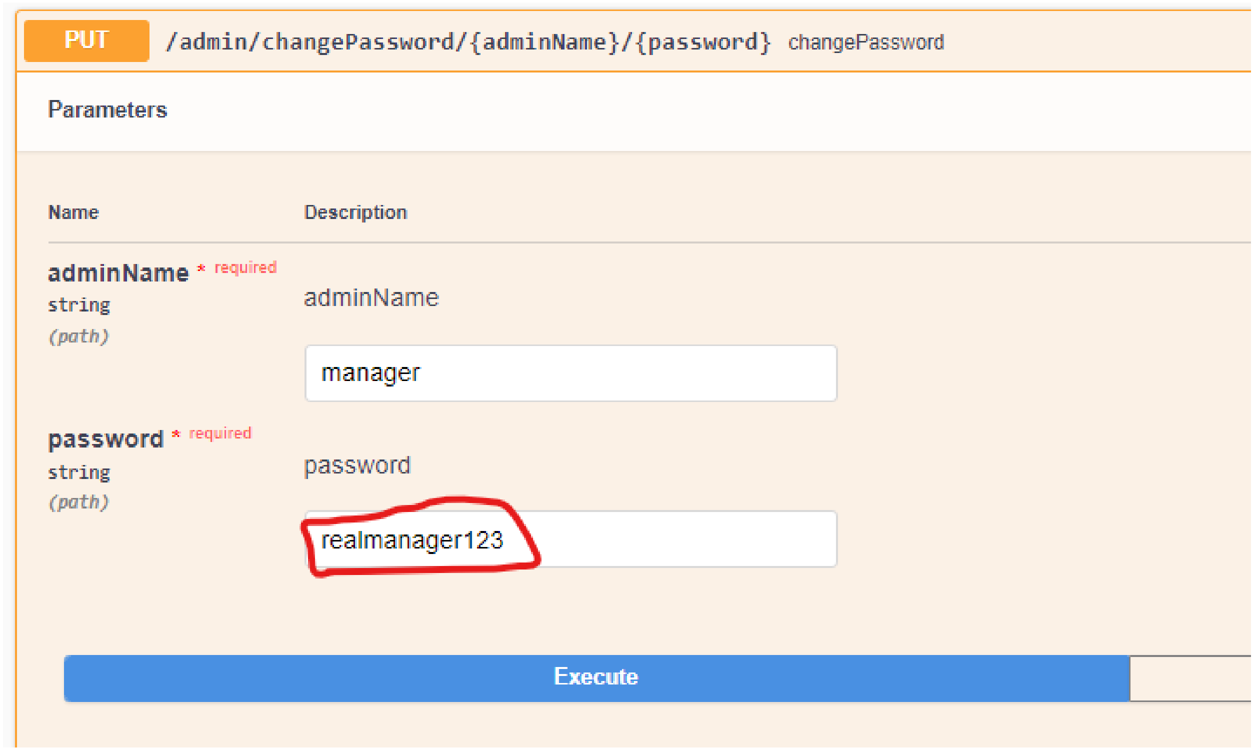


```
http://localhost:8080/admin/signOut
```

```
Admin is succesfully Signed Out.
```

Picture 40- Output of /admin/signOut

To change the password of admin, admin have option of change password in the swagger2.

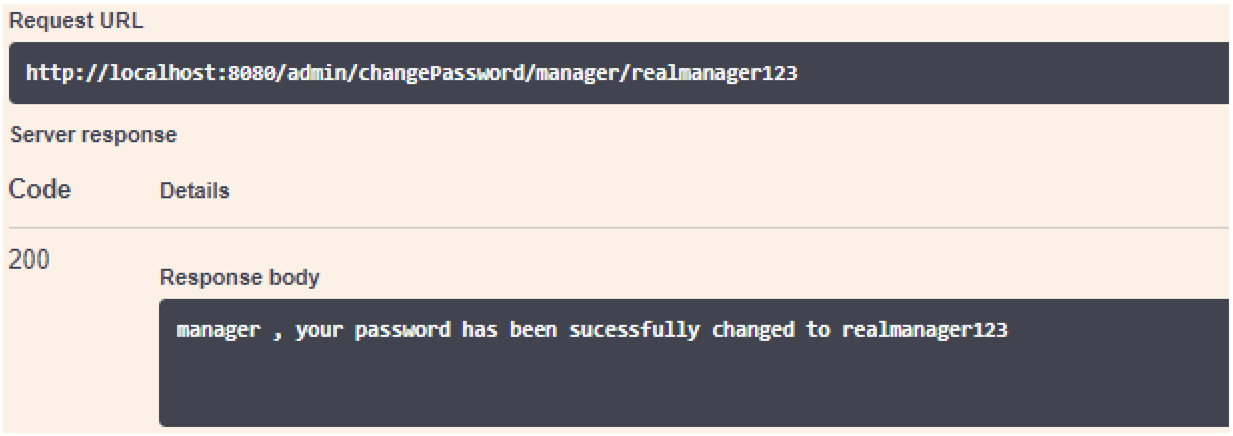


PUT /admin/changePassword/{adminName}/{password} changePassword

Name	Description
adminName * required string (path)	adminName: manager
password * required string (path)	password: realmanager123

Picture 41- to change password provide change password and adminname whos password you want to change

Here password of admin:manager will change to password:**realmanager123** and also changes reflect in the table **admin_table.**

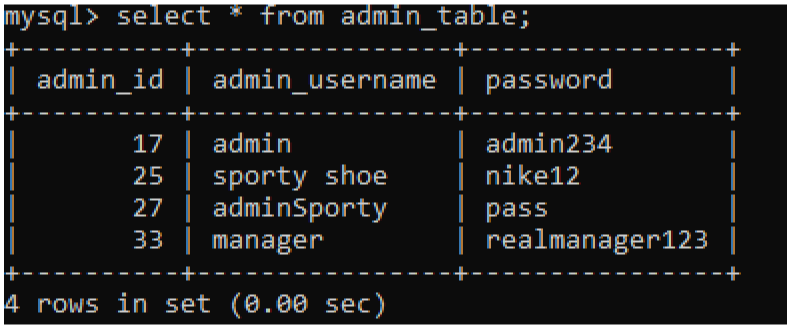


```
http://localhost:8080/admin/changePassword/manager/realmanager123
```

200

```
manager , your password has been sucessfully changed to realmanager123
```

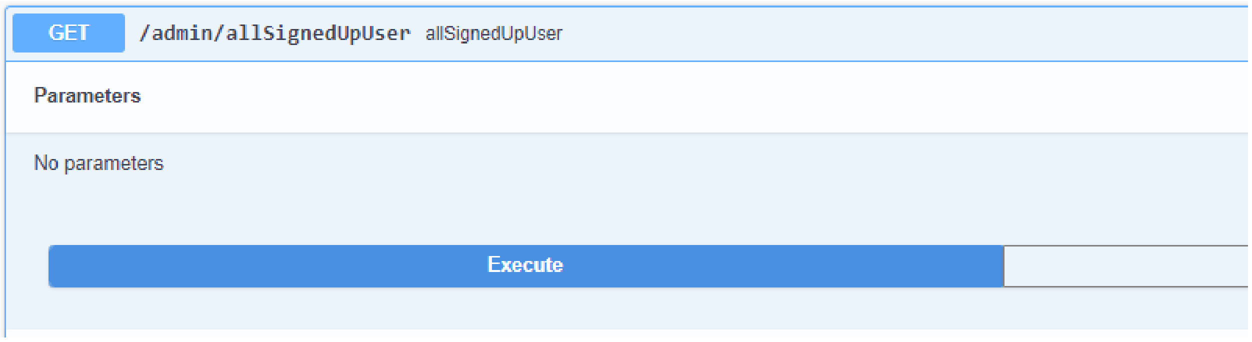
Picture 42-Output of /admin/changePassword/manager/realmanager123



```
mysql> select * from admin_table;
+----------+----------------+-----------------+
| admin_id | admin_username | password        |
+----------+----------------+-----------------+
|       17 | admin          | admin234        |
|       25 | sporty shoe    | nike12          |
|       27 | adminSporty    | pass            |
|       33 | manager        | realmanager123  |
+----------+----------------+-----------------+
4 rows in set (0.00 sec)
```

Picture 43- change of password reflect in table admin_table

If admin want to see, all the signed Up user in the site SportyShoes. The admin can see that by using the **admin/allSignedUpUser** in swagger2.

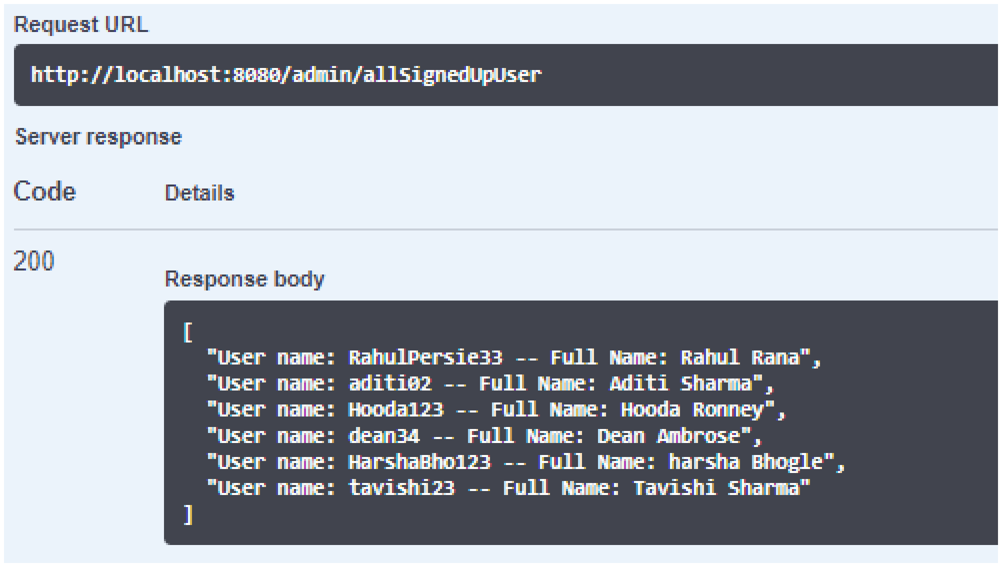


GET /admin/allSignedUpUser allSignedUpUser

Parameters

No parameters

Picture 44- Execute the /admin/allSignedUpUser



Request URL

```
http://localhost:8080/admin/allSignedUpUser
```

Server response

Code Details

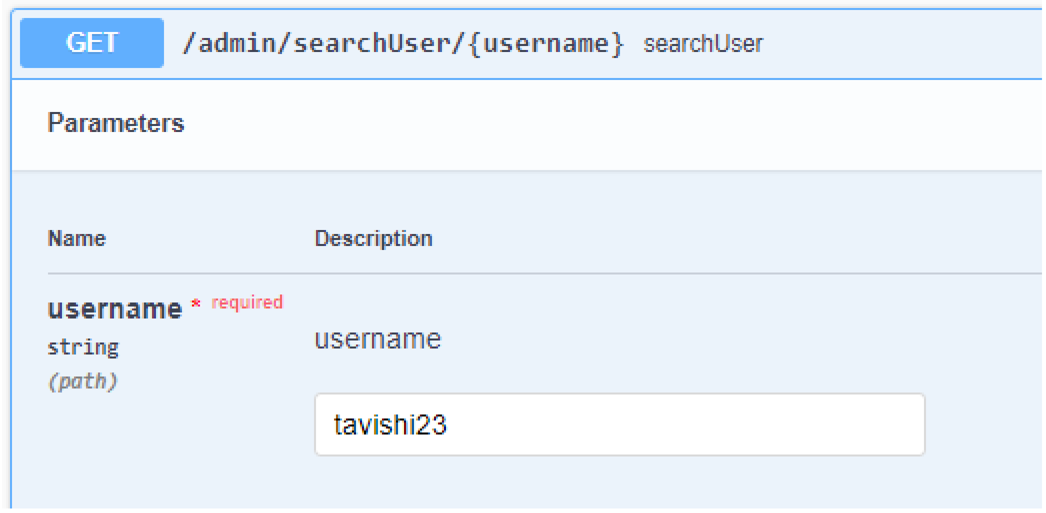
200

Response body

```
[
  "User name: RahulPersie33 -- Full Name: Rahul Rana",
  "User name: aditi02 -- Full Name: Aditi Sharma",
  "User name: Hooda123 -- Full Name: Hooda Ronney",
  "User name: dean34 -- Full Name: Dean Ambrose",
  "User name: HarshaBho123 -- Full Name: harsha Bhogle",
  "User name: tavishi23 -- Full Name: Tavishi Sharma"
]
```

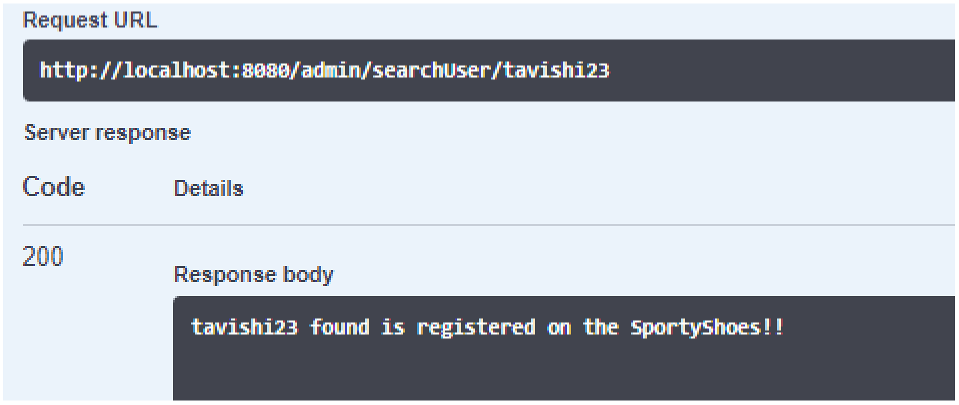
Picture 45- Output of /admin/allSignedIpUser

If admin want to search for a specific user is signed up, he can directly check that by entering username of user.



```
tavishi23
```

Picture 46- Execute /admin/searchUser/{username}

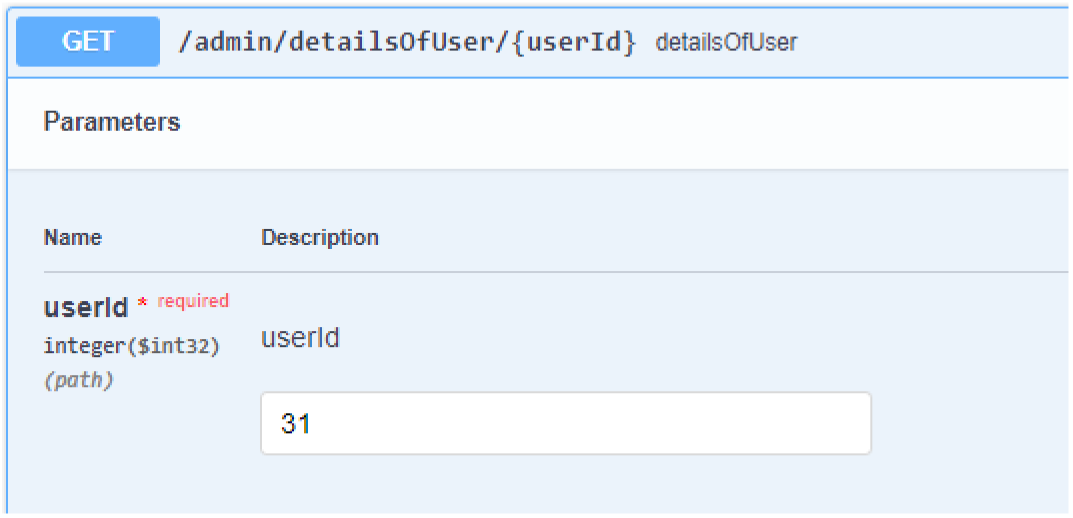


```
http://localhost:8080/admin/searchUser/tavishi23
```

```
tavishi23 found is registered on the SportyShoes!!
```

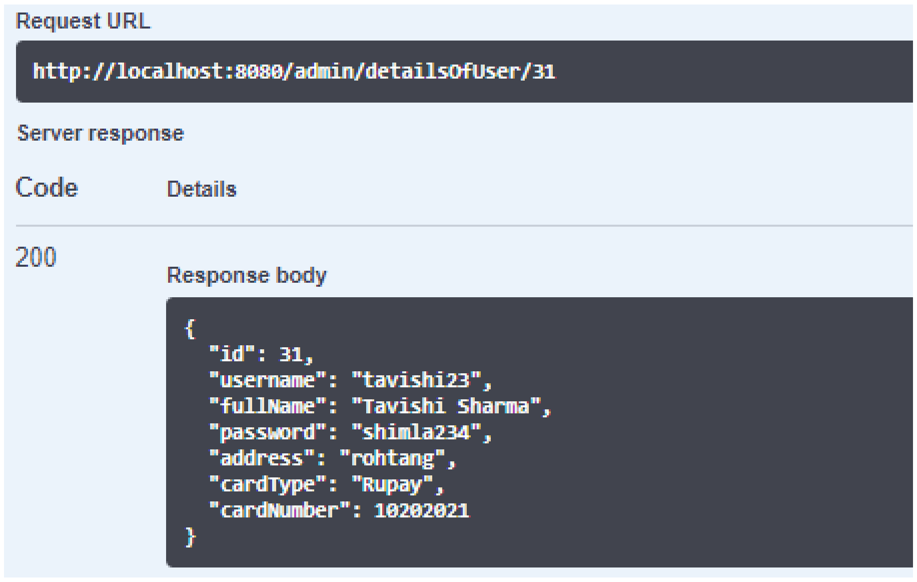
Picture 47- Output /admin/searchUser/tavishi23

If admin want to see details of particular user by giving userId.



```
31
```

Picture 48- Execute /admin/detailsOfUser/{userId}

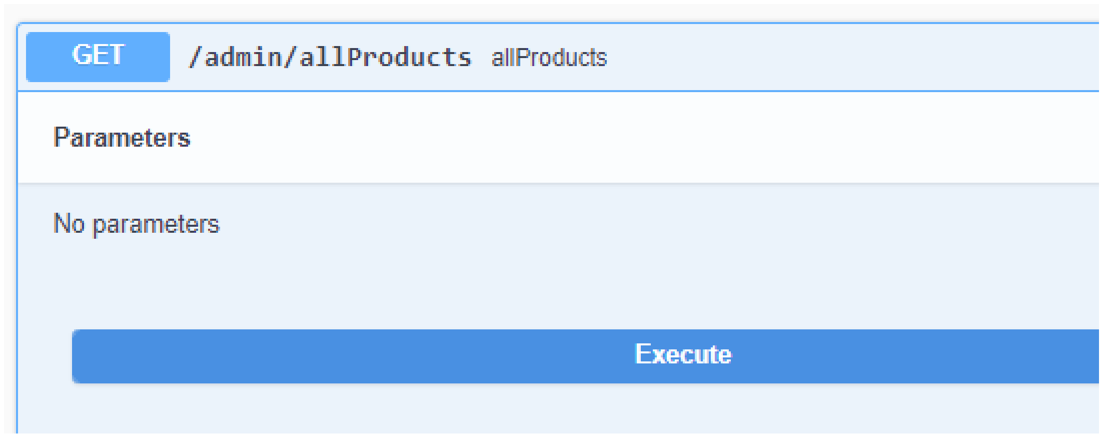


```
http://localhost:8080/admin/detailsOfUser/31
```

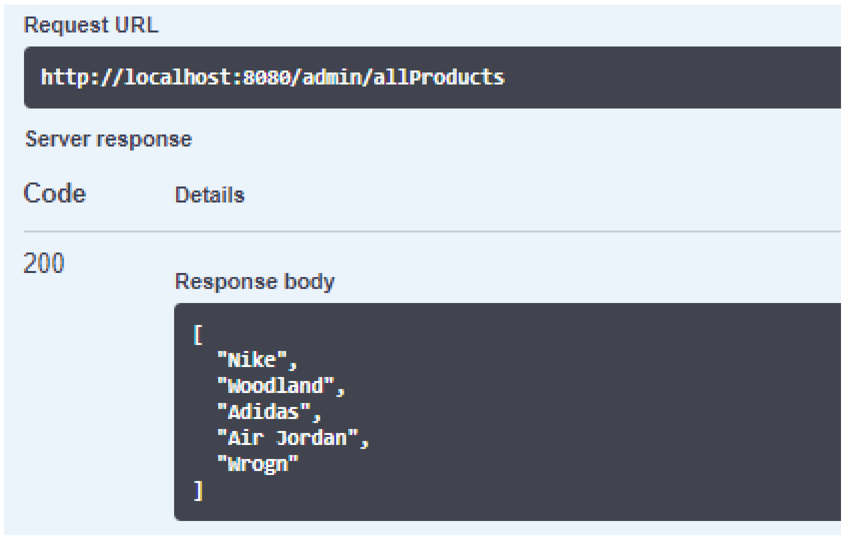
```
{
  "id": 31,
  "username": "tavishi23",
  "fullName": "Tavishi Sharma",
  "password": "shimla234",
  "address": "rohtang",
  "cardType": "Rupay",
  "cardNumber": 10202021
}
```

Picture 49- Output of /admin/detailsOfUser/31

If admin want to view all products names then admin can select **/admin/allProducts** in swagger2.



Picture 50 - Execute /admin/allProducts



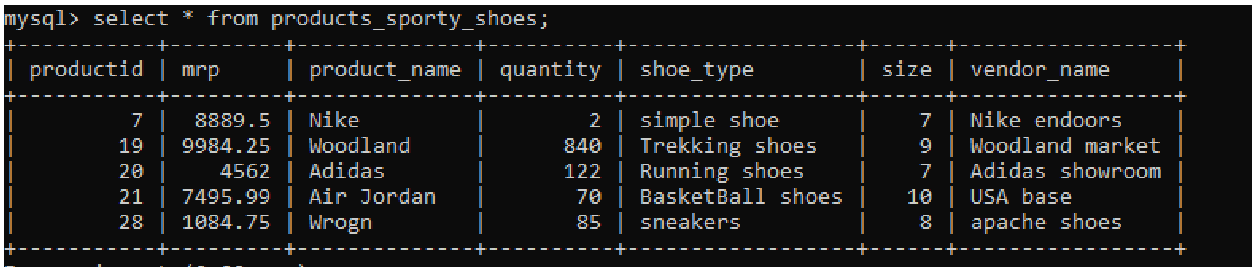
```
http://localhost:8080/admin/allProducts
```

200

```
[
  "Nike",
  "Woodland",
  "Adidas",
  "Air Jordan",
  "Wrogn"
]
```

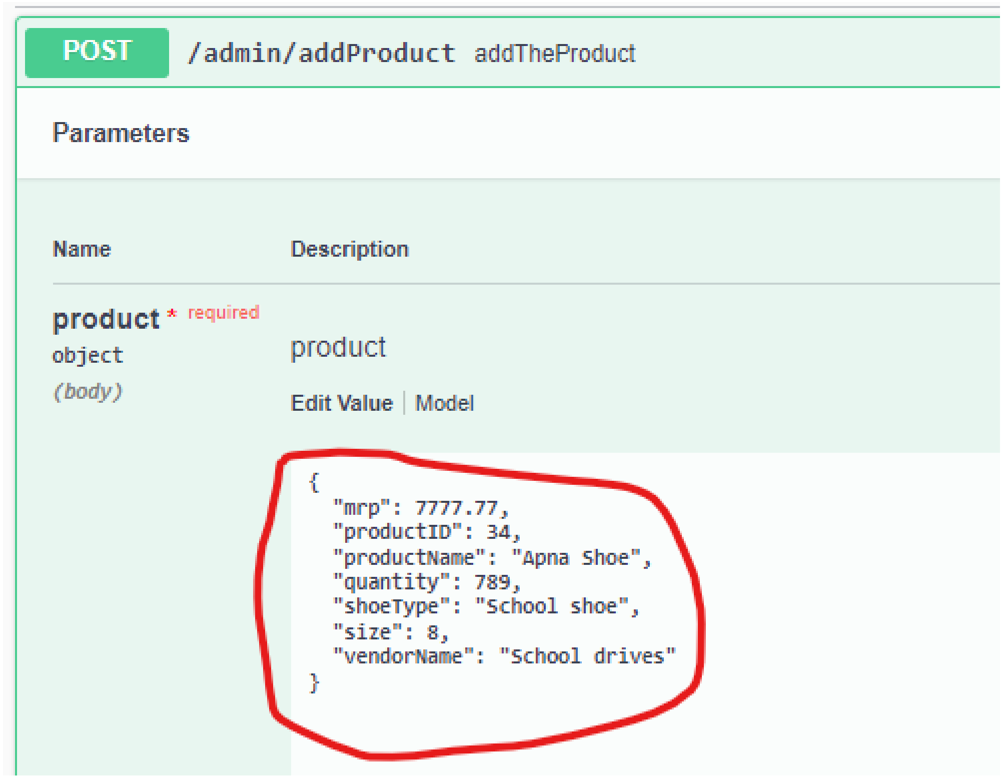
Picture 51 – Output of /admin/allProducts which will display all ProductNames

If admin want to add the product in the **Products_sporty_shoes** table he can use **/admin/addProduct** using RequestBody.



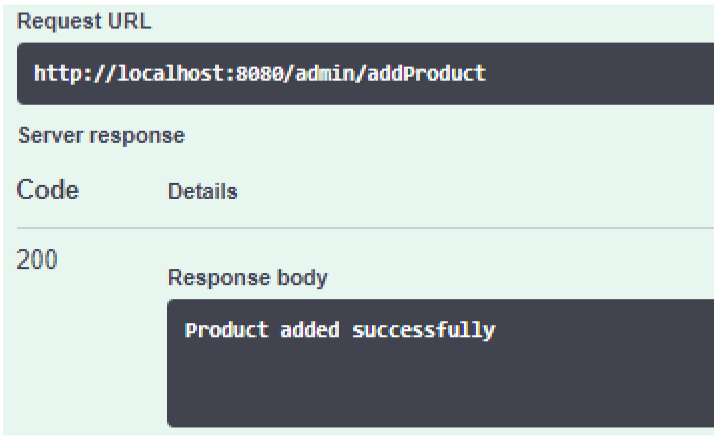
```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+------------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type        | size | vendor_name     |
+-----------+---------+--------------+----------+------------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe      |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes   |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes    |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes |   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers         |    8 | apache shoes    |
+-----------+---------+--------------+----------+------------------+------+-----------------+
```

Picture 52- Product table before adding the product



```
{
  "mrp": 7777.77,
  "productID": 34,
  "productName": "Apna Shoe",
  "quantity": 789,
  "shoeType": "School shoe",
  "size": 8,
  "vendorName": "School drives"
}
```

Picture 53- Execute /admin/addProduct

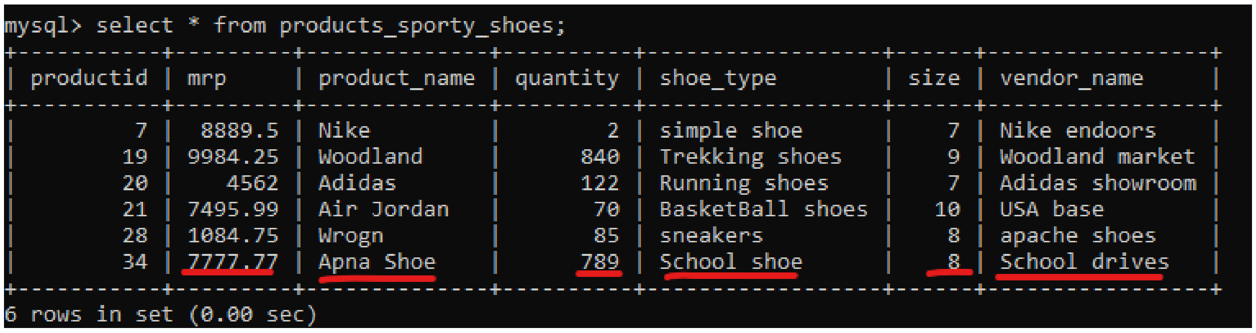


```
http://localhost:8080/admin/addProduct
```

200

```
Product added successfully
```

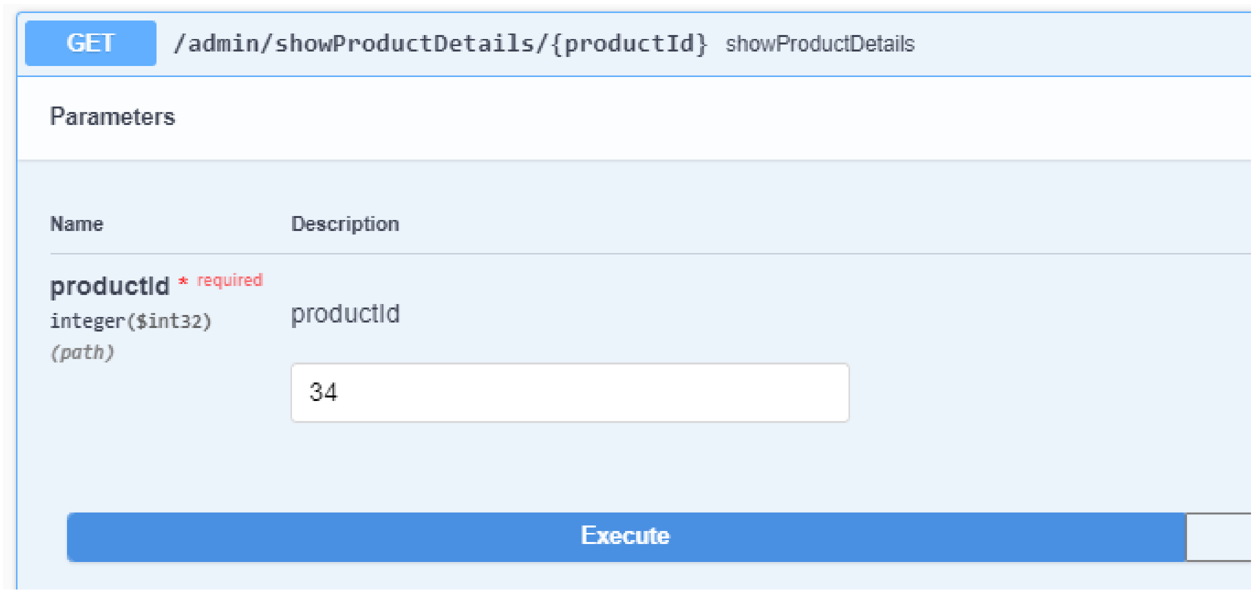
Picture 54- Output /admin/addProduct using json



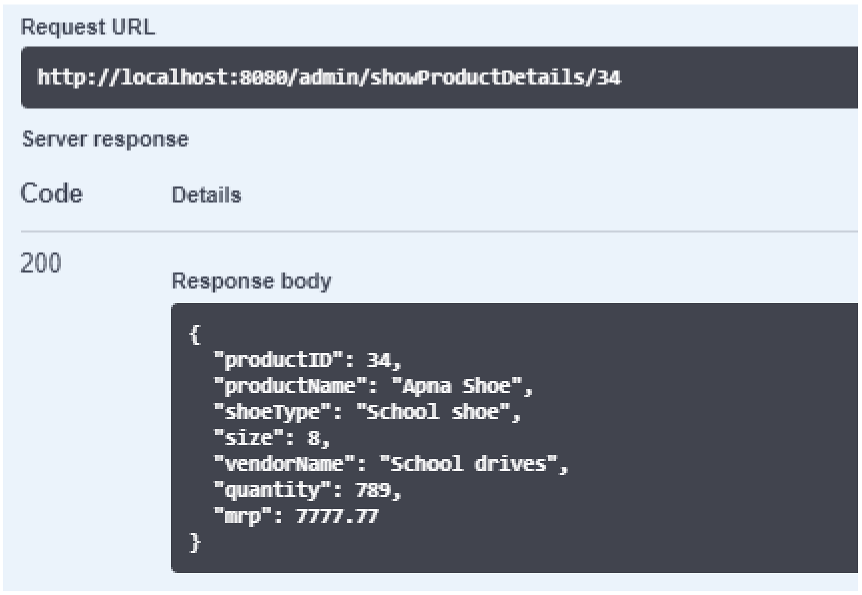
```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+------------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type        | size | vendor_name     |
+-----------+---------+--------------+----------+------------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe      |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes   |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes    |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes |   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers         |    8 | apache shoes    |
|        34 | 7777.77 | Apna Shoe    |      789 | School shoe      |    8 | School drives   |
+-----------+---------+--------------+----------+------------------+------+-----------------+
6 rows in set (0.00 sec)
```

Picture 55- In product table new product added successfully

Show all the product details using productId in swagger2.



Picture 56-Execute /admin/showProductDetails/{productId}

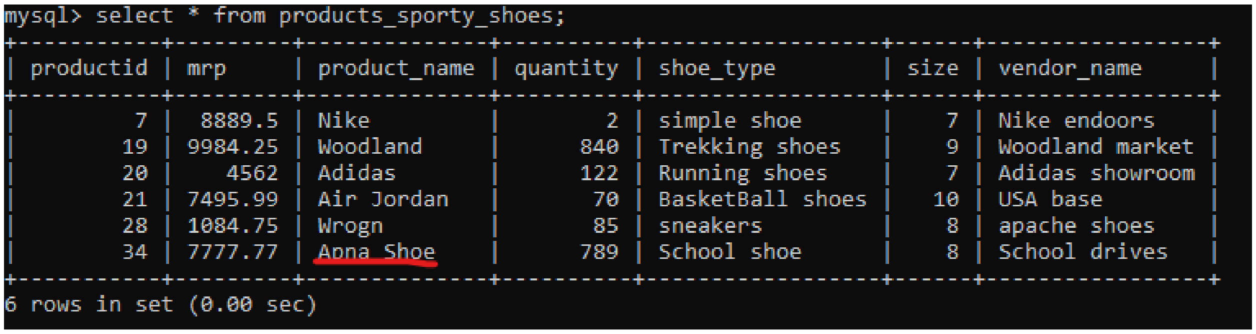


```
http://localhost:8080/admin/showProductDetails/34
```

```
{
  "productID": 34,
  "productName": "Apna Shoe",
  "shoeType": "School shoe",
  "size": 8,
  "vendorName": "School drives",
  "quantity": 789,
  "mrp": 7777.77
}
```

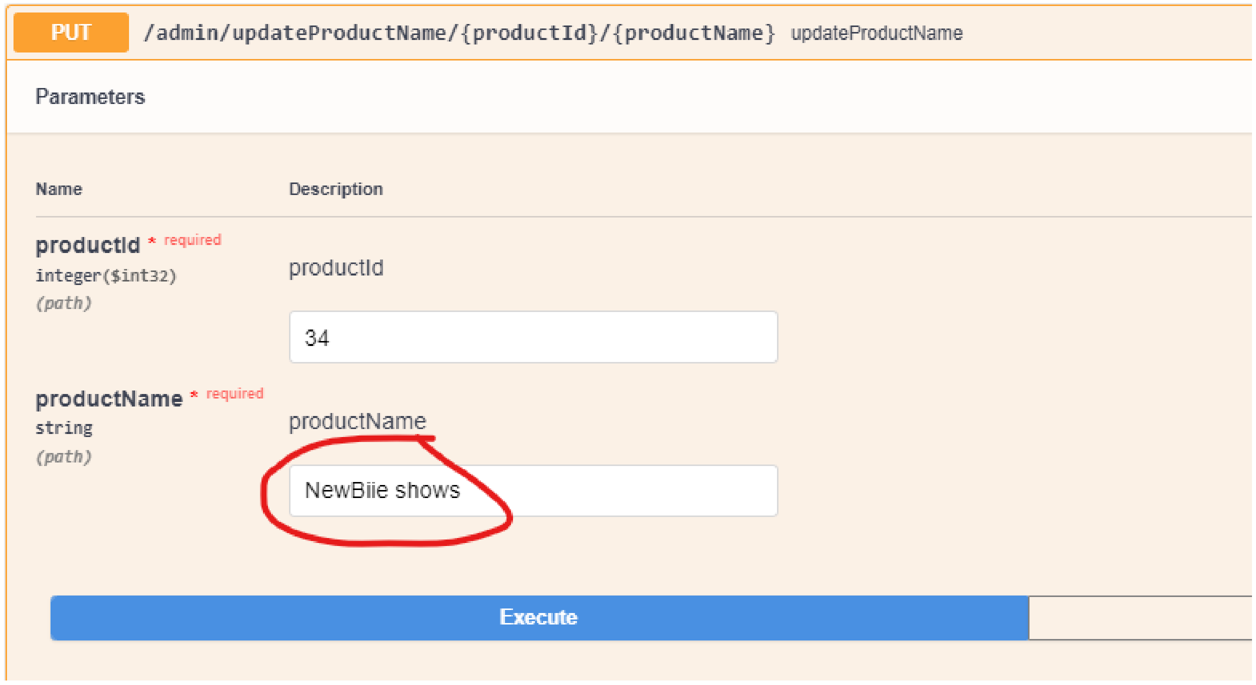
Picture 57- Output /admin/showProductDetails/34

To update the name of the product in the product table, here is the option **/admin/updateProductName/{productId}/{productName}.**



```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers        |    8 | apache shoes    |
|        34 | 7777.77 | Apna Shoe    |      789 | School shoe     |    8 | School drives   |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

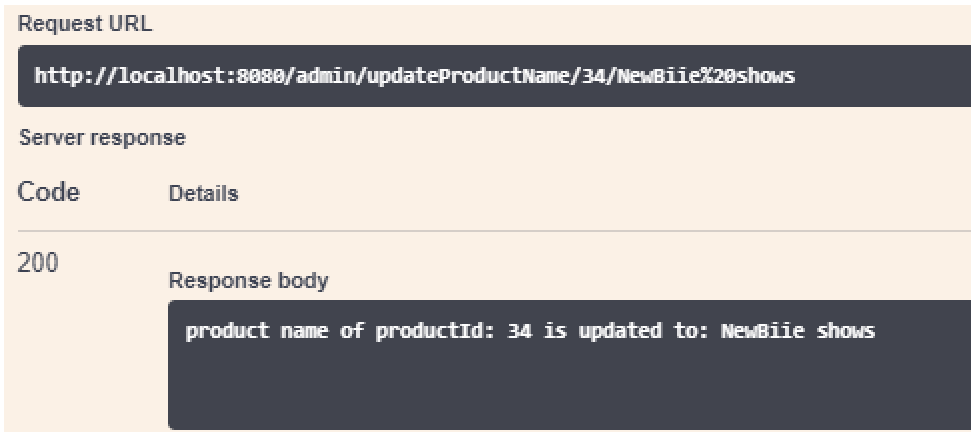
Picture 58- product table before update productName



productId: 34

productName: NewBiie shows

Picture 59- Execute /admin/updateProductName/{productId}/{productName}

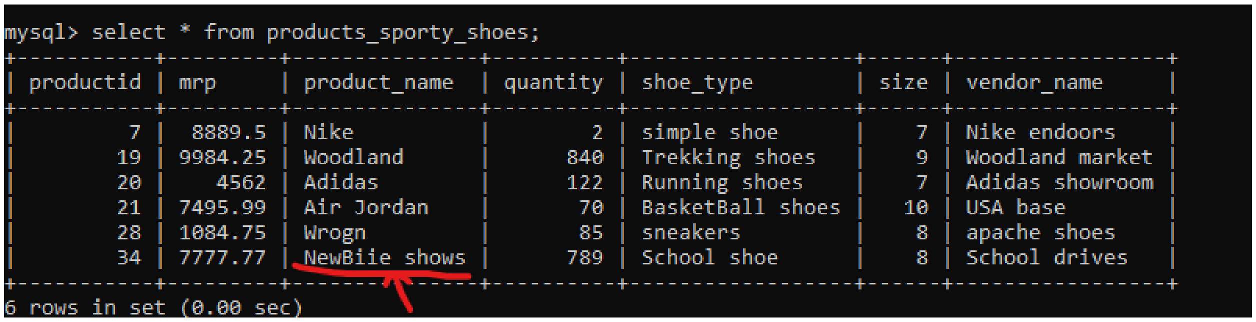


```
http://localhost:8080/admin/updateProductName/34/NewBiie%20shows
```

200

```
product name of productId: 34 is updated to: NewBiie shows
```

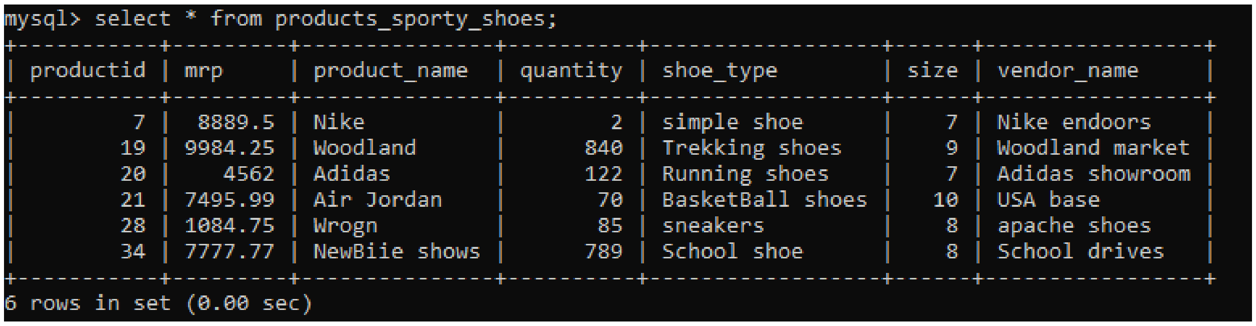
Picture 60- Output of /admin/updateProductName/{productId}/{productName}



```
mysql> select * from products_sporty_shoes;
+-----------+---------+---------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name  | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike          |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland      |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas        |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan    |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn         |       85 | sneakers        |    8 | apache shoes    |
|        34 | 7777.77 | NewBiie shows |      789 | School shoe     |    8 | School drives   |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

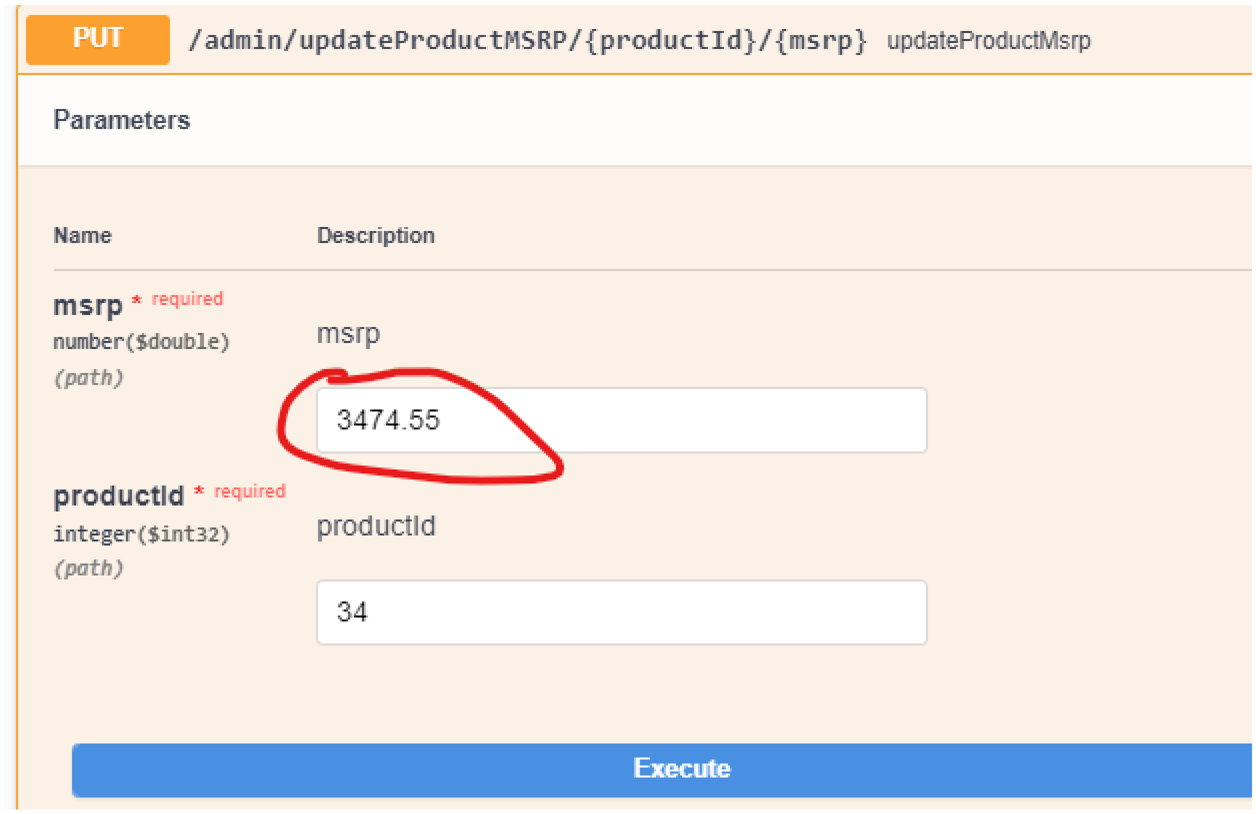
Picture 61- Product table after update productName

To update the msrp of the product in the product table, here is the option **/admin/updateProductMSRP/{productId}/{msrp}.**

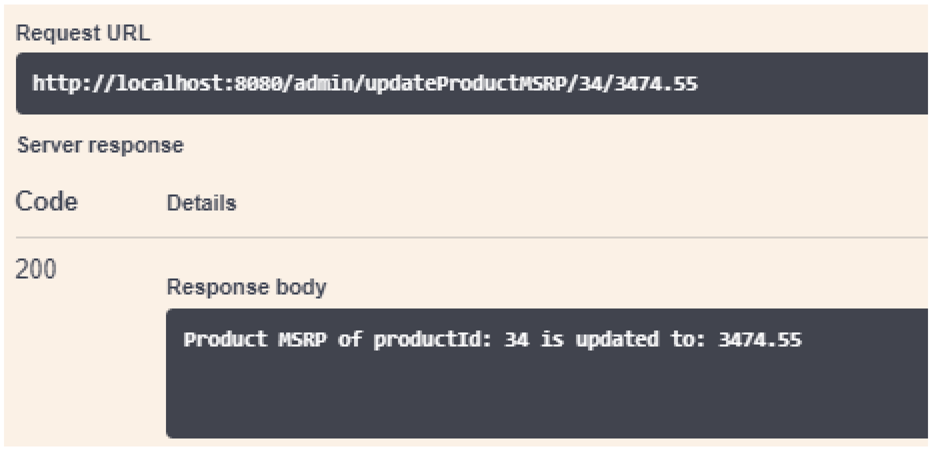


```
mysql> select * from products_sporty_shoes;
+-----------+---------+---------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name  | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike          |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland      |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas        |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan    |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn         |       85 | sneakers        |    8 | apache shoes    |
|        34 | 7777.77 | NewBiie shows |      789 | School shoe     |    8 | School drives   |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

Picture 62- product table before update MSRP



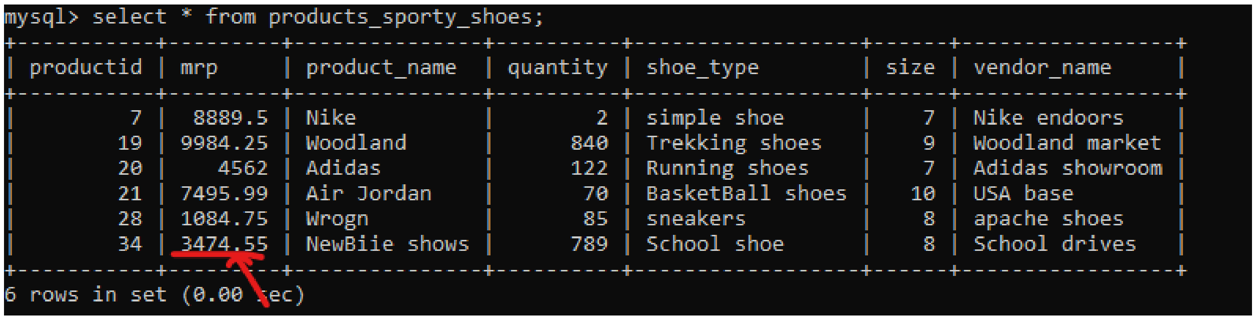
Picture 63- Execute /admin/updateProductMSRP/{productId}/{msrp}



```
http://localhost:8080/admin/updateProductMSRP/34/3474.55
```

```
Product MSRP of productId: 34 is updated to: 3474.55
```

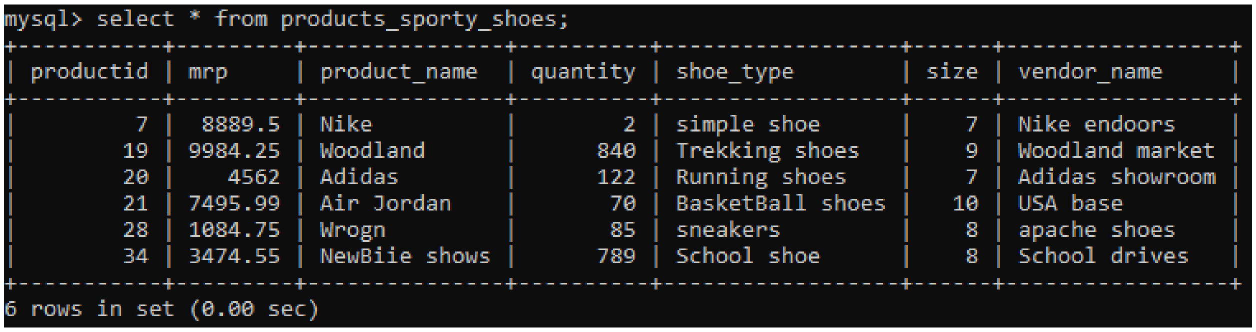
Picture 64- Output of /admin/updateProductMSRP/{productId}/{msrp}



```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes |   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers        |    8 | apache shoes    |
|        34 | 3474.55 | NewBiie shows |     789 | School shoe     |    8 | School drives   |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

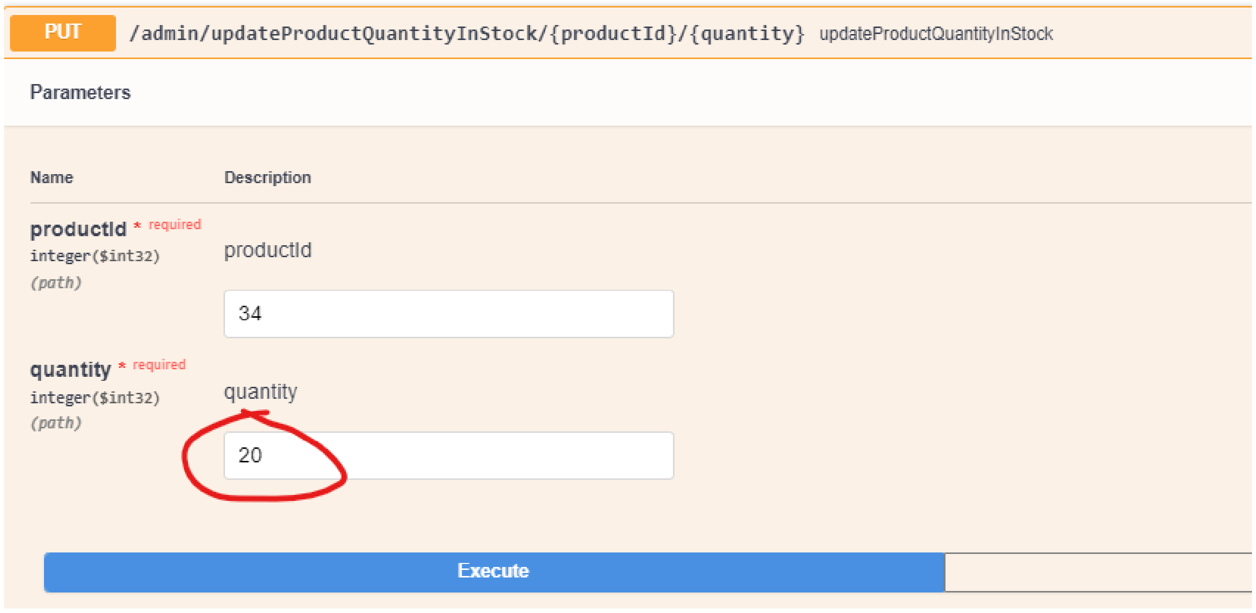
Picture 65- Product table after update ProductMSRP

To update the quantity of the product in the product table, here is the option **/admin/updateProductMSRP/{productId}/{msrp}.**



```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes |   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers        |    8 | apache shoes    |
|        34 | 3474.55 | NewBiie shows |     789 | School shoe     |    8 | School drives   |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

Picture 66- product table before update Quantity of product in the stock

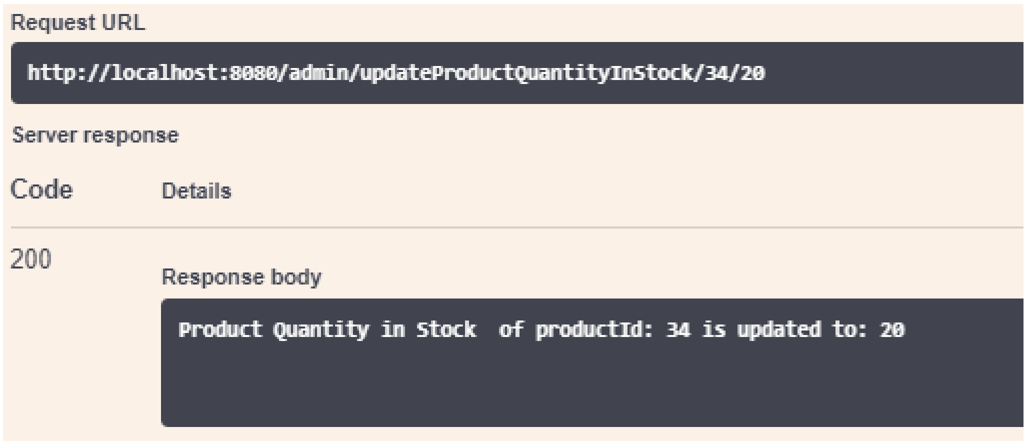


PUT /admin/updateProductQuantityInStock/{productId}/{quantity} updateProductQuantityInStock

Parameters

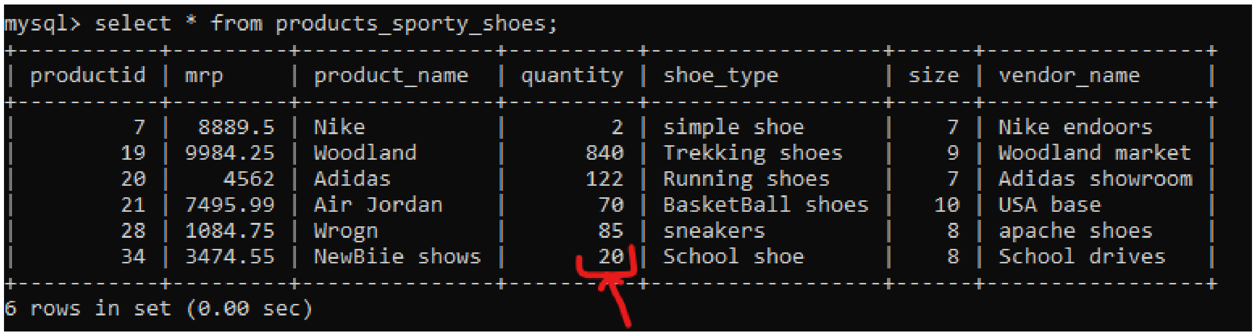
Name	Description
productId * required integer($int32) (path)	productId: 34
quantity * required integer($int32) (path)	quantity: 20

Picture 67- Execute /admin/updateProductQuantityInStock/{productId}/{quantity}



```
Request URL
http://localhost:8080/admin/updateProductQuantityInStock/34/20
Server response
Code    Details
200     Response body
        Product Quantity in Stock  of productId: 34 is updated to: 20
```

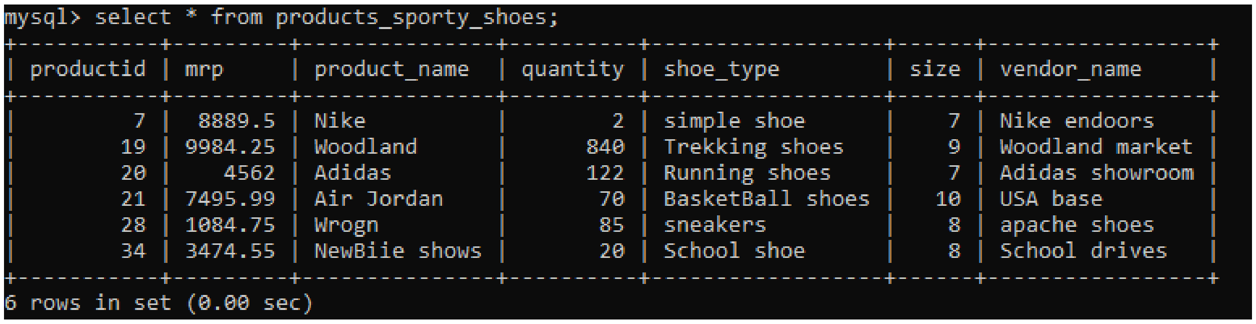
Picture 68- Output of /admin/updateProductQuantityInStock/{productId}/{quantity}



```
mysql> select * from products_sporty_shoes;
+-----------+---------+---------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name  | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike          |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland      |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas        |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan    |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn         |       85 | sneakers        |    8 | apache shoes    |
|        34 | 3474.55 | NewBiie shows |       20 | School shoe     |    8 | School drives   |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

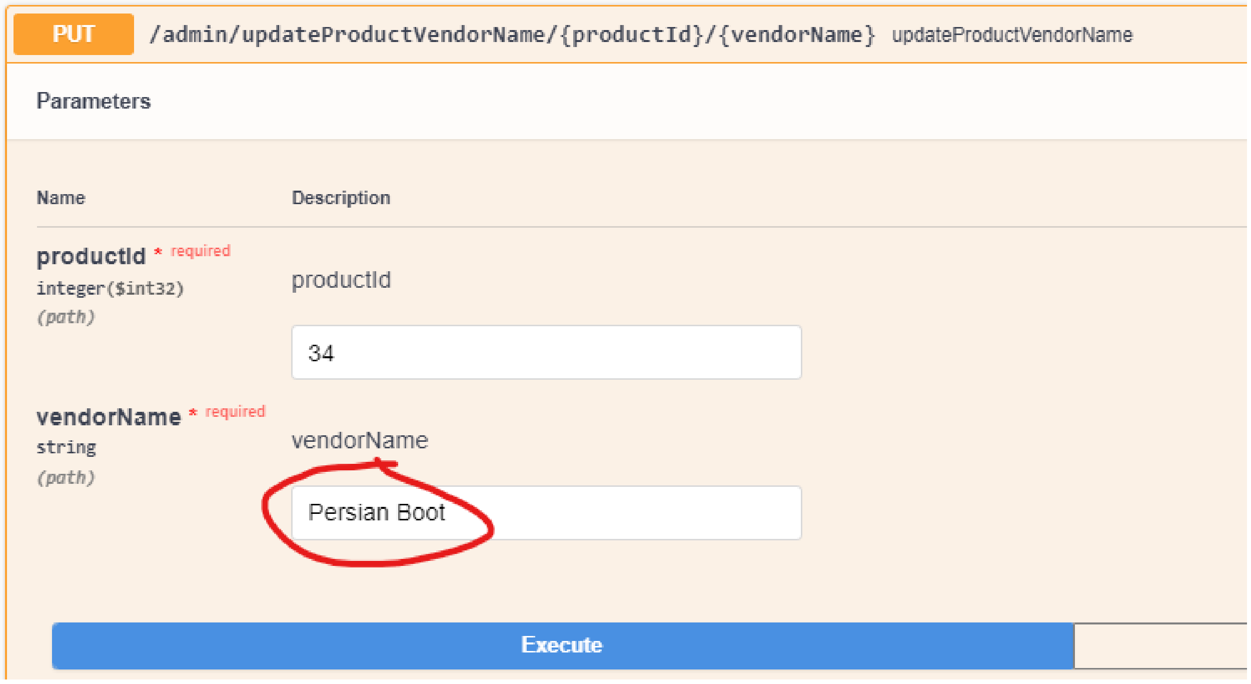
Picture 69- Product table after update ProductQuantityInStock

To update the product Vendor name of the product in the product table, here is the option **/admin/updateProductVendorName/{productId}/{vendorName}.**



```
mysql> select * from products_sporty_shoes;
+-----------+---------+---------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name  | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike          |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland      |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas        |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan    |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn         |       85 | sneakers        |    8 | apache shoes    |
|        34 | 3474.55 | NewBiie shows |       20 | School shoe     |    8 | School drives   |
+-----------+---------+---------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

Picture 70- product table before update Vendor Name

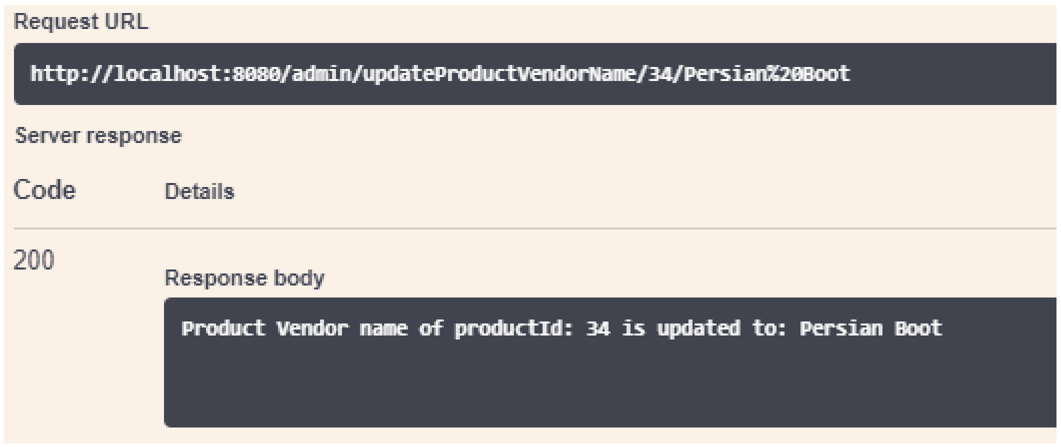


PUT /admin/updateProductVendorName/{productId}/{vendorName} updateProductVendorName

Parameters

Name	Description
productId * required integer($int32) (path)	productId 34
vendorName * required string (path)	vendorName Persian Boot

Picture 71- Execute /admin/updateProductVendorName/{productId}/{vendorName}



Request URL

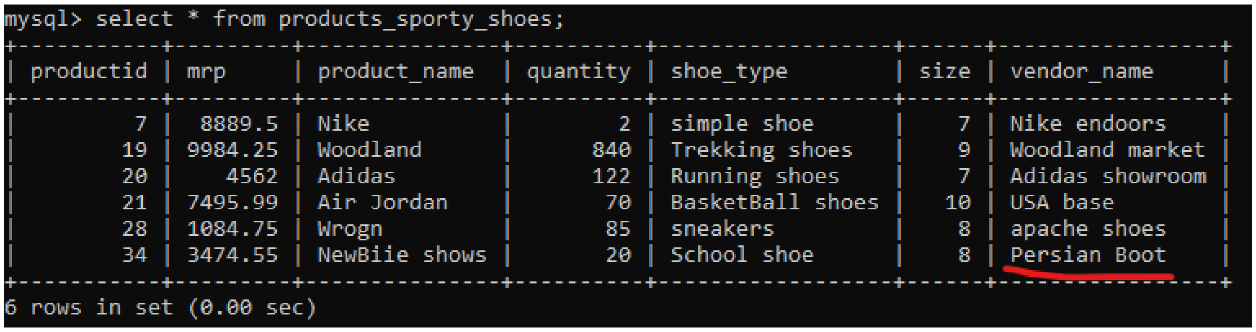
```
http://localhost:8080/admin/updateProductVendorName/34/Persian%20Boot
```

Server response

Code	Details
200	Response body

```
Product Vendor name of productId: 34 is updated to: Persian Boot
```

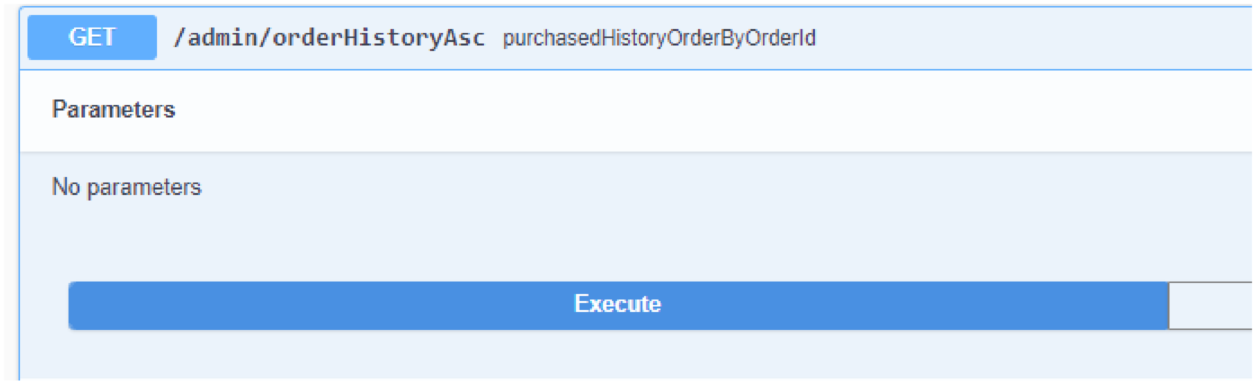
Picture 72- Output of /admin/updateProductVendorName/{productId}/{vendorName}



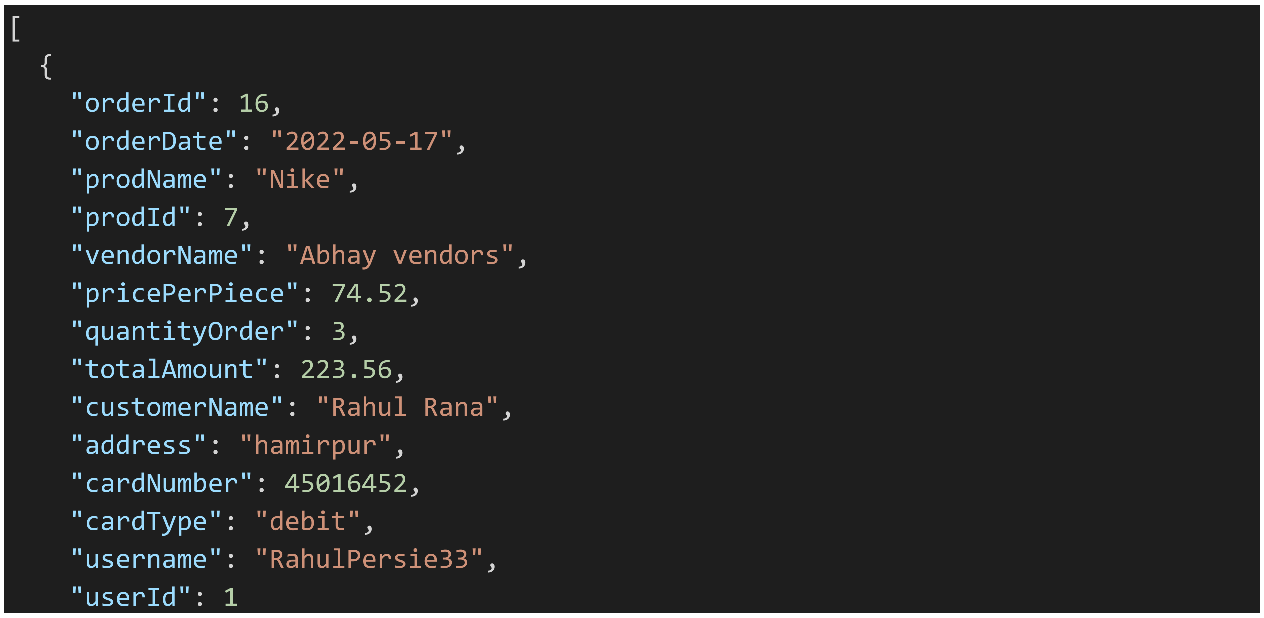
```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes |  10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers        |    8 | apache shoes    |
|        34 | 3474.55 | NewBiie shows |      20 | School shoe     |    8 | Persian Boot    |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
6 rows in set (0.00 sec)
```

Picture 73- Product table after update ProductVendorName

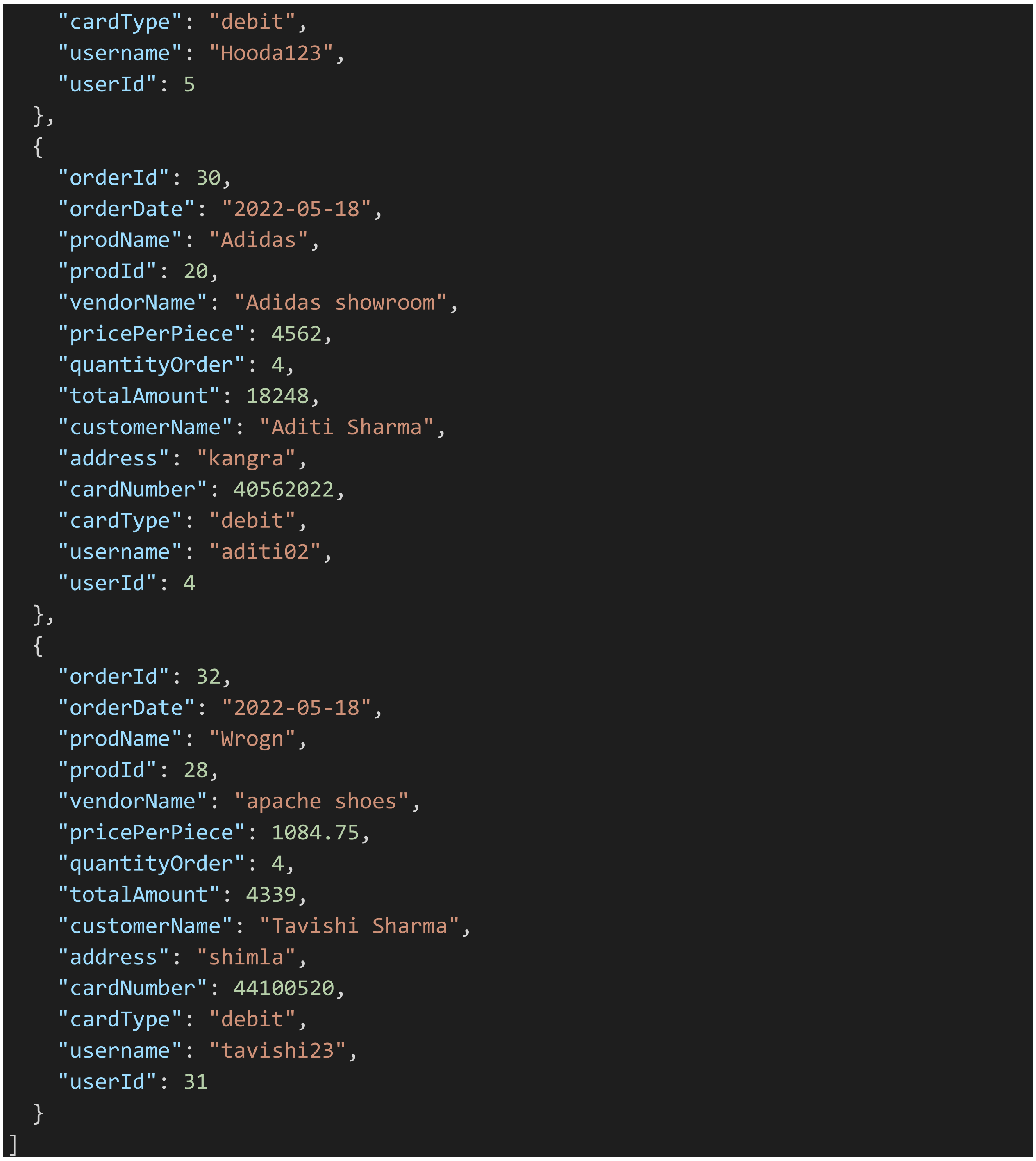
If admin want to view all Orders placed in OrderId in ascending, then select option **/admin/orderHistoryAsc** in swagger2.



Picture 74-Execute admin/orderHistoryAsc



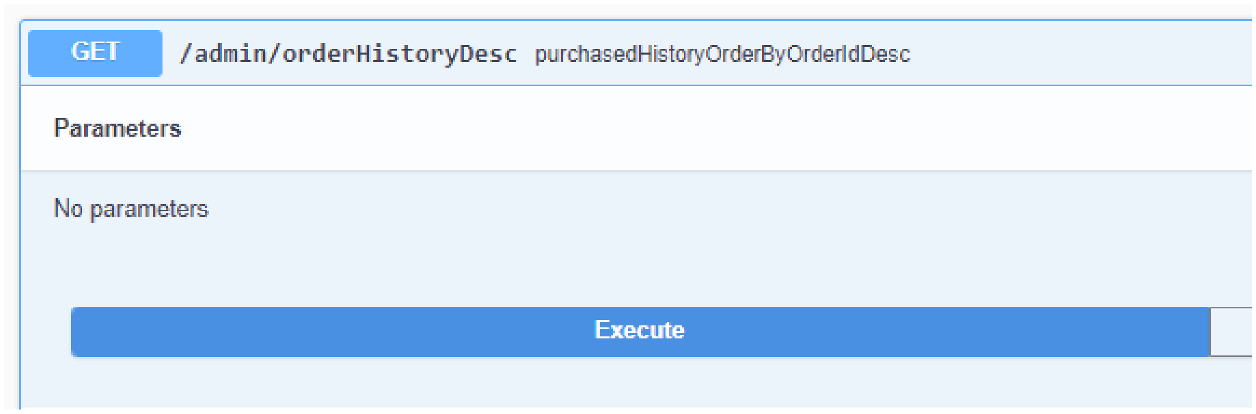
```
[
  {
    "orderId": 16,
    "orderDate": "2022-05-17",
    "prodName": "Nike",
    "prodId": 7,
    "vendorName": "Abhay vendors",
    "pricePerPiece": 74.52,
    "quantityOrder": 3,
    "totalAmount": 223.56,
    "customerName": "Rahul Rana",
    "address": "hamirpur",
    "cardNumber": 45016452,
    "cardType": "debit",
    "username": "RahulPersie33",
    "userId": 1
  },
  {
    "orderId": 22,
    "orderDate": "2022-05-18",
    "prodName": "Woodland",
    "prodId": 19,
    "vendorName": "Woodland market",
    "pricePerPiece": 9984.25,
    "quantityOrder": 1,
    "totalAmount": 9984.25,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 23,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 2,
    "totalAmount": 9124,
    "customerName": "Dean Ambrose",
    "address": "north carolina",
    "cardNumber": 40406985,
    "cardType": "credit",
    "username": "dean34",
    "userId": 6
  },
  {
    "orderId": 24,
    "orderDate": "2022-05-18",
    "prodName": "Air Jordan",
    "prodId": 21,
    "vendorName": "USA base",
    "pricePerPiece": 7495.99,
    "quantityOrder": 5,
    "totalAmount": 37479.95,
    "customerName": "Hooda Ronney",
    "address": "manchester",
    "cardNumber": 30540004,
```



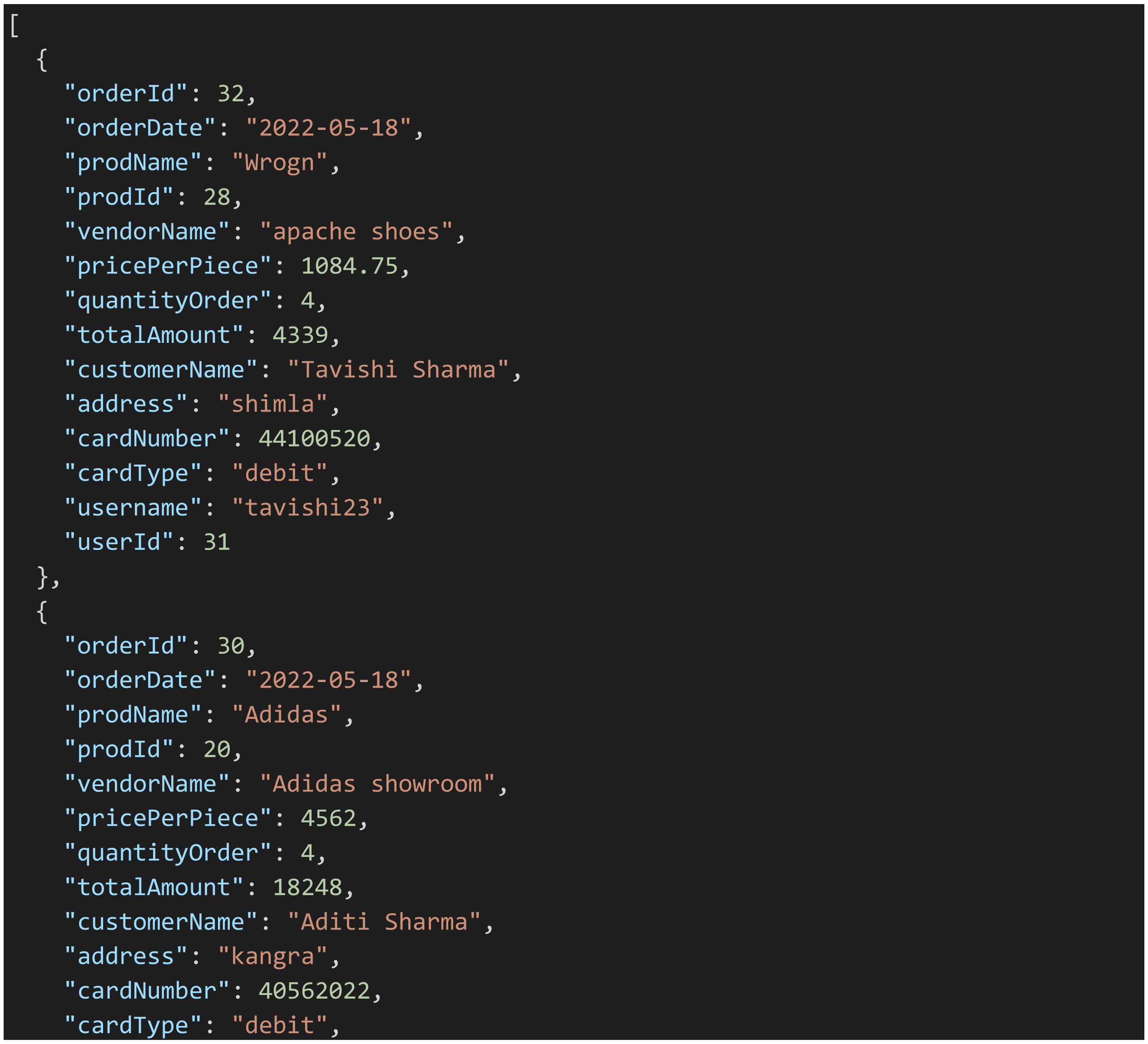
```
    "cardType": "debit",
    "username": "Hooda123",
    "userId": 5
  },
  {
    "orderId": 30,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 4,
    "totalAmount": 18248,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 32,
    "orderDate": "2022-05-18",
    "prodName": "Wrogn",
    "prodId": 28,
    "vendorName": "apache shoes",
    "pricePerPiece": 1084.75,
    "quantityOrder": 4,
    "totalAmount": 4339,
    "customerName": "Tavishi Sharma",
    "address": "shimla",
    "cardNumber": 44100520,
    "cardType": "debit",
    "username": "tavishi23",
    "userId": 31
  }
]
```

Picture 75- Output of admin/orderHistoryAsc

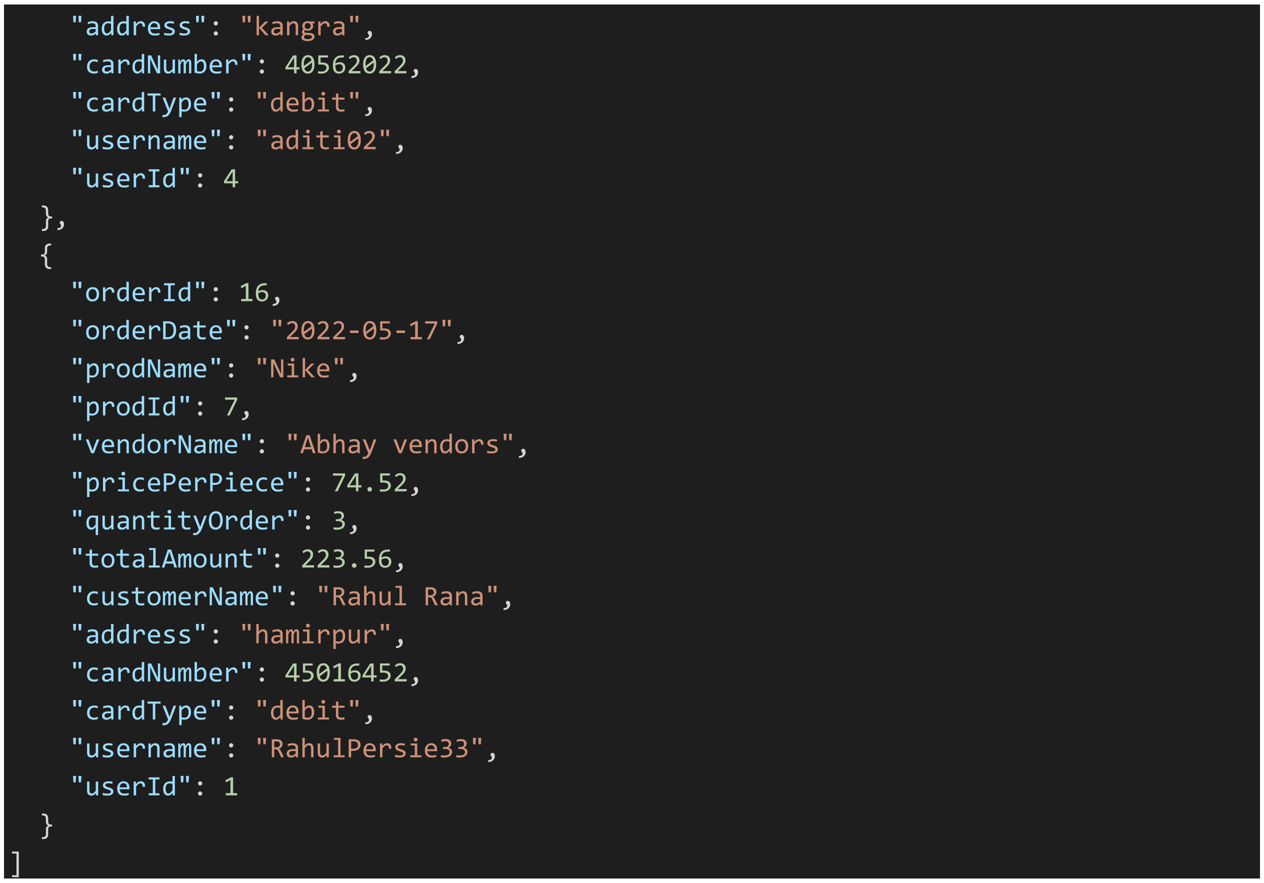
If admin want to view all Orders placed in OrderId in descending, then select option **/admin/orderHistoryDesc** in swagger2.



Picture 76-Execute admin/orderHistoryDesc



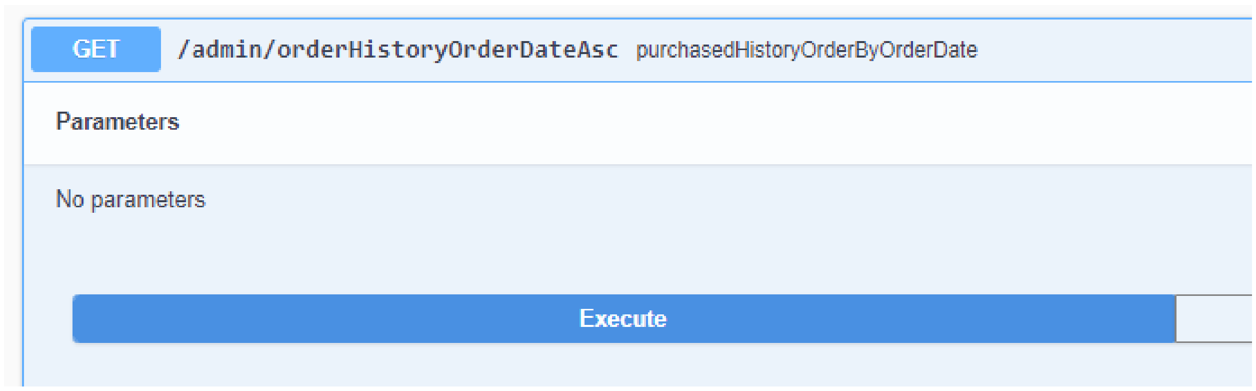
```
[
  {
    "orderId": 32,
    "orderDate": "2022-05-18",
    "prodName": "Wrogn",
    "prodId": 28,
    "vendorName": "apache shoes",
    "pricePerPiece": 1084.75,
    "quantityOrder": 4,
    "totalAmount": 4339,
    "customerName": "Tavishi Sharma",
    "address": "shimla",
    "cardNumber": 44100520,
    "cardType": "debit",
    "username": "tavishi23",
    "userId": 31
  },
  {
    "orderId": 30,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 4,
    "totalAmount": 18248,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 24,
    "orderDate": "2022-05-18",
    "prodName": "Air Jordan",
    "prodId": 21,
    "vendorName": "USA base",
    "pricePerPiece": 7495.99,
    "quantityOrder": 5,
    "totalAmount": 37479.95,
    "customerName": "Hooda Ronney",
    "address": "manchester",
    "cardNumber": 30540004,
    "cardType": "debit",
    "username": "Hooda123",
    "userId": 5
  },
  {
    "orderId": 23,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 2,
    "totalAmount": 9124,
    "customerName": "Dean Ambrose",
    "address": "north carolina",
    "cardNumber": 40406985,
    "cardType": "credit",
    "username": "dean34",
    "userId": 6
  },
  {
    "orderId": 22,
    "orderDate": "2022-05-18",
    "prodName": "Woodland",
    "prodId": 19,
    "vendorName": "Woodland market",
    "pricePerPiece": 9984.25,
    "quantityOrder": 1,
    "totalAmount": 9984.25,
    "customerName": "Aditi Sharma",
```



```
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 16,
    "orderDate": "2022-05-17",
    "prodName": "Nike",
    "prodId": 7,
    "vendorName": "Abhay vendors",
    "pricePerPiece": 74.52,
    "quantityOrder": 3,
    "totalAmount": 223.56,
    "customerName": "Rahul Rana",
    "address": "hamirpur",
    "cardNumber": 45016452,
    "cardType": "debit",
    "username": "RahulPersie33",
    "userId": 1
  }
]
```

Picture 77- Output of admin/orderHistoryDesc in json format

If admin want to view all Orders placed by OrderDate in ascending, then select option **/admin/orderHistoryOrderDateAsc** in swagger2.

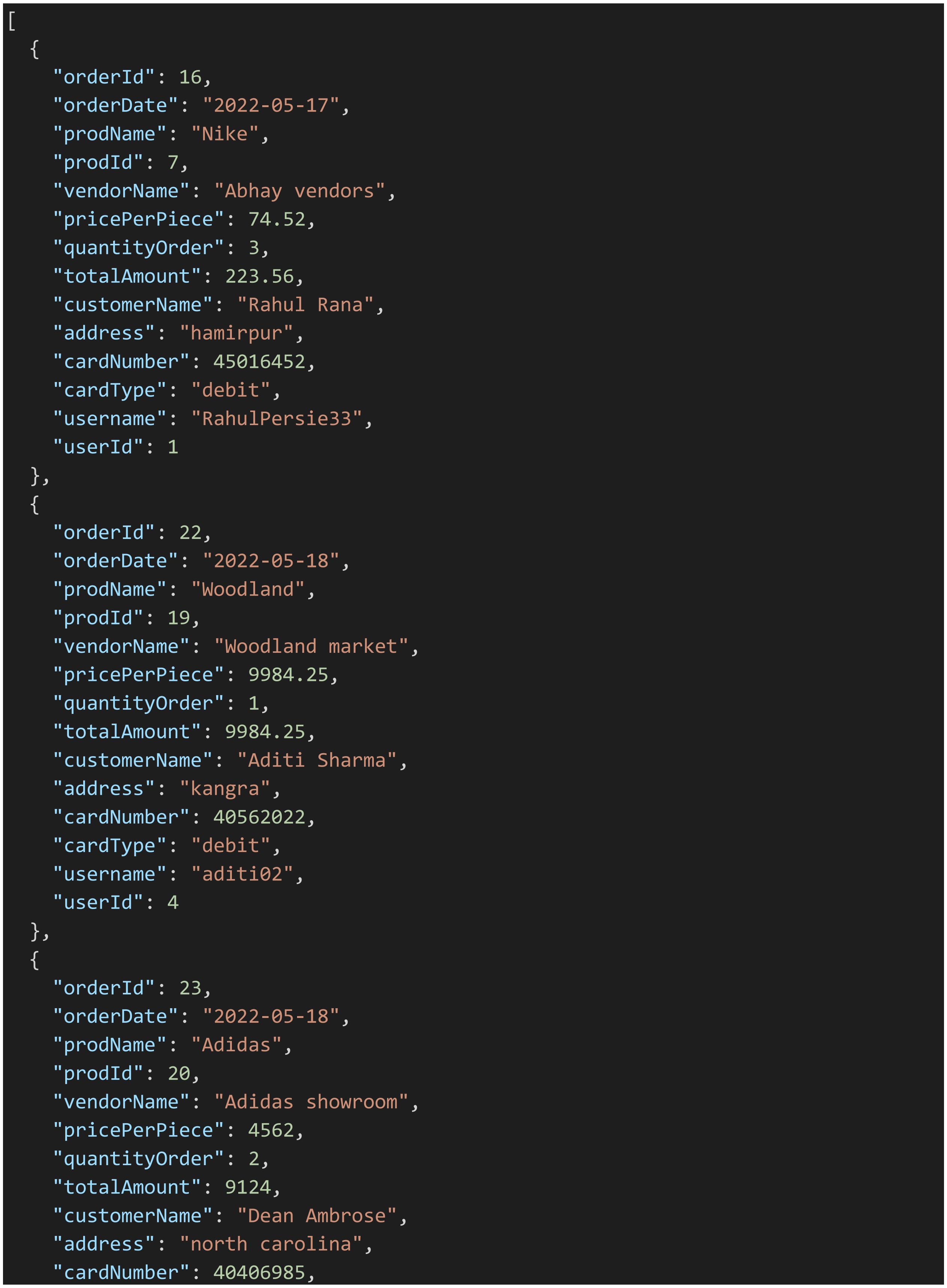


GET /admin/orderHistoryOrderDateAsc purchasedHistoryOrderByOrderDate

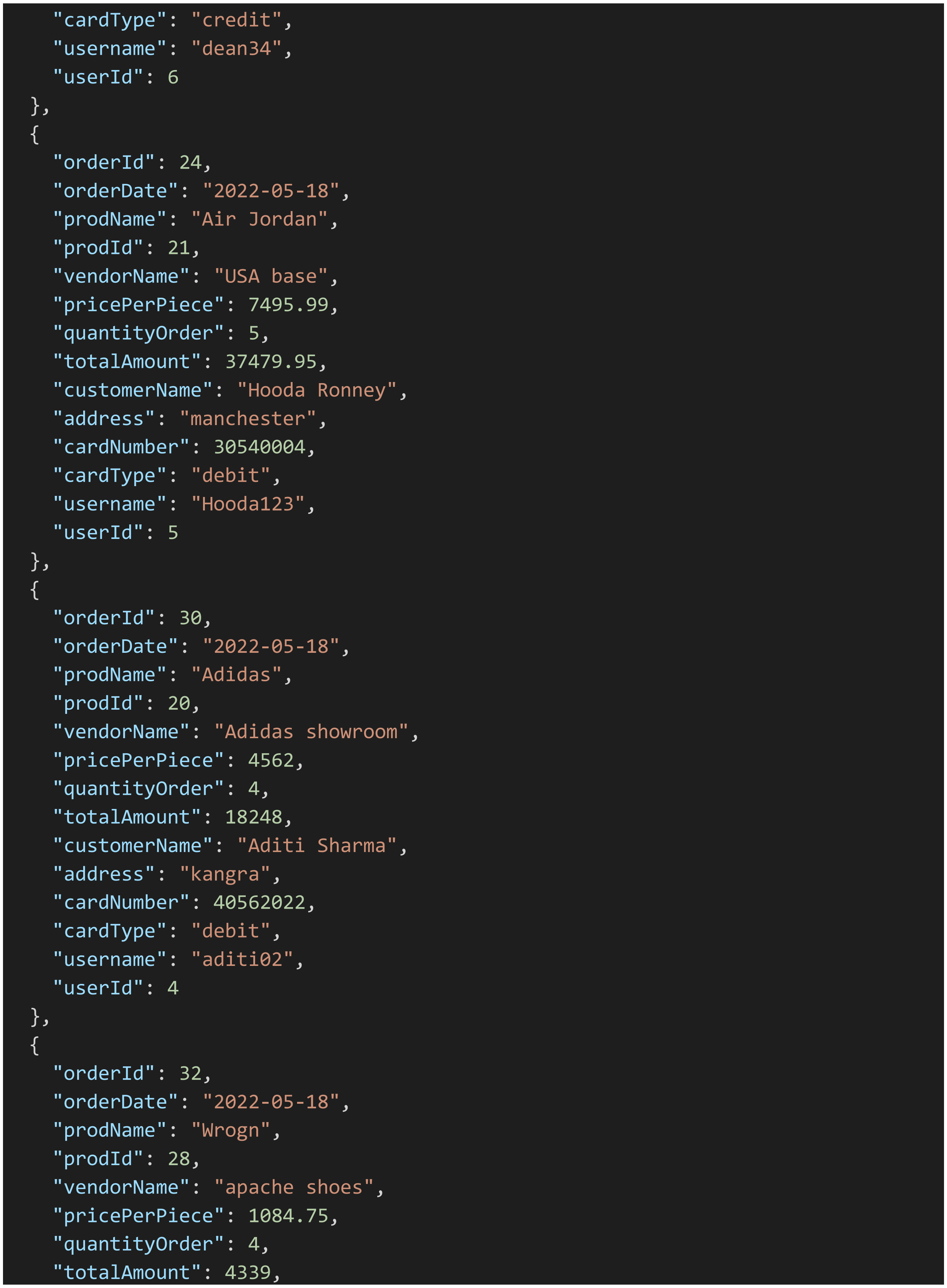
Parameters

No parameters

Picture 78-Execute admin/orderHistoryOrderDateAsc



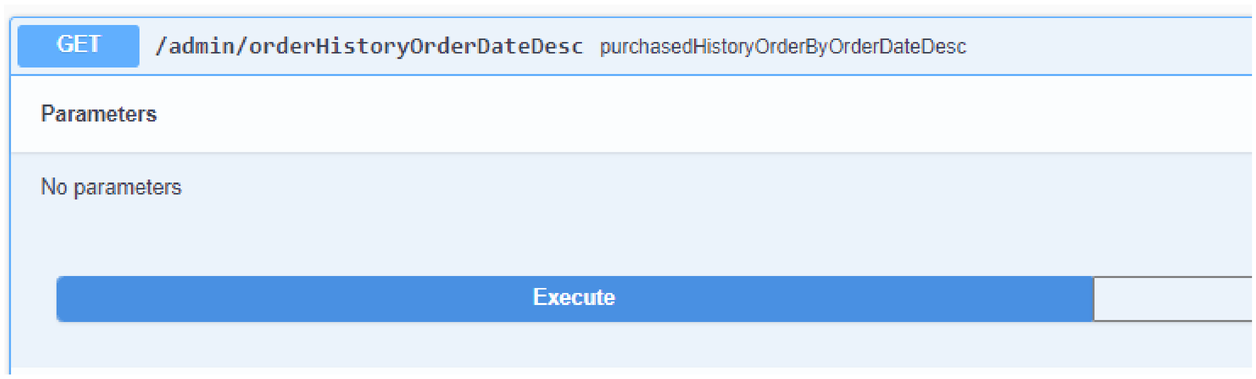
```json
[
  {
    "orderId": 16,
    "orderDate": "2022-05-17",
    "prodName": "Nike",
    "prodId": 7,
    "vendorName": "Abhay vendors",
    "pricePerPiece": 74.52,
    "quantityOrder": 3,
    "totalAmount": 223.56,
    "customerName": "Rahul Rana",
    "address": "hamirpur",
    "cardNumber": 45016452,
    "cardType": "debit",
    "username": "RahulPersie33",
    "userId": 1
  },
  {
    "orderId": 22,
    "orderDate": "2022-05-18",
    "prodName": "Woodland",
    "prodId": 19,
    "vendorName": "Woodland market",
    "pricePerPiece": 9984.25,
    "quantityOrder": 1,
    "totalAmount": 9984.25,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 23,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 2,
    "totalAmount": 9124,
    "customerName": "Dean Ambrose",
    "address": "north carolina",
    "cardNumber": 40406985,
```



```
    "cardType": "credit",
    "username": "dean34",
    "userId": 6
  },
  {
    "orderId": 24,
    "orderDate": "2022-05-18",
    "prodName": "Air Jordan",
    "prodId": 21,
    "vendorName": "USA base",
    "pricePerPiece": 7495.99,
    "quantityOrder": 5,
    "totalAmount": 37479.95,
    "customerName": "Hooda Ronney",
    "address": "manchester",
    "cardNumber": 30540004,
    "cardType": "debit",
    "username": "Hooda123",
    "userId": 5
  },
  {
    "orderId": 30,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 4,
    "totalAmount": 18248,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 32,
    "orderDate": "2022-05-18",
    "prodName": "Wrogn",
    "prodId": 28,
    "vendorName": "apache shoes",
    "pricePerPiece": 1084.75,
    "quantityOrder": 4,
    "totalAmount": 4339,
    "customerName": "Tavishi Sharma",
    "address": "shimla",
    "cardNumber": 44100520,
    "cardType": "debit",
    "username": "tavishi23",
    "userId": 31
  }
]
```

Picture 79- Output admin/orderHistoryOrderDateAsc

If admin want to view all Orders placed by OrderDate in descending, then select option **/admin/orderHistoryOrderDateDesc** in swagger2.


GET /admin/orderHistoryOrderDateDesc purchasedHistoryOrderByOrderDateDesc

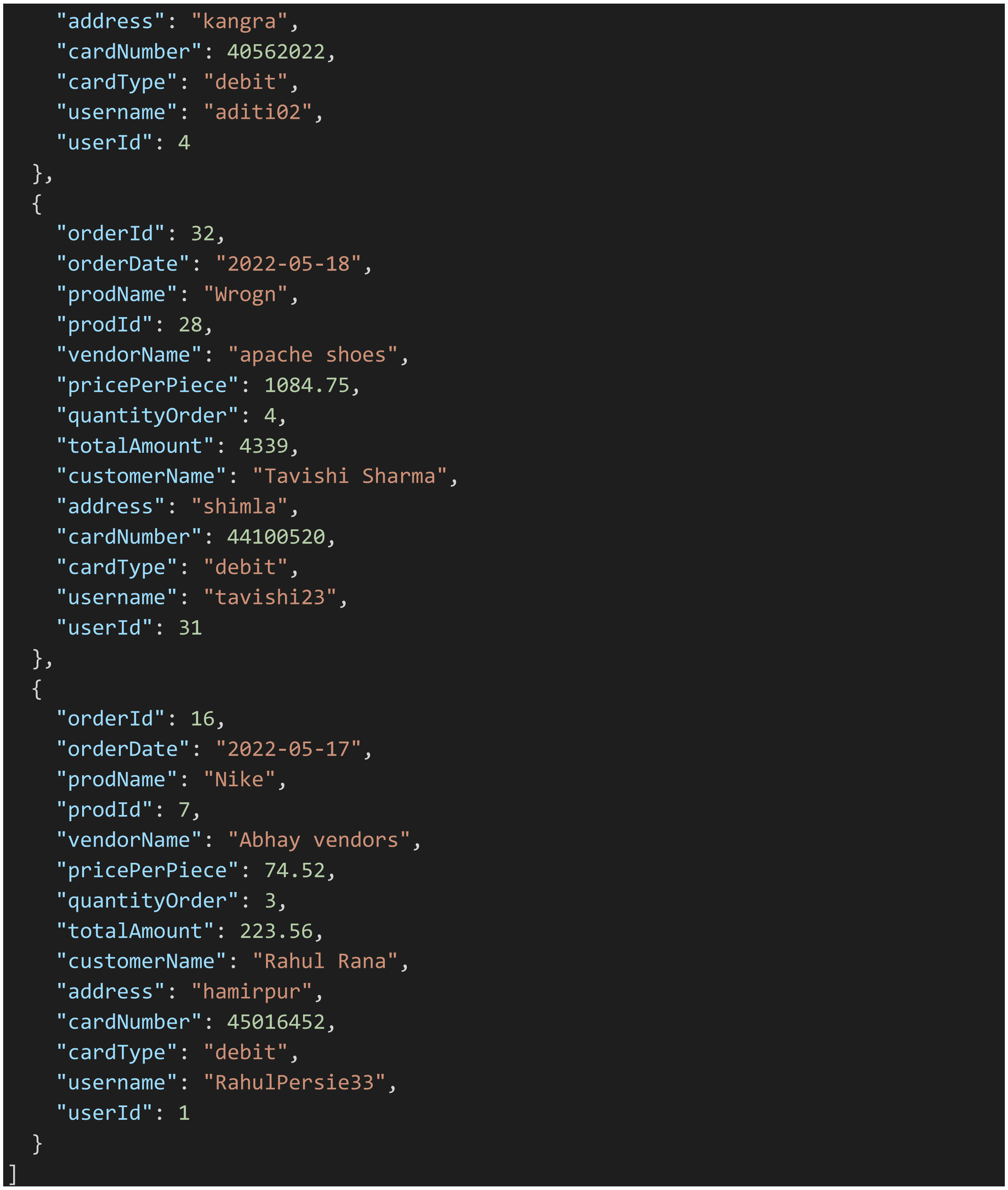
Parameters

No parameters

Execute

Picture 80-Execute admin/orderHistoryOrderDateDesc

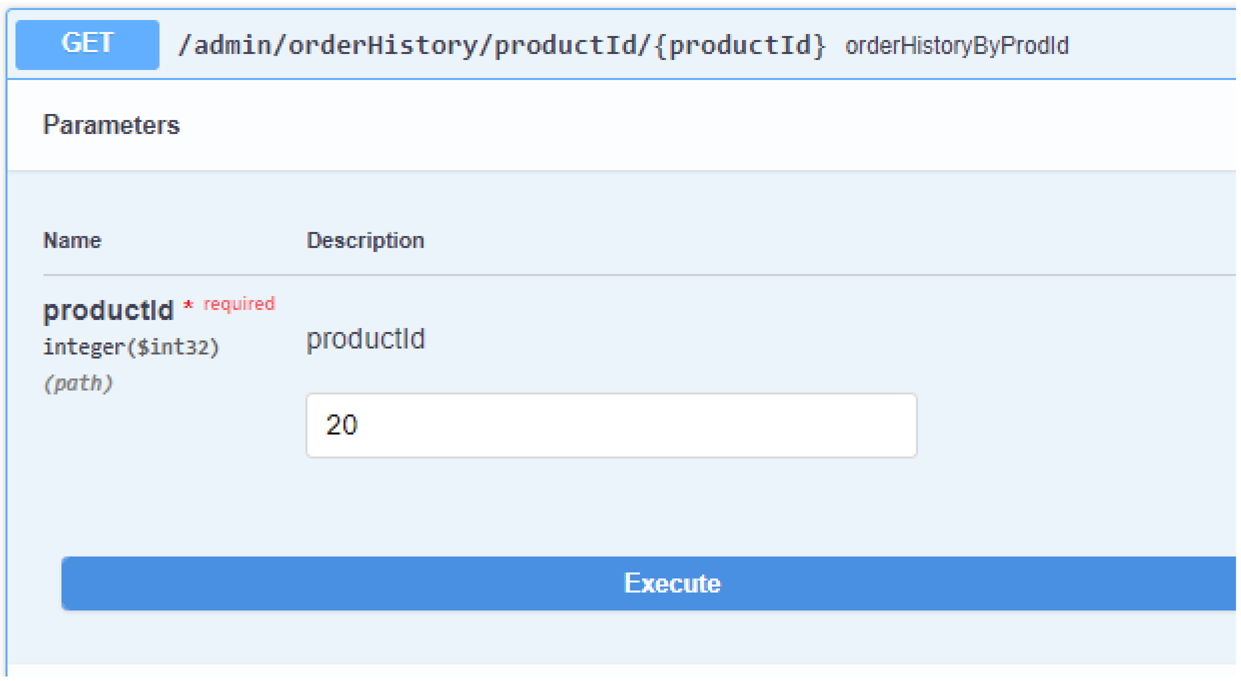
```
[
  {
    "orderId": 22,
    "orderDate": "2022-05-18",
    "prodName": "Woodland",
    "prodId": 19,
    "vendorName": "Woodland market",
    "pricePerPiece": 9984.25,
    "quantityOrder": 1,
    "totalAmount": 9984.25,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 23,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 2,
    "totalAmount": 9124,
    "customerName": "Dean Ambrose",
    "address": "north carolina",
    "cardNumber": 40406985,
    "cardType": "credit",
    "username": "dean34",
    "userId": 6
  },
  {
    "orderId": 24,
    "orderDate": "2022-05-18",
    "prodName": "Air Jordan",
    "prodId": 21,
    "vendorName": "USA base",
    "pricePerPiece": 7495.99,
    "quantityOrder": 5,
    "totalAmount": 37479.95,
    "customerName": "Hooda Ronney",
    "address": "manchester",
    "cardNumber": 30540004,
    "cardType": "debit",
    "username": "Hooda123",
    "userId": 5
  },
  {
    "orderId": 30,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 4,
    "totalAmount": 18248,
    "customerName": "Aditi Sharma",
```



```
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  },
  {
    "orderId": 32,
    "orderDate": "2022-05-18",
    "prodName": "Wrogn",
    "prodId": 28,
    "vendorName": "apache shoes",
    "pricePerPiece": 1084.75,
    "quantityOrder": 4,
    "totalAmount": 4339,
    "customerName": "Tavishi Sharma",
    "address": "shimla",
    "cardNumber": 44100520,
    "cardType": "debit",
    "username": "tavishi23",
    "userId": 31
  },
  {
    "orderId": 16,
    "orderDate": "2022-05-17",
    "prodName": "Nike",
    "prodId": 7,
    "vendorName": "Abhay vendors",
    "pricePerPiece": 74.52,
    "quantityOrder": 3,
    "totalAmount": 223.56,
    "customerName": "Rahul Rana",
    "address": "hamirpur",
    "cardNumber": 45016452,
    "cardType": "debit",
    "username": "RahulPersie33",
    "userId": 1
  }
]
```

Picture 81- Output admin/orderHistoryOrderDateDesc

If admin want to view orders for productId pid, the admin can go for **/admin/orderHistory/productId/{productId}** in swagger2.

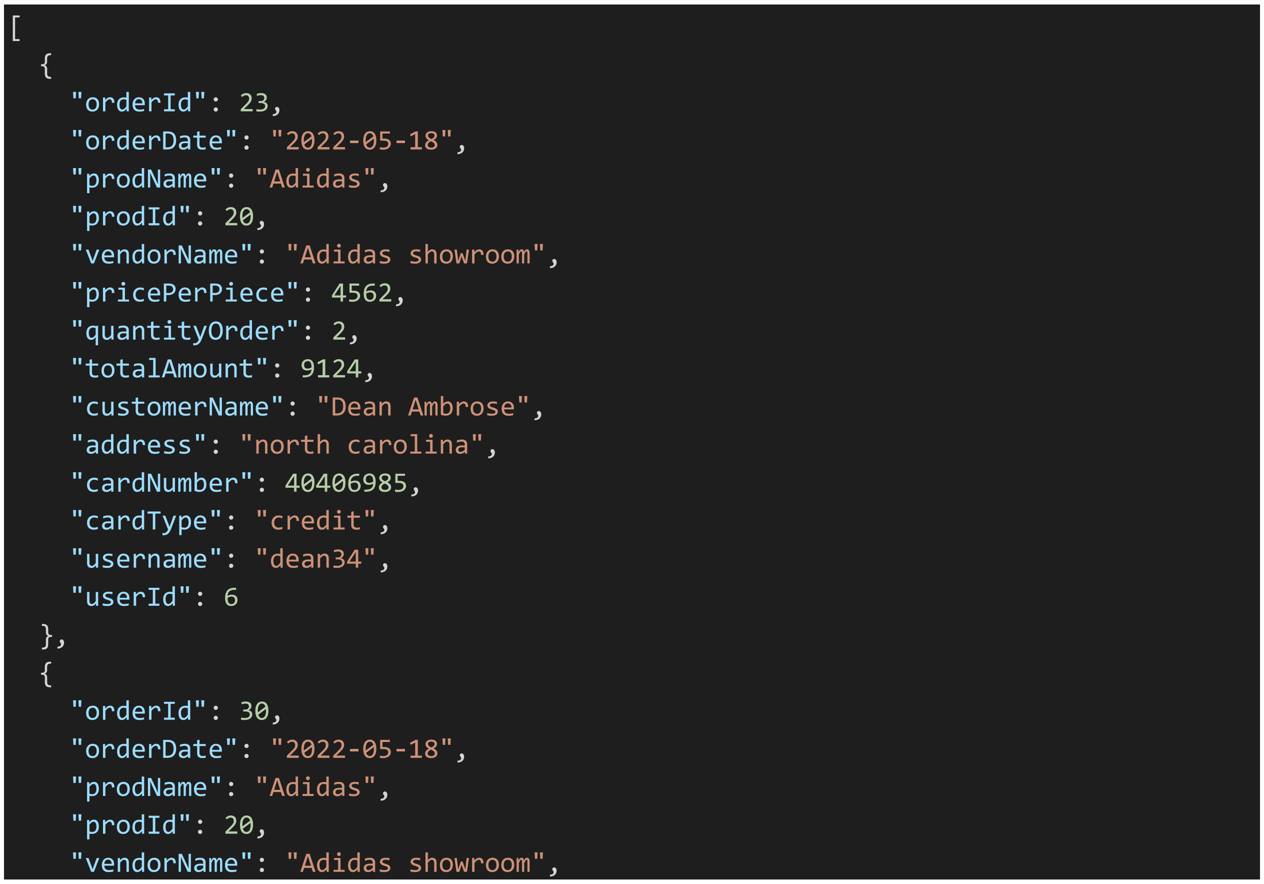


GET /admin/orderHistory/productId/{productId} orderHistoryByProdId

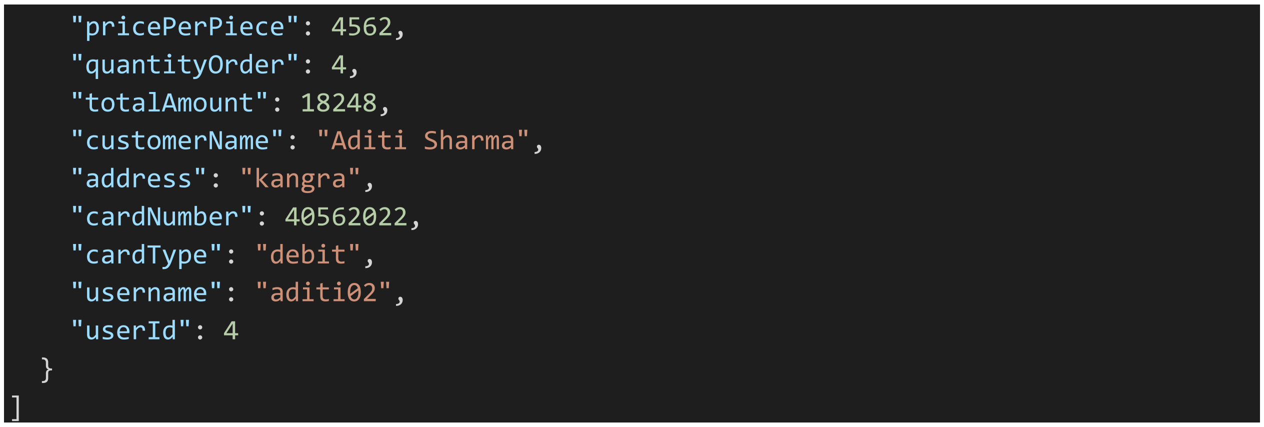
Parameters

Name	Description
productId * required integer($int32) *(path)*	productId 20

Picture 82- Execute /admin/orderHistory/productId/{productId}



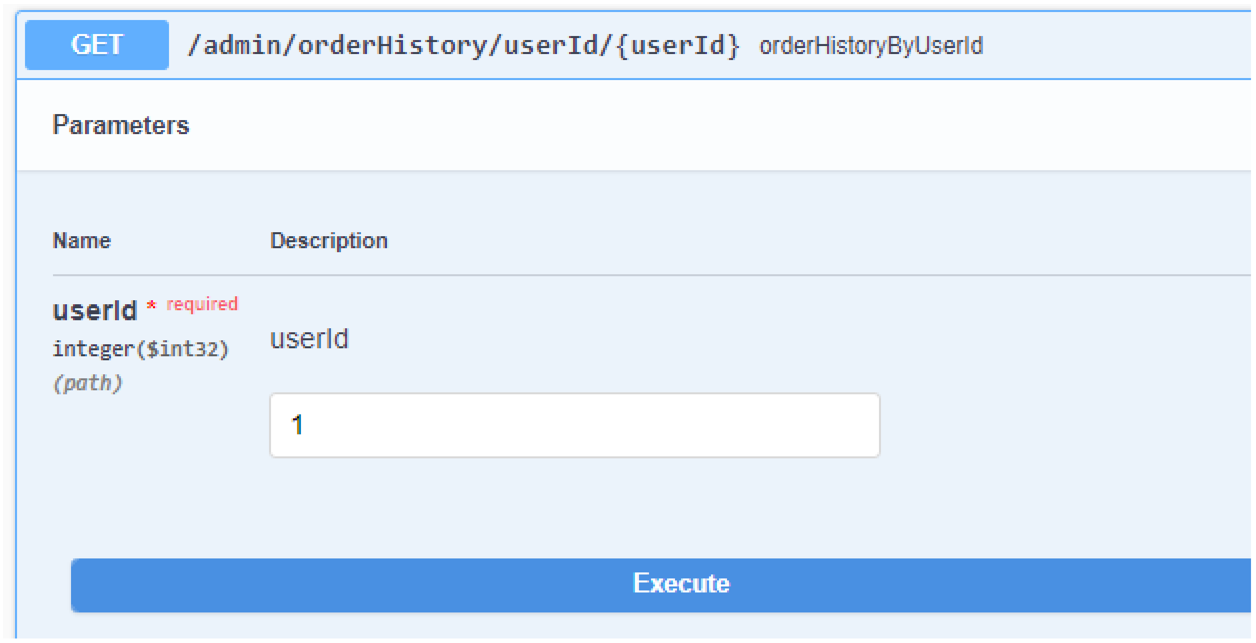
```
[
  {
    "orderId": 23,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
    "pricePerPiece": 4562,
    "quantityOrder": 2,
    "totalAmount": 9124,
    "customerName": "Dean Ambrose",
    "address": "north carolina",
    "cardNumber": 40406985,
    "cardType": "credit",
    "username": "dean34",
    "userId": 6
  },
  {
    "orderId": 30,
    "orderDate": "2022-05-18",
    "prodName": "Adidas",
    "prodId": 20,
    "vendorName": "Adidas showroom",
```



```
    "pricePerPiece": 4562,
    "quantityOrder": 4,
    "totalAmount": 18248,
    "customerName": "Aditi Sharma",
    "address": "kangra",
    "cardNumber": 40562022,
    "cardType": "debit",
    "username": "aditi02",
    "userId": 4
  }
]
```

Picture 83- Output of /admin/orderHistory/productId/20

If admin want to view orders for userId uid, the admin can go for **/admin/orderHistory/productId/{userId}** in swagger2.

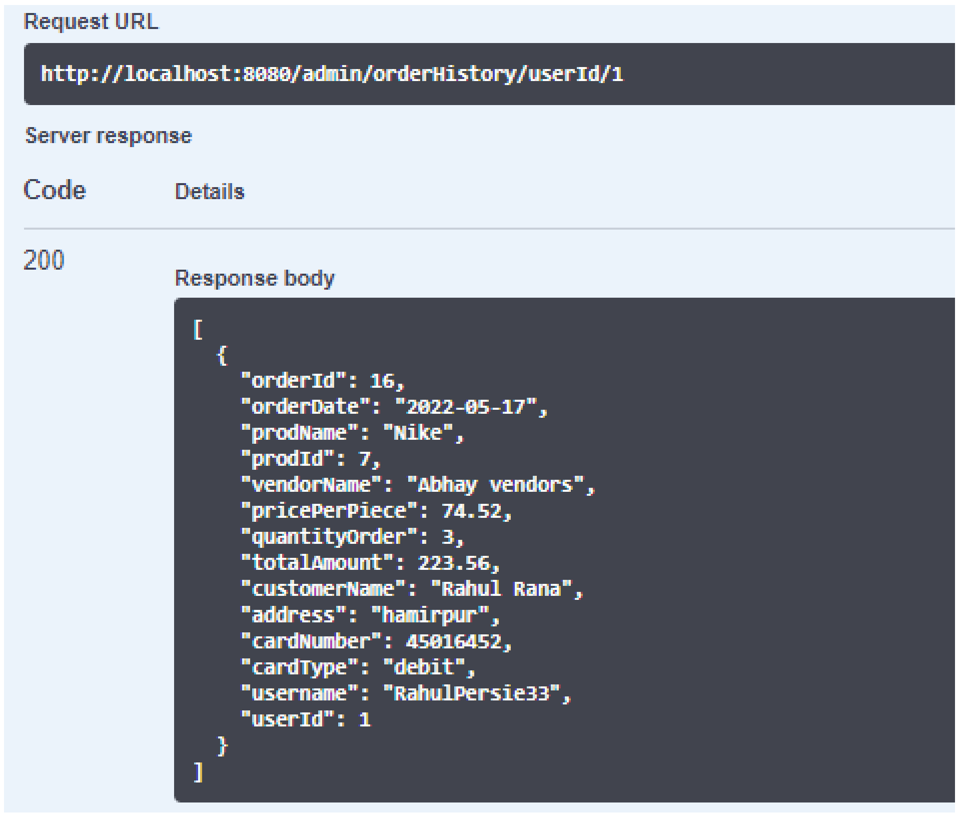


GET `/admin/orderHistory/userId/{userId}` orderHistoryByUserId

Parameters

Name	Description
userId * required integer($int32) *(path)*	userId 1

Picture 84- Execute /admin/orderHistory/userId/{userId}


Request URL

```
http://localhost:8080/admin/orderHistory/userId/1
```

Server response

Code Details

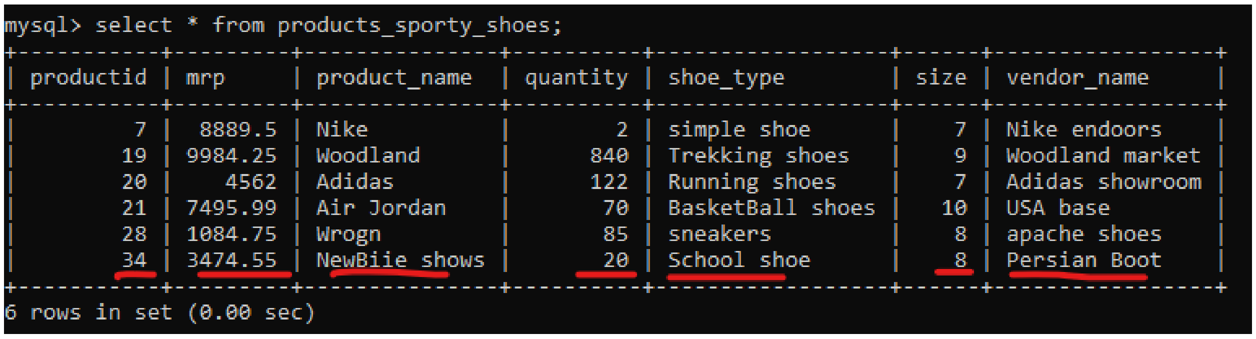
200

Response body

```
[
  {
    "orderId": 16,
    "orderDate": "2022-05-17",
    "prodName": "Nike",
    "prodId": 7,
    "vendorName": "Abhay vendors",
    "pricePerPiece": 74.52,
    "quantityOrder": 3,
    "totalAmount": 223.56,
    "customerName": "Rahul Rana",
    "address": "hamirpur",
    "cardNumber": 45016452,
    "cardType": "debit",
    "username": "RahulPersie33",
    "userId": 1
  }
]
```

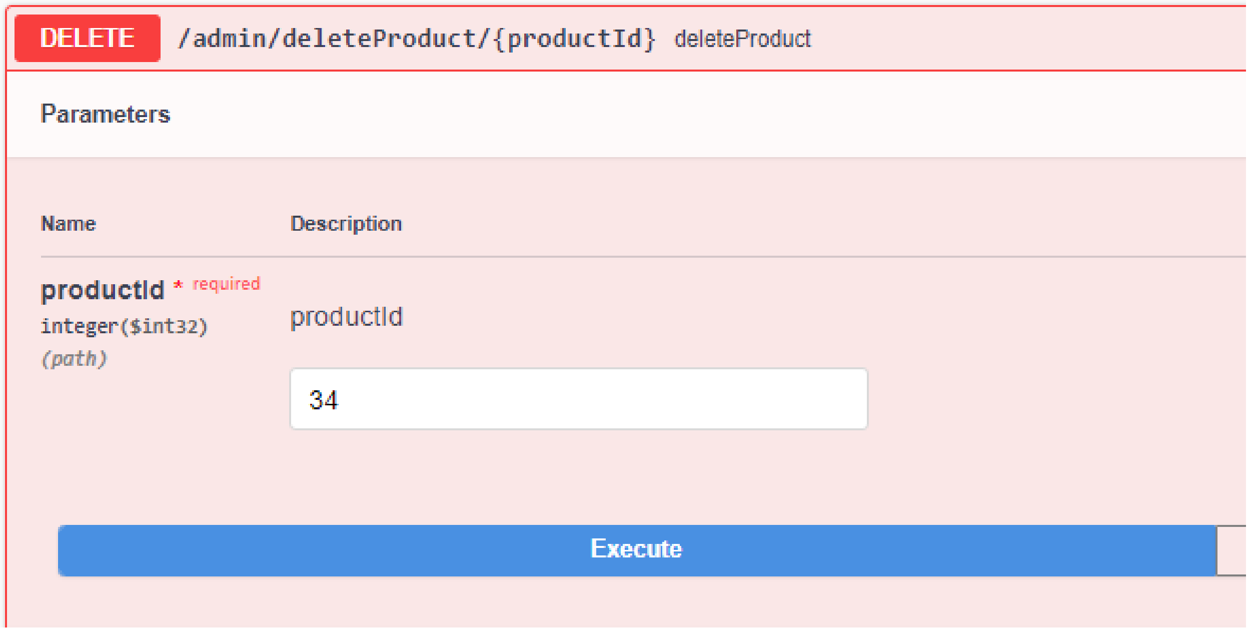
Picture 85- Output of /admin/orderHistory/userId/1

If admin want to delete a specific product admin can delete it using productId in swagger 2 option **/admin/deleteProduct/{productId}.**



```
mysql> select * from products_sporty_shoes;
+-----------+---------+---------------+----------+------------------+------+------------------+
| productid | mrp     | product_name  | quantity | shoe_type        | size | vendor_name      |
+-----------+---------+---------------+----------+------------------+------+------------------+
|         7 |  8889.5 | Nike          |        2 | simple shoe      |    7 | Nike endoors     |
|        19 | 9984.25 | Woodland      |      840 | Trekking shoes   |    9 | Woodland market  |
|        20 |    4562 | Adidas        |      122 | Running shoes    |    7 | Adidas showroom  |
|        21 | 7495.99 | Air Jordan    |       70 | BasketBall shoes |   10 | USA base         |
|        28 | 1084.75 | Wrogn         |       85 | sneakers         |    8 | apache shoes     |
|        34 | 3474.55 | NewBiie shows |       20 | School shoe      |    8 | Persian Boot     |
+-----------+---------+---------------+----------+------------------+------+------------------+
6 rows in set (0.00 sec)
```

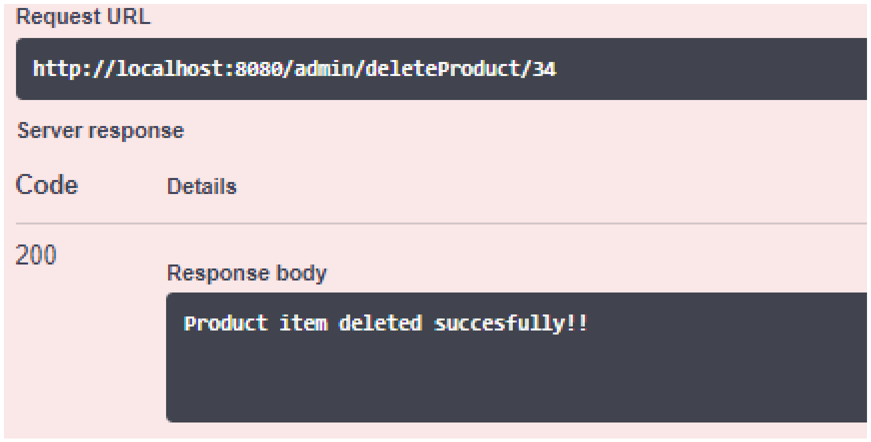
Picture 86-Product table before deletion



```
/admin/deleteProduct/{productId}
```

```
34
```

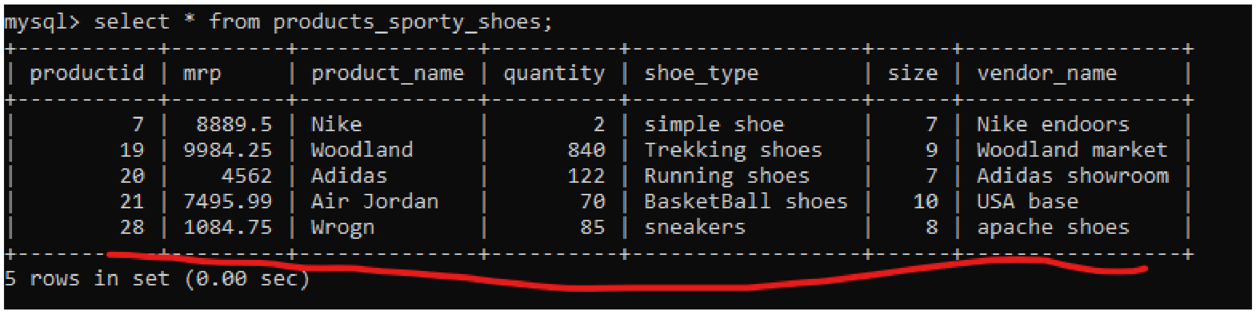
Picture 87- Execute /admin/deleteProduct/{productId}



```
http://localhost:8080/admin/deleteProduct/34
```

```
Product item deleted succesfully!!
```

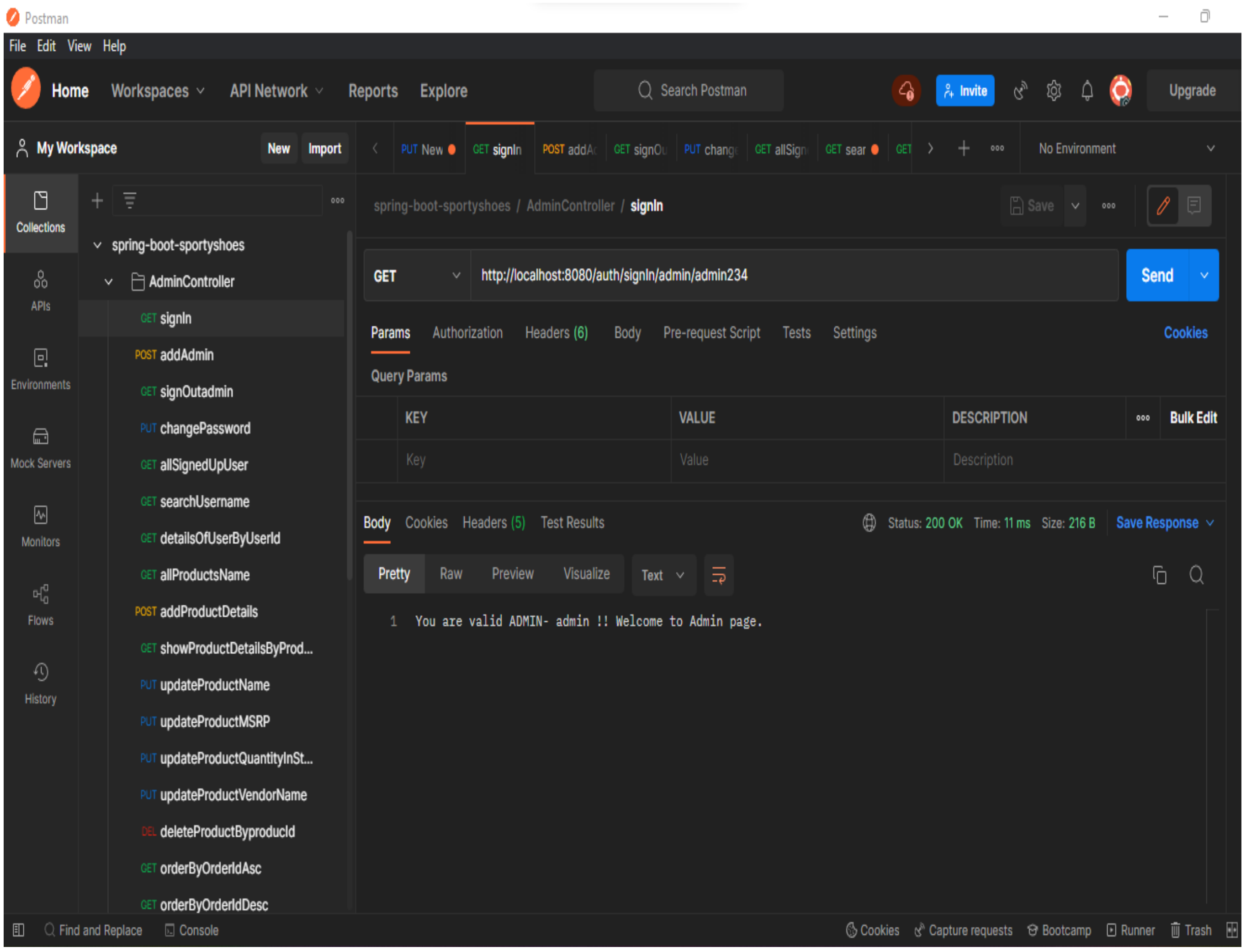
Picture 89- Output of /admin/deleteProduct/{productId}



```
mysql> select * from products_sporty_shoes;
+-----------+---------+--------------+----------+-----------------+------+-----------------+
| productid | mrp     | product_name | quantity | shoe_type       | size | vendor_name     |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
|         7 |  8889.5 | Nike         |        2 | simple shoe     |    7 | Nike endoors    |
|        19 | 9984.25 | Woodland     |      840 | Trekking shoes  |    9 | Woodland market |
|        20 |    4562 | Adidas       |      122 | Running shoes   |    7 | Adidas showroom |
|        21 | 7495.99 | Air Jordan   |       70 | BasketBall shoes|   10 | USA base        |
|        28 | 1084.75 | Wrogn        |       85 | sneakers        |    8 | apache shoes    |
+-----------+---------+--------------+----------+-----------------+------+-----------------+
5 rows in set (0.00 sec)
```

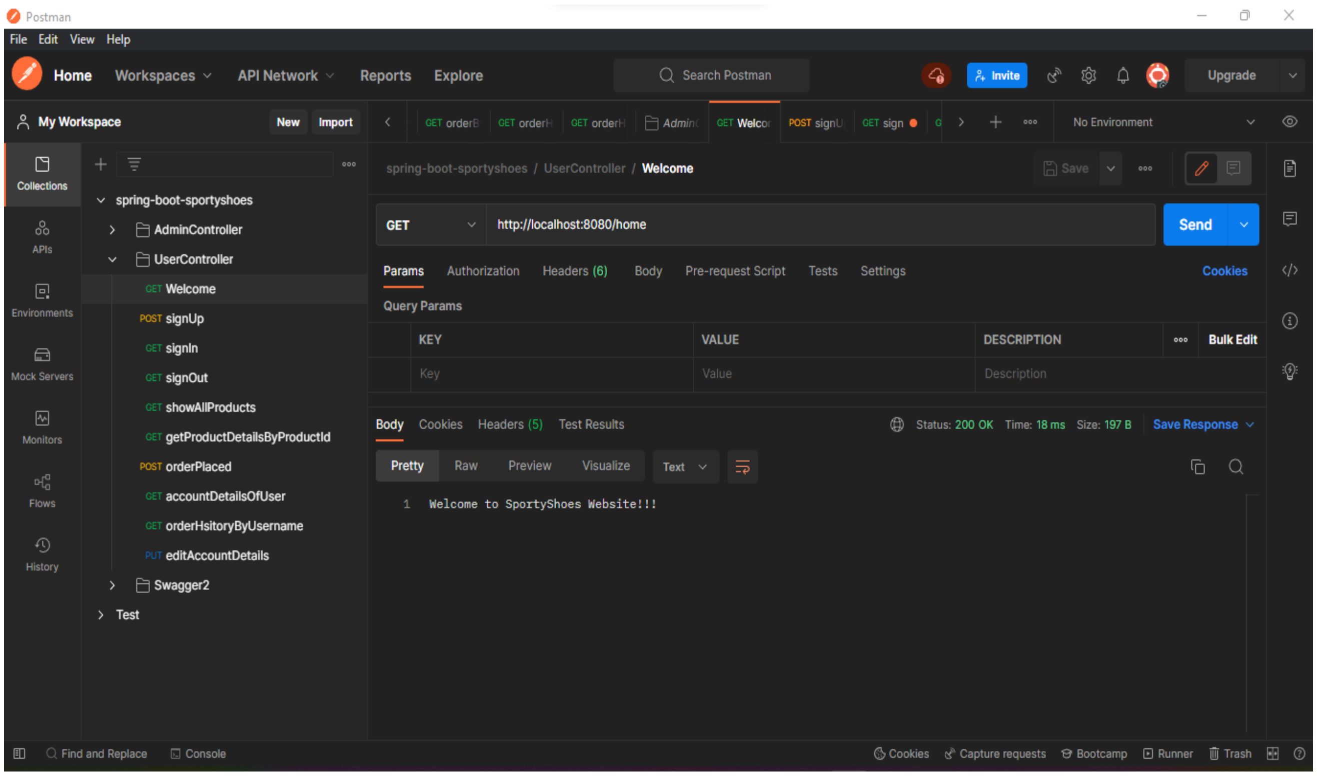
Picture 90- delete the Pid-34 row from table Product is deleted

All these function of admin and user controller can be done via postman by providing them a url for each function.


Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite Upgrade
My Workspace New Import
No Environment
Collections APIs Environments Mock Servers Monitors Flows History
spring-boot-sportyshoes
AdminController
GET signIn
POST addAdmin
GET signOutadmin
PUT changePassword
GET allSignedUpUser
GET searchUsername
GET detailsOfUserByUserId
GET allProductsName
POST addProductDetails
GET showProductDetailsByProd...
PUT updateProductName
PUT updateProductMSRP
PUT updateProductQuantityInSt...
PUT updateProductVendorName
DEL deleteProductByproducId
GET orderByOrderIdAsc
GET orderByOrderIdDesc
spring-boot-sportyshoes / AdminController / signIn
Save
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 11 ms Size: 216 B Save Response
Pretty Raw Preview Visualize Text
Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

GET http://localhost:8080/auth/signIn/admin/admin234

```
You are valid ADMIN- admin !! Welcome to Admin page.
```



```
GET http://localhost:8080/home
```

Status: 200 OK Time: 18 ms Size: 197 B

```
Welcome to SportyShoes Website!!!
```

Picture 91 – Postman view of all functions in one screenshot