Project Setup and API Documentation

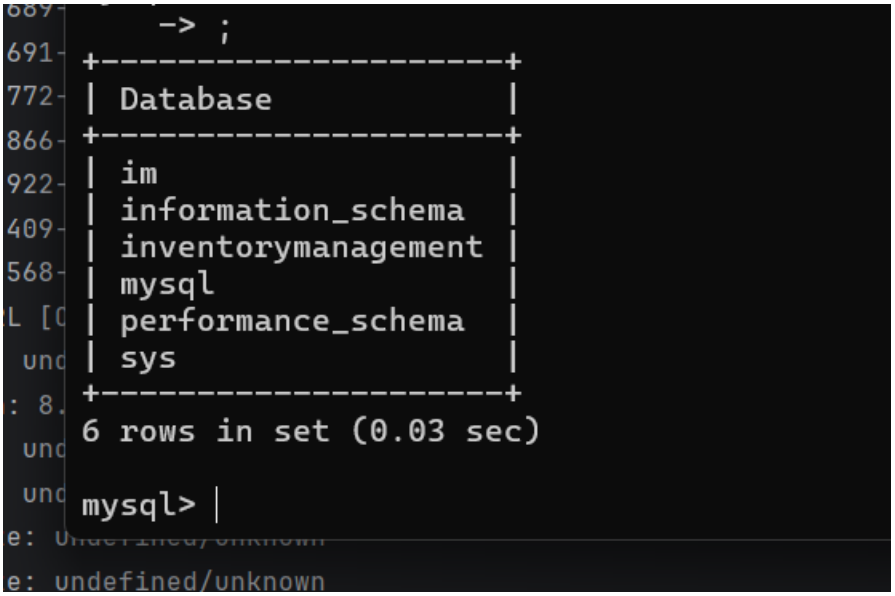
Maven Dependencies that were used.

```
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <version>1.18.36</version>
            <scope>provided</scope>
        </dependency>
```

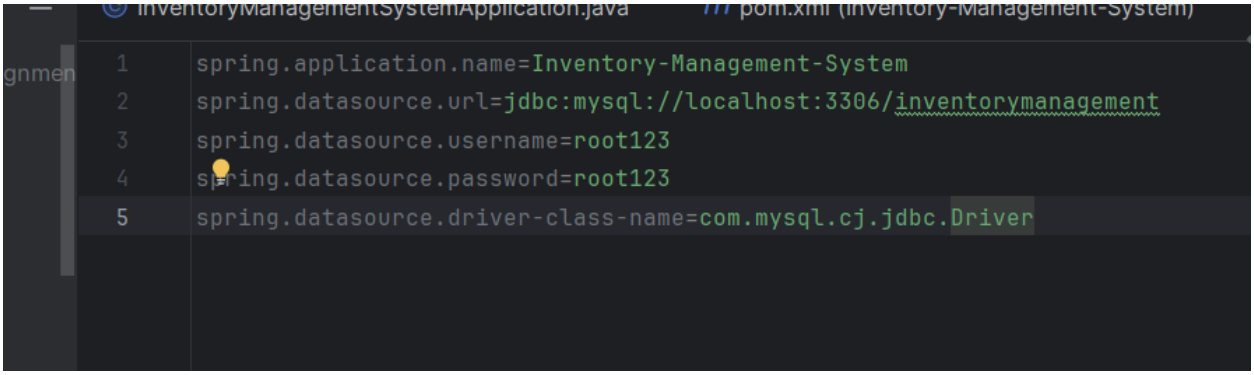
```
59          <dependency>
60              <groupId>org.springframework.boot</groupId>
61              <artifactId>spring-boot-starter-validation</artifactId>
62          </dependency>
63      </dependencies>
64
65      <build>
66          <plugins>
67              <plugin>
68                  <groupId>org.springframework.boot</groupId>
69                  <artifactId>spring-boot-maven-plugin</artifactId>
70              </plugin>
71          </plugins>
72      </build>
73
74  </project>
```

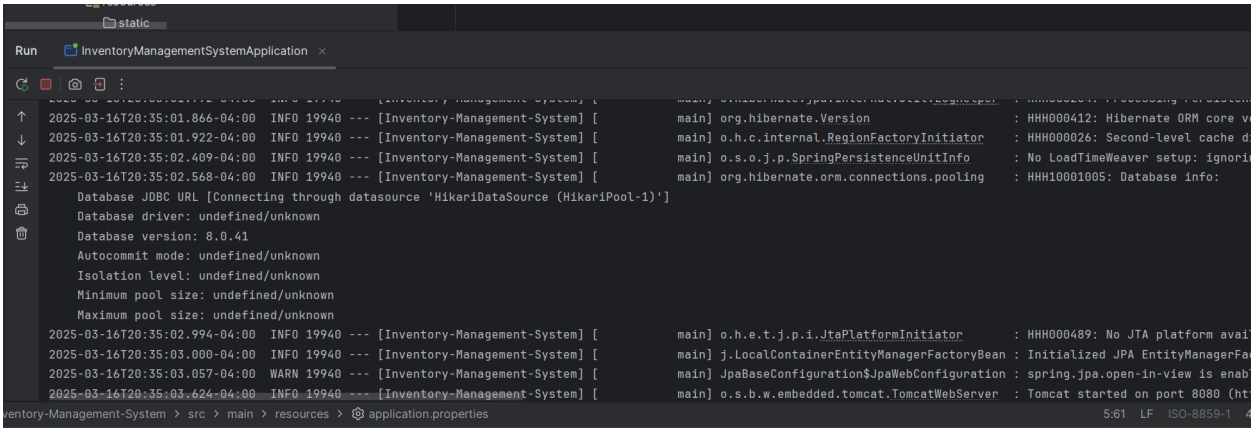
MySQL Workbench started on localhost:3306

Created database with name inventory management



Application.properites Configuration





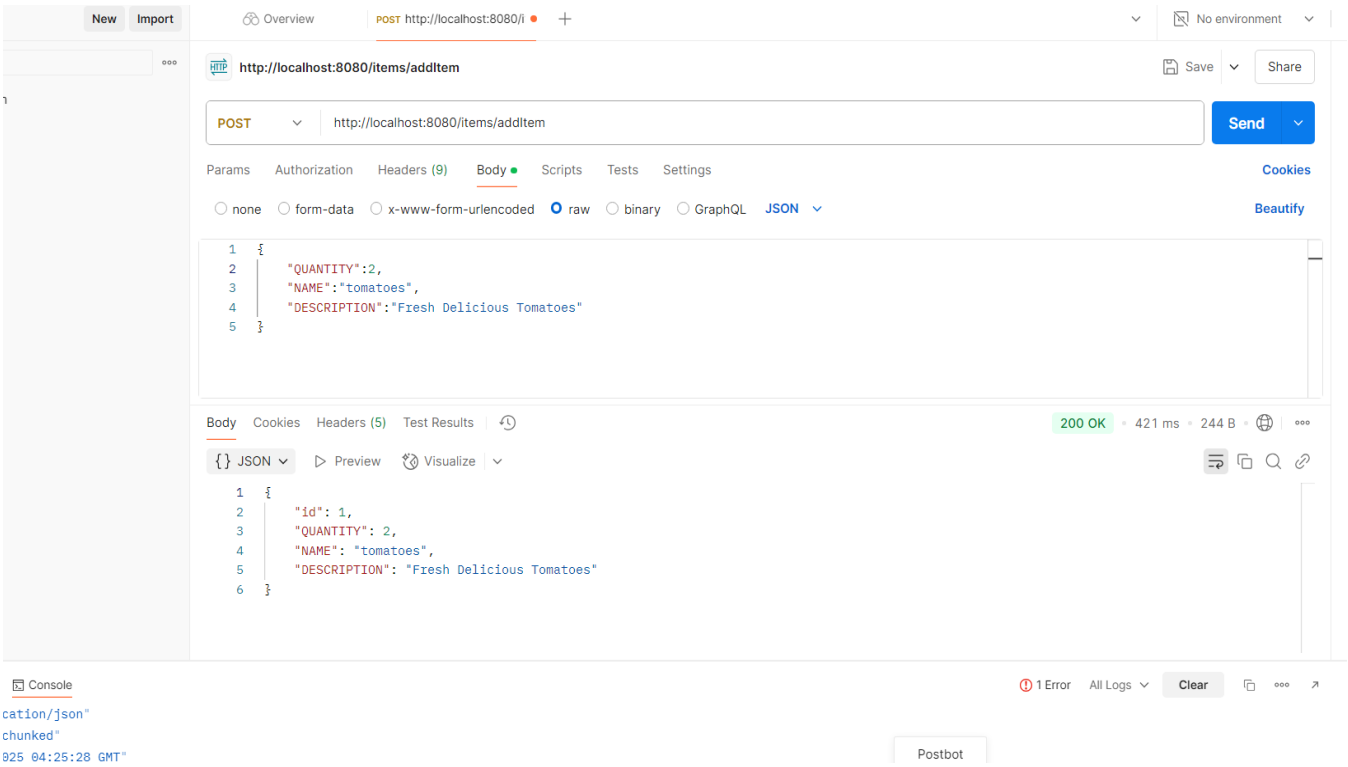
Controller Class mappings

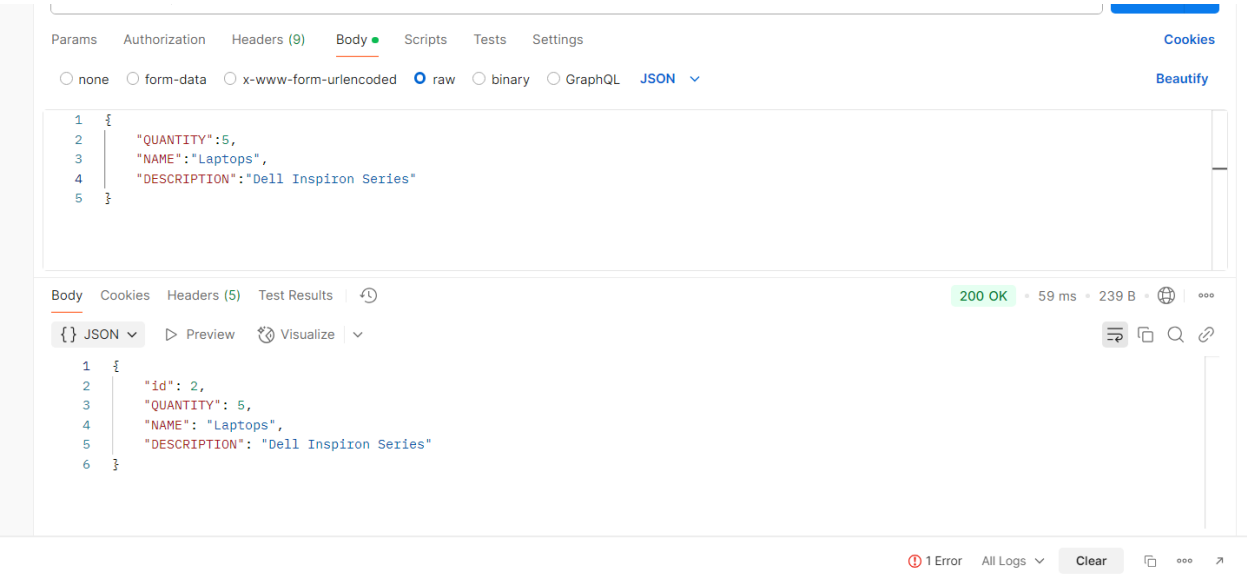
```java
@RestController  no usages
@RequestMapping("/items")
@Validated
public class ItemController {

    @Autowired  7 usages
    public ItemService itemService;

    @GetMapping("/getItems")  no usages
   List<Item> getAllItems(){
        return itemService.getallItems();
   }

   @GetMapping("/getItem/{id}")  no usages
    public Optional<Item> getItemById(@PathVariable int id) {
        return itemService.getItembyID(id);
   }

   @PostMapping("/addItem")  no usages
    public Item saveitem(@Valid @RequestBody Item item) {
        return itemService.saveItem(item);
   }

   @PutMapping("/updateItemQuantity")  no usages
    public Item updateitem(@Valid @RequestBody Item item) {
        return itemService.updateItemQuantity(item);
   }
```

```java
@PutMapping("/udateItemById/{id}")  no usages
 public Item updatewholeItem(@PathVariable int id, @Valid @RequestBody Item item) {
     return itemService.updateItementity(id,item);
}

@DeleteMapping("/deleteById/{id}")  no usages
 public void deleteItembyid(@PathVariable int id) {
     itemService.deleteItem(id);
}

@DeleteMapping("/deleteAllItems")  no usages
 public void deleteallitems() {
     itemService.deleteAll();
}
```

API Endpoints

POST url **/addItem**

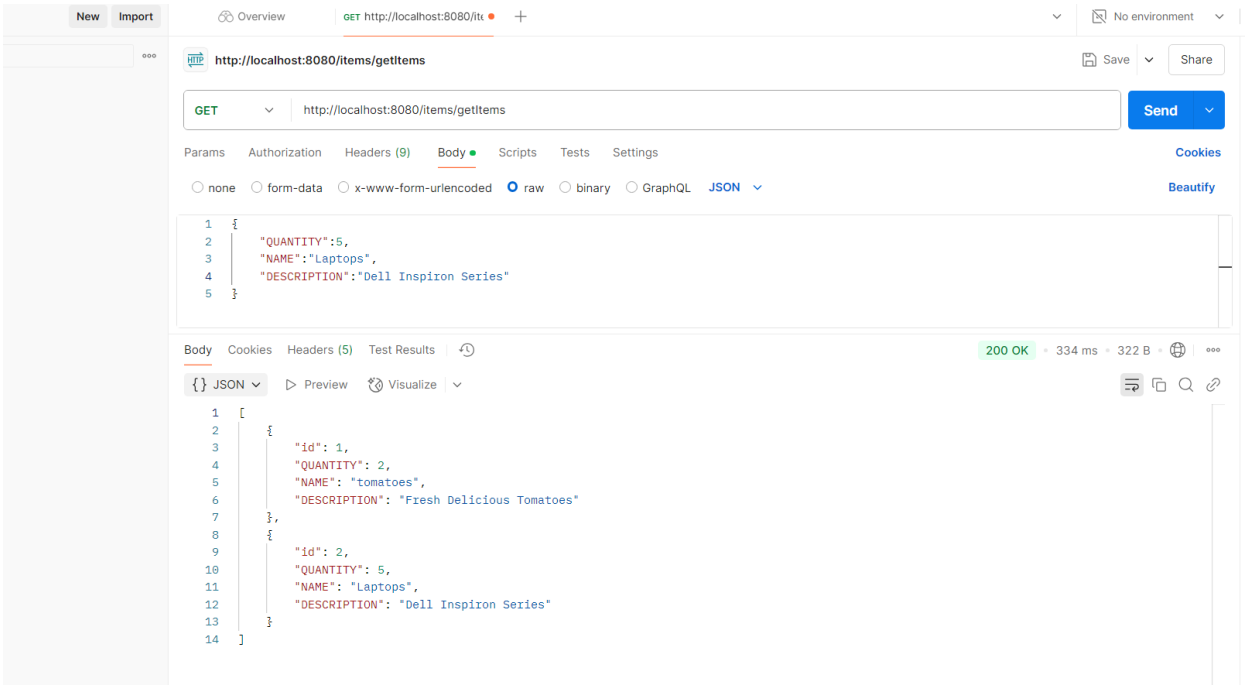




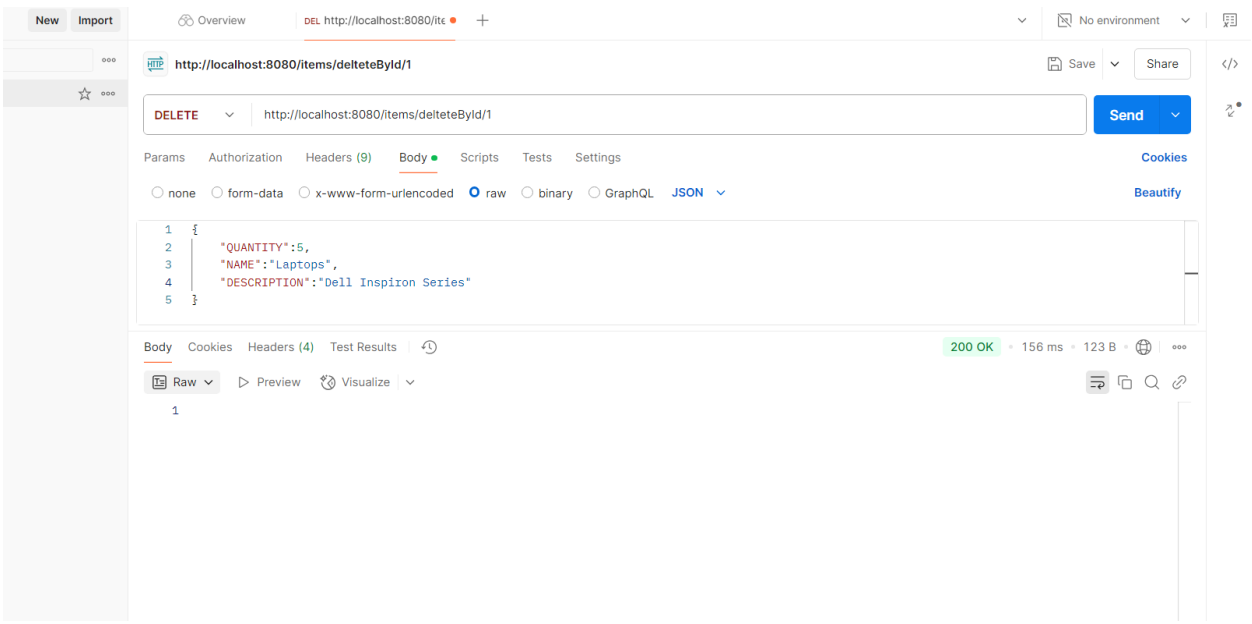
Data Loaded in MYSQL From POST call

```
mysql> select * from item
    -> ;
+----+----------+--------------------------+----------+
| id | quantity | description              | name     |
+----+----------+--------------------------+----------+
|  1 |        2 | Fresh Delicious Tomatoes | tomatoes |
|  2 |        5 | Dell Inspiron Series     | Laptops  |
+----+----------+--------------------------+----------+
2 rows in set (0.01 sec)

mysql>
```

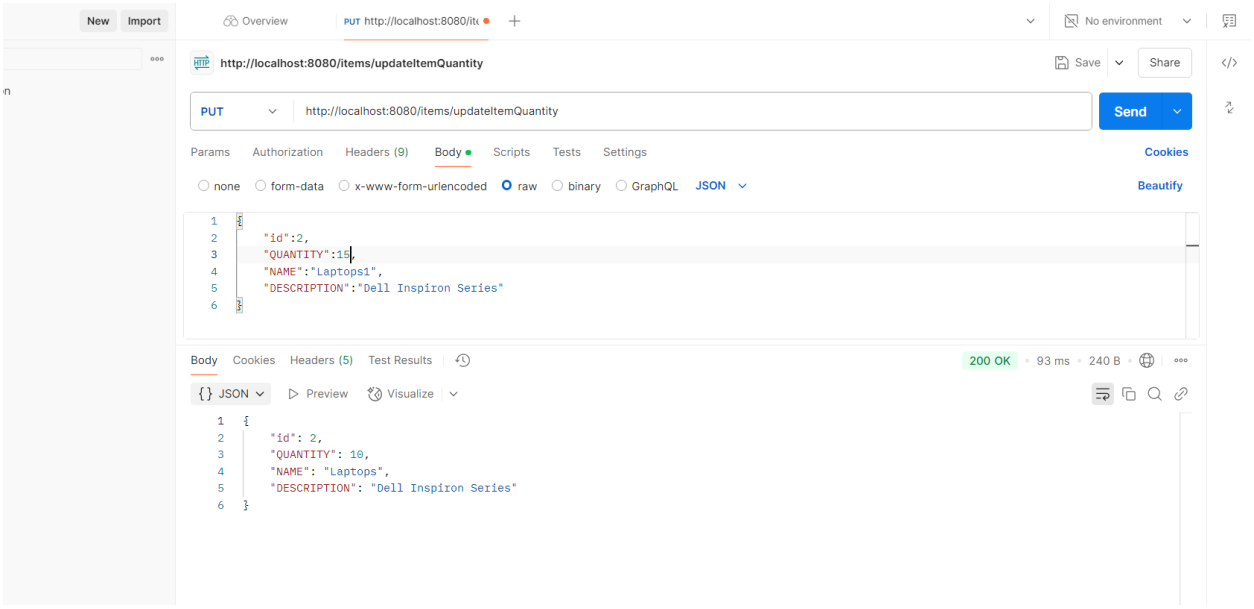
GET API endpoint – to get all the items /getItems



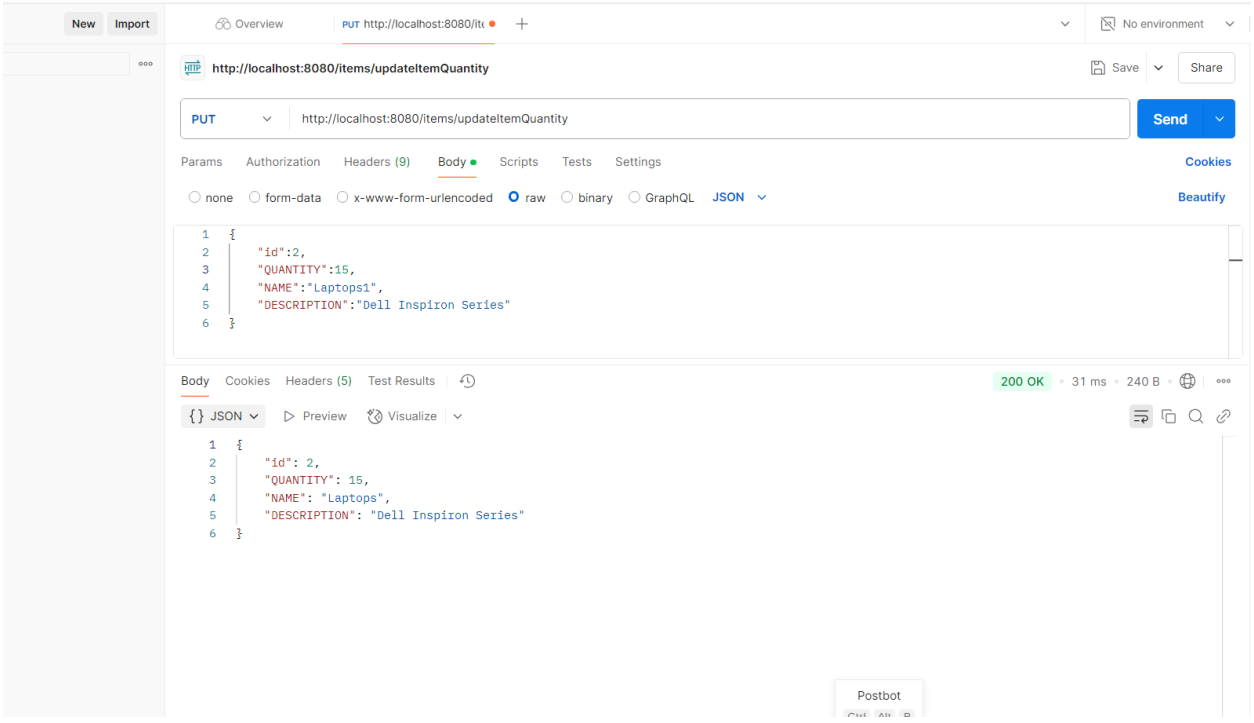
DELETE API endpoint – Delete by ID



PUT API endpoint – update only the Item Quantity here ... /updateItemQuantity



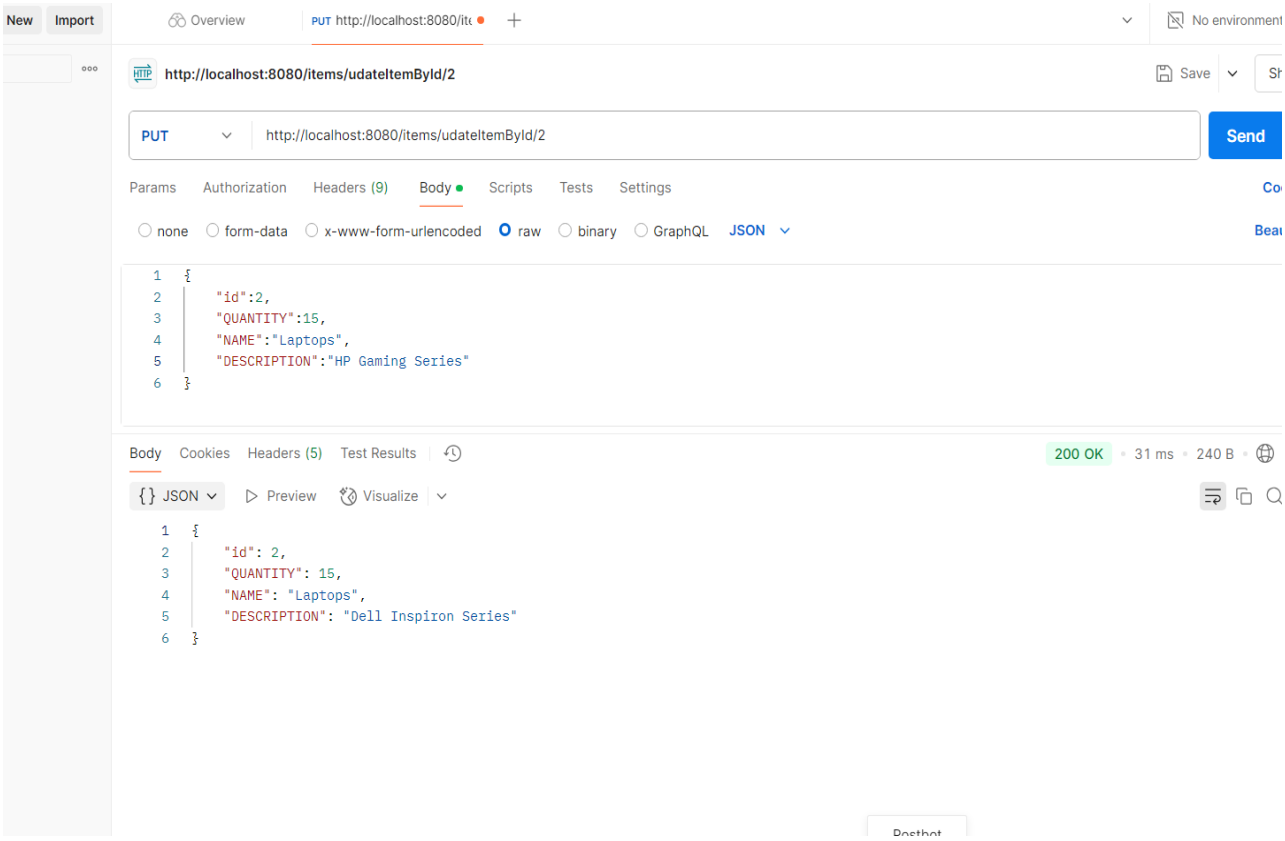
Response Returned, only the quantity Is updated



MYSQL Update data

```
mysql> select * from item;
+----+----------+----------------------+---------+
| id | quantity | description          | name    |
+----+----------+----------------------+---------+
|  2 |       15 | Dell Inspiron Series | Laptops |
+----+----------+----------------------+---------+
1 row in set (0.00 sec)

mysql>
```

PUT mapping Update, update the entire Record or data - /updateItemById/{id}



http://localhost:8080/items/udateItemById/2

PUT http://localhost:8080/items/udateItemById/2

Params Authorization Headers (9) Body Scripts Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "id":2,
    "QUANTITY":15,
    "NAME":"Laptops",
    "DESCRIPTION":"HP Gaming Series"
}
```

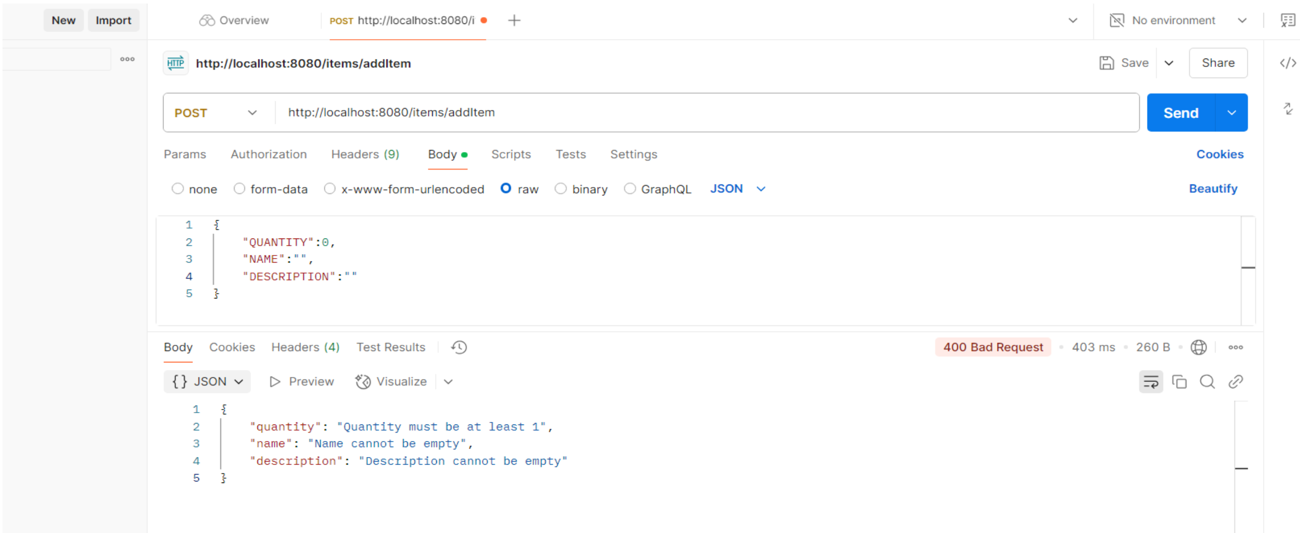
Body Cookies Headers (5) Test Results

{} JSON Preview Visualize

```
{
    "id": 2,
    "QUANTITY": 15,
    "NAME": "Laptops",
    "DESCRIPTION": "HP Gaming Series"
}
```

```
mysql> select * from item;
+----+----------+------------------+---------+
| id | quantity | description      | name    |
+----+----------+------------------+---------+
|  2 |       15 | HP Gaming Series | Laptops |
+----+----------+------------------+---------+
1 row in set (0.01 sec)

mysql>
```

Validating the Errors for Empty POST request



Number of characters can be at most 50