

Great Learning

Project Screenshots

Contains the screenshots of the Graded Coding Assignment 6

Shubham Kumar Shukla

Table of Contents

Problem Statement	2
SCREENSHOTS.....	3
ADMIN Account.....	3
1. Adding role.....	3
2. Adding User which can be used for authentication purposes.....	4
3. List of details of all Employees.....	4
4. Get details of an Employee based upon its ID	5
5. Adding details of an Employee	5
6. Updating an Employee's Detail.....	6
7. Deleting an Employee's Detail	6
8. Searching Employee based upon the First Name	7
9. List of details of all Employees in Ascending Order	7
10. List of details of all Employees in Descending Order	8
USER Account	9
1. List of details of all Employees.....	9
2. Get details of an Employee based upon its ID	9
3. Adding details of an Employee	10
4. Updating an Employee's Detail.....	10
5. Deleting an Employee's Detail	11
6. Searching Employee based upon the First Name	11
7. List of details of all Employees in Ascending Order	12
8. List of details of all Employees in Descending Order	12

Employee Management

Problem Statement

Create a Employee Management Rest Api based Web application, where you will be developing CRUD(Create,Read,Update and Delete) functionality along with Sorting and some concepts of security.

Also, the application should be able to add roles in the database dynamically and should be able to add Users in the db which can be used for authentication purposes.

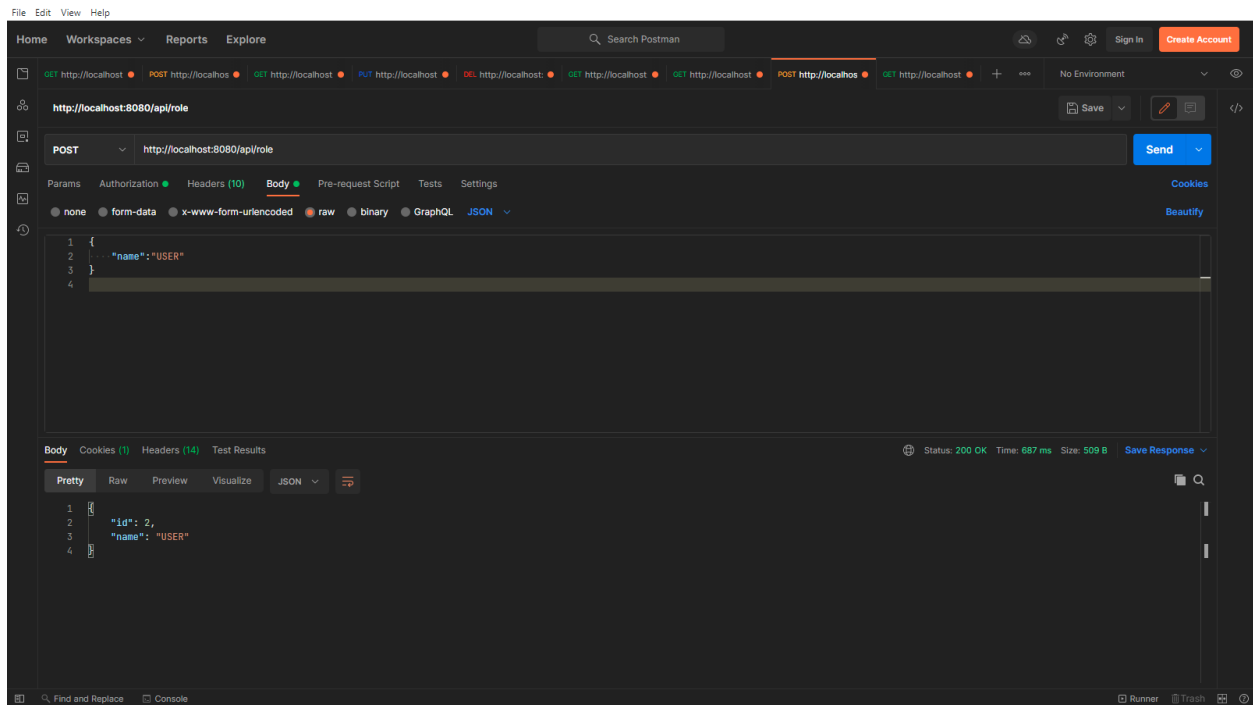
NOTE

A screen recorded demonstration of all required operations is available on the README render.

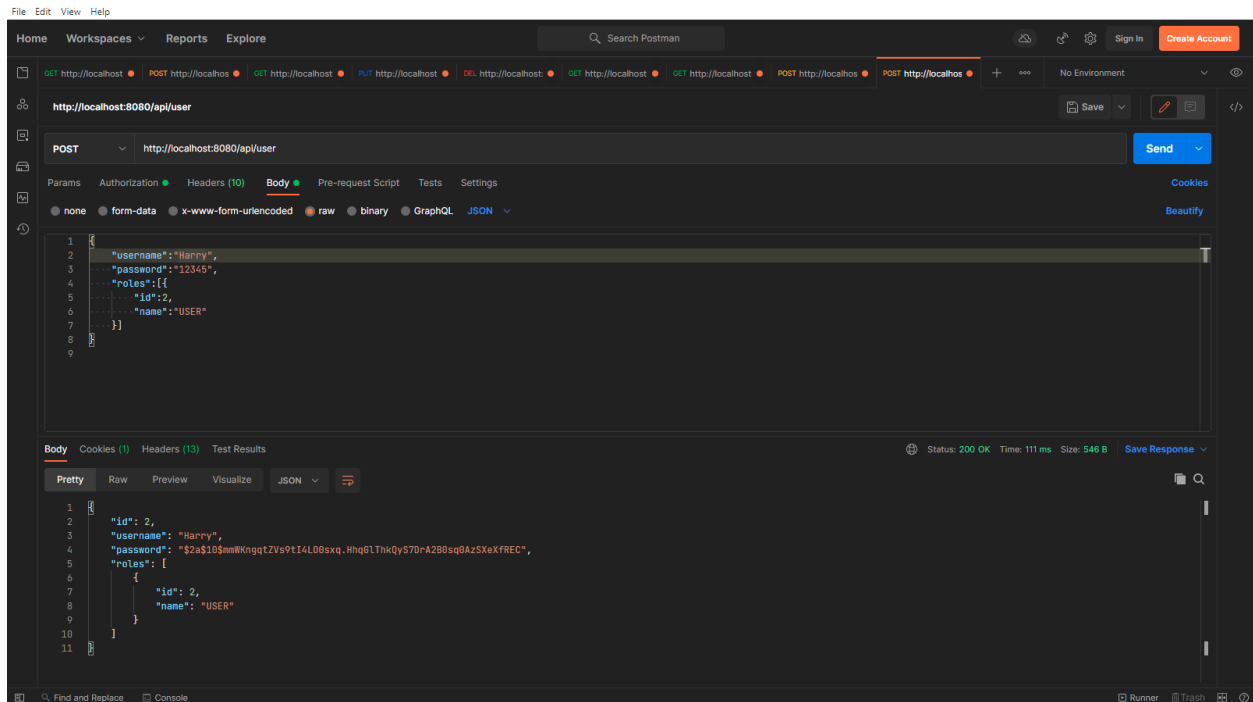
SCREENSHOTS

ADMIN Account

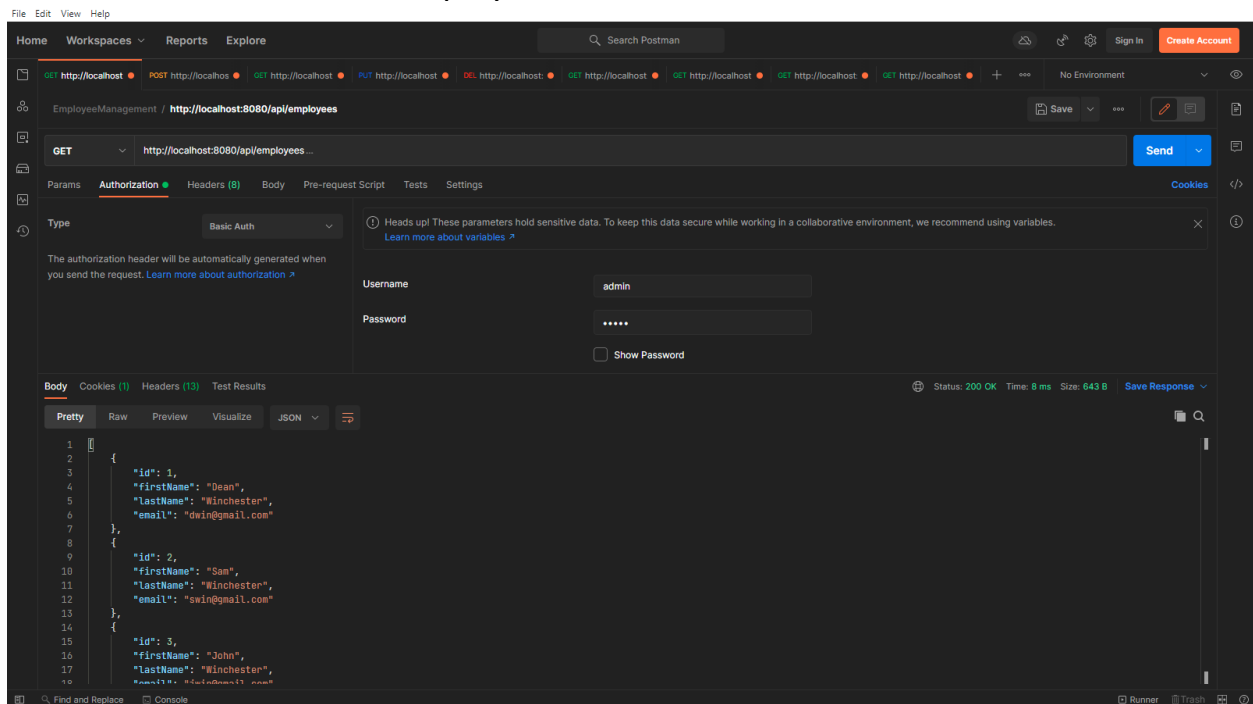
1. Adding role



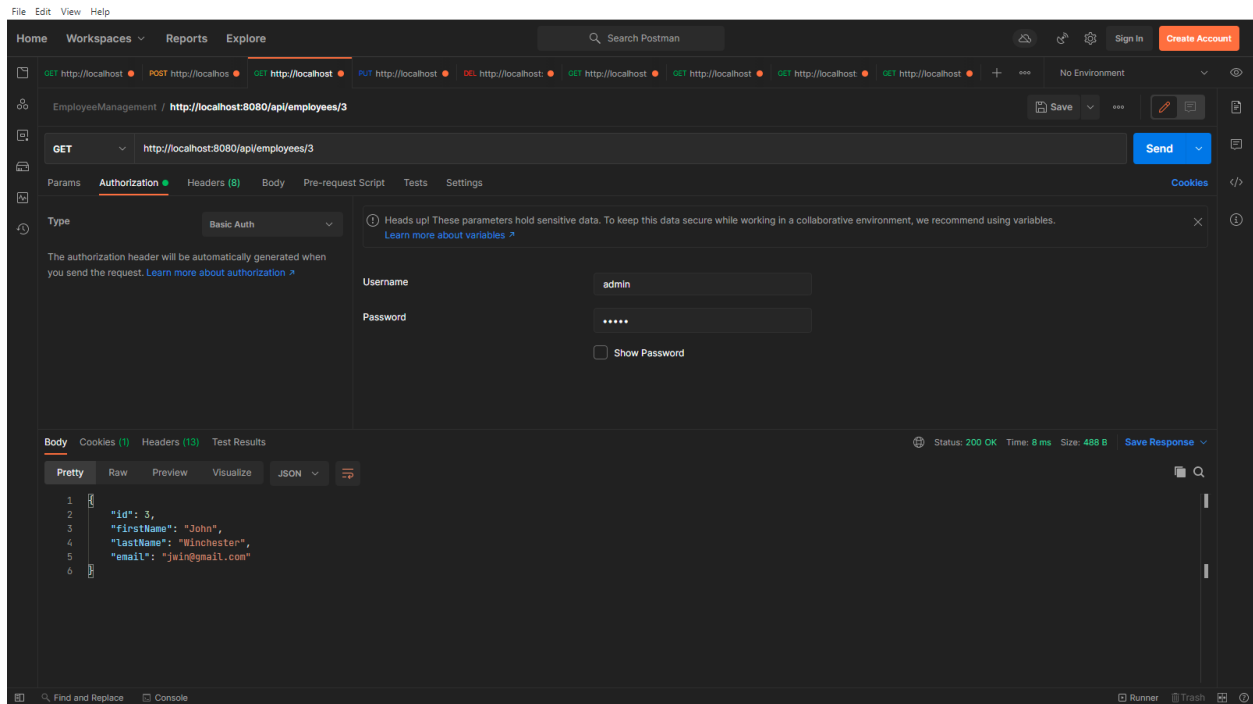
2. Adding User which can be used for authentication purposes



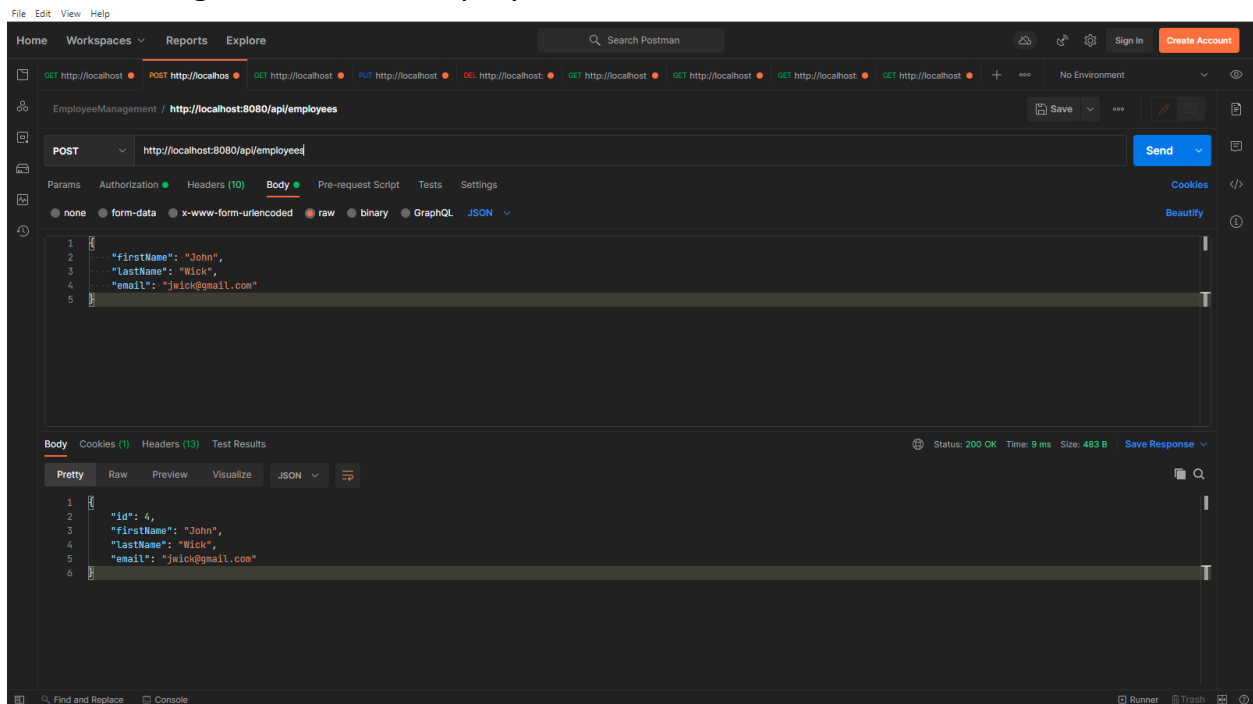
3. List of details of all Employees



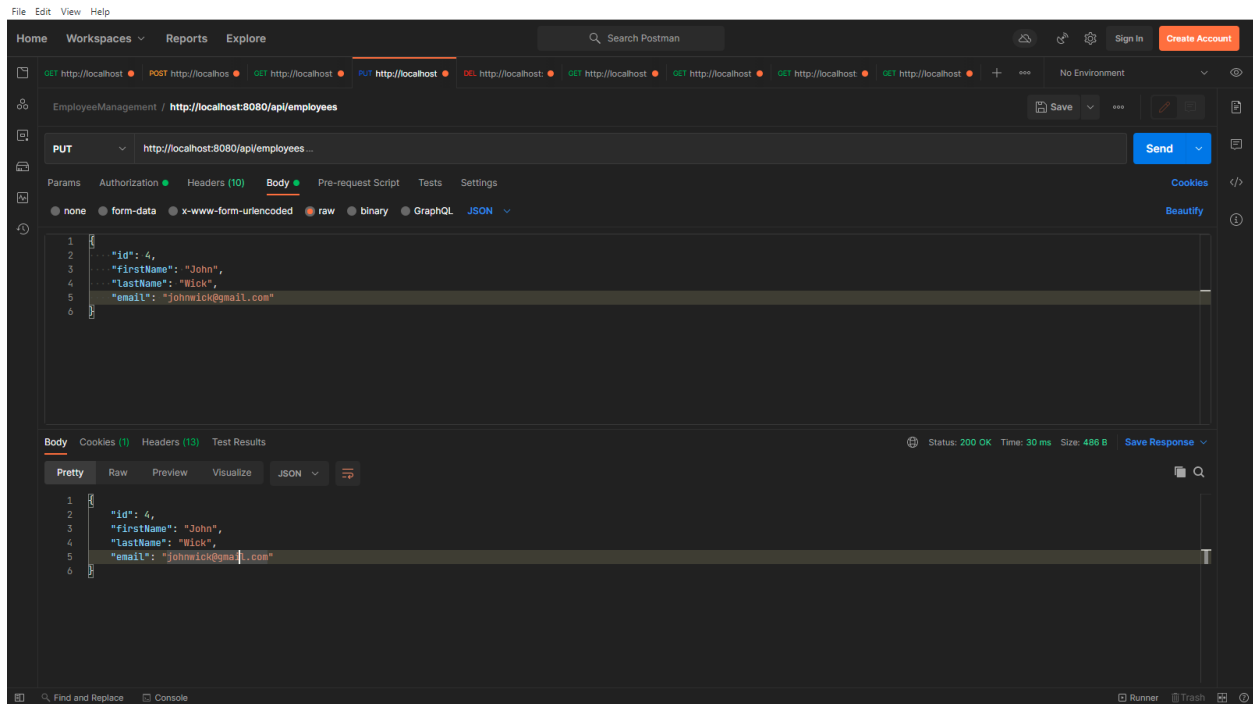
4. Get details of an Employee based upon its ID



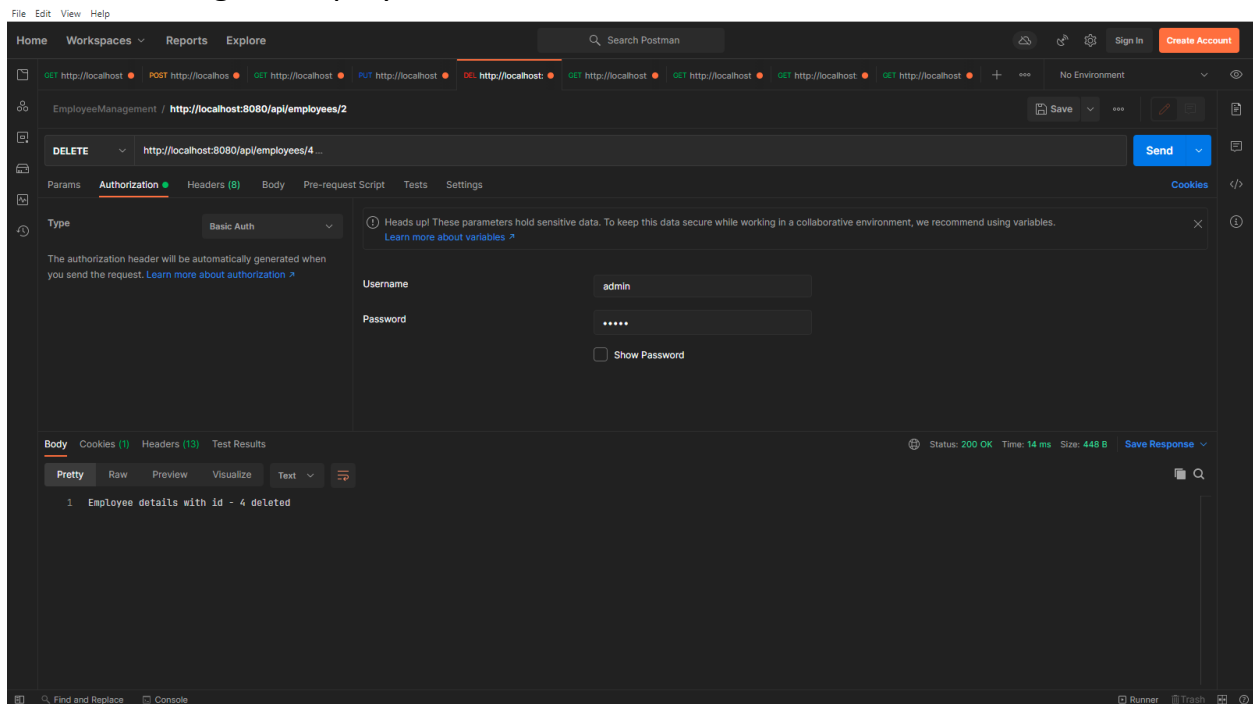
5. Adding details of an Employee



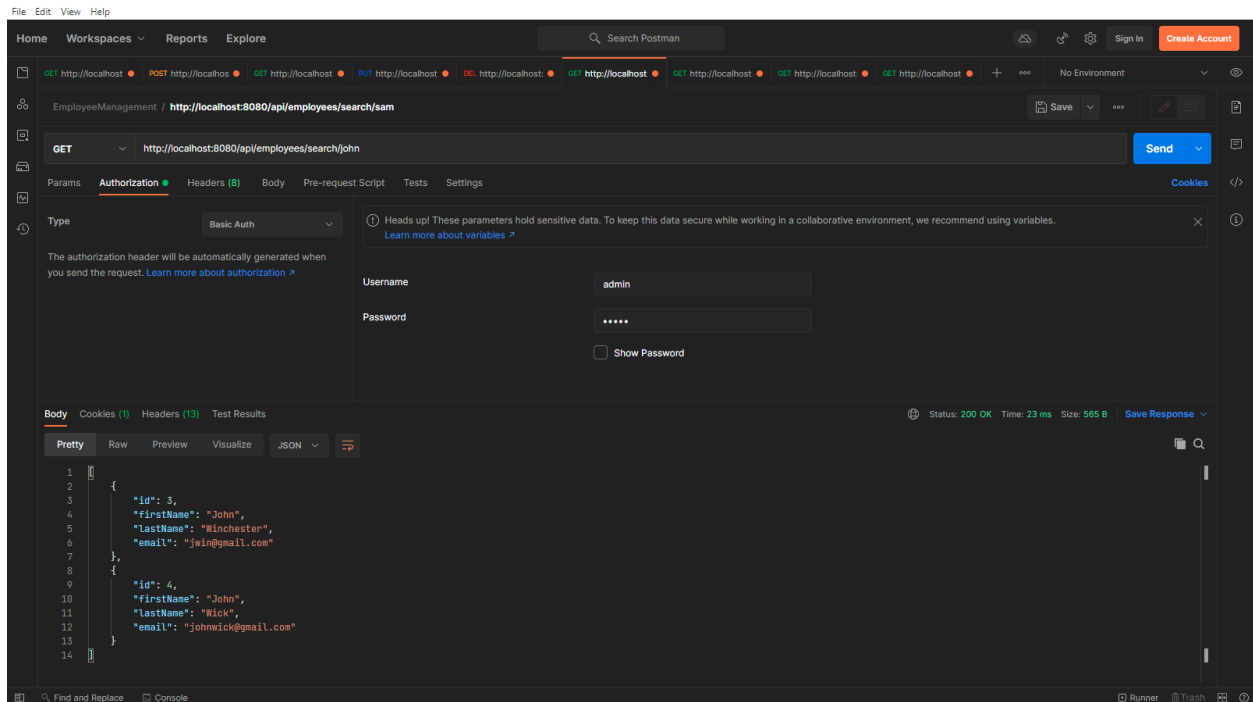
6. Updating an Employee's Detail



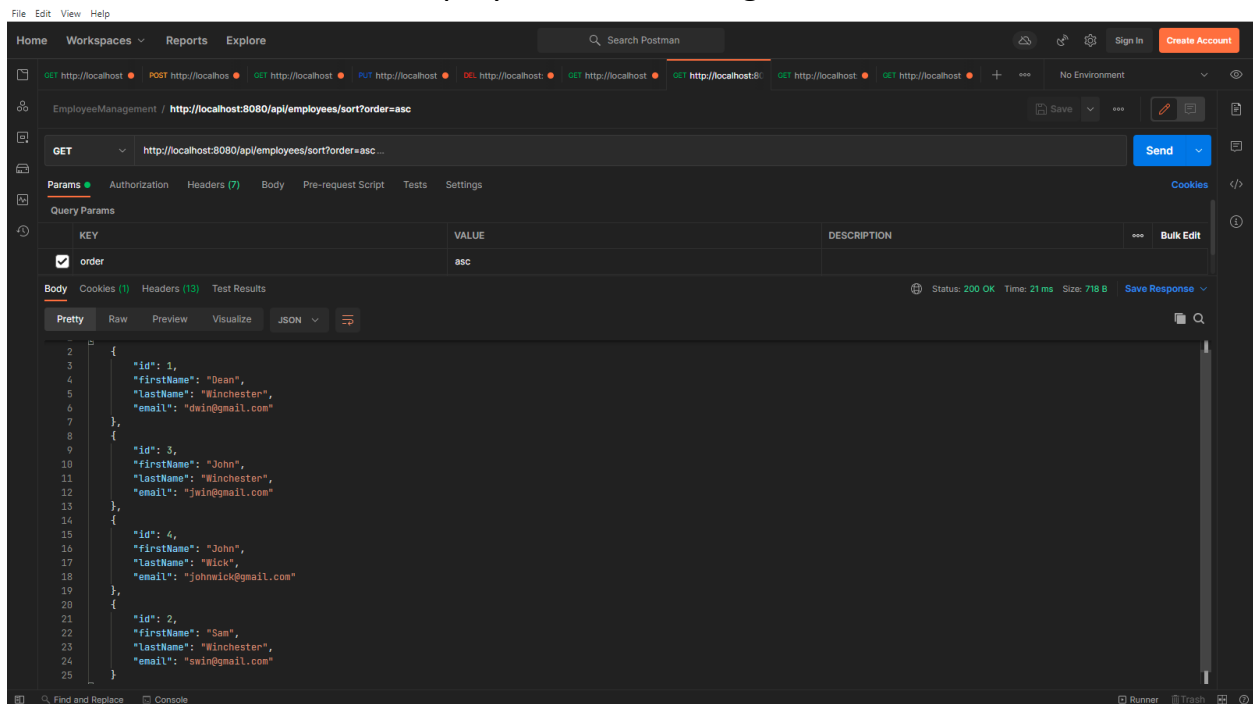
7. Deleting an Employee's Detail



8. Searching Employee based upon the First Name



9. List of details of all Employees in Ascending Order



10. List of details of all Employees in Descending Order

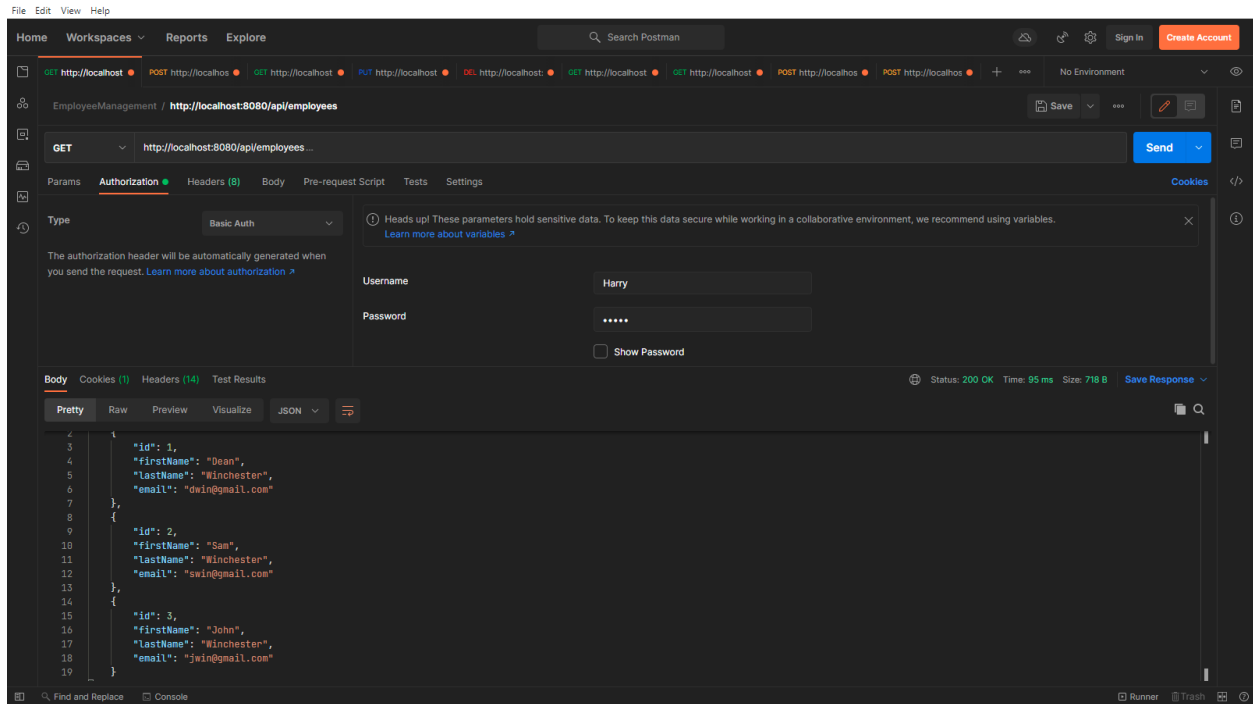
The screenshot shows the Postman interface with a GET request to `http://localhost:8080/api/employees/sort?order=desc`. The response is a JSON array of employee details, sorted by id in descending order. The response status is 200 OK, with a time of 9 ms and a size of 718 B.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> order	desc	

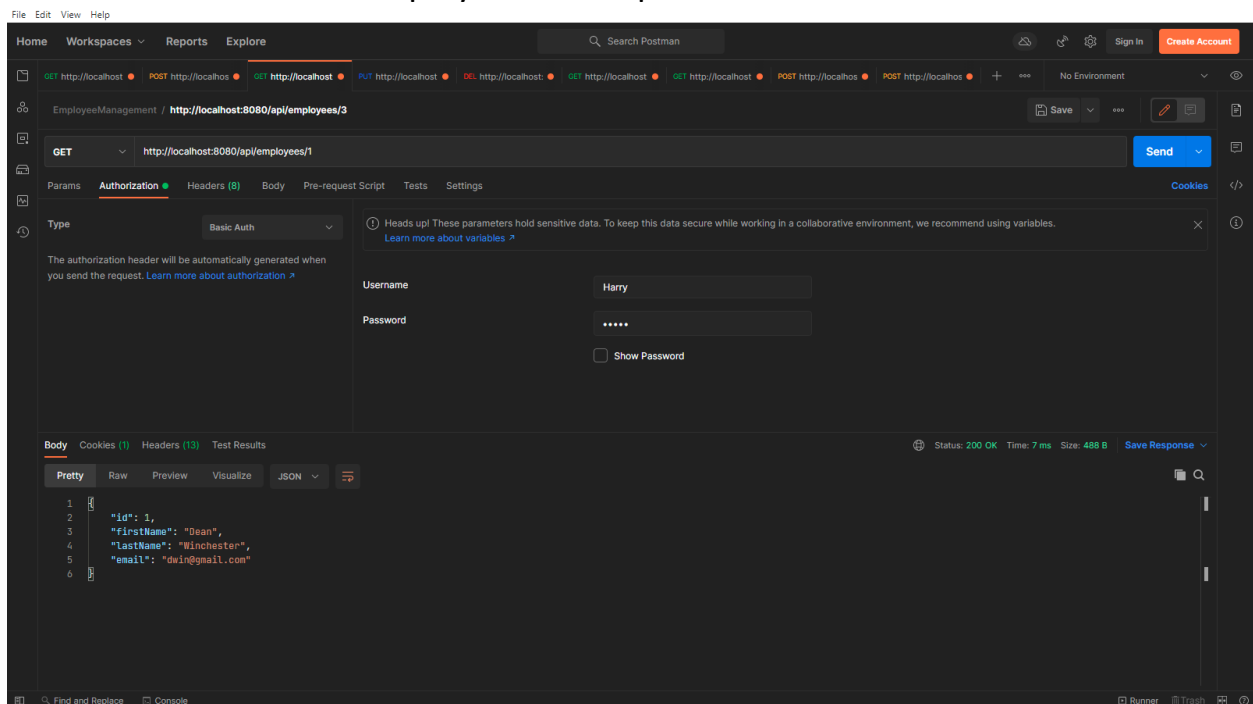
```
{
  "id": 2,
  "firstName": "Sam",
  "lastName": "Winchester",
  "email": "swin@gmail.com"
},
{
  "id": 3,
  "firstName": "John",
  "lastName": "Winchester",
  "email": "jwin@gmail.com"
},
{
  "id": 4,
  "firstName": "John",
  "lastName": "Wick",
  "email": "johnwick@gmail.com"
},
{
  "id": 1,
  "firstName": "Dean",
  "lastName": "Winchester",
  "email": "dwin@gmail.com"
}
```

USER Account

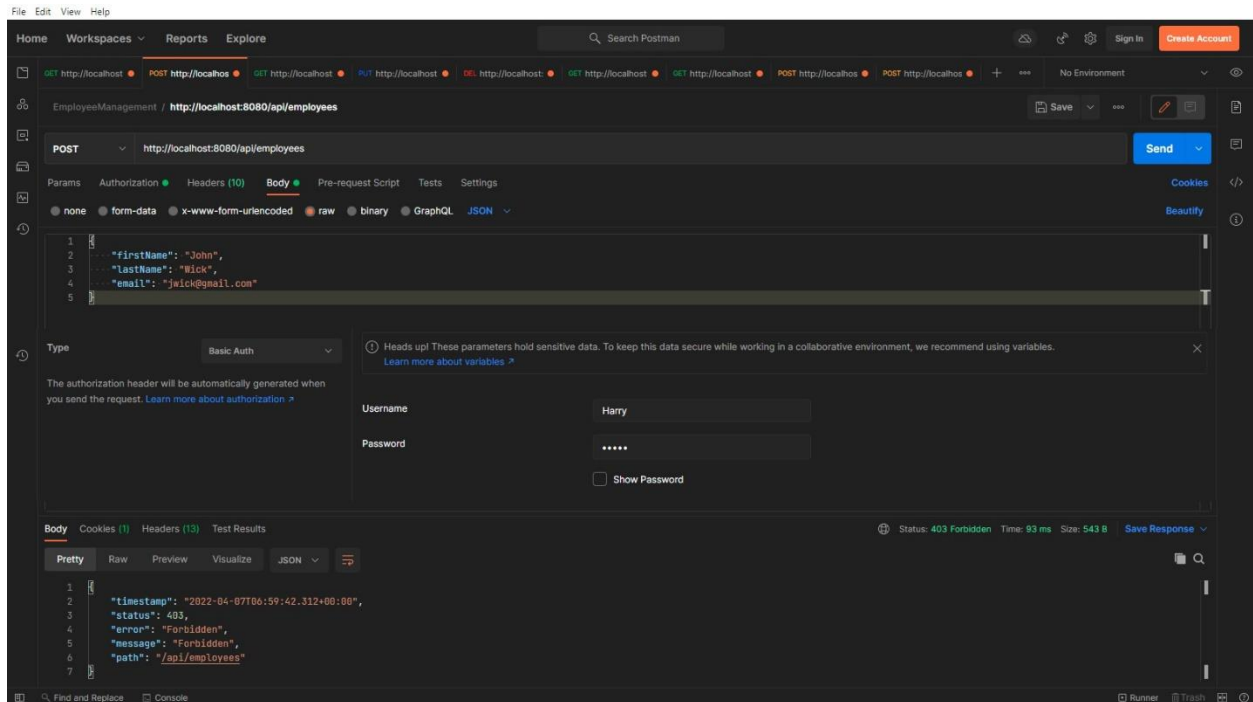
1. List of details of all Employees



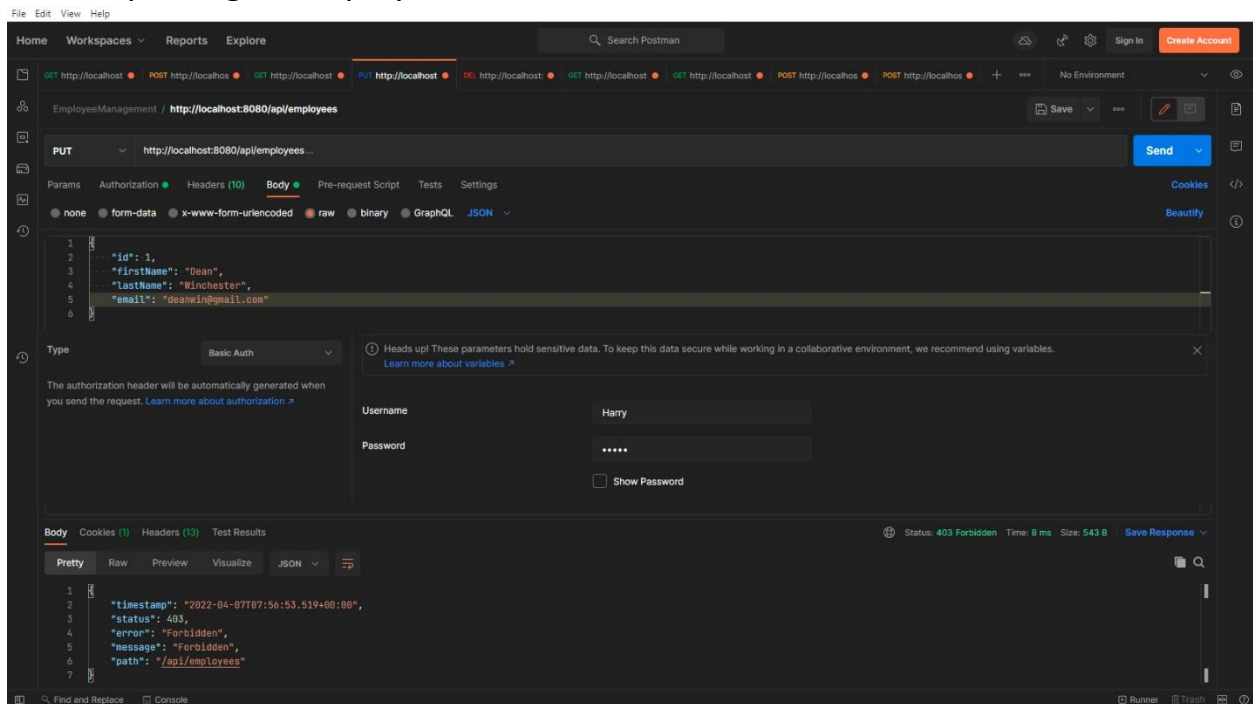
2. Get details of an Employee based upon its ID



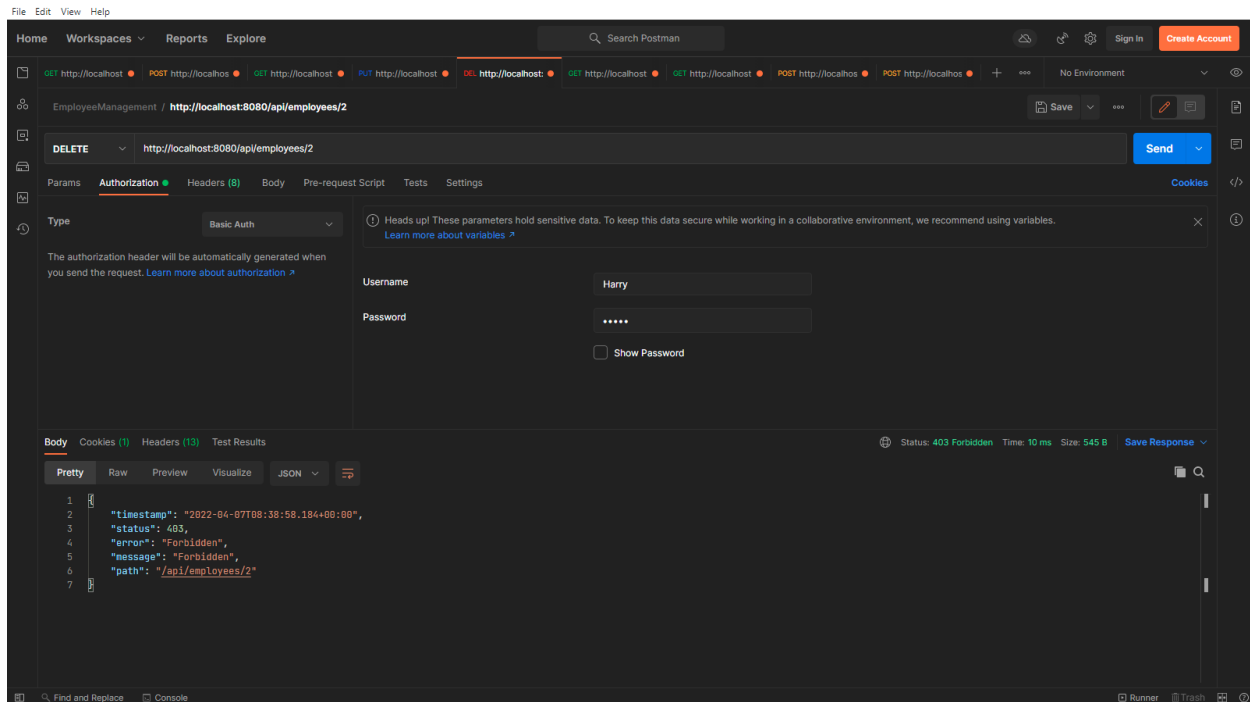
3. Adding details of an Employee



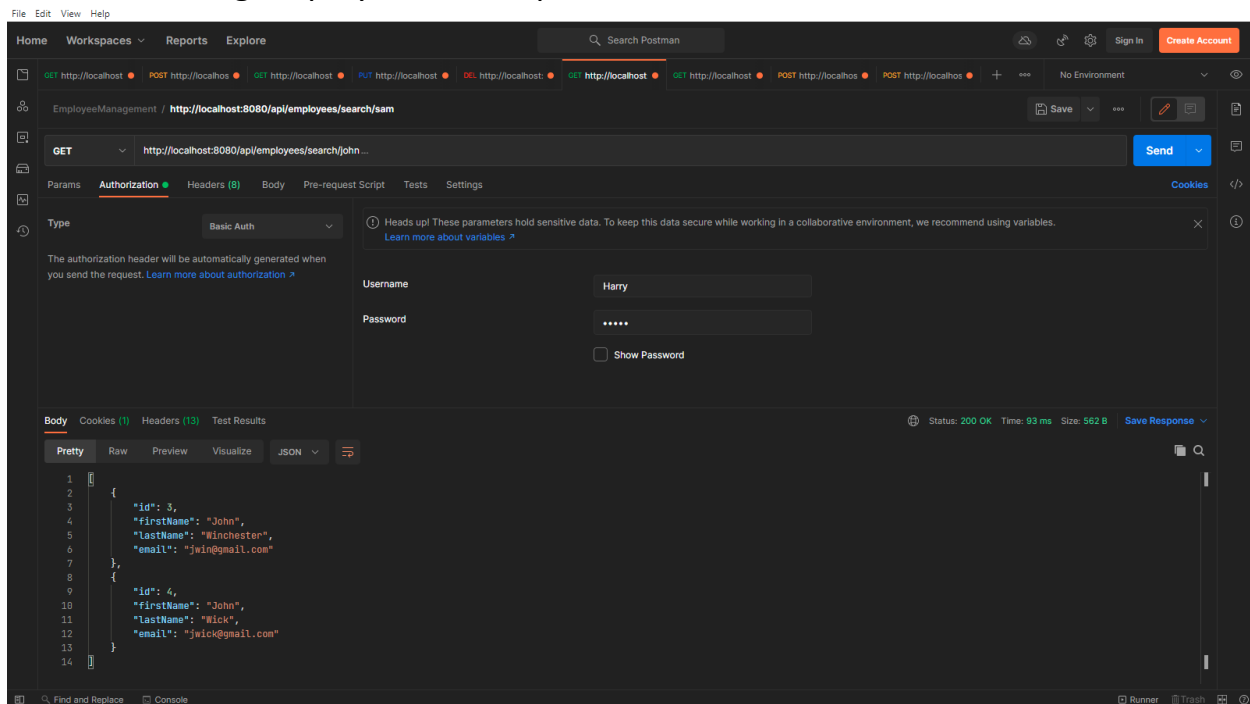
4. Updating an Employee's Detail



5. Deleting an Employee's Detail



6. Searching Employee based upon the First Name



7. List of details of all Employees in Ascending Order

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/employees/sort?order=asc`. The response status is 200 OK. The response body is a JSON array of employee objects, sorted by ID in ascending order.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> order	asc	

```
{
  "id": 1,
  "firstName": "Dean",
  "lastName": "Winchester",
  "email": "dwin@gmail.com"
},
{
  "id": 3,
  "firstName": "John",
  "lastName": "Winchester",
  "email": "jwin@gmail.com"
},
{
  "id": 4,
  "firstName": "John",
  "lastName": "Wick",
  "email": "jwick@gmail.com"
},
{
  "id": 2,
  "firstName": "Sam",
  "lastName": "Winchester",
  "email": "swin@gmail.com"
}
```

8. List of details of all Employees in Descending Order

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/employees/sort?order=desc`. The response status is 200 OK. The response body is a JSON array of employee objects, sorted by ID in descending order.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> order	desc	

```
{
  "id": 2,
  "firstName": "Sam",
  "lastName": "Winchester",
  "email": "swin@gmail.com"
},
{
  "id": 3,
  "firstName": "John",
  "lastName": "Winchester",
  "email": "jwin@gmail.com"
},
{
  "id": 4,
  "firstName": "John",
  "lastName": "Wick",
  "email": "jwick@gmail.com"
},
{
  "id": 1,
  "firstName": "Dean",
  "lastName": "Winchester",
  "email": "dwin@gmail.com"
}
```