EMPLOYEE MANAGEMENT WEB REST APPLICATION

1. **Add Roles Dynamically in the database**

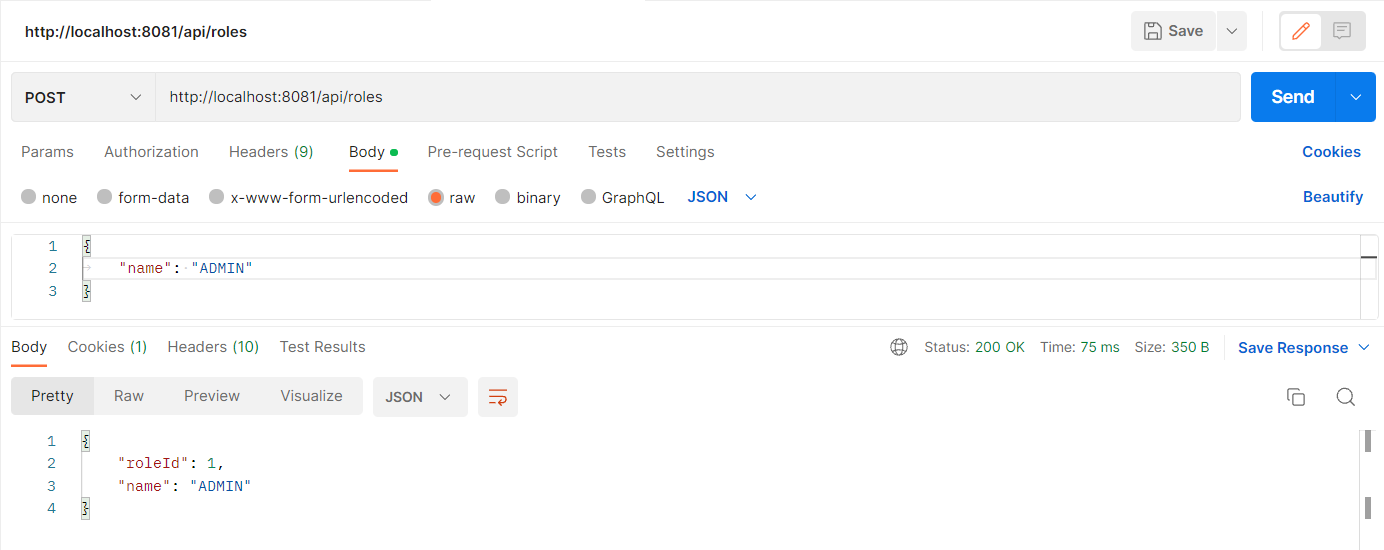
Fetching and Creating roles are available in the endpoint

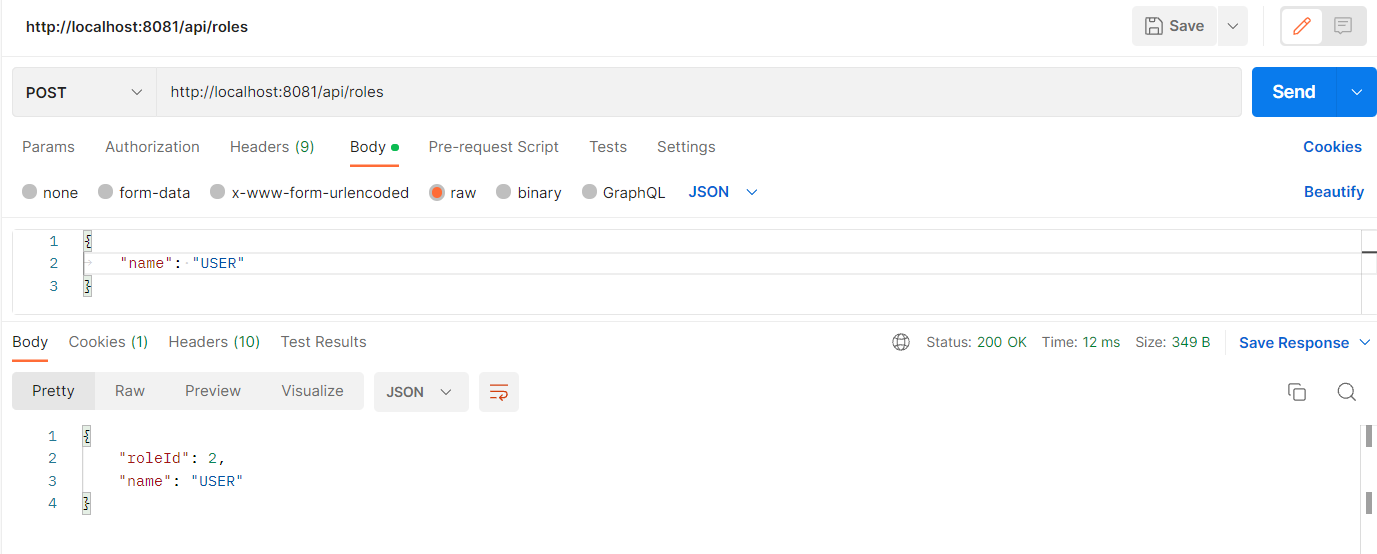
<http://localhost:8081/api/roles>

Two Roles are being added as shown below:

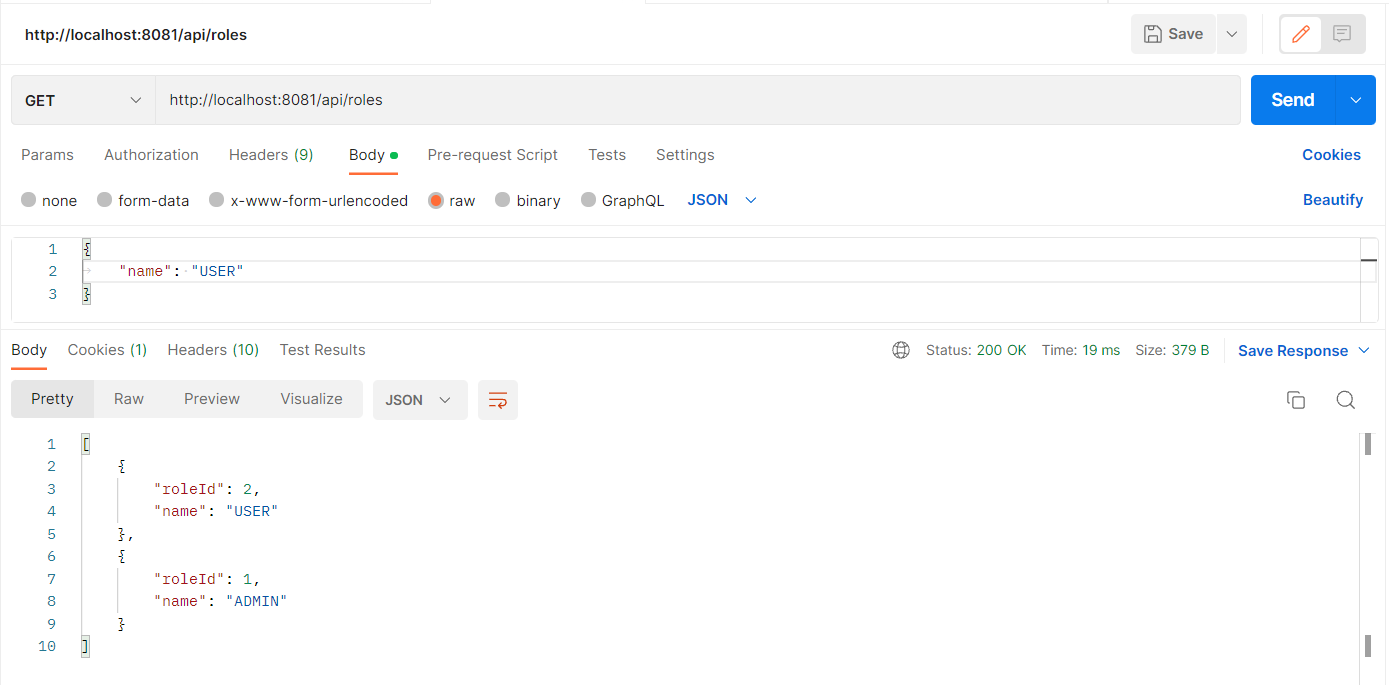
USER and ADMIN

Creating roles using POST





Fetching Roles using GET Request

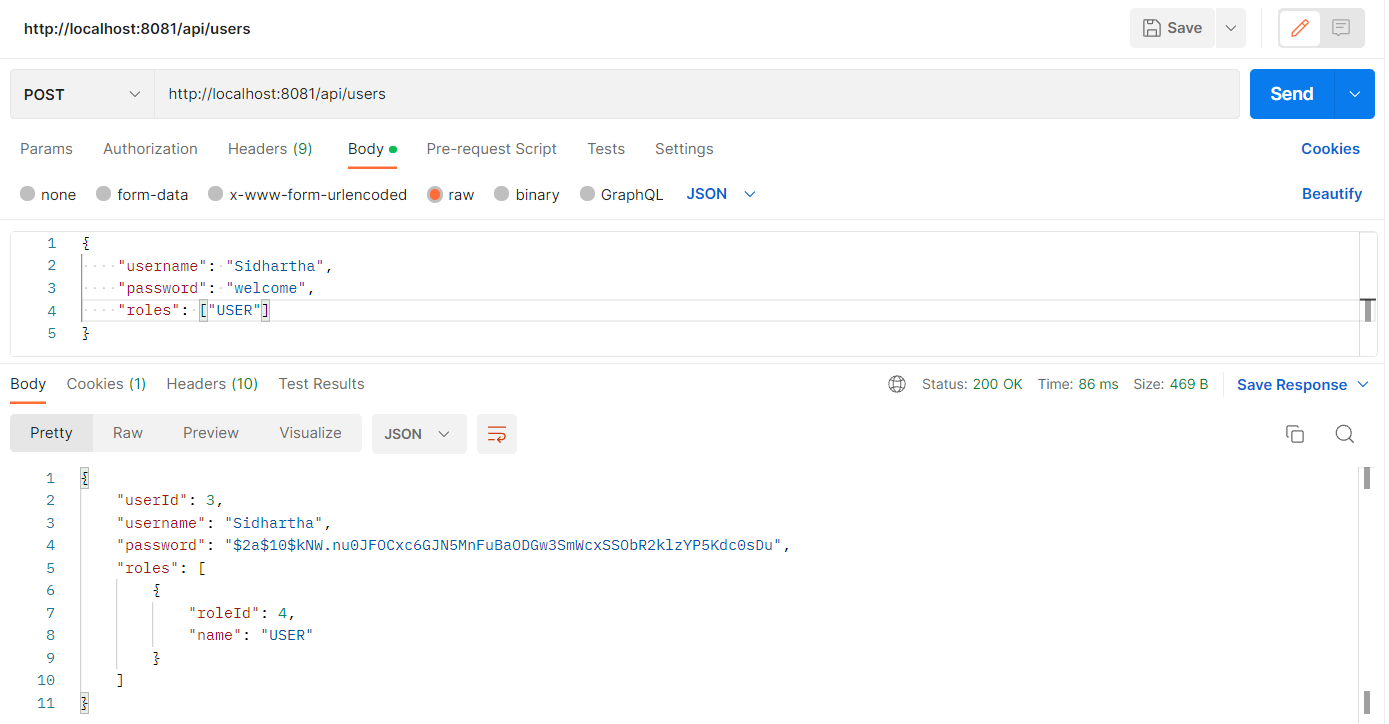
****

1. **Add Users in the DB dynamically**

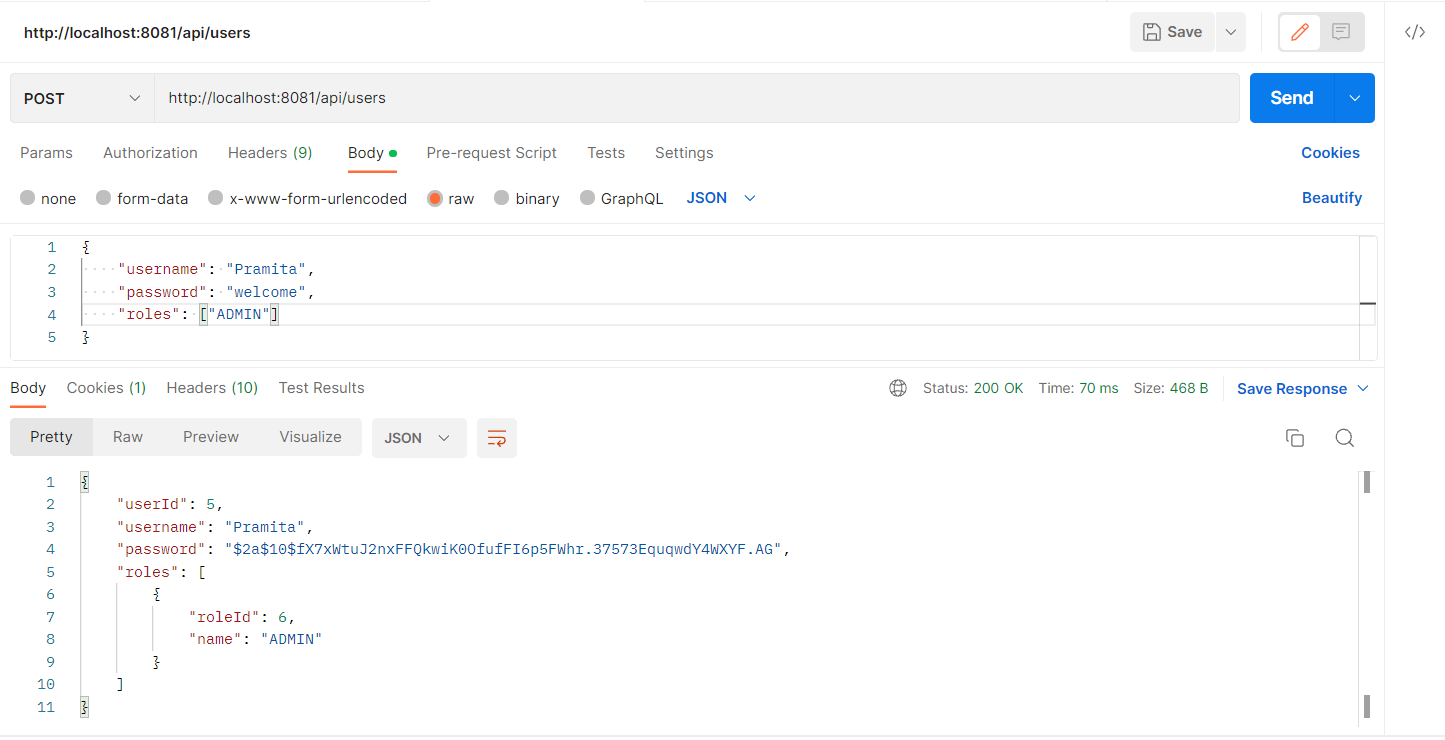
The fetching and creating of Users are available in the end-point

http://localhost:8081/api/users

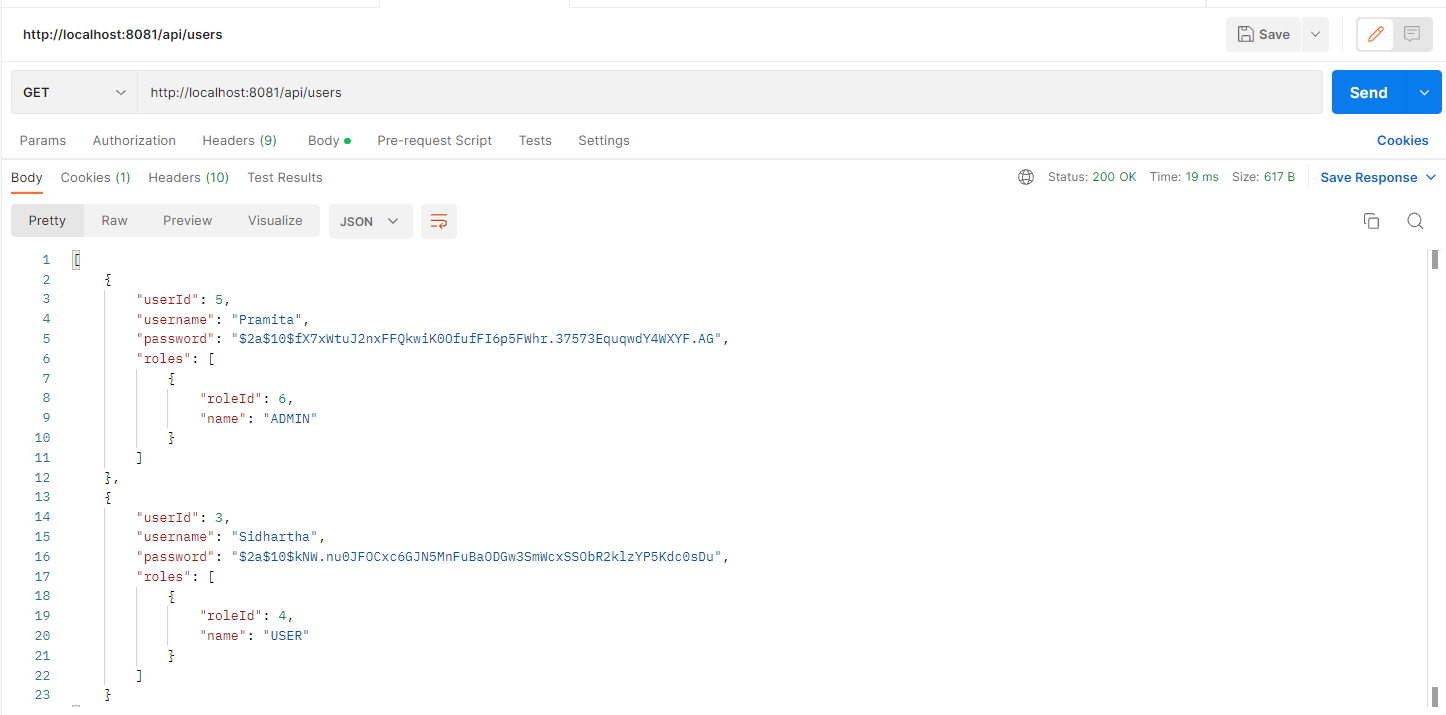
Adding a User with USER role



Adding a User with ADMIN Role



Fetching the Users using GET Request



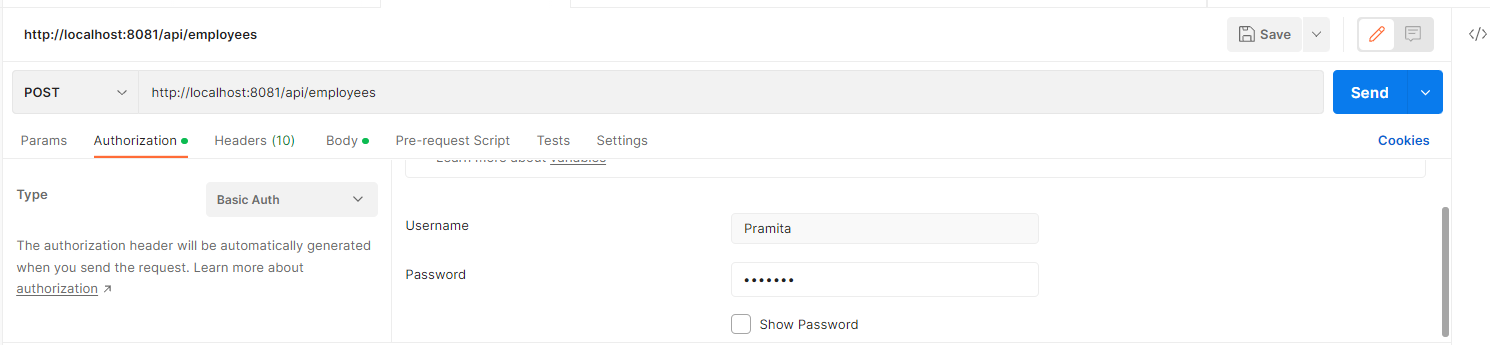
1. **Add Employees to the DB using a user having Admin role. The user has to be authenticated using Username and password in Postman**

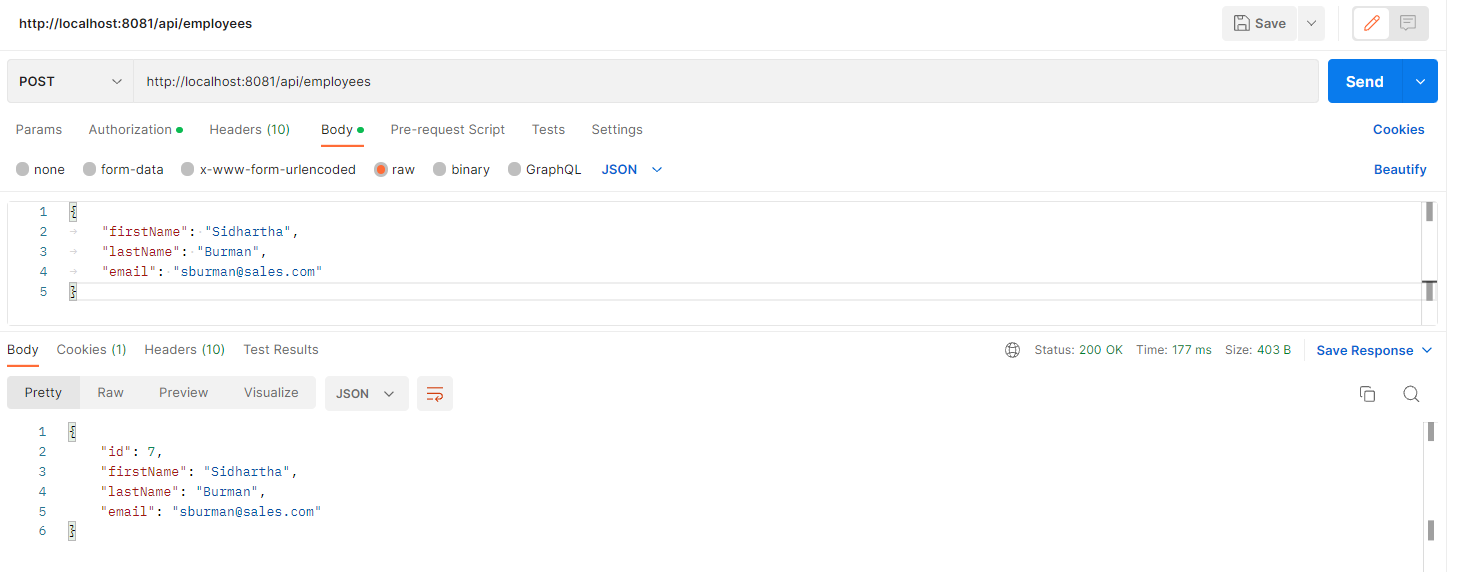
The employees can be added/fetched from the end point

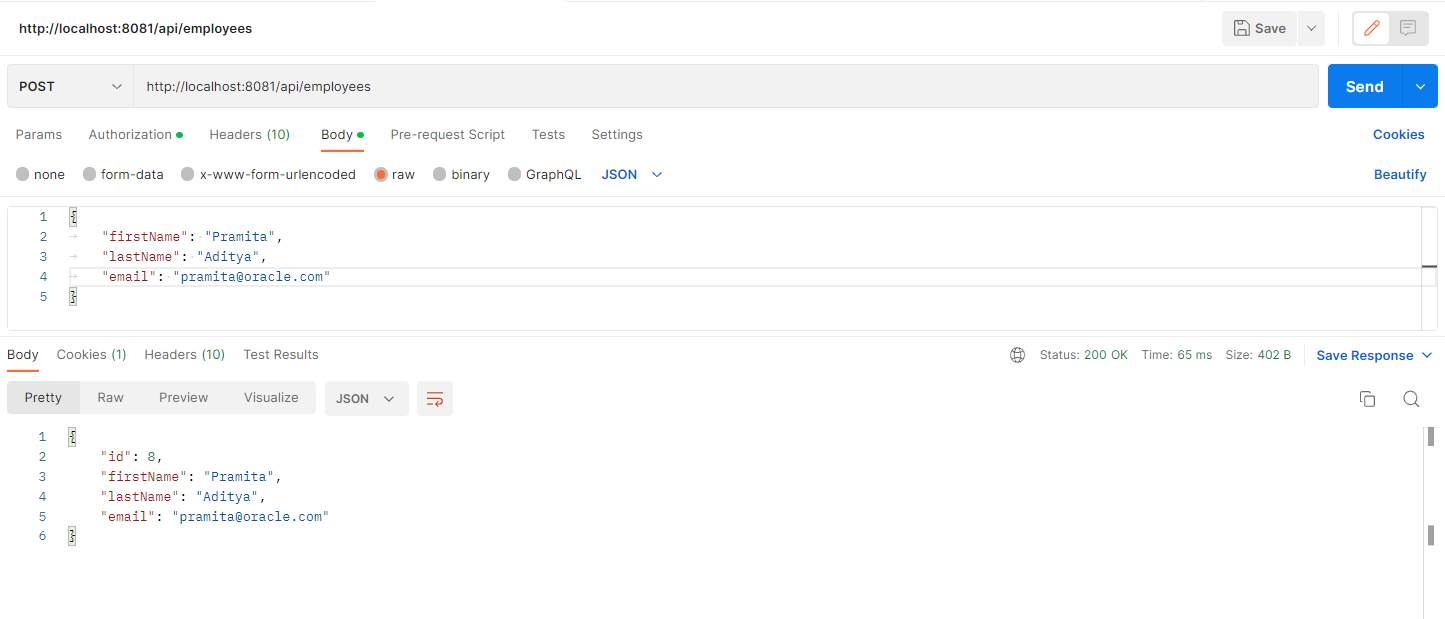
http://localhost:8081/api/employees

Here, in this example I’m using User: Pramita having admin role to POST employees

Authentication



Posting Employees



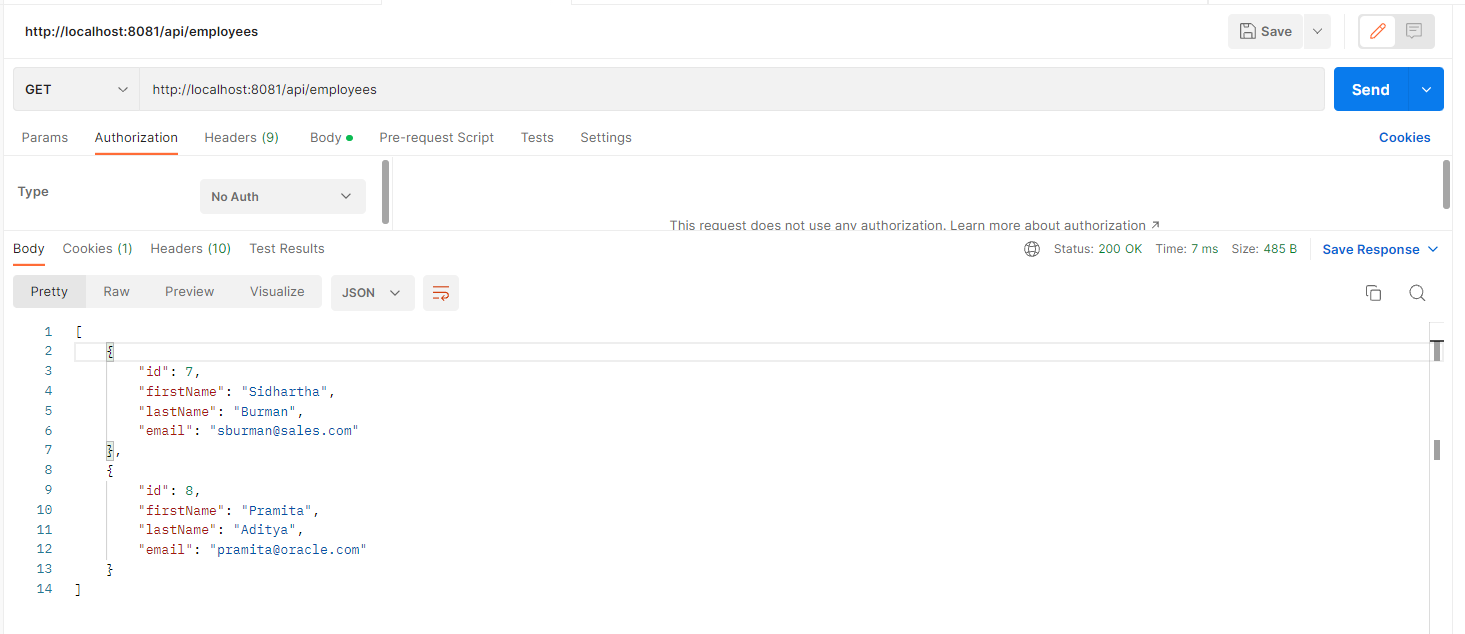
1. **List all the employees in the database**

The employees can be fetched from the endpoint

http://localhost:8081/api/employees

Fetching the added Employees using GET Method. This returns the employees added in Step 3 above.

Note : No authorisation is required to fetch the employees from the DB. It is an open end-point for fetching



1. **Fetching an employee by Id**

An employee can be fetched by passing the employee id in the end point as shown below

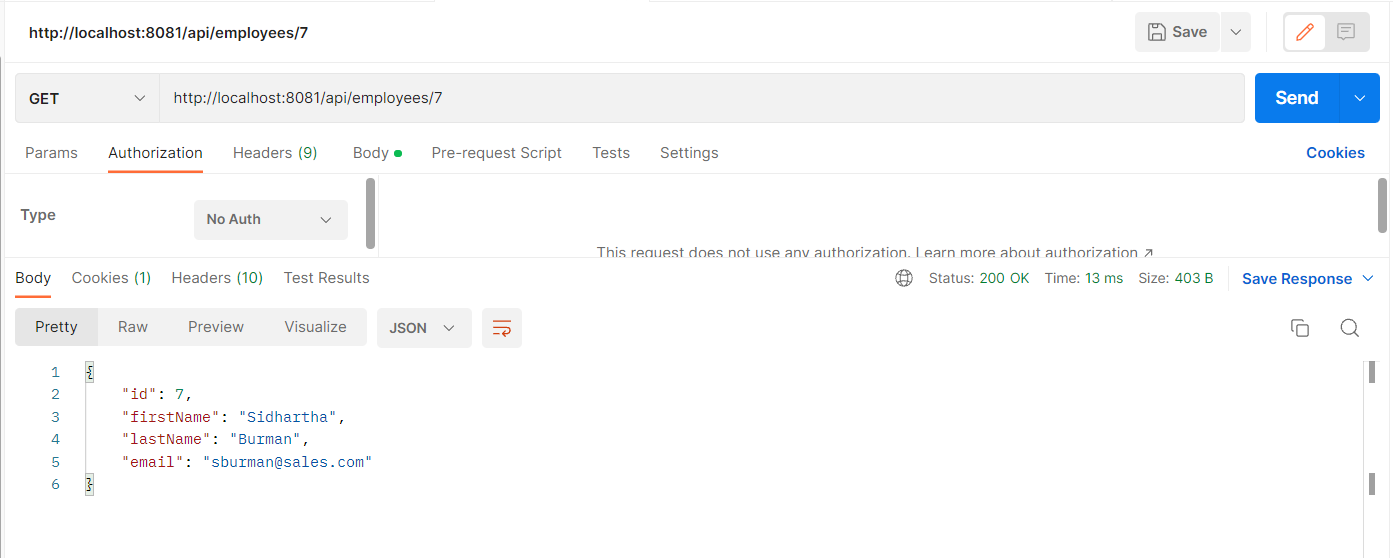
<http://localhost:8081/api/employees/7>

Here, 7 is the employee id of the Employee: Sidhartha. So, hitting the above end-point will return details of employee Sidhartha.

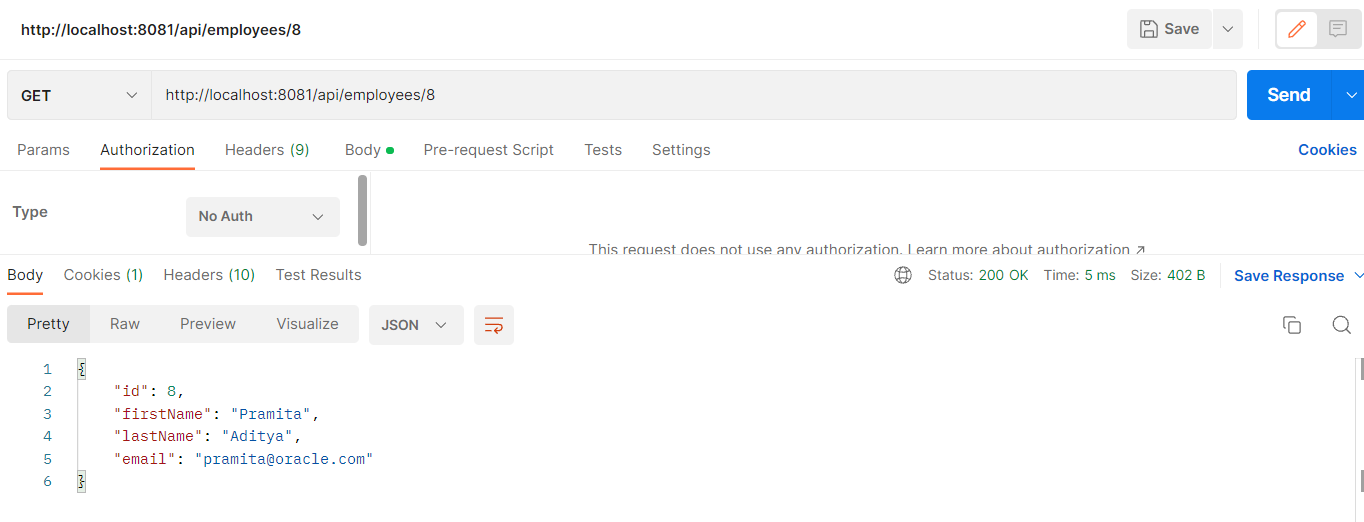
Similarly, passing 8 as parameter to end-point will return details of Employee Pramita.

The above are demonstrated below.

Employee id: 7



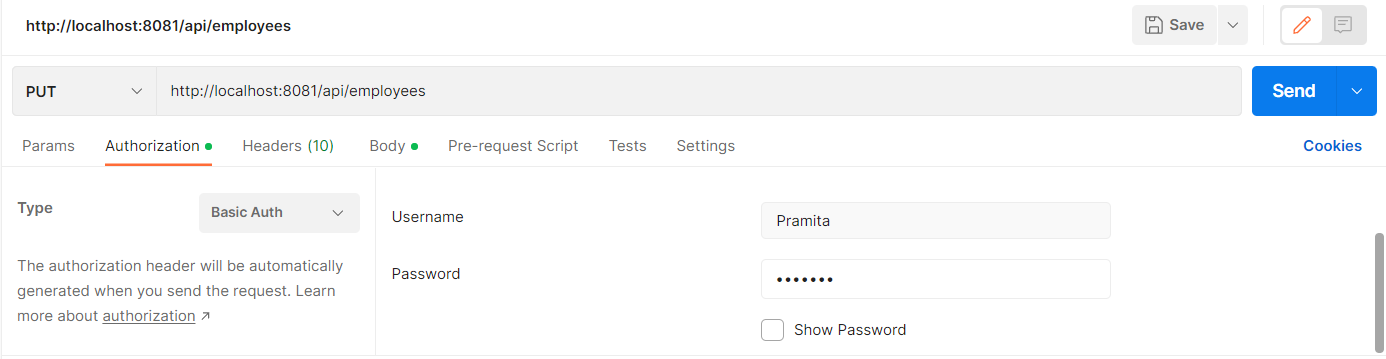
Employee id: 8



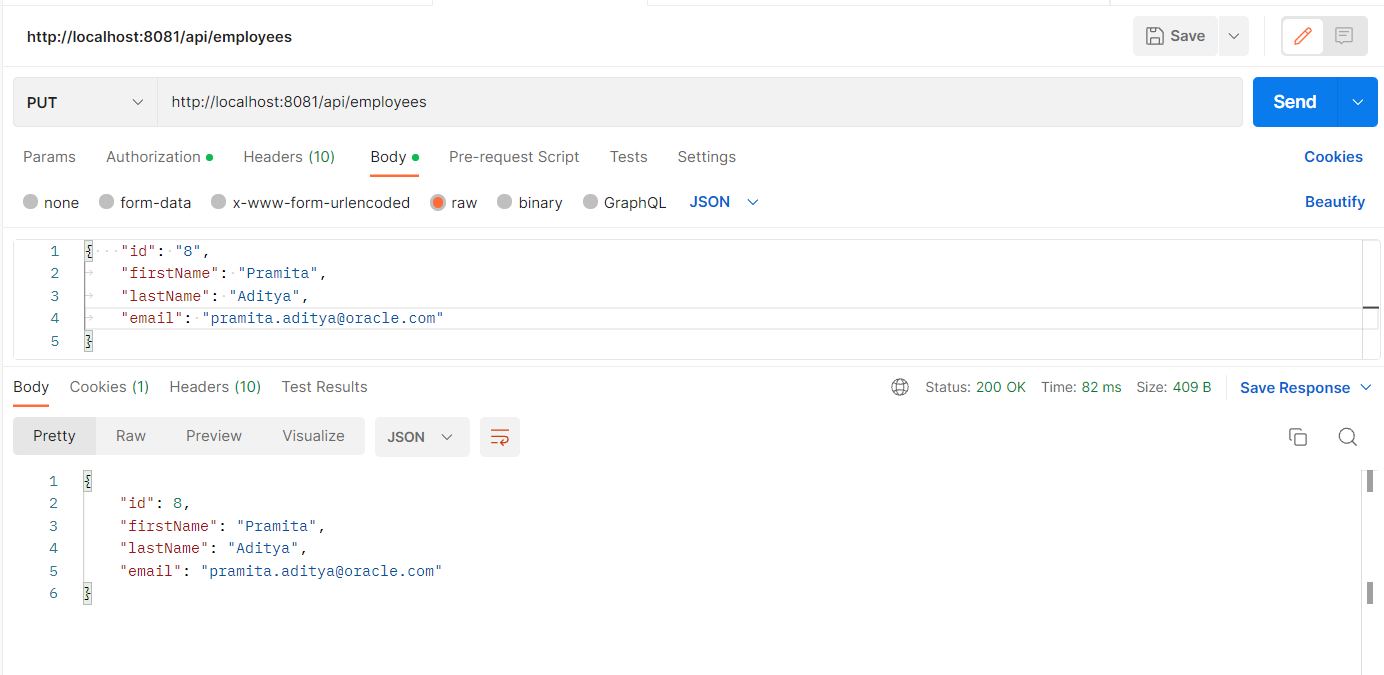
1. **Update an existing employee record with the given updated json object.**

An existing employee can be updated by passing the json object with employee id.

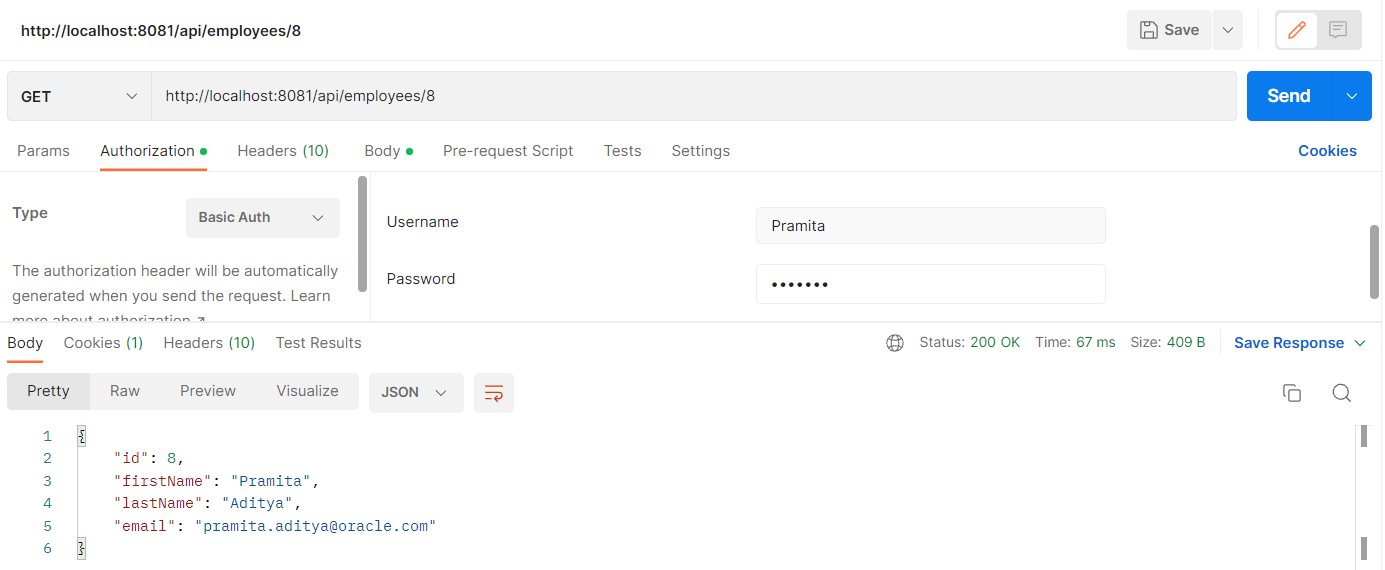
PUT request will need Authentication since it updates employee data. Have used the User: Pramita to update the same.



Updating the Employee Pramita’s email to pramita.aditya@oracle.com

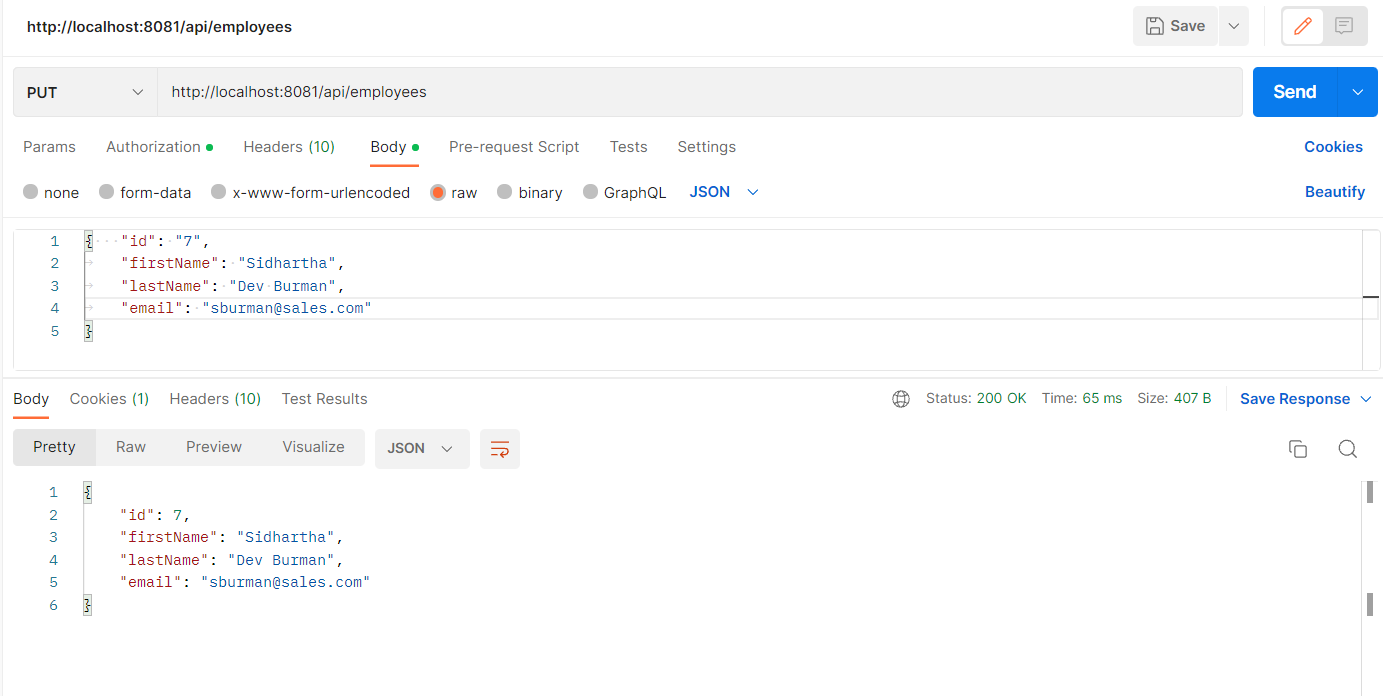


Fetching the updated Employee details using <http://localhost:8081/api/employees/8>

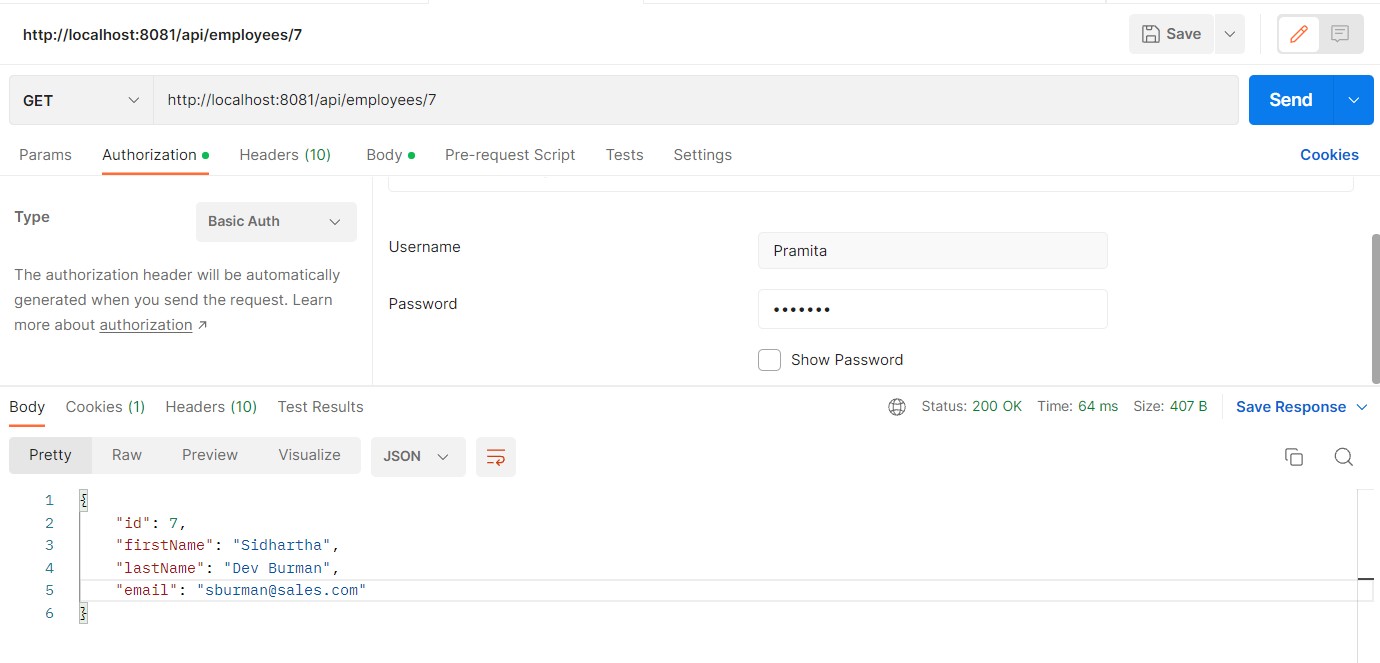


Similarly, updating Employee’s: Sidhartha lastname to : Dev Burman

Updating



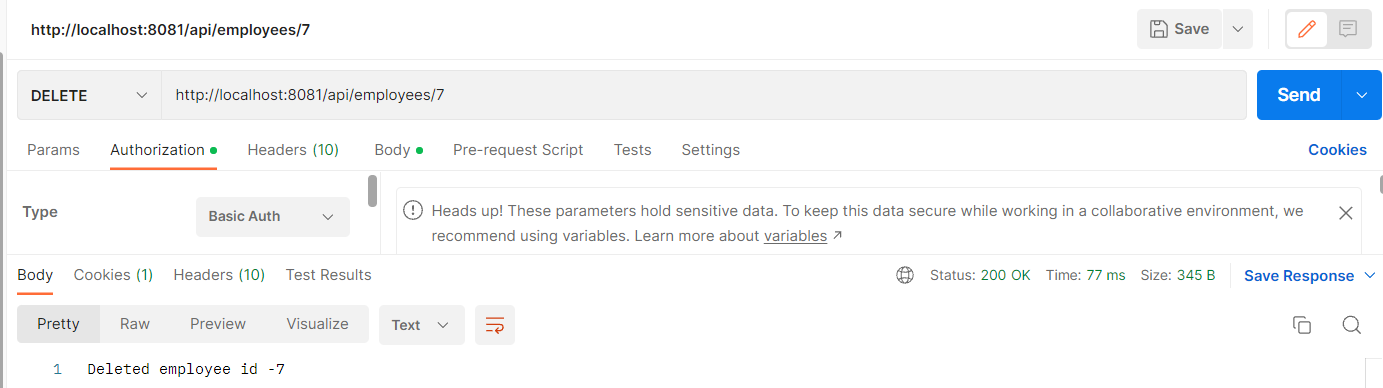
Fetching the updated employee



1. **Delete an existing employee**

An existing employee can be deleted by passing empid as a parameter and using DELETE method in postman as shown. The end point is :

http://localhost:8081/api/employees/7



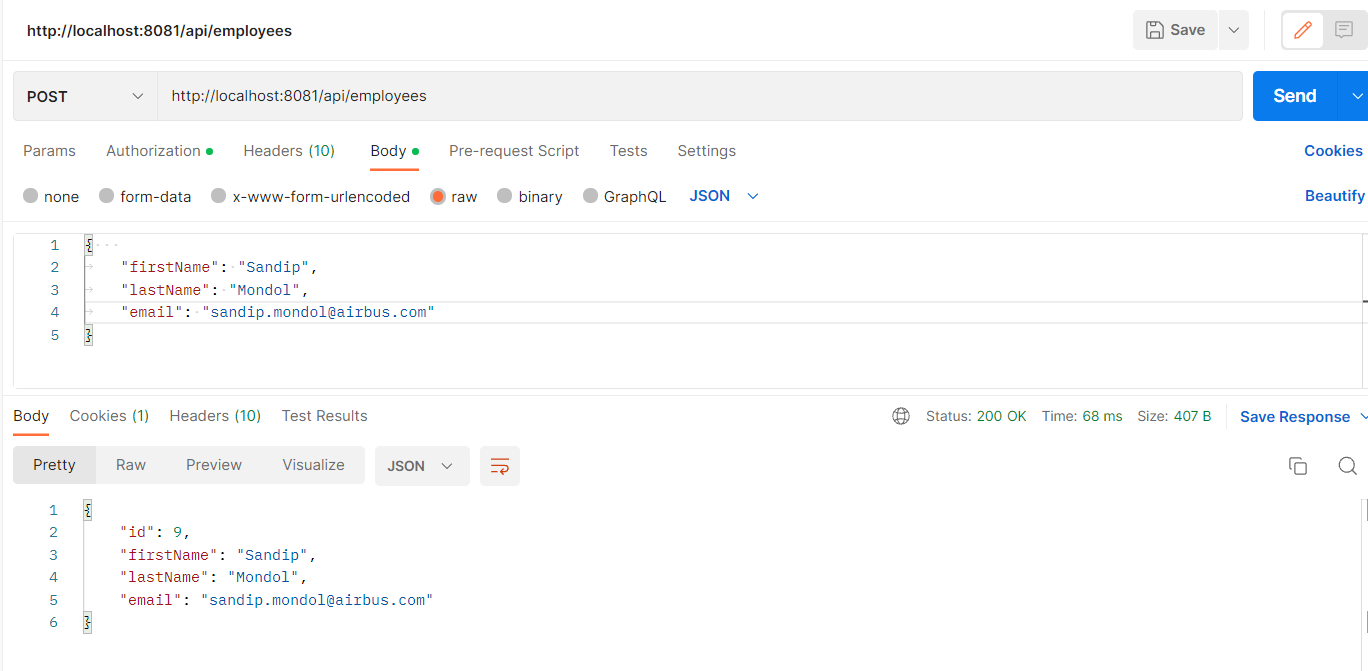
1. Fetching Employees by First Name

The endpoint for searching employees is

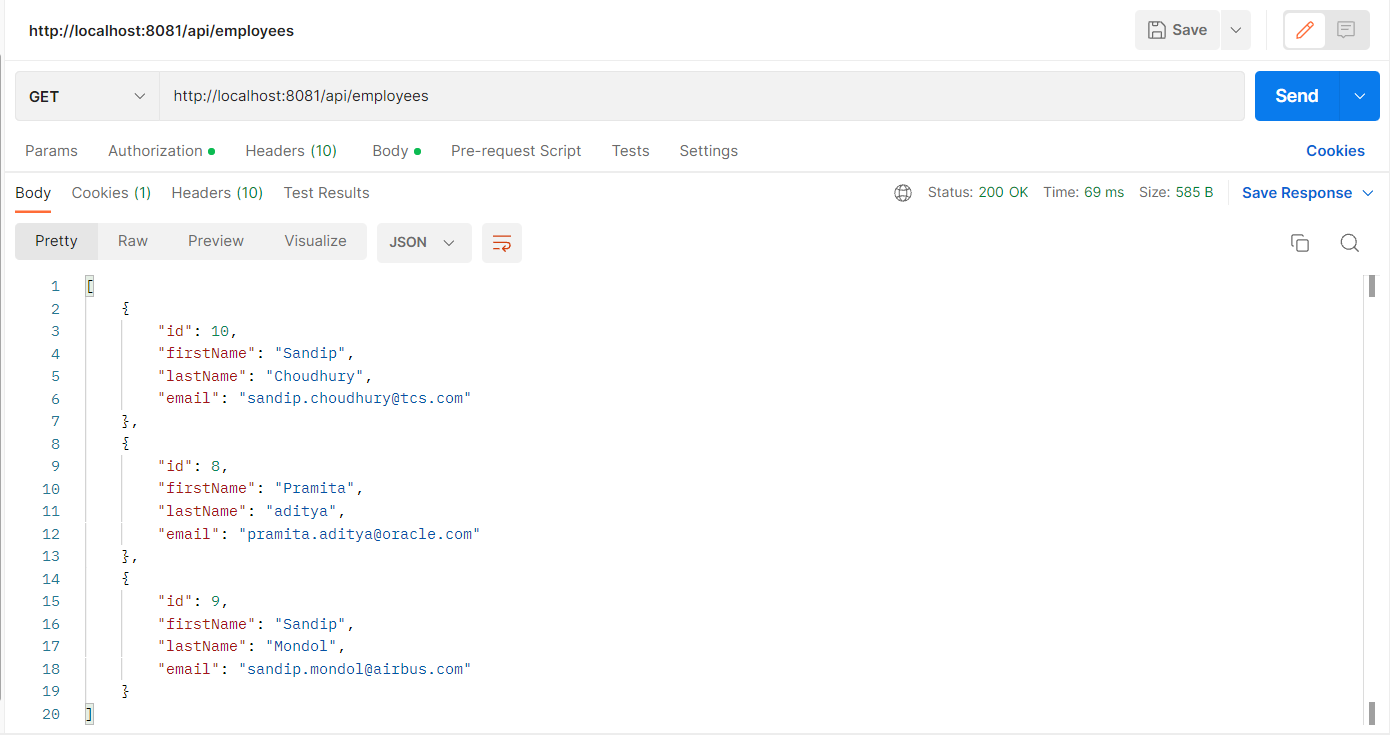
<http://localhost:8081/api/employees/search/Pramita>

Here, Pramita is the search parameter, i.e the Firstname

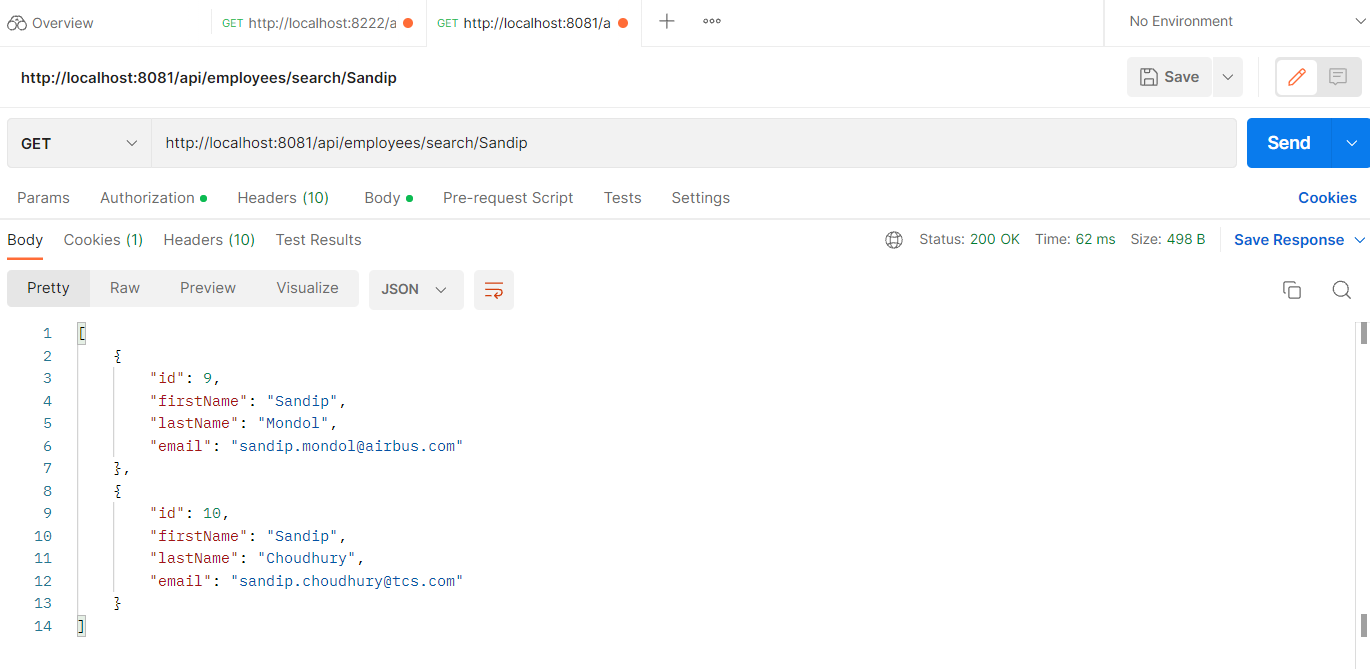
To demonstrate the above, creating 2 employees by firstname: Sandip



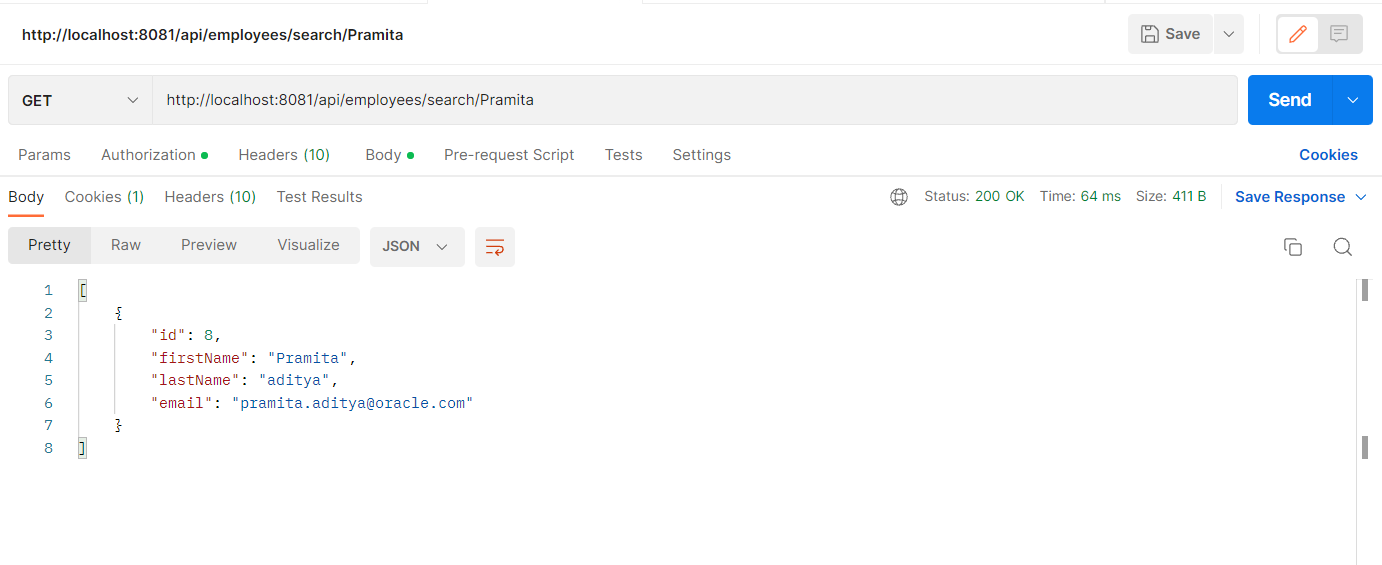
After addition of two employees, the GET fetches the below results



Now searching by Firstname: Sandip

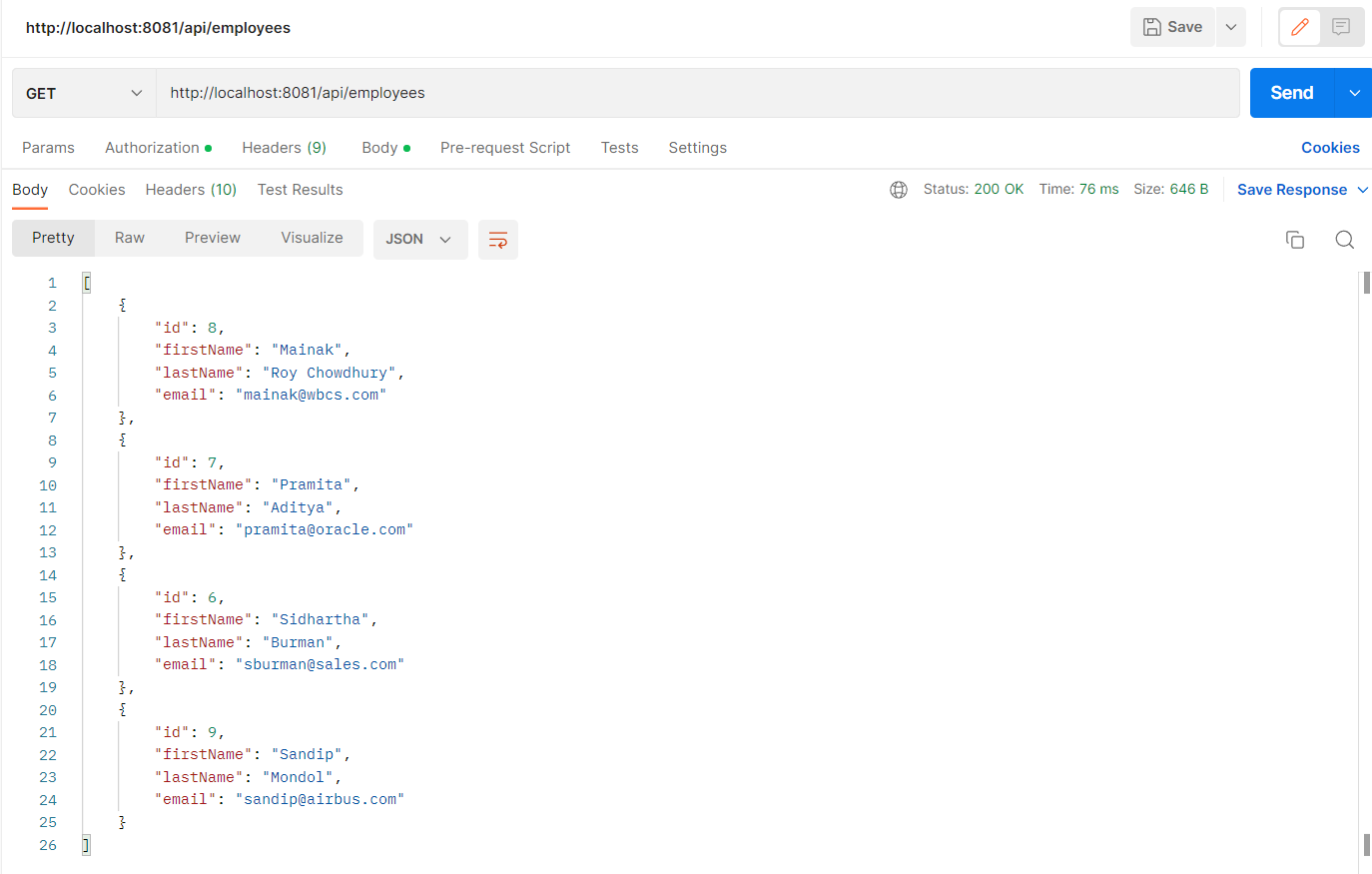


Now Searching by Firstname: Pramita



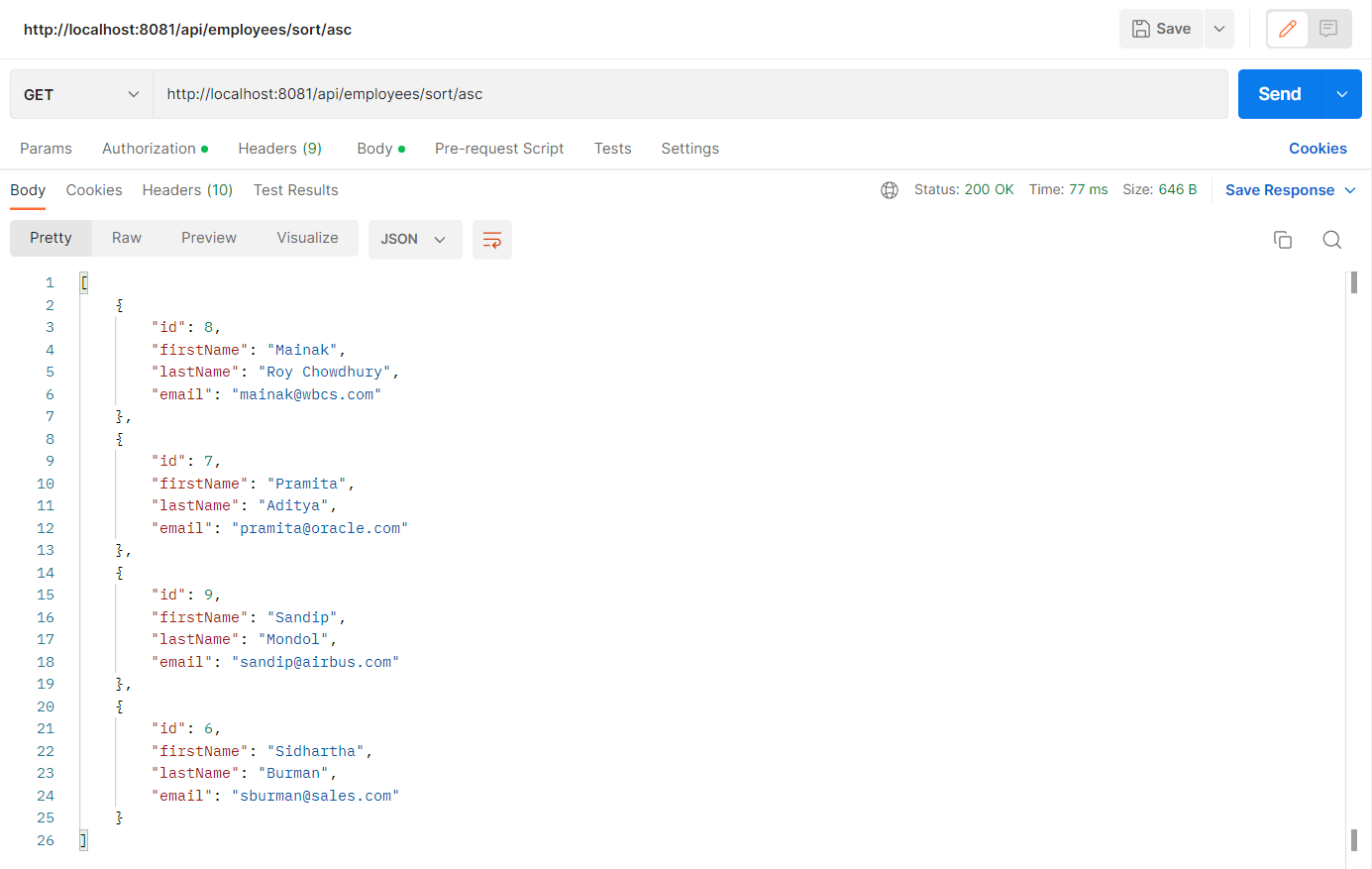
1. **Sorting Employees on First Name in Ascending or Descending order.**

Added few employees as shown below to test the Sorting end point. Employees fetched using GET



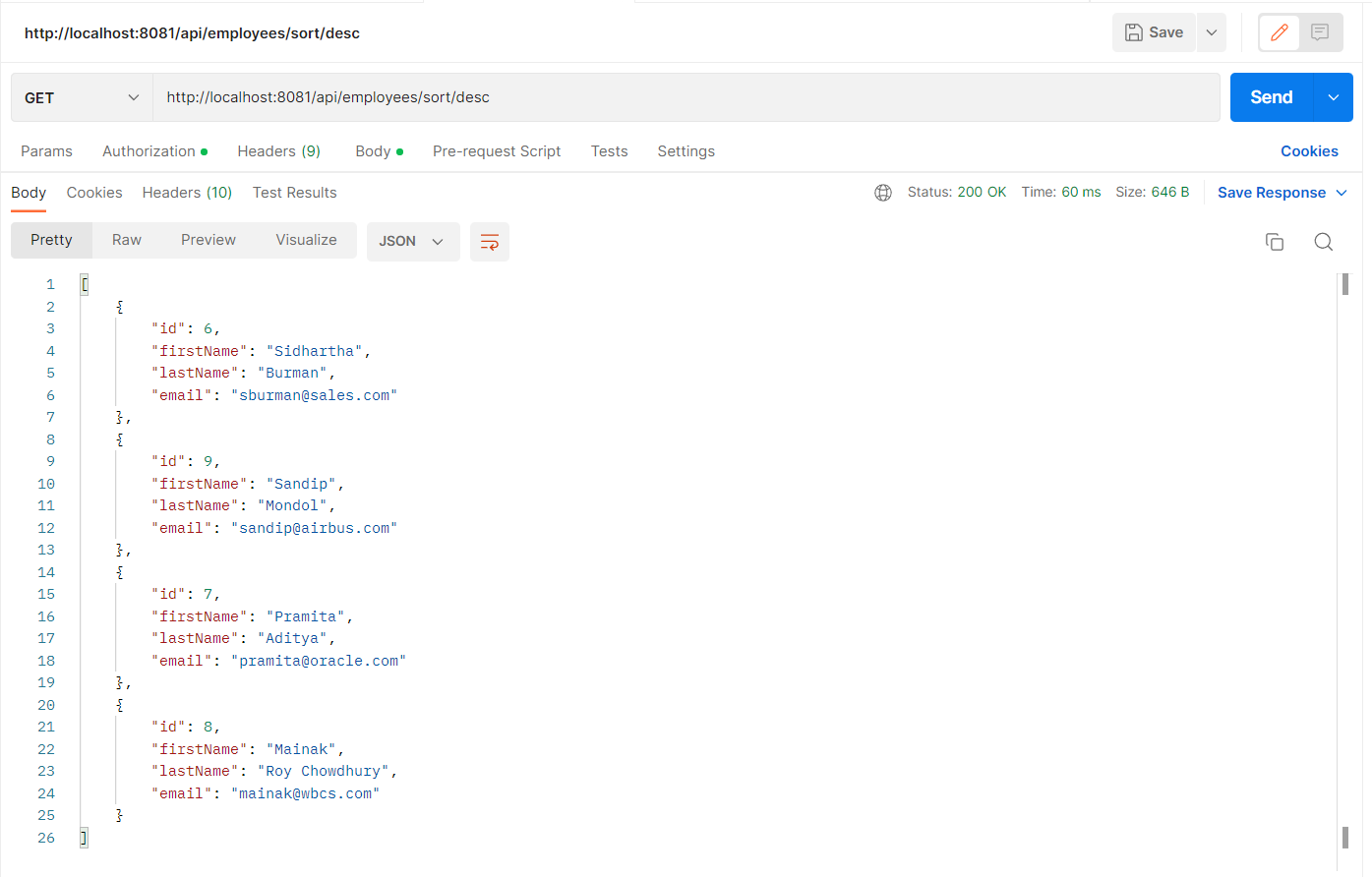
Endpoint for Sorting Employees by FirstName in Ascending order

http://localhost:8081/api/employees/sort/asc



Endpoint for Sorting Employees by FirstName in Descending order

http://localhost:8081/api/employees/sort/desc



---END--