**Environmental setup:**

1. Before executing the project we need to install the applications like xampp,postman,composer

1.1 Please find the xampp application download link below.

(<https://www.apachefriends.org/download.html>) preferred version **7.4.10**

1.1.1 After installing the application we need to start services Apache,MySQL using xampp control.

Mysql version **10.4.14-MariaDB**

1.2 Please find the postman application download link below.

(<https://www.postman.com/downloads>)

1.3 Please find the composer application download link below.

(<https://getcomposer.org/download/>)

**Project Execution Steps.**

1. Download or clone the project using this link (<https://github.com/tswaamy/tswaamy-Microservice-RestApi-s.git>)
2. After downloading the project extract files into htdocs/Project Name
3. Upload sample database available in document/techapi.sql.
4. Project environmental setup using .env
5. In .env file listed below are the main things for the project database connections.

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=techapi

DB\_USERNAME=root

DB\_PASSWORD=""

1. After .env file changes we need to run some commands using terminal.
2. In the terminal we need to change the path to the project folder.

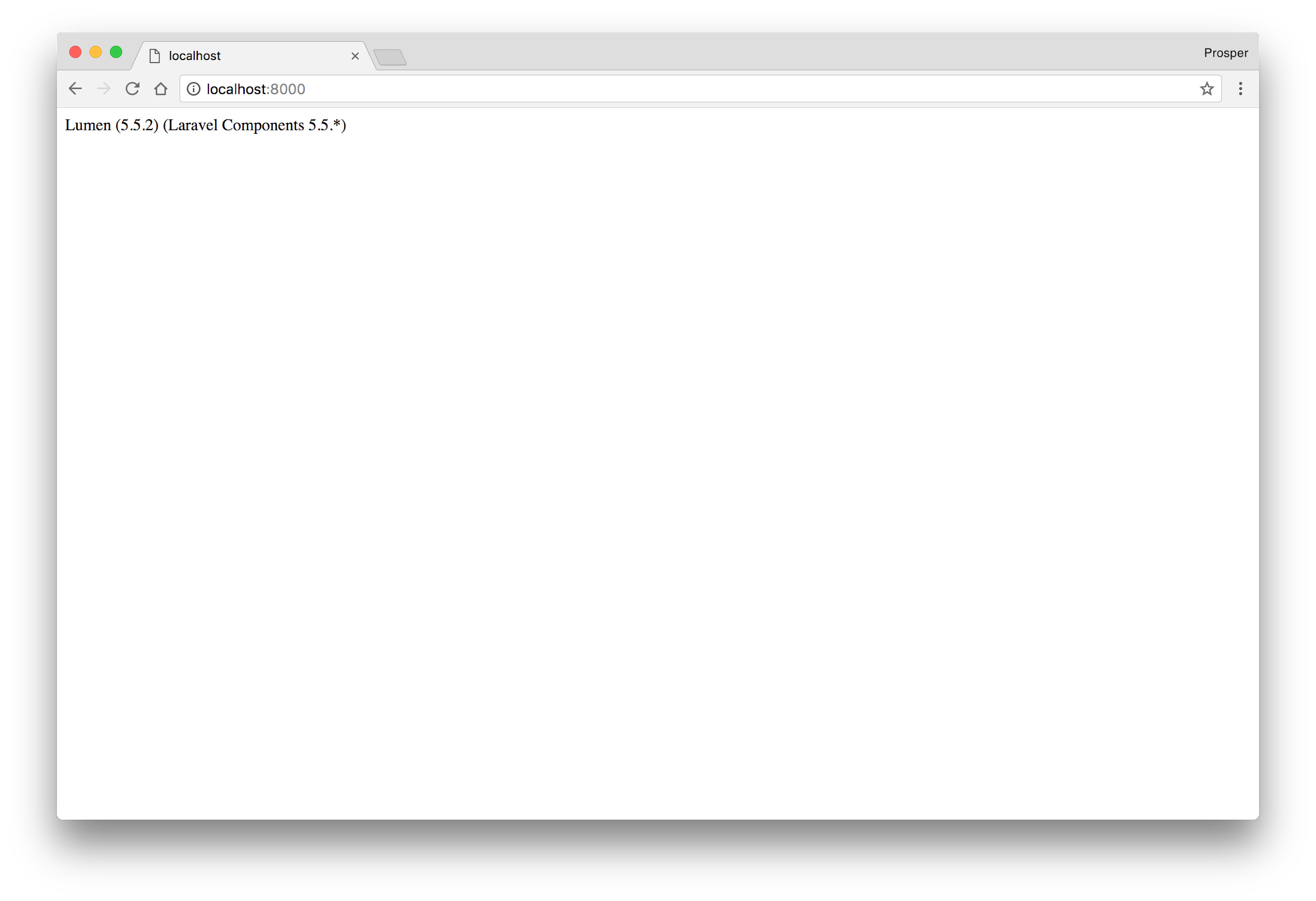
**Cmd: composer update**

(it helps for creating or updating required packages for the project)

**Cmd: php artisan cache:clear**

(it helps for clearing cache for the project)

1. Now, run **php -S localhost:8000 -t public** to serve the project. Head over to your browser. You should see the index page like so:



**Code Explanation**

1. Created a migration file in database/migrations/2020\_10\_08\_044321\_create\_emp\_details\_table.php this file helps for creating a data table.
2. Created a Model file in app/EmployeeDetail.php this file helps for creating or getting data from a data table.
3. Created a Controller file in app/Http/controllers/ExampleController.php this file helps for controlling the logic of methods.

3.1 **showAllEmployees** method helps for getting all employees data using EmployeeDetail model.

3.2 **showOneEmployee** method helps for getting selected employee data.

3.3 **create** method helps for creating or updating for employees data with the mandatory fields like (name,email,location,skills,emp\_id)

3.4 **delete** method helps for soft deleting employee data using EmployeeDetails model.

**Routes**

1. All related routes are added in routes/web.php

**Api:** url: <http://localhost:8000/api/employees>

**method:**Post

Sample data for creating multiple employee

{

"data":[{"id":"","emp\_id":"10012","name":"Shiva","email":"test1@accionlabs.com","location":"Bangalore","skills":"PHP,Laravel","status":"1"},

{"id":"","emp\_id":"10021","name":"Sai","email":"test2@accionlabs.com","location":"Bangalore","skills":"Python,AWS","status":"1"}

]

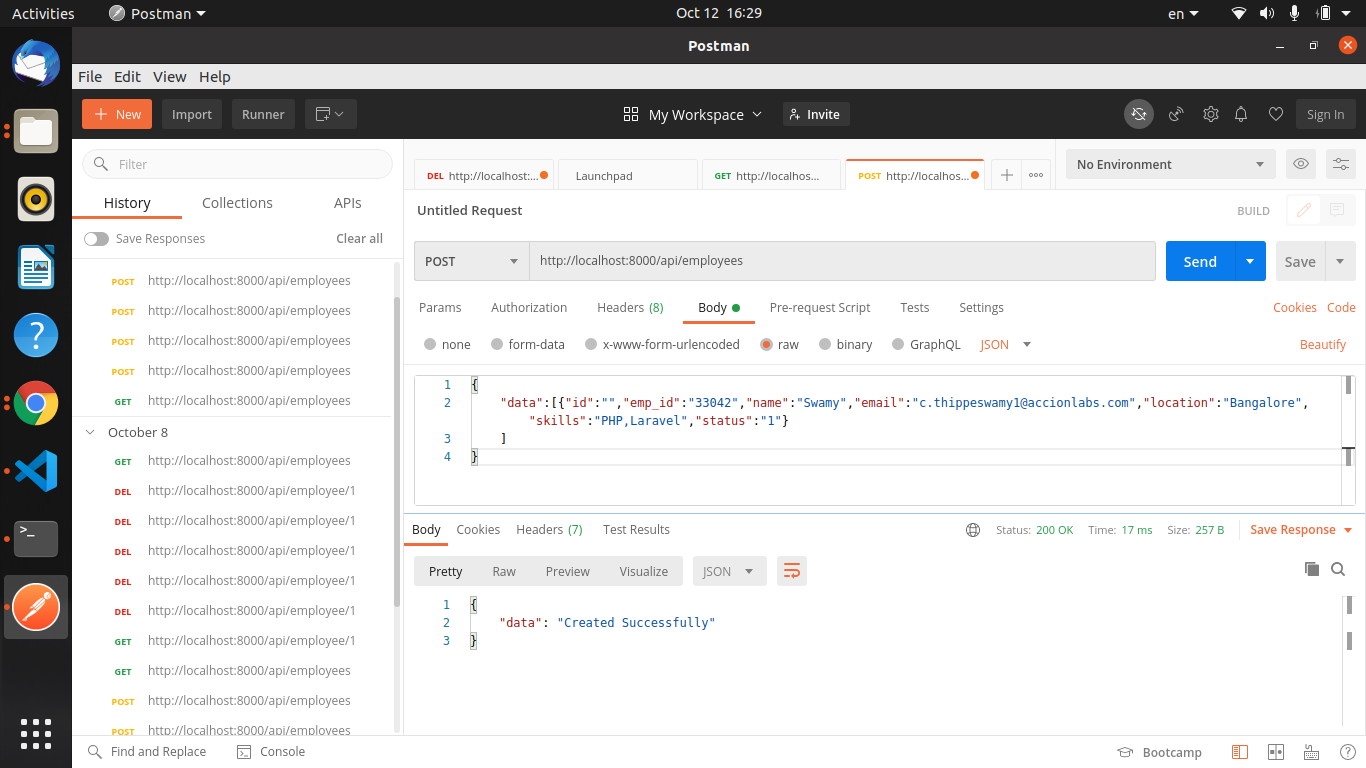
}

Sample data for creating single employee

{

"data":[{"id":"","emp\_id":"33042","name":"Swamy","email":"test@accionlabs.com","location":"Bangalore","skills":"PHP,Laravel","status":"1"}]

}

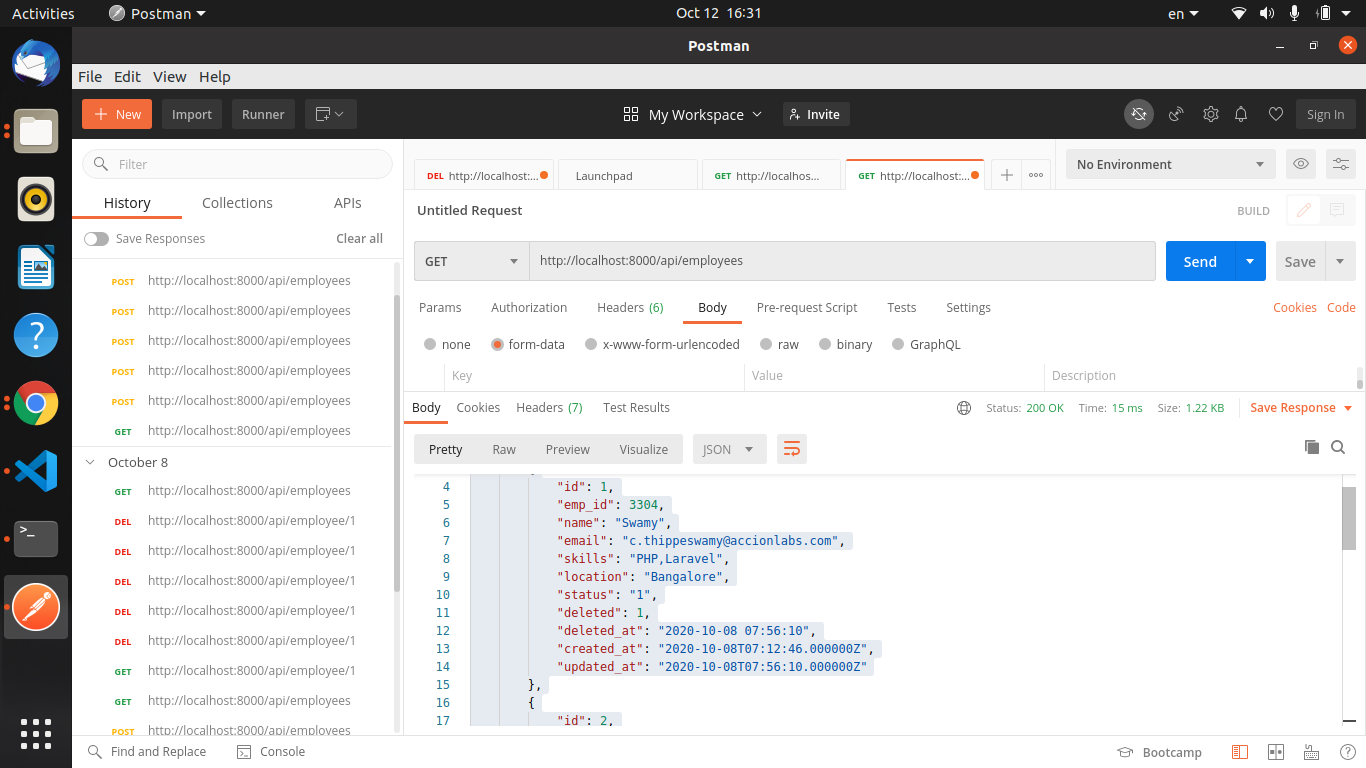


Sample data for get multiple employees

**Api:** url: <http://localhost:8000/api/employees>

**method:**Get

We are not passing any request data from body

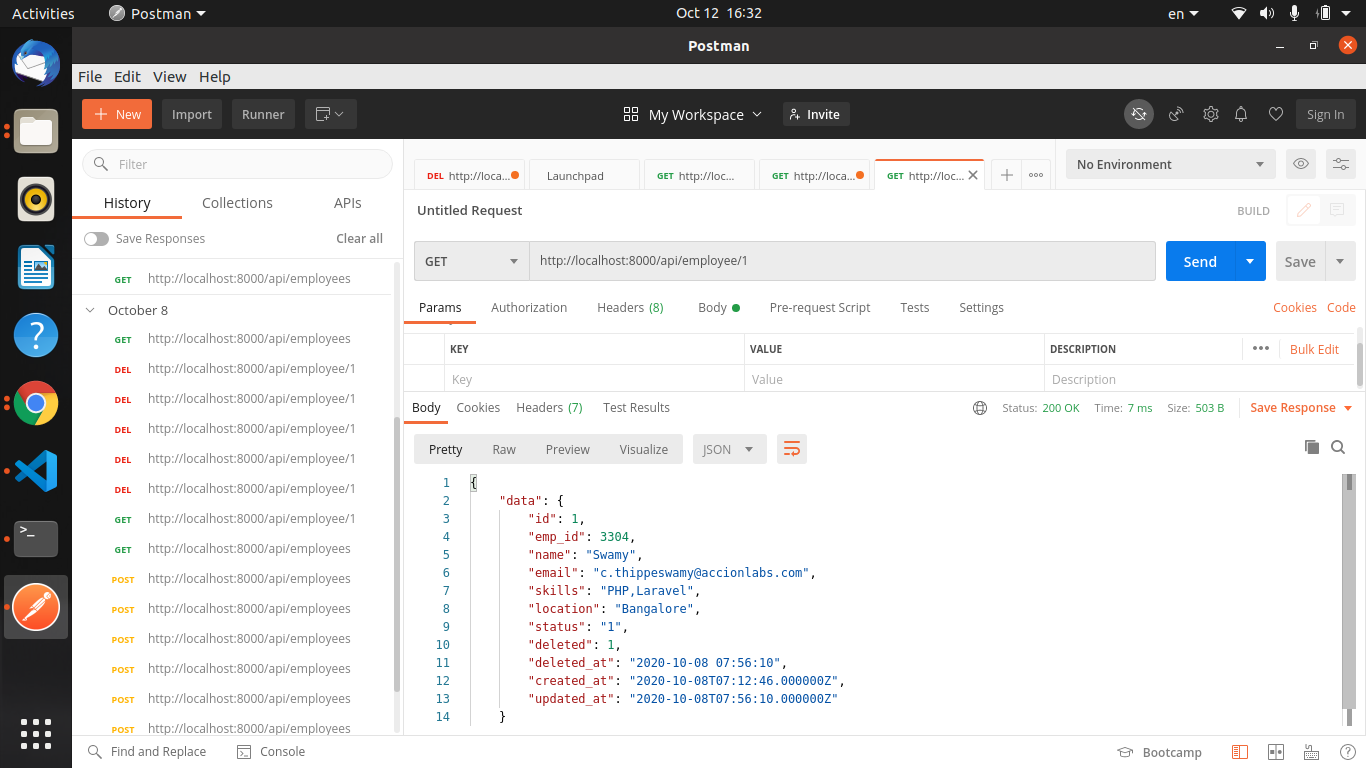


Sample data for get selected employee

**Api:** url: [http://localhost:8000/api/employee](http://localhost:8000/api/employees)/{id}

**method:**Get

We are not passing any request data from body

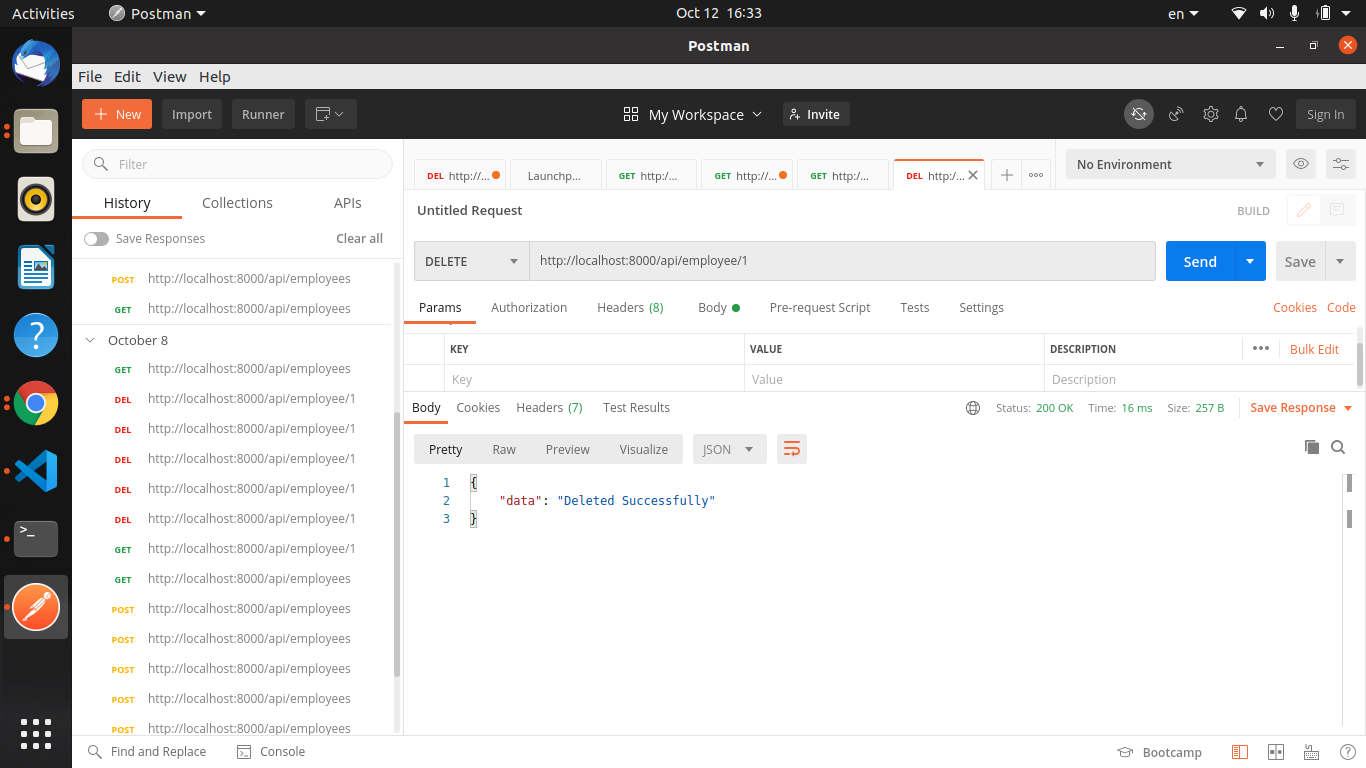


Sample data for soft delete selected employee

**Api:** url: [http://localhost:8000/api/employee](http://localhost:8000/api/employees)/{id}

**method:**DELETE

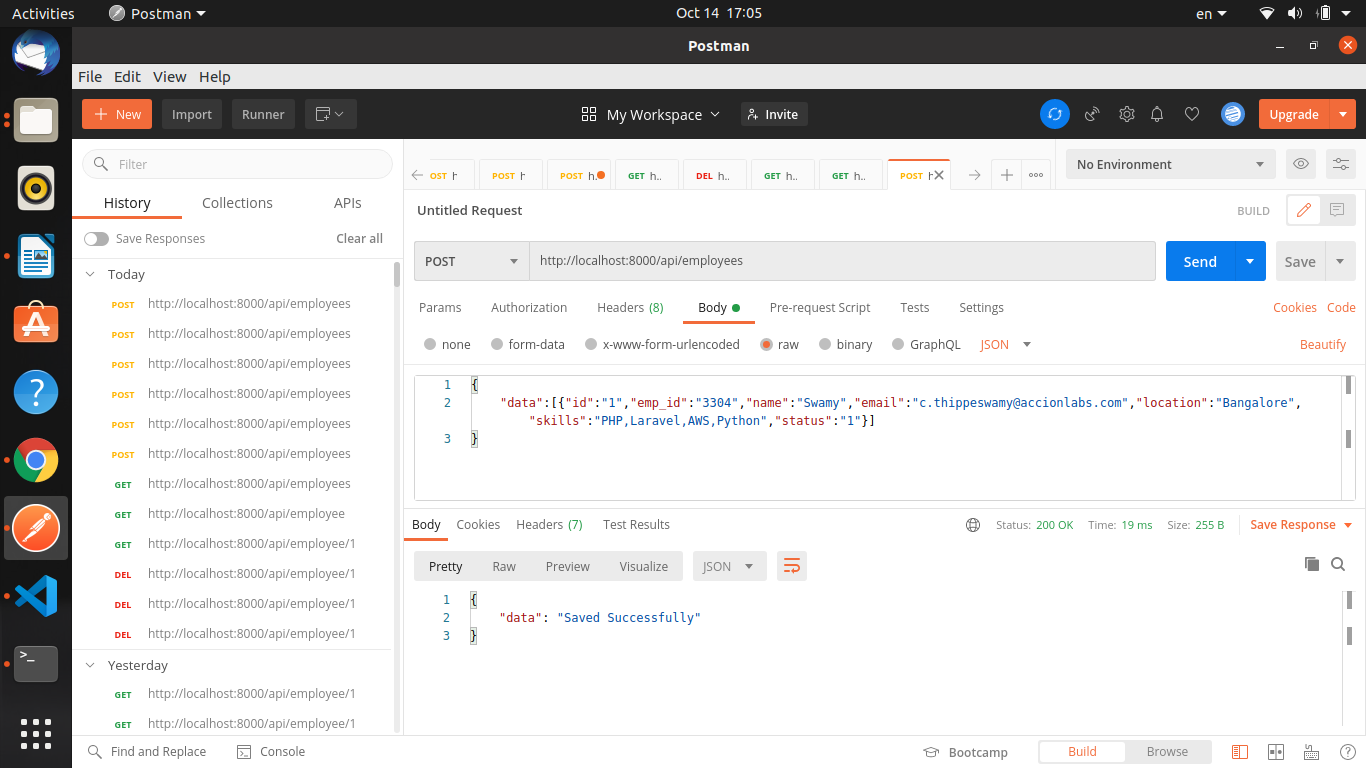
We are not passing any request data from body



Update api

**Api:** url: <http://localhost:8000/api/employees>

**method:**Post



Api’s Collections

url:<https://www.getpostman.com/collections/187683a506f8ca104e95>