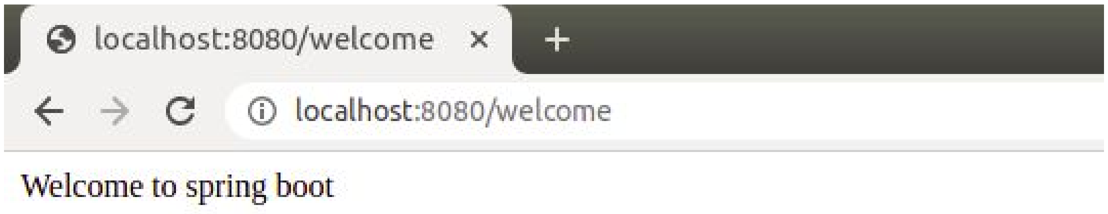
1. **Create a simple RESTful service in Spring Boot which returns the Response "Welcome to spring boot".**

```
package com.bootcamp.demo;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class Welcome {

@GetMapping(path = "/welcome")
public String welcome1(){
  return "Welcome to spring boot";
}

}
```



```
Welcome to spring boot
```

2. **Create an Employee Bean(id, name, age) and service to perform different operations related to employee.**

//Employee Class

```
package com.bootcamp.demo.employee;

public class Employee {
   private Integer id,age;
   private String name;


   public Integer getId() {
       return id;
   }

   public void setId(Integer id) {
       this.id = id;
   }

   public Integer getAge() {
       return age;
   }

   public void setAge(Integer age) {
       this.age = age;
   }

   public String getName() {
       return name;
   }

   public void setName(String name) {
       this.name = name;
   }

   public Employee(Integer id, Integer age, String name) {
       this.id = id;
       this.age = age;
       this.name = name;
   }
}
```

//EmployeeService Class

```
package com.bootcamp.demo.employee;

import org.springframework.stereotype.Component;

import java.util.ArrayList;
import java.util.List;

@Component
public class EmployeeService {
  private static List<Employee> emp=new ArrayList<>();

  static {
     emp.add(new Employee(1,23,"Gaurav"));
     emp.add(new Employee(2,33,"Rahul"));
     emp.add(new Employee(3,25,"Megha"));
        }
  public List<Employee> findAll(){
     return emp;
        }
  public Employee empAdd(Employee e){
     emp.add(e);
     return e;
  }
  public Employee findOne(int id){
     for(Employee e:emp){
        if(e.getId()==id) return e;
        }
     return null;
  }
}
```

//EmployeeRequest Class

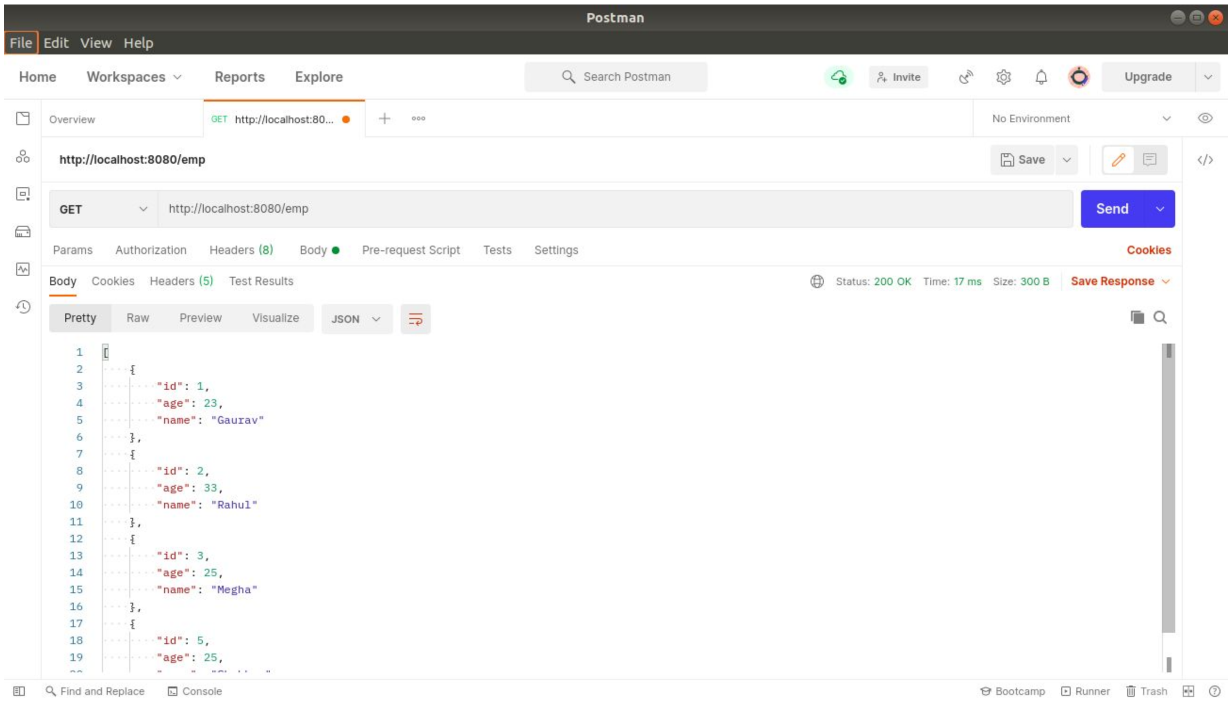
```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
public class EmployeeRequest {
  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
     return service.findAll();
  }

  @GetMapping(path = "emp/get-one/{id}")
  public Employee getOne(@PathVariable int id)
  {
     return service.findOne(id);
  }

  @PostMapping(path = "emp")
  public Employee add(@RequestBody Employee e){
     Employee a= service.empAdd(e);
     return e;
  }
}
```

3. Implement GET http request for Employee to get list of employees.

//EmployeeRequest Class

```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
public class EmployeeRequest {

  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
     return service.findAll();
  }
```

```
    @GetMapping(path = "emp/get-one/{id}")
    public Employee getOne(@PathVariable int id)
    {
        return service.findOne(id);
    }

    @PostMapping(path = "emp")
    public Employee add(@RequestBody Employee e){
        Employee a= service.empAdd(e);
        return e;
    }
}
```



Postman

GET http://localhost:8080/emp

Status: 200 OK Time: 17 ms Size: 300 B

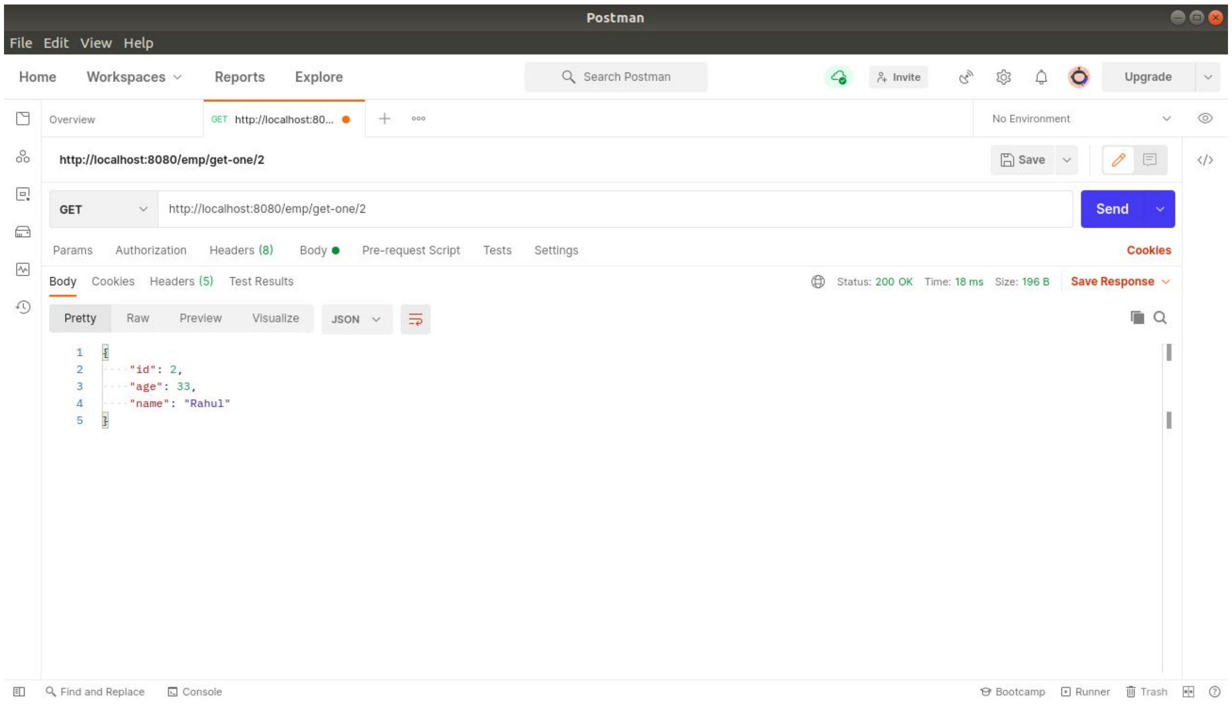
```
[
    {
        "id": 1,
        "age": 23,
        "name": "Gaurav"
    },
    {
        "id": 2,
        "age": 33,
        "name": "Rahul"
    },
    {
        "id": 3,
        "age": 25,
        "name": "Megha"
    },
    {
        "id": 5,
        "age": 25,
```

4. Implement GET http request using path variable to get one employee

//EmployeeRequest Class

```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
```

```
public class EmployeeRequest {

  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
      return service.findAll();
  }

  @GetMapping(path = "emp/get-one/{id}")
  public Employee getOne(@PathVariable int id)
  {
      return service.findOne(id);
  }

  @PostMapping(path = "emp")
  public Employee add(@RequestBody Employee e){
      Employee a= service.empAdd(e);
      return e;
  }
}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
GET http://localhost:80...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace Console
Bootcamp Runner Trash

http://localhost:8080/emp/get-one/2

GET http://localhost:8080/emp/get-one/2

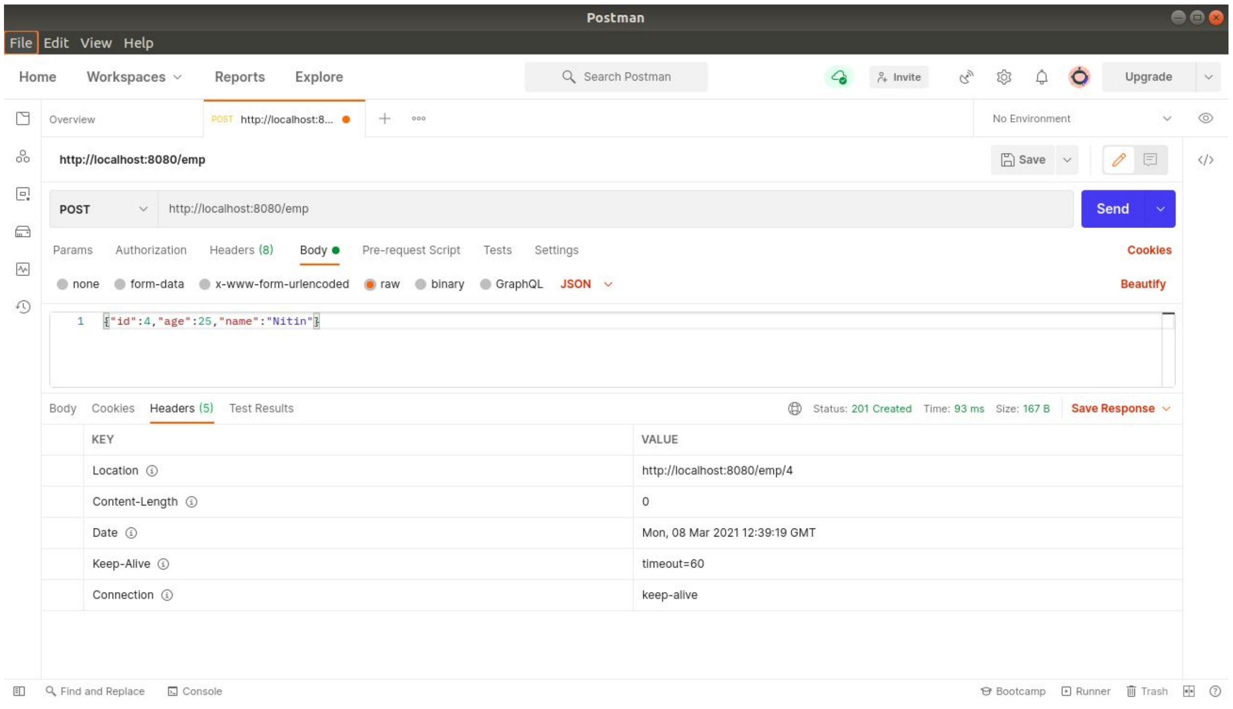
Status: 200 OK Time: 18 ms Size: 196 B

```
{
    "id": 2,
    "age": 33,
    "name": "Rahul"
}
```

5. Implement POST http request for Employee to create a new employee.

//EmployeeRequest Class

```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
public class EmployeeRequest {

  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
     return service.findAll();
  }

  @GetMapping(path = "emp/{id}")
  public Employee getOne(@PathVariable int id)
  {
     return service.findOne(id);
  }

  @PostMapping(path = "emp")
  public Employee add(@RequestBody Employee e){
     Employee a= service.empAdd(e);

URI location= ServletUriComponentsBuilder
.fromCurrentRequest().path("/{id}").buildAndExpand(a.getId()).toUri();

return ResponseEntity.created(location).build();

  }
}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
POST http://localhost:8...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (5) Test Results
Save Response
Find and Replace Console
Bootcamp Runner Trash

http://localhost:8080/emp

POST http://localhost:8080/emp

```
{"id":4,"age":25,"name":"Nitin"}
```

Status: 201 Created Time: 93 ms Size: 167 B

KEY	VALUE
Location	http://localhost:8080/emp/4
Content-Length	0
Date	Mon, 08 Mar 2021 12:39:19 GMT
Keep-Alive	timeout=60
Connection	keep-alive

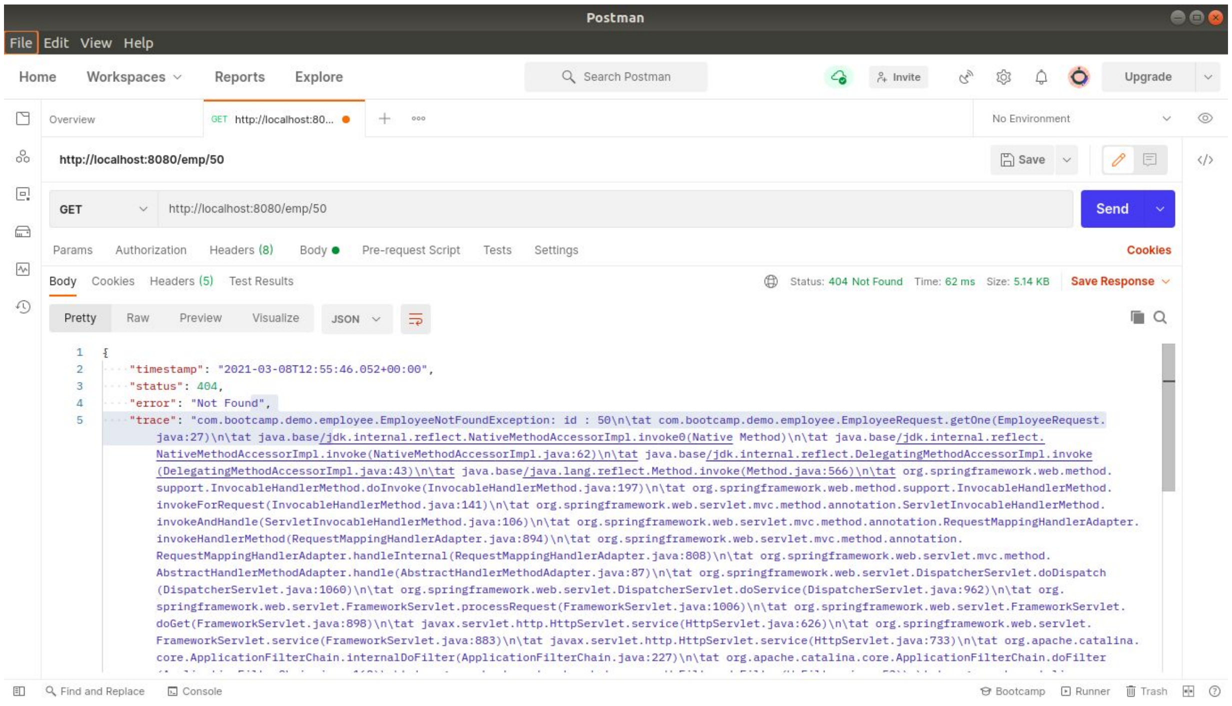
6. Implement Exception Handling for resource not found

//EmployeeRequest Class

```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.HttpClientErrorException;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import java.net.URI;
import java.util.List;

@RestController
public class EmployeeRequest {

  @Autowired
  private EmployeeService service;
  @GetMapping(path = "emp")
  public List<Employee> getAll(){
     return service.findAll();
  }

  @GetMapping(path = "emp/{id}")
  public Employee getOne(@PathVariable int id)  {
     Employee obj = service.findOne(id);
     if(obj==null){
        throw new EmployeeNotFoundException("id : "+id);
     }
     return obj;
  }

  @PostMapping(path = "emp")
  public ResponseEntity<Object> add(@RequestBody Employee e){
     Employee a= service.empAdd(e);

     URI location=
ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(a.getId()).toUri(
);

     return ResponseEntity.created(location).build();

  }

}
```

//EmployeeNotFoundException Class

```
package com.bootcamp.demo.employee;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(HttpStatus.NOT_FOUND)
public class EmployeeNotFoundException extends RuntimeException {
  public EmployeeNotFoundException(String s) {
     super(s);
  }
}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
GET http://localhost:80...
No Environment
Save
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace
Console
Bootcamp Runner Trash

http://localhost:8080/emp/50

GET http://localhost:8080/emp/50 Send

Status: 404 Not Found Time: 62 ms Size: 5.14 KB

```
{
    "timestamp": "2021-03-08T12:55:46.052+00:00",
    "status": 404,
    "error": "Not Found",
    "trace": "com.bootcamp.demo.employee.EmployeeNotFoundException: id : 50\n\tat com.bootcamp.demo.employee.EmployeeRequest.getOne(EmployeeRequest.
        java:27)\n\tat java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)\n\tat java.base/jdk.internal.reflect.
        NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)\n\tat java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke
        (DelegatingMethodAccessorImpl.java:43)\n\tat java.base/java.lang.reflect.Method.invoke(Method.java:566)\n\tat org.springframework.web.method.
        support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:197)\n\tat org.springframework.web.method.support.InvocableHandlerMethod.
        invokeForRequest(InvocableHandlerMethod.java:141)\n\tat org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.
        invokeAndHandle(ServletInvocableHandlerMethod.java:106)\n\tat org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.
        invokeHandlerMethod(RequestMappingHandlerAdapter.java:894)\n\tat org.springframework.web.servlet.mvc.method.annotation.
        RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:808)\n\tat org.springframework.web.servlet.mvc.method.
        AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87)\n\tat org.springframework.web.servlet.DispatcherServlet.doDispatch
        (DispatcherServlet.java:1060)\n\tat org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:962)\n\tat org.
        springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1006)\n\tat org.springframework.web.servlet.FrameworkServlet.
        doGet(FrameworkServlet.java:898)\n\tat javax.servlet.http.HttpServlet.service(HttpServlet.java:626)\n\tat org.springframework.web.servlet.
        FrameworkServlet.service(FrameworkServlet.java:883)\n\tat javax.servlet.http.HttpServlet.service(HttpServlet.java:733)\n\tat org.apache.catalina.
        core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:227)\n\tat org.apache.catalina.core.ApplicationFilterChain.doFilter
```

7. Implement DELETE http request for Employee to delete employee

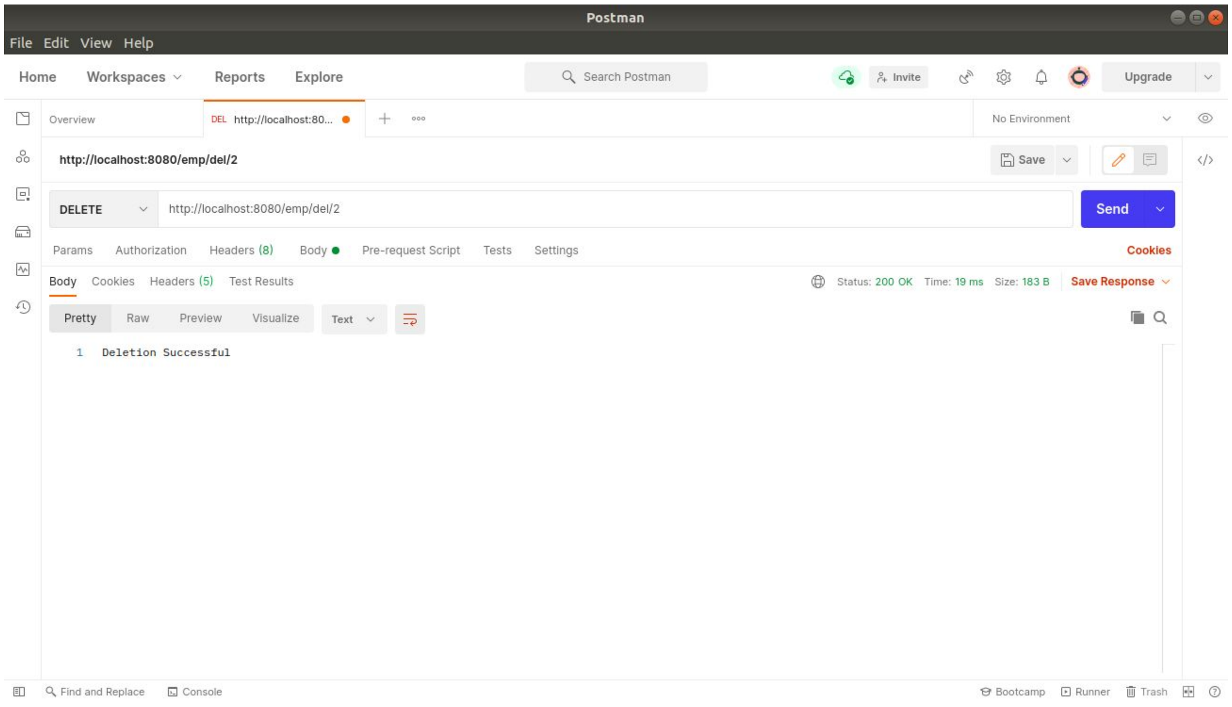
//EmployeeService

```
package com.bootcamp.demo.employee;
import org.springframework.stereotype.Component;
import java.util.ArrayList;
import java.util.List;

@Component
public class EmployeeService {
  private static List<Employee> emp=new ArrayList<>();

  static {
     emp.add(new Employee(1,23,"Gaurav"));
     emp.add(new Employee(2,33,"Rahul"));
     emp.add(new Employee(3,25,"Megha"));
        }
  public List<Employee> findAll(){
     return emp;
        }
  public Employee empAdd(Employee e){
     emp.add(e);
     return e;
  }
  public Employee findOne(int id){
     for(Employee e:emp){
        if(e.getId()==id) return e;

        }
     return null;
  }
  public String delEmployee(int id){
     for(Employee e:emp){
        if(e.getId()==id){
           emp.remove(e);
           return "Deletion Successful";
        }
     }
     return null;
  }
}
```

//EmployeeRequest

```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.HttpClientErrorException;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import javax.validation.Valid;
import java.net.URI;
import java.util.List;

@RestController
public class EmployeeRequest {

  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
     return service.findAll();
  }

  @GetMapping(path = "emp/{id}")
  public Employee getOne(@PathVariable int id)  {
     Employee obj = service.findOne(id);
     if(obj==null){
        throw new EmployeeNotFoundException("id : "+id);
     }
     return obj;
  }

  @PostMapping(path = "emp")
  public ResponseEntity<Object> add(@Valid @RequestBody Employee e){
     Employee a= service.empAdd(e);

     URI location=
ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(a.getId()).toUri(
);

     return ResponseEntity.created(location).build();
```

```
   }
   @DeleteMapping(path="emp/del/{id}")
   public String delEmp(@PathVariable int id){
       String s=service.delEmployee(id);
       if(s==null){
           throw new EmployeeNotFoundException("ID : "+id);
       }
       return s;
   }
}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
DEL http://localhost:80...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize Text
Find and Replace Console
Bootcamp Runner Trash

http://localhost:8080/emp/del/2

DELETE http://localhost:8080/emp/del/2

Status: 200 OK Time: 19 ms Size: 183 B

```
Deletion Successful
```

8. Implement PUT http request for Employee to update employee

//EmployeeService

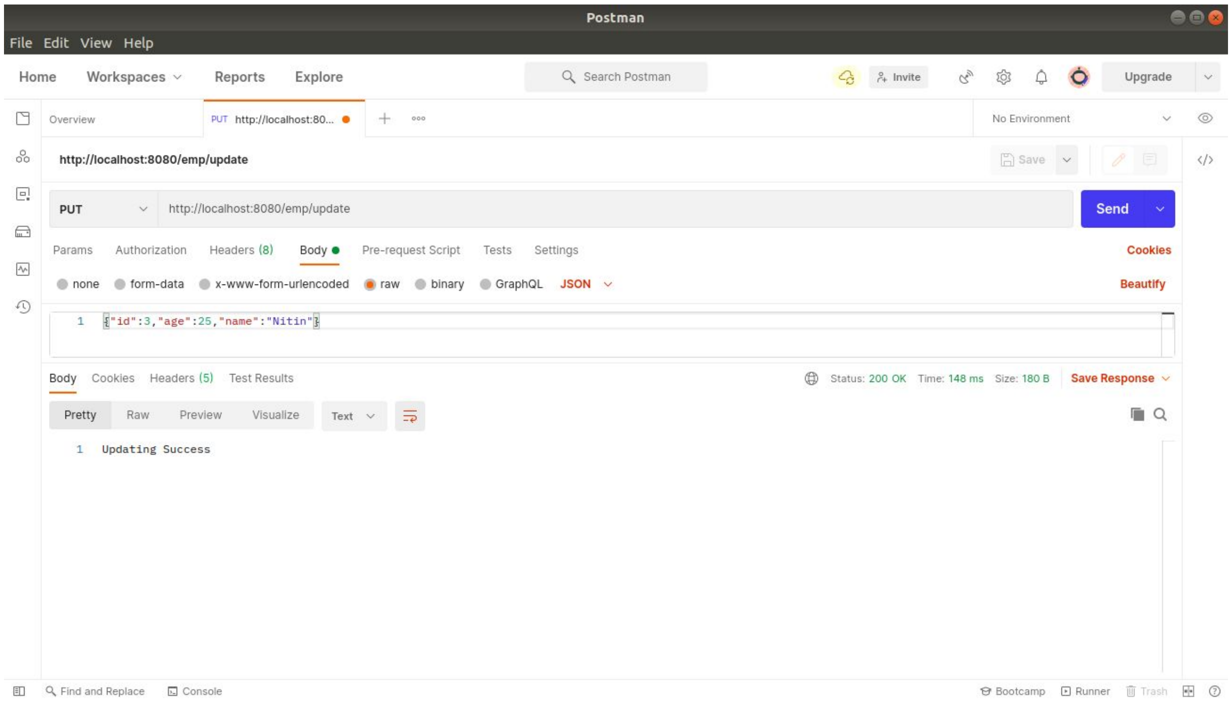
```
package com.bootcamp.demo.employee;

import org.springframework.stereotype.Component;

import java.util.ArrayList;
import java.util.List;

@Component
public class EmployeeService {
   private static List<Employee> emp=new ArrayList<>();

   static {
       emp.add(new Employee(1,23,"Gaurav"));
       emp.add(new Employee(2,33,"Rahul"));
       emp.add(new Employee(3,25,"Megha"));
           }
   public List<Employee> findAll(){
       return emp;
           }
   public Employee empAdd(Employee e){
       emp.add(e);
       return e;
   }
   public Employee findOne(int id){
       for(Employee e:emp){
           if(e.getId()==id) return e;

           }
       return null;
   }
   public String delEmployee(int id){
       for(Employee e:emp){
           if(e.getId()==id){
               emp.remove(e);
               return "Deletion Successful";
           }
       }
       return null;
   }
```

```
  public Employee updateEmployeeDetails(Employee employee){
      for(Employee ep1:emp){
          if(ep1.getId() == employee.getId()){
              ep1.setName(employee.getName());
              ep1.setAge(employee.getAge());
              return ep1;
          }
      }
      return null;
  }
}
```

//EmployeeRequest Class

```
package com.bootcamp.demo.employee;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.HttpClientErrorException;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import javax.validation.Valid;
import java.net.URI;
import java.util.List;

@RestController
public class EmployeeRequest {

  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
      return service.findAll();
  }

  @GetMapping(path = "emp/{id}")
  public Employee getOne(@PathVariable int id)  {
      Employee obj = service.findOne(id);
      if(obj==null){
          throw new EmployeeNotFoundException("id : "+id);
      }
```

```
        return obj;
    }

    @PostMapping(path = "emp")
    public ResponseEntity<Object> add(@Valid @RequestBody Employee e){
        Employee a= service.empAdd(e);

        URI location=
ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(a.getId()).toUri(
);

        return ResponseEntity.created(location).build();

    }
    @DeleteMapping(path="emp/del/{id}")
    public String delEmp(@PathVariable int id){
        String s=service.delEmployee(id);
        if(s==null){
            throw new EmployeeNotFoundException("ID : "+id);
        }
        return s;
    }

    @PutMapping("emp/update")
    public String updateEmployee(@Valid @RequestBody Employee employee) {
        Employee employee1 = service.updateEmployeeDetails(employee);
        if (employee1 == null)
            throw new EmployeeNotFoundException("ID not found : "+employee.getId());

        return "Updating Success";
    }

}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
PUT http://localhost:80...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize Text
Find and Replace Console
Bootcamp Runner Trash

http://localhost:8080/emp/update

PUT http://localhost:8080/emp/update

```
{"id":3,"age":25,"name":"Nitin"}
```

Status: 200 OK Time: 148 ms Size: 180 B

```
Updating Success
```

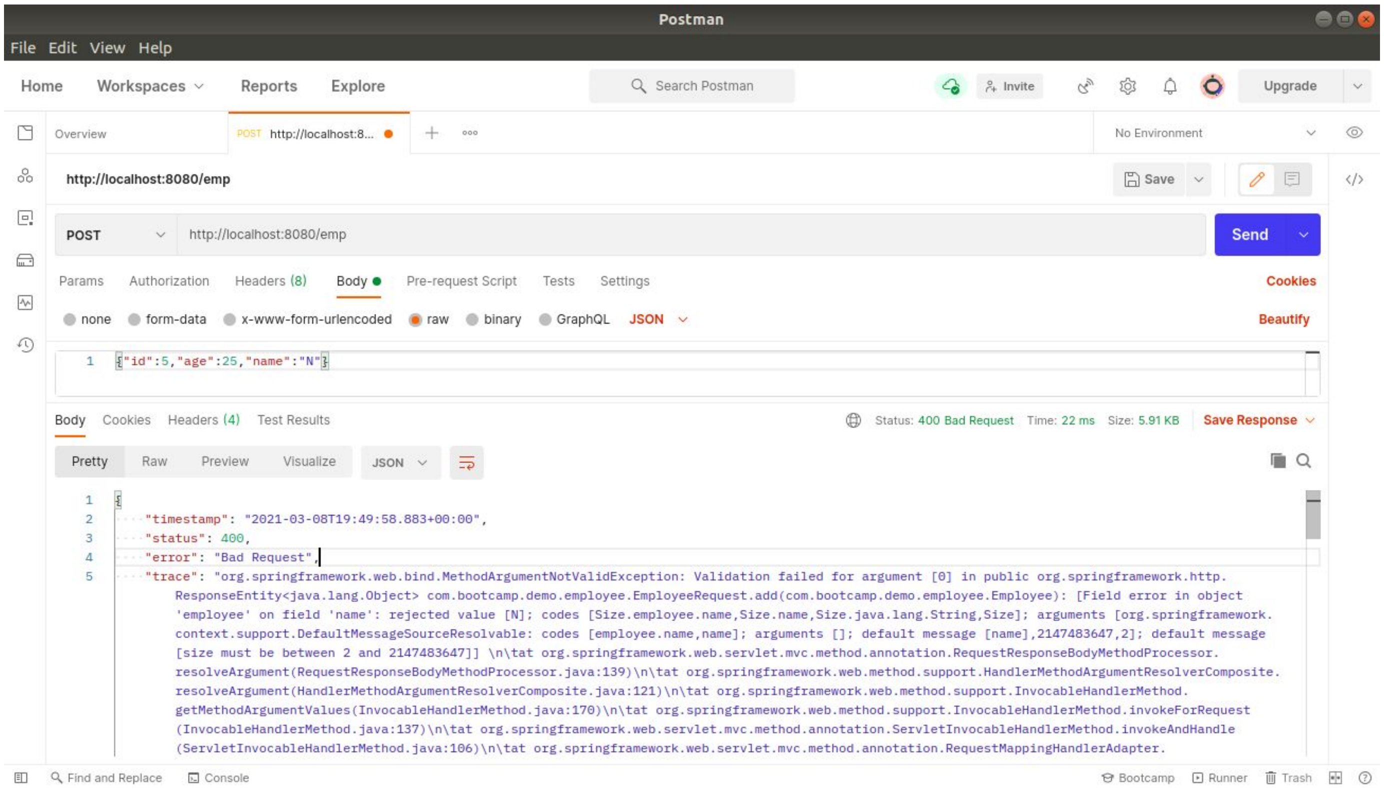
9. Apply validation while creating a new employee using POST http Request.

//EmployeeRequest Class

```
package com.bootcamp.demo.employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.HttpClientErrorException;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import javax.validation.Valid;
import java.net.URI;
import java.util.List;

@RestController
public class EmployeeRequest {
  @Autowired
  private EmployeeService service;

  @GetMapping(path = "emp")
  public List<Employee> getAll(){
     return service.findAll();
  }
  @GetMapping(path = "emp/{id}")
  public Employee getOne(@PathVariable int id)  {
     Employee obj = service.findOne(id);
     if(obj==null){
        throw new EmployeeNotFoundException("id : "+id);
     }
     return obj;
  }
  @PostMapping(path = "emp")
  public ResponseEntity<Object> add(@Valid @RequestBody Employee e){
     Employee a= service.empAdd(e);

     URI location= ServletUriComponentsBuilder
         .fromCurrentRequest()
         .path("/{id}")
         .buildAndExpand(a.getId()).toUri();
     return ResponseEntity.created(location).build();

  }
  @DeleteMapping(path="emp/del/{id}")
  public String delEmp(@PathVariable int id){
     String s=service.delEmployee(id);
     if(s==null){
        throw new EmployeeNotFoundException("ID : "+id);
     }
     return s;
  }
}
```

//Employee Class

```
package com.bootcamp.demo.employee;

import javax.validation.constraints.Min;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;

public class Employee {

@NotNull(message = "ID cannot be null")
@Min(value=0,message = "ID should be greater than zero")
private Integer id;

@Min(value=0,message = "Age should be greater than 0")
private Integer age;

@Size(min = 2,message = "Invalid Name ")
private String name;



   public Integer getId() {
       return id;
   }

   public void setId(Integer id) {
       this.id = id;
   }

   public Integer getAge() {
       return age;
   }

   public void setAge(Integer age) {
       this.age = age;
   }

   public String getName() {
       return name;
   }

   public void setName(String name) {
       this.name = name;
   }

   public Employee(Integer id, Integer age, String name) {
       this.id = id;
       this.age = age;
       this.name = name;
   }
}
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite Upgrade
Overview
POST http://localhost:8...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (4) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace Console
Bootcamp Runner Trash

http://localhost:8080/emp

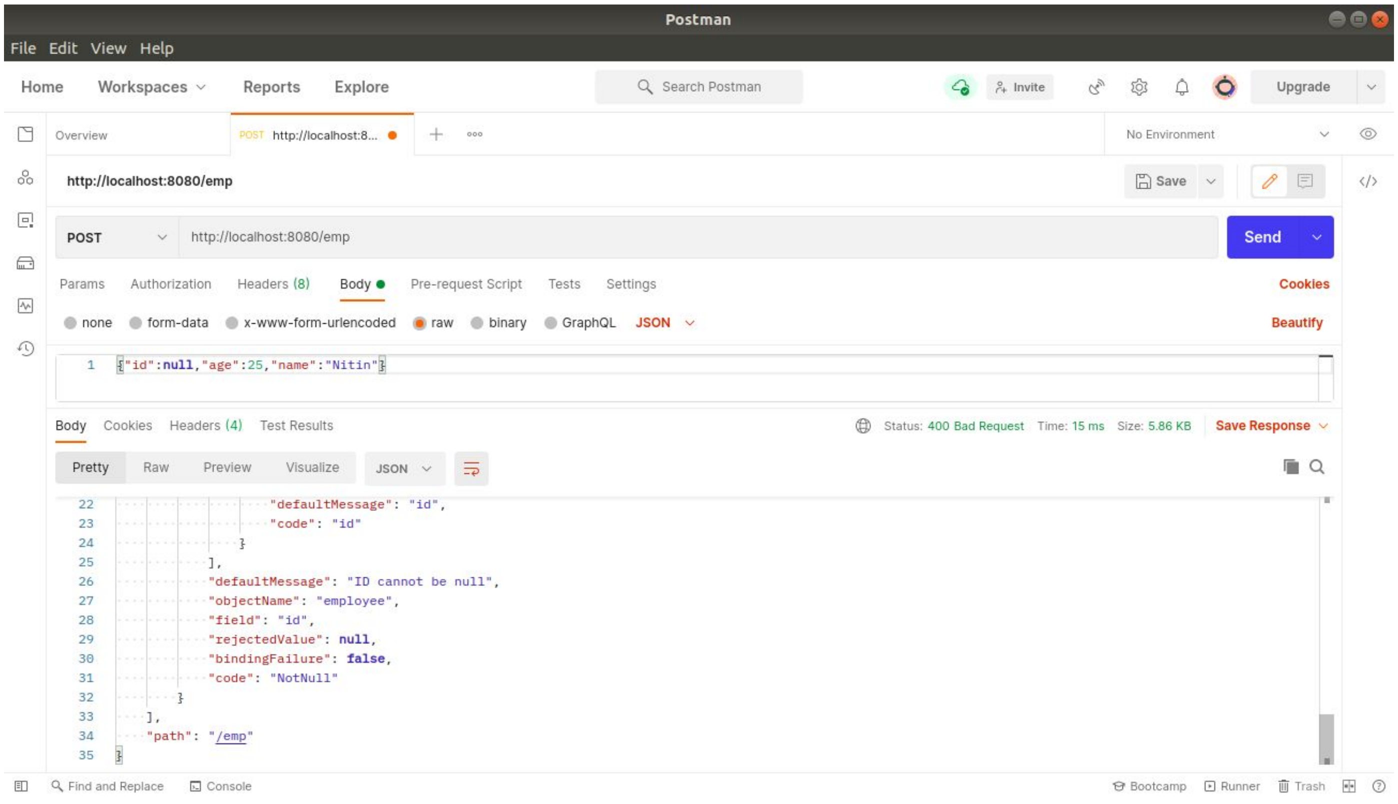
POST http://localhost:8080/emp

Body: raw, JSON

```
{"id":5,"age":25,"name":"N"}
```

Status: 400 Bad Request Time: 22 ms Size: 5.91 KB

```
{
    "timestamp": "2021-03-08T19:49:58.883+00:00",
    "status": 400,
    "error": "Bad Request",
    "trace": "org.springframework.web.bind.MethodArgumentNotValidException: Validation failed for argument [0] in public org.springframework.http.
        ResponseEntity<java.lang.Object> com.bootcamp.demo.employee.EmployeeRequest.add(com.bootcamp.demo.employee.Employee): [Field error in object
        'employee' on field 'name': rejected value [N]; codes [Size.employee.name,Size.name,Size.java.lang.String,Size]; arguments [org.springframework.
        context.support.DefaultMessageSourceResolvable: codes [employee.name,name]; arguments []; default message [name],2147483647,2]; default message
        [size must be between 2 and 2147483647]] \n\tat org.springframework.web.servlet.mvc.method.annotation.RequestResponseBodyMethodProcessor.
        resolveArgument(RequestResponseBodyMethodProcessor.java:139)\n\tat org.springframework.web.method.support.HandlerMethodArgumentResolverComposite.
        resolveArgument(HandlerMethodArgumentResolverComposite.java:121)\n\tat org.springframework.web.method.support.InvocableHandlerMethod.
        getMethodArgumentValues(InvocableHandlerMethod.java:170)\n\tat org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest
        (InvocableHandlerMethod.java:137)\n\tat org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle
        (ServletInvocableHandlerMethod.java:106)\n\tat org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite Upgrade
Overview
POST http://localhost:8...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (4) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace Console
Bootcamp Runner Trash

http://localhost:8080/emp

POST http://localhost:8080/emp

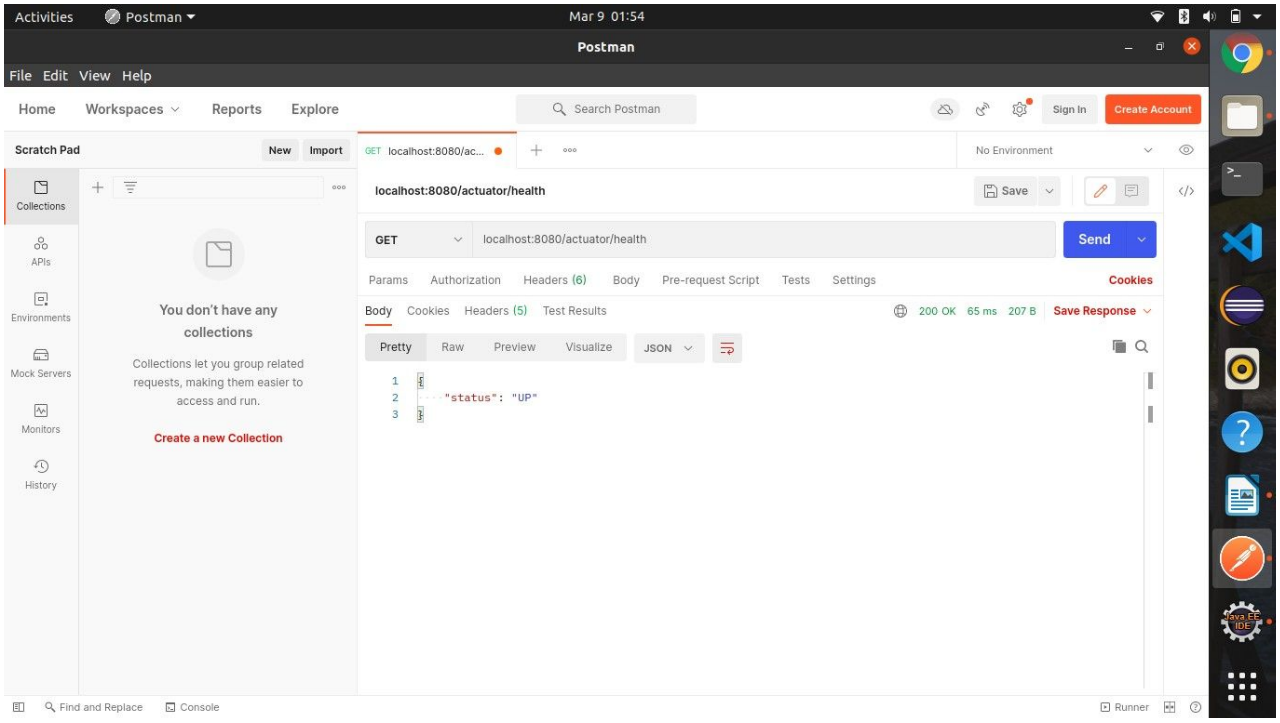
Body: raw, JSON

```
{"id":null,"age":25,"name":"Nitin"}
```

Status: 400 Bad Request Time: 15 ms Size: 5.86 KB

```
                        "defaultMessage": "id",
                        "code": "id"
                    }
                ],
                "defaultMessage": "ID cannot be null",
                "objectName": "employee",
                "field": "id",
                "rejectedValue": null,
                "bindingFailure": false,
                "code": "NotNull"
            }
        ],
    "path": "/emp"
}
```

10. **Configure actuator in your project to check the health of application and get the information about various beans configured in your application**


Activities
Postman
Mar 9 01:54
Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Sign In
Create Account
Scratch Pad
New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create a new Collection
No Environment
Save
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace
Console
Runner

GET localhost:8080/actuator/health

200 OK 65 ms 207 B

```
{
    "status": "UP"
}
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
  <version>2.3.1.RELEASE</version>
</dependency>
```