

q-1 Create a simple REST ful service in Spring Boot which returns the Response "Welcome to spring boot".

```
package com.example.demo.q1;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
public class as2q1 {
    @GetMapping(path="/Welcome-Page")
    public String welcome(){
        return "Welcome to Spring Boot";
    }
}
```



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

```
package com.example.demo.q2;
import javax.validation.constraints.Size;
public class q2emp {
    private Integer id;
    @Size (min=2)
    private String name;
    private Integer age;
    protected q2emp()
    {
    }
    public q2emp(Integer id, String name, Integer age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }
    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
    @Override
    public String toString() {
        return "q2emp{" +
            "id=" + id +
            ", name='" + name + '\'' +

```

```

        ", age=" + age +
        '}'';
    }
}

```

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

@Component

```

public class q2empdata {
    private static int usercount=3;
    private static List<q2emp> emp=new ArrayList<>();
    static
    {
        emp.add(new q2emp(1,"Siddharth",23));
        emp.add(new q2emp(2,"Mohan",25));
        emp.add(new q2emp(3,"Praveen",26));
    }
    public List<q2emp> findAll()
    {
        return emp;
    }
}

```

```

public class q2empmethods {
    @Autowired
    private q2empdata data;
    @GetMapping("/emp")
    public List<q2emp> retrieveAllEmp()
    {
        return data.findAll();
    }
}

```

The screenshot shows a REST client interface with the following details:

- Request:** GET http://localhost:8080/emp
- Status:** 200 OK, Time: 32ms, Size: 270 B
- Response Body (JSON):**

```

[
  {
    "id": 1,
    "name": "Siddharth",
    "age": 23
  },
  {
    "id": 2,
    "name": "Mohan",
    "age": 25
  },
  {
    "id": 3,
    "name": "Praveen",
    "age": 26
  }
]

```

q-4 Implement GET http request using path variable top get one employee

`@GetMapping("/find_one/{Top}")`

```
public q2emp retrieveOneEmp(@PathVariable int Top)
{
    q2emp a= data.find_One(Top);
    if(a==null)
    {
        throw new exception_handel("Top");
    }
    return a;
}
```

`public q2emp find_One(int id)`

```
{
    for(q2emp e : emp) {
        if(e.getId()==id)
        {
            return e;
        }
    }
    return null;
}
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/find\_one/1
- Status:** 200 OK
- Time:** 17ms
- Size:** 200 B
- Response Body (JSON):**

```
{
  "id": 1,
  "name": "Siddharth",
  "age": 23
}
```

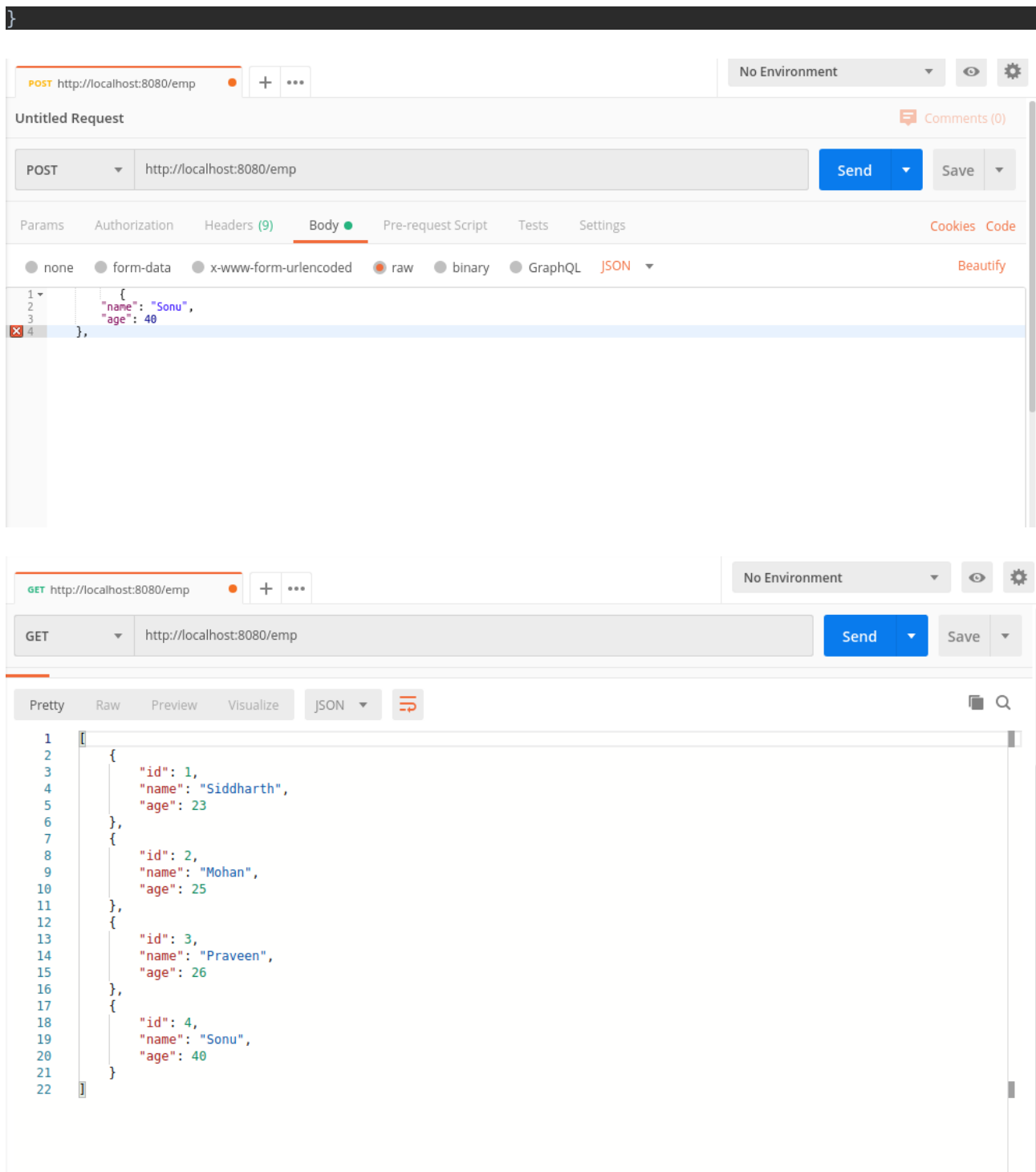
q-5 Implement POST http request for Employee to create a new employee.

`public q2emp save_emp(q2emp e)`

```
{
    if(e.getId()==null)
    {
        e.setId(++usercount);
    }
    emp.add(e);
    return e;
}
```

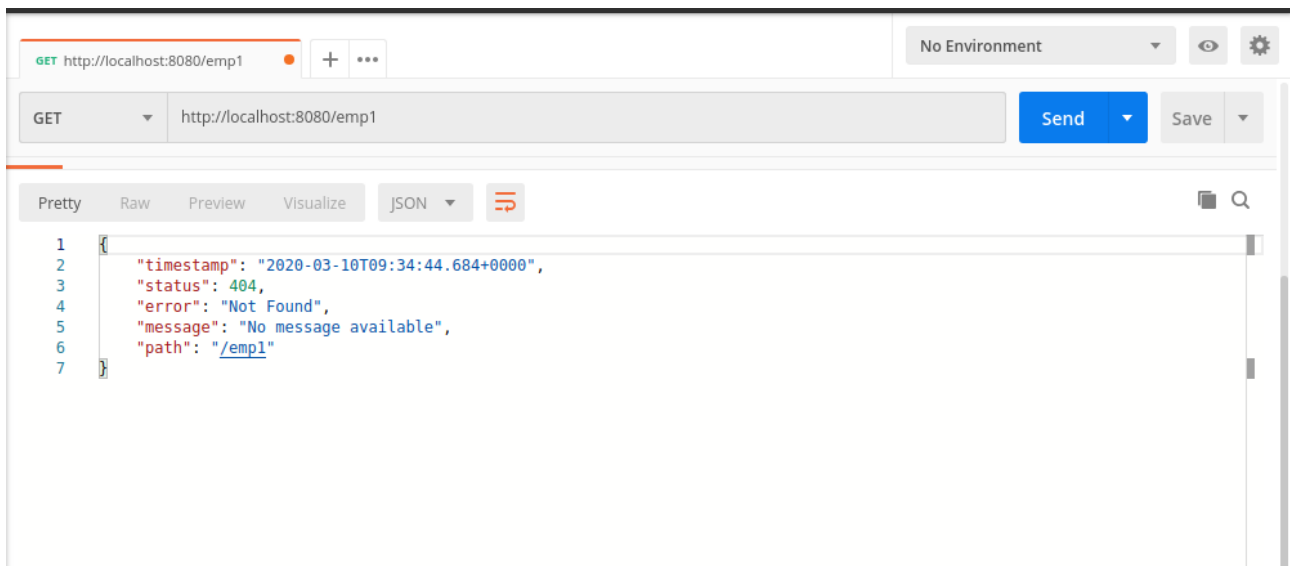
`@PostMapping("/emp")`

```
public void createEmp(@Valid @RequestBody q2emp e)
{
    q2emp saved=data.save_emp(e);
}
```



q-6 Imploement Exception Handling for resource not found

```
package com.example.demo.q2;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ResponseStatus;
@ResponseStatus(HttpStatus.NOT_FOUND)
public class exception_handel extends RuntimeException {
    public exception_handel(String message) {
        super(message);
    }
    /* @Override
    protected ResponseEntity<Object> myexception()*/
}
```



q-7 Implement DELETE http request for Employee to delete employee

```
public q2emp deleteById(int id)
```

```

{
    Iterator<q2emp> itr=emp.iterator();
    while(itr.hasNext())
    {
        q2emp emp=itr.next();
        if(emp.getId()==id)
        {
            itr.remove();
            return emp;
        }
    }
    return null;
}

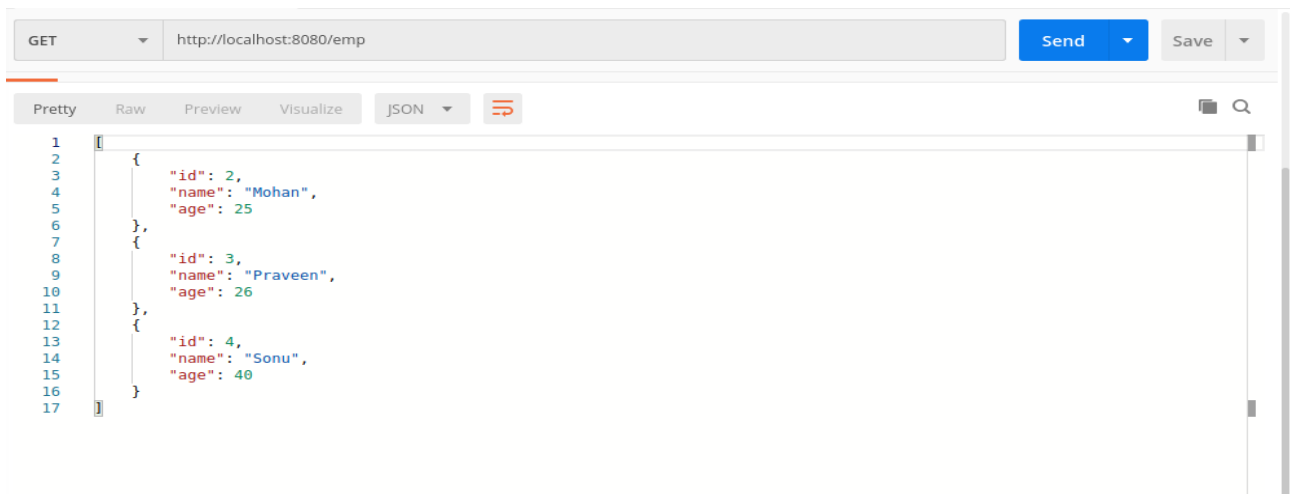
```

```
@DeleteMapping("/del/{id}")
```

```

public void deleteusr(@PathVariable int id)
{
    q2emp e=data.deleteById(id);
    if(e==null)
    {
        throw new exception_handel("id"+id);
    }
}

```



q-8 Implement PUT http request for Employee to update employee

q-9 Apply validation while create a new employee using POST http Request.

See q2

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

