

Rest API-CA1

Question No:1

Controller:

```
package com.example.demo.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.example.demo.model.Address;
import com.example.demo.repository.AddressRepo;

@RestController
public class ApiController {

    @Autowired
    AddressRepo repository;

    @GetMapping("/address")
    public List<Address> getAddress()
    {
        return repository.findAll();
    }

    @PostMapping("/address")
    public String addAddress(@RequestBody Address address)
    {
        repository.save(address);
        return "Added";
    }

    @GetMapping("/address/{address}")
    public List<Address> get(@PathVariable String address)
    {
        return repository.findByAddress(address);
    }
}
```

Model:

```
package com.example.demo.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;

@Entity
public class Address {

    private String employeeName;
    private String address;
    @Id
    private int pincode;
    public String getEmployeeName() {
        return employeeName;
    }
    public void setEmployeeName(String employeeName) {
        this.employeeName = employeeName;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public int getPincode() {
        return pincode;
    }
    public void setPincode(int pincode) {
        this.pincode = pincode;
    }
    @Override
    public String toString() {
        return "Address [employeeName=" + employeeName + ", address=" +
address + ", pincode=" + pincode + "]";
    }

}
```

Repository:

```
package com.example.demo.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.example.demo.model.Address;

@Repository
public interface AddressRepo extends JpaRepository<Address,Integer>{

    public List<Address> findAll();

    public List<Address> findByAddress(String address);

}
```

Output:

The screenshot displays a REST client interface. At the top, a dropdown menu shows 'GET' and the URL 'http://localhost:8080/address'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', and 'Pre-request Script'. The 'Body' tab is selected, and it shows a list of body types: 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (selected), and 'binary'. The body content is a JSON array with two objects, displayed in a code editor with line numbers 1 through 5. Below the code editor, there are tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results'. The 'Body' tab is selected, and it shows a 'Pretty' view of the JSON data. The JSON is a list of two objects, each with 'employeeName', 'address', and 'pincode' fields. The first object is for 'Ravi' in 'Coimbatore' with pincode '637212'. The second object is for 'Niki' in 'Trichy' with pincode '637521'. The 'Pretty' view is selected, and the JSON is displayed in a code editor with line numbers 1 through 12.

```
1 [
2   {
3     "employeeName": "Ravi",
4     "address": "Coimbatore",
5     "pincode": 637212
6   },
7   {
8     "employeeName": "Niki",
9     "address": "Trichy",
10    "pincode": 637521
11  }
12 ]
```

GET ⌵ http://localhost:8080/address/Trichy

Params Authorization Headers (9) Body ● Pre-r

Query Params

	KEY	VAL
	Key	Val

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ⌵

```
1  [
2    {
3      "employeeName": "Niki",
4      "address": "Trichy",
5      "pincode": 637521
6    }
7  ]
```

- 1 • create database cal;
- 2 • use cal;
- 3 • desc address;
- 4 • select * from address;

pincode	address	employee_name
637212	Coimbatore	Ravi
637521	Trichy	Niki
NULL	NULL	NULL

Question 2:

Controller:

```
package com.example.demo.controller;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import com.example.demo.model.Student;
import com.example.demo.service.ApiService;

@RestController
public class ApiController {

    @Autowired
    ApiService service;

    @PostMapping("/stu")
    public String addStudent(@RequestBody Student student)
    {
        return service.addStudent(student);
    }

    @GetMapping("/stu")
    public List<Student> read()
    {
        return service.getStudent();
    }

    @PutMapping("/stu")
    public Student update(@RequestBody Student student)
    {
        return service.updateStudent(student);
    }

    @DeleteMapping("/stu/{id}")
    public String delete(@PathVariable int id)
    {
        return service.deleteStudentById(id);
    }
}
```

Model:

```
package com.example.demo.model;
```

```

import jakarta.persistence.Entity;
import jakarta.persistence.Id;

@Entity
public class Student {
    @Id
    private int id;
    private String studentName;
    private String departmentName;
    private String section;
    private long mobile;
    private String mailid;
    private String address;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getStudentName() {
        return studentName;
    }
    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }
    public String getDepartmentName() {
        return departmentName;
    }
    public void setDepartmentName(String departmentName) {
        this.departmentName = departmentName;
    }
    public String getSection() {
        return section;
    }
    public void setSection(String section) {
        this.section = section;
    }
    public long getMobile() {
        return mobile;
    }
    public void setMobile(long mobile) {
        this.mobile = mobile;
    }
    public String getMailid() {
        return mailid;
    }
    public void setMailid(String mailid) {
        this.mailid = mailid;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    @Override
    public String toString() {
        return "Student [id=" + id + ", studentName=" + studentName +
            ", departmentName=" + departmentName

```

```

        + ", section=" + section + ", mobile=" + mobile +
        ", mailid=" + mailid + ", address=" + address + "];"
    }

}

```

Repository:

```

package com.example.demo.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.example.demo.model.Student;

@Repository
public interface StudentRepository extends JpaRepository<Student,Integer>{

}

```

Service:

```

package com.example.demo.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.demo.model.Student;
import com.example.demo.repository.StudentRepository;

@Service
public class ApiService {
    @Autowired
    StudentRepository repository;

    public String addStudent(Student student) {
        repository.save(student);
        return "Added";
    }

    public List<Student> getStudent() {
        return repository.findAll();
    }

    public Student updateStudent(Student student) {
        repository.save(student);
        return student;
    }
}

```

```

    }
    public String deleteStudentById(int id) {
        repository.deleteById(id);
        return "Deleted!...";
    }
}

```

Output:

Post:

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

- Method:** POST
- URL:** http://localhost:8080/stu
- Body Type:** raw (selected)
- Body Content:**

```

1  {
2    "id":1,
3    "studentName":"Rasika",
4    "departmentName":"IT",
5    "section":"C",
6    "mobile":9334003222,
7    "mailid":"rasika@gamil.com",
8    "address":"Namakkal"
9  }

```
- Response Section:**
 - Body:** Pretty (selected), Raw, Preview, Visualize
 - Text:** Text (selected)
 - Content:** 1 Added

Get:

GET



http://localhost:8080/stu

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (5)

Test Results



Pretty

Raw

Preview

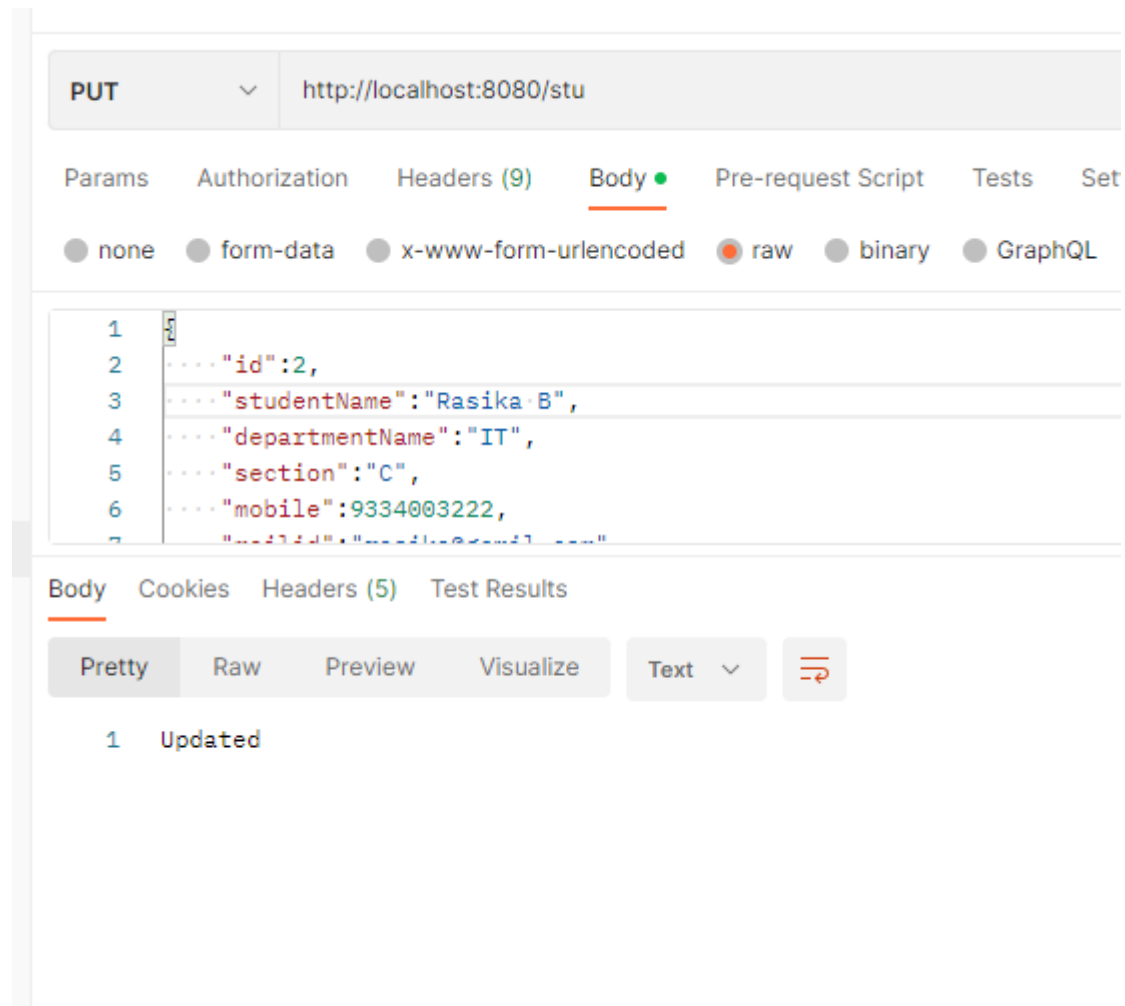
Visualize

JSON



```
1  [
2    {
3      "id": 1,
4      "studentName": "Rasika",
5      "departmentName": "IT",
6      "section": "C",
7      "mobile": 9334003222,
8      "mailid": "rasika@gamil.com",
9      "address": "Namakkal"
10   }
11 ]
```

Put:



Delete:

DELETE

▼

http://localhost:8080/stu/2

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Test Results

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text ▼

1

Deleted!!!

Question 3:

Controller:

```
package com.example.demo.controller;

import java.util.List;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.example.demo.model.Family;
import com.example.demo.repository.FamilyRepo;

@RestController
public class ApiController {

    @Autowired
    FamilyRepo repository;

    @GetMapping("/family")
    public List<Family> getFamily()
    {
        return repository.findAll();
    }

    @PostMapping("/family")
    public String addFamily(@RequestBody Family family)
    {
        repository.save(family);
        return "Added";
    }
}

```

Model:

```

package com.example.demo.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;

@Entity
public class Family {
    @Id
    private int id;
    private String name;
    private int age;
    private String relation;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
}

```

```

    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getRelation() {
        return relation;
    }
    public void setRelation(String relation) {
        this.relation = relation;
    }
    @Override
    public String toString() {
        return "Family [id=" + id + ", name=" + name + ", age=" + age +
            ", relation=" + relation + "]";
    }
}

```

Repository:

```

package com.example.demo.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.example.demo.model.Family;

public interface FamilyRepo extends JpaRepository<Family, Integer>{

    public List<Family> findAll();

}

```

Output:

Post:

Get:

GET

http://localhost:8080/family

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

[

{

"id": 1,

"name": "Rasika B",

"age": 20,

"relation": "Student"

}

]

Question 4:

Controller:

```
package com.example.demo.controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```

@RestController
public class Name {
    private String YourName="Rasika";
    @RequestMapping("/name")
    public String name()
    {
        return "Hi "+YourName+"!";
    }
}

```

Output:

The screenshot shows a REST client interface. At the top, a GET request is configured for the URL `http://localhost:8080/name`. Below this, several tabs are visible: **Params**, **Authorization**, **Headers (7)**, **Body**, **Pre-request Script**, and **Test**. The **Params** tab is selected, showing a table for Query Params with columns **KEY** and **VALUE**. The table contains one entry: **Key** and **Value**. Below the table, there are tabs for **Body**, **Cookies**, **Headers (5)**, and **Test Results**. The **Body** tab is selected, showing a text input field with the value `Hi Rasika!`. The **Text** format is selected for the response, and a refresh button is visible.

Question 5:

Controller:

```
package com.example.demo.controller;
```



```

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import com.example.demo.model.Student;
import com.example.demo.service.ApiService;

@RestController
public class ApiController {

    @Autowired
    ApiService service;

    @PostMapping("/stu")
    public String addStudent(@RequestBody Student student)
    {
        return service.addStudent(student);
    }

    @GetMapping("/stu")
    public List<Student> read()
    {
        return service.getStudent();
    }

}

```

Model:

```

package com.example.demo.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;

@Entity
public class Student {
    @Id
    private int id;
    private String studentName;
    private String departmentName;
    private String section;
    private long mobile;
    private String mailid;
    private String address;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getStudentName() {
        return studentName;
    }
}

```

```

    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }
    public String getDepartmentName() {
        return departmentName;
    }
    public void setDepartmentName(String departmentName) {
        this.departmentName = departmentName;
    }
    public String getSection() {
        return section;
    }
    public void setSection(String section) {
        this.section = section;
    }
    public long getMobile() {
        return mobile;
    }
    public void setMobile(long mobile) {
        this.mobile = mobile;
    }
    public String getMailid() {
        return mailid;
    }
    public void setMailid(String mailid) {
        this.mailid = mailid;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    @Override
    public String toString() {
        return "Student [id=" + id + ", studentName=" + studentName +
", departmentName=" + departmentName
        + ", section=" + section + ", mobile=" + mobile +
", mailid=" + mailid + ", address=" + address + "]\n";
    }
}

```

Repository:

```

package com.example.demo.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.example.demo.model.Student;

@Repository
public interface StudentRepository extends JpaRepository<Student,Integer>{

```

```
}
```

Service:

```
package com.example.demo.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.demo.model.Student;
import com.example.demo.repository.StudentRepository;

@Service
public class ApiService {
    @Autowired
    StudentRepository repository;

    public String addStudent(Student student) {
        repository.save(student);
        return "Added";
    }

    public List<Student> getStudent() {
        return repository.findAll();
    }
}
```

Output:

Get:

GET

▼

http://localhost:8080/stu

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

≡

```
1  [
2    {
3      "id": 1,
4      "studentName": "Rasika",
5      "departmentName": "IT",
6      "section": "C",
7      "mobile": 9334003222,
8      "mailid": "rasika@gamil.com",
9      "address": "Namakkal"
10   }
11 ]
```

Post:

POST

▼

http://localhost:8080/stu

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Snippets

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

```
1 {
2   ... "id":1,
3   ... "studentName":"Rasika",
4   ... "departmentName":"IT",
5   ... "section":"C",
6   ... "mobile":9334003222,
7   ... "mailid":"rasika@gamil.com",
8   ... "address":"Namakkal"
9 }
```

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text ▼

↺

1

Added

