

Microservices using Spring boot

PersonEntity.java:

```
package com.example.test;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class PersonEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", updatable = false, nullable = false)
    private Integer personId;
    @Column
    private String name;
    @Column
    private Integer age;

    public PersonEntity() {
        super();
    }
    public PersonEntity(Integer personId, String name, Integer age) {
        super();
        this.personId = personId;
        this.name = name;
        this.age = age;
    }
    public Integer getPersonId() {
        return personId;
    }
    public void setPersonId(Integer personId) {
        this.personId = personId;
    }
    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;
    }
}
```

```
}
```

PersonController.java:

```
package com.example.test;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path = "/webapione")
public class PersonController {

    @Autowired
    PersonService personService;

    @RequestMapping("/person/{personId}")
    public PersonResponse getPerson(@PathVariable int personId) {
        return personService.getPerson(personId);
    }

    @RequestMapping(method=RequestMethod.POST, value="/person")
    public void addPerson(@RequestBody PersonEntity pe ) {
        personService.addPerson(pe);
    }
}
```

PersonRepository.java:

```
package com.example.test;

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

@Repository
public interface PersonRepository extends JpaRepository<PersonEntity, Integer> {

}
```

PersonResponse.java:

```
package com.example.test;

public class PersonResponse {

    private Integer personId;
    private String name;
    private Integer age;
    private String hobby;
    public Integer getPersonId() {
        return personId;
    }
}
```

```

    }
    public void setPersonId(Integer personId) {
        this.personId = personId;
    }
    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;
    }
    public String getHobby() {
        return hobby;
    }
    public void setHobby(String result) {
        this.hobby = result;
    }
}

```

PersonService.java:

```

package com.example.test;

import java.util.HashMap;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

@Service
public class PersonService {

    @Autowired
    PersonRepository personRepository;

    RestTemplate restTemplate = new RestTemplate();

    public PersonResponse getPerson(int personId){
        final String uri =
            "http://localhost:8082/webapitwo/hobby/{personId}";

        Map<String, Integer> params = new HashMap<String, Integer>();
        params.put("personId", personId);

        String result = restTemplate.getForObject(uri, String.class,
params);
        PersonEntity pe=personRepository.findById(personId).get();
        PersonResponse pr=new PersonResponse();
        pr.setPersonId(pe.getPersonId());
        pr.setName(pe.getName());
        pr.setAge(pe.getAge());
        pr.setHobby(result);
    }
}

```

```

        return pr;
    }
    public void addPerson(PersonEntity pe){
        personRepository.save(pe);
    }
}

```

application.prpoerties:

```

server.port=8081
spring.application.name=RestApiOne

```

RestApiTwo Spring boot application:

HobbyEntity.java:

```

package com.example.test;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class HobbyEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", updatable = false, nullable = false)
    private Integer id;
    @Column
    private Integer personId;
    @Column
    private String name;

    public HobbyEntity() {
        super();
    }
    public HobbyEntity(Integer personId, String name) {
        super();
        this.personId = personId;
        this.name = name;
    }
    public Integer getPersonId() {
        return personId;
    }
    public void setPersonId(Integer personId) {
        this.personId = personId;
    }
    public String getName() {
        return name;
    }
}

```

```

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

```

HobbyController.java:

```

package com.example.test;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path = "/webapitwo")
public class HobbyController {

    @Autowired
    HobbyService hobbyService;

    @RequestMapping("/hobby/{personid}")
    public String findByPersonId(@PathVariable int personid) {
        return hobbyService.findByPersonId(personid);
    }

    @RequestMapping(method=RequestMethod.POST, value="/hobby")
    public void addHobby(@RequestBody HobbyEntity he ) {
        hobbyService.addHobby(he);
    }

}

```

HobbyRepository.java:

```

package com.example.test;

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

@Repository
public interface HobbyRepository extends JpaRepository<HobbyEntity, Integer> {

    @Query("SELECT h.name FROM HobbyEntity h WHERE h.personId=:personId")
    public String findByPersonId(Integer personId);

}

```

HobbyService.java:

```

package com.example.test;

```

```

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

@Service
public class HobbyService {

    @Autowired
    HobbyRepository hobbyRepository;

    public String findByPersonId(int personid) {
        return hobbyRepository.findByPersonId(personid);
    }
    public void addHobby(HobbyEntity he) {
        hobbyRepository.save(he);
    }
}

```

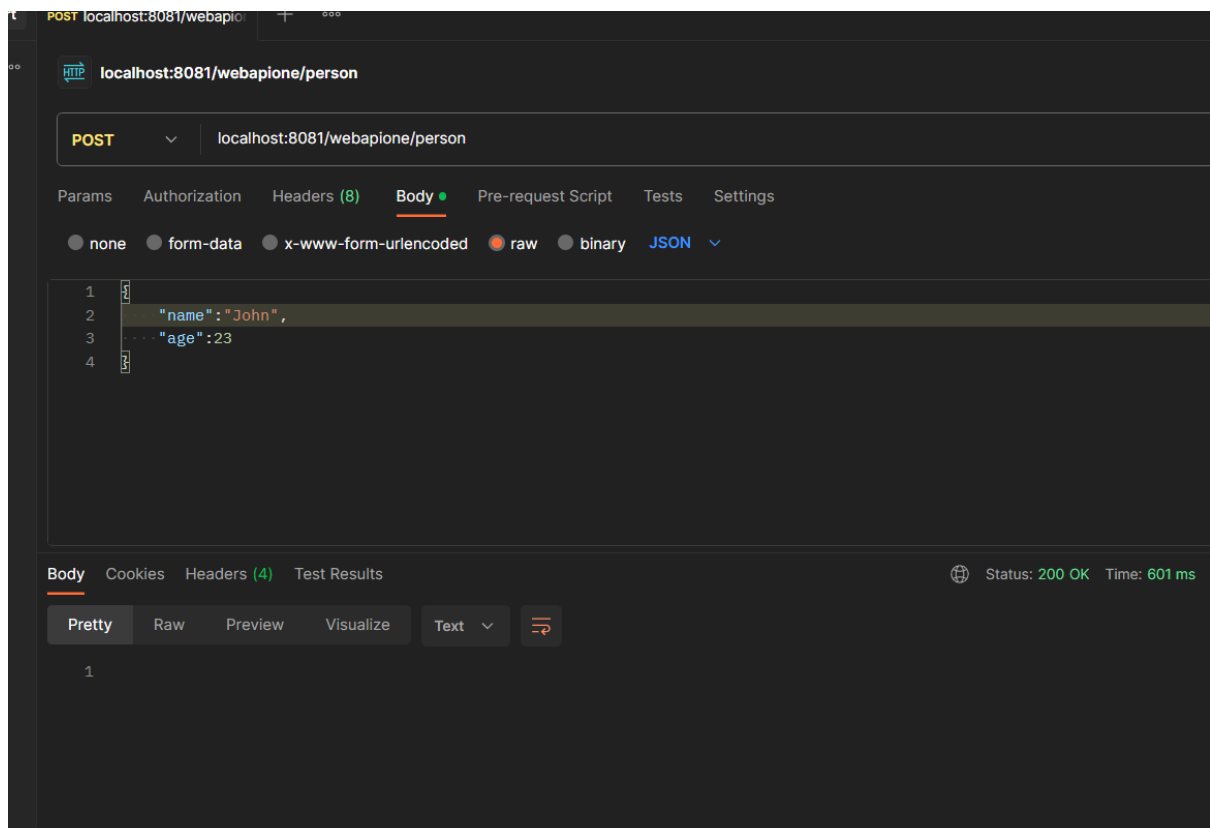
application.properties:

```

server.port=8082
spring.application.name=RestApiTwo

```

Output:



POST localhost:8082/webapi/

localhost:8082/webapitwo/hobby

POST localhost:8082/webapitwo/hobby

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "personId": 1,
3   "name": "Badminton"
4 }
5
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 640 ms

Pretty Raw Preview Visualize Text

1

POST localhost:8082/webapi/ POST localhost:8081/webapi/

localhost:8081/webapione/person

POST localhost:8081/webapione/person/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

Key	Value
-----	-------

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 374 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "personId": 1,
3   "name": "John",
4   "age": 23,
5   "hobby": "Badminton"
6 }
```

POST localhost:8082/webapitwo/hobby

POST localhost:8081/webapitwo/hobby

POST localhost:8082/webapitwo/hobby

+

...

localhost:8082/webapitwo/hobby

POST localhost:8082/webapitwo/hobby/1

ParamsAuthorizationHeaders (8)Body ●Pre-request ScriptTestsSettings

Query Params

	Key	Value
	Key	Value

BodyCookiesHeaders (5)Test Results

Status: 200 OK

PrettyRawPreviewVisualizeText

1 Badminton