Spring REST Service - Students

Create a New Service

Create a New Service

- Return a list of students

Create a New Service

- Return a list of students

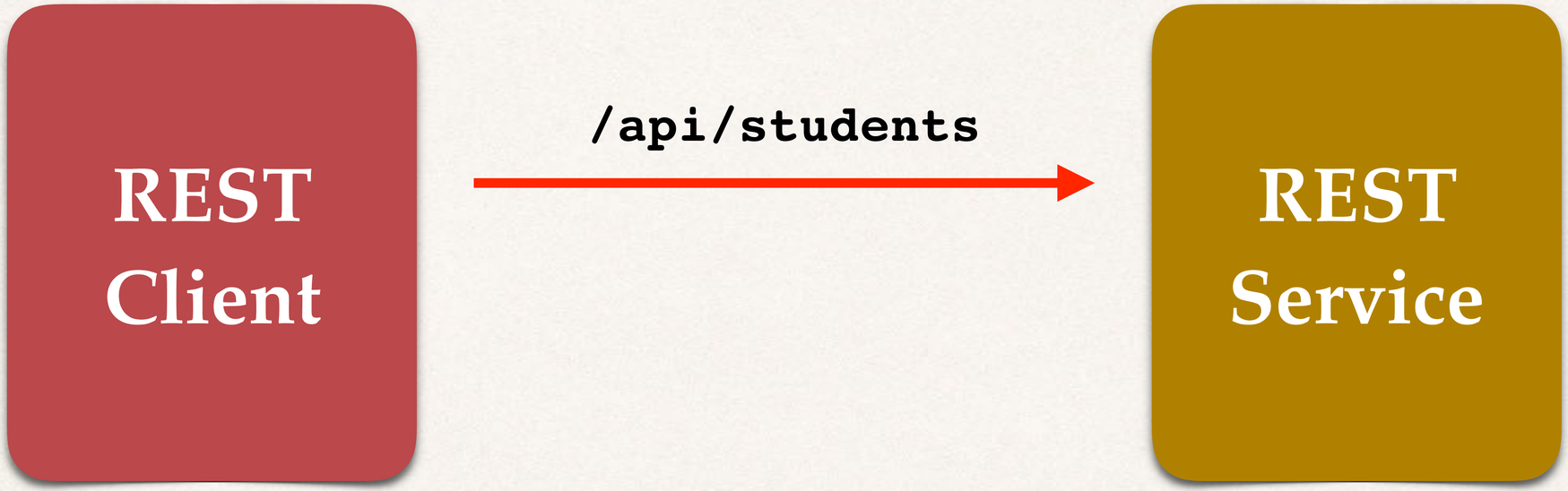
GET `/api/students` *Returns a list of students*

Spring REST Service

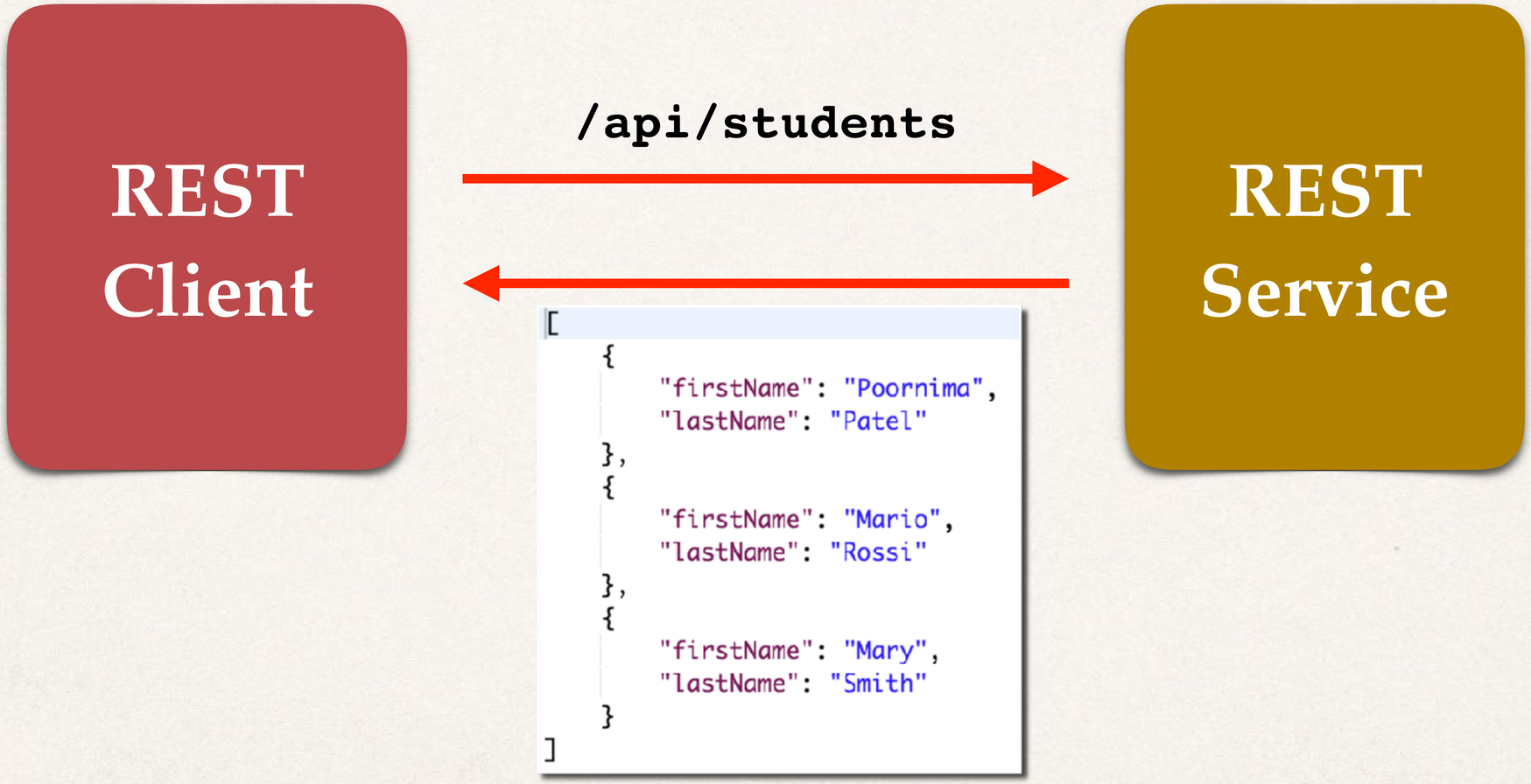
Spring REST Service

Spring REST Service

Spring REST Service



Spring REST Service



Spring REST Service

We will write this code

REST Client

/api/students

REST Service

Spring REST Service

We will write this code

REST Client

/api/students

REST Service

Web Browser Or Postman

Convert Java POJO to JSON

Convert Java POJO to JSON

- Our REST Service will return **`List<Student>`**

Convert Java POJO to JSON

- Our REST Service will return **`List<Student>`**

- Need to convert **`List<Student>`** to JSON

Convert Java POJO to JSON

- Our REST Service will return **`List<Student>`**
- Need to convert **`List<Student>`** to JSON
- **Jackson** can help us out with this …

Spring and Jackson Support

Remember

Spring and Jackson Support

- Spring will automatically handle **Jackson** integration

Spring and Jackson Support

Remember

- Spring will automatically handle **Jackson** integration

- As long as the **Jackson** project is on the classpath or pom.xml

Spring and Jackson Support

- Spring will automatically handle **Jackson** integration
- As long as the **Jackson** project is on the classpath or pom.xml
- JSON data being passed to REST controller is converted to Java POJO

Spring and Jackson Support

- Spring will automatically handle **Jackson** integration
- As long as the **Jackson** project is on the classpath or pom.xml
- JSON data being passed to REST controller is converted to Java POJO
- Java POJO being returned from REST controller is converted to JSON

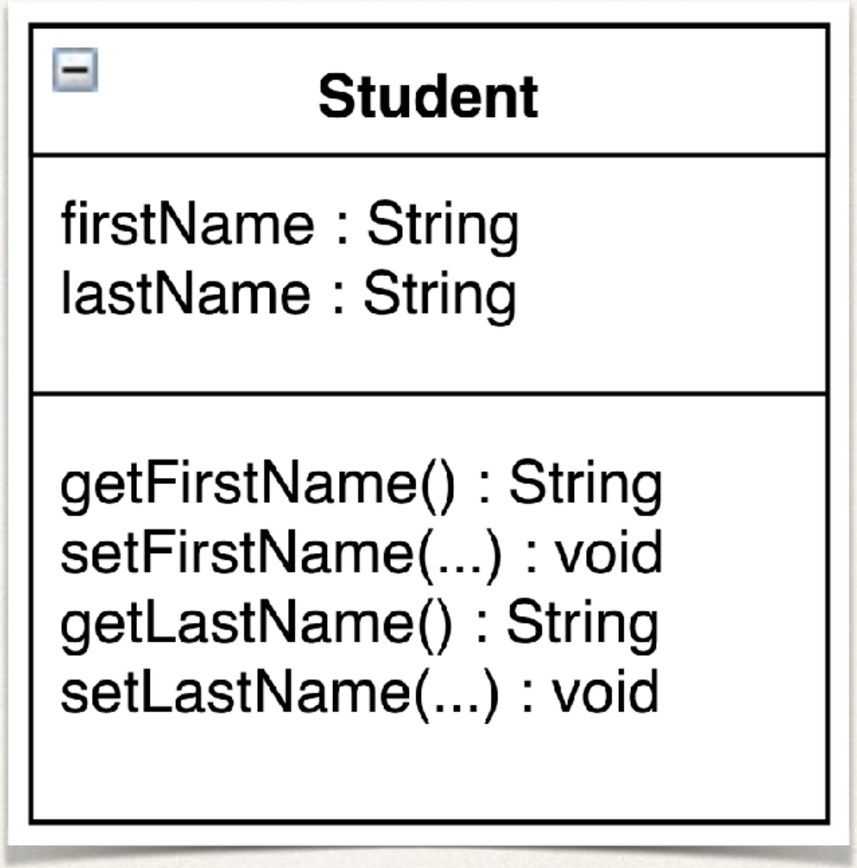
Spring and Jackson Support

Happens automatically behind the scenes

- Spring will automatically handle **Jackson** integration
- As long as the **Jackson** project is on the classpath or pom.xml
- JSON data being passed to REST controller is converted to Java POJO
- Java POJO being returned from REST controller is converted to JSON

Student POJO (class)

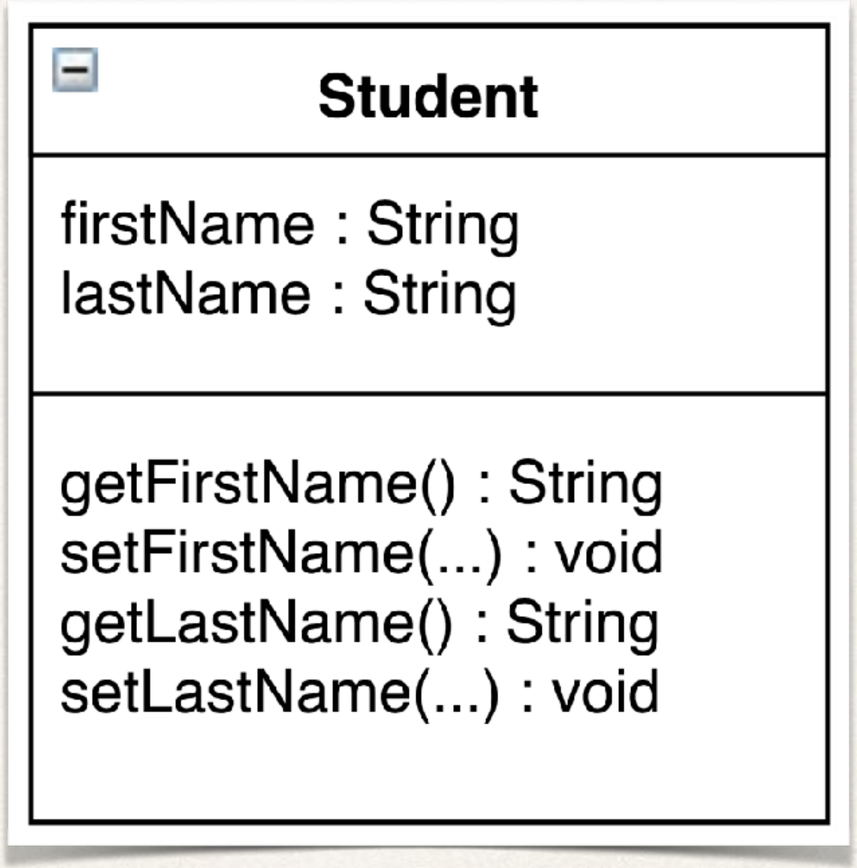
Student POJO (class)



Student POJO (class)

Java
POJO

Student



Jackson Data Binding

Remember

Jackson Data Binding

- Jackson will call appropriate getter/setter method

Jackson Data Binding

Remember

- Jackson will call appropriate getter/setter method

```
{
  "id": 14,

  "firstName": "Mario",

  "lastName": "Rossi",

  "active": true
}
```

Jackson Data Binding

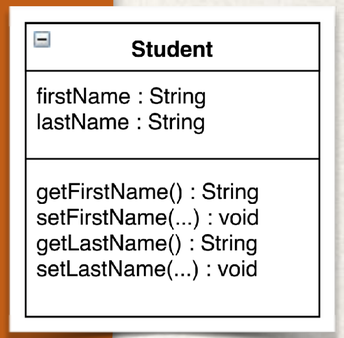
Remember

- Jackson will call appropriate getter/setter method

```
{
  "id": 14,

  "firstName": "Mario",

  "lastName": "Rossi",

  "active": true
}
```

Java
POJO

Student



Jackson Data Binding

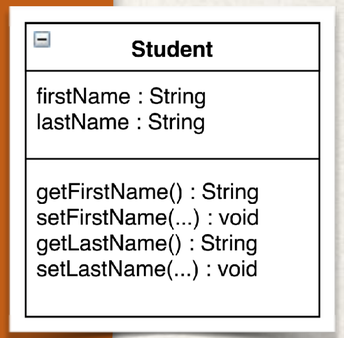
Remember

- Jackson will call appropriate getter/setter method

```
{
  "id": 14,

  "firstName": "Mario",

  "lastName": "Rossi",

  "active": true
}
```

Java
POJO

Student



Jackson Data Binding

Remember

- Jackson will call appropriate getter/setter method

```
{
  "id": 14,

  "firstName": "Mario",

  "lastName": "Rossi",

  "active": true
}
```

Java
POJO

Student

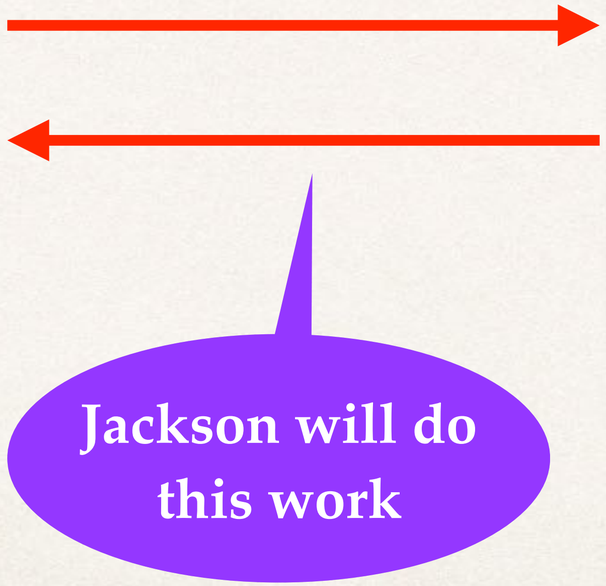
Student
firstName : String lastName : String
getFirstName() : String setFirstName(...) : void getLastName() : String setLastName(...) : void

Jackson Data Binding

Remember

- Jackson will call appropriate getter/setter method

```
{
  "id": 14,

  "firstName": "Mario",

  "lastName": "Rossi",

  "active": true
}
```



Java POJO

Student

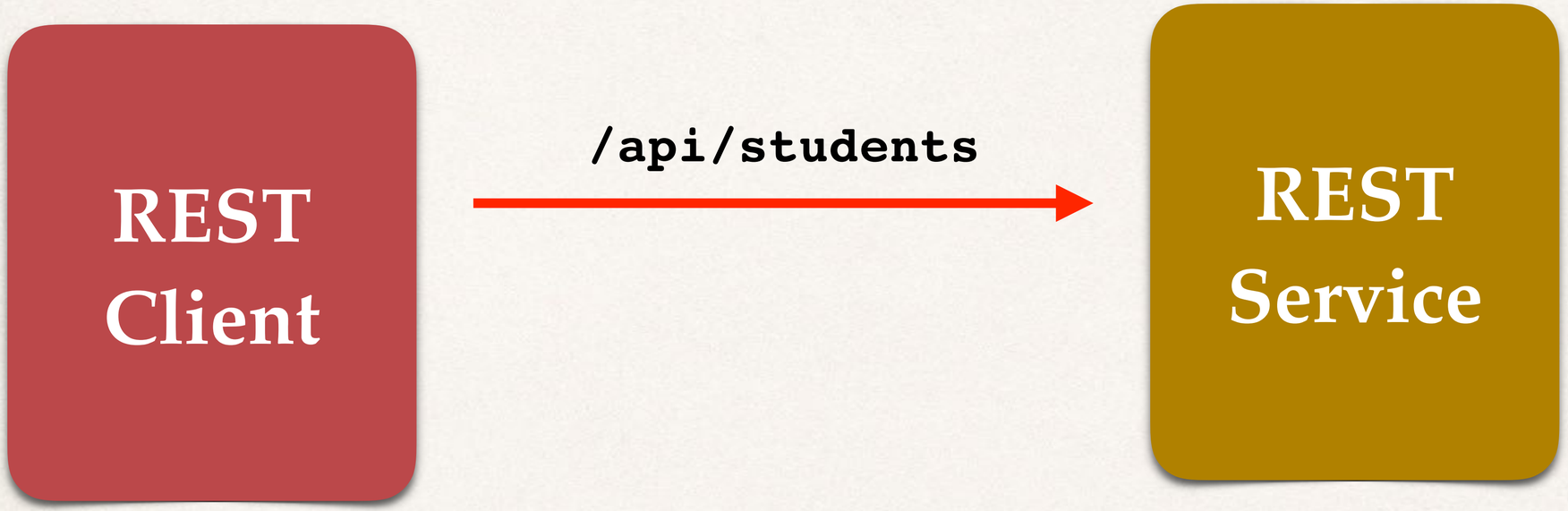
Student
firstName : String lastName : String
getFirstName() : String setFirstName(...) : void getLastName() : String setLastName(...) : void

Spring REST Service

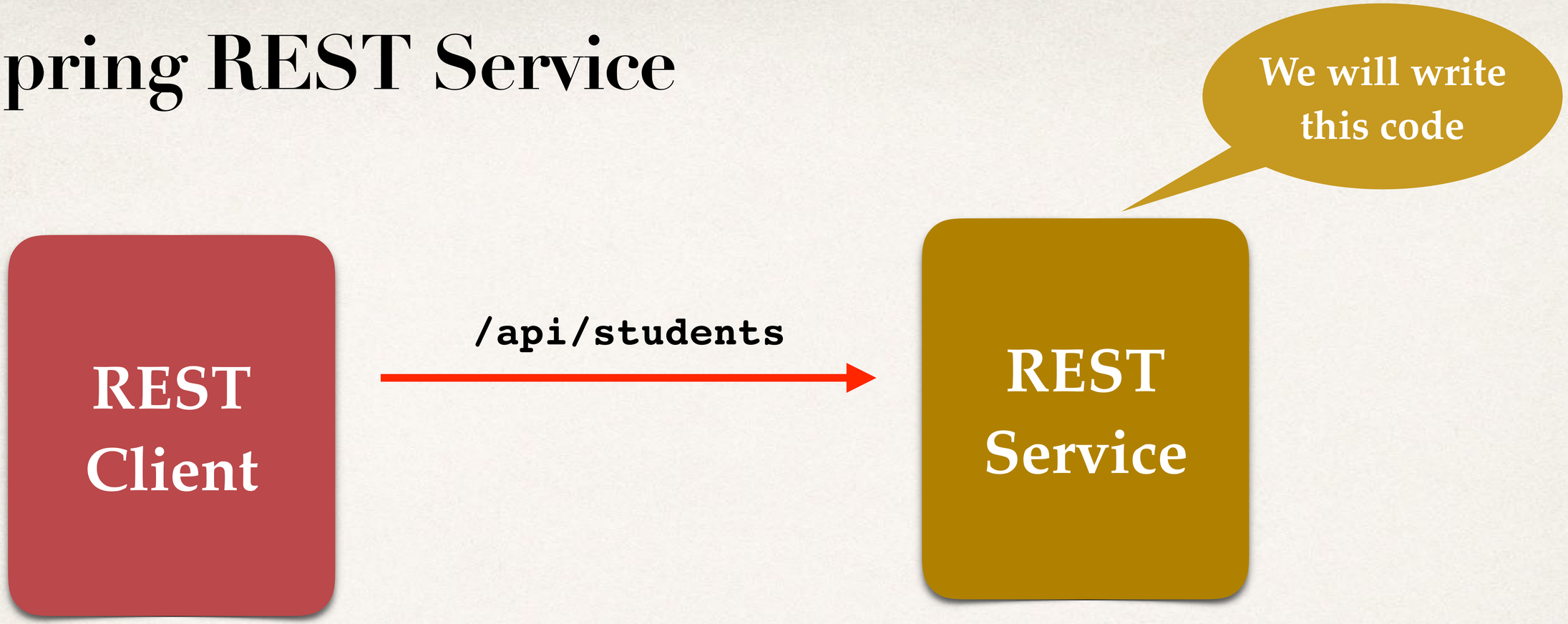
Spring REST Service

Spring REST Service

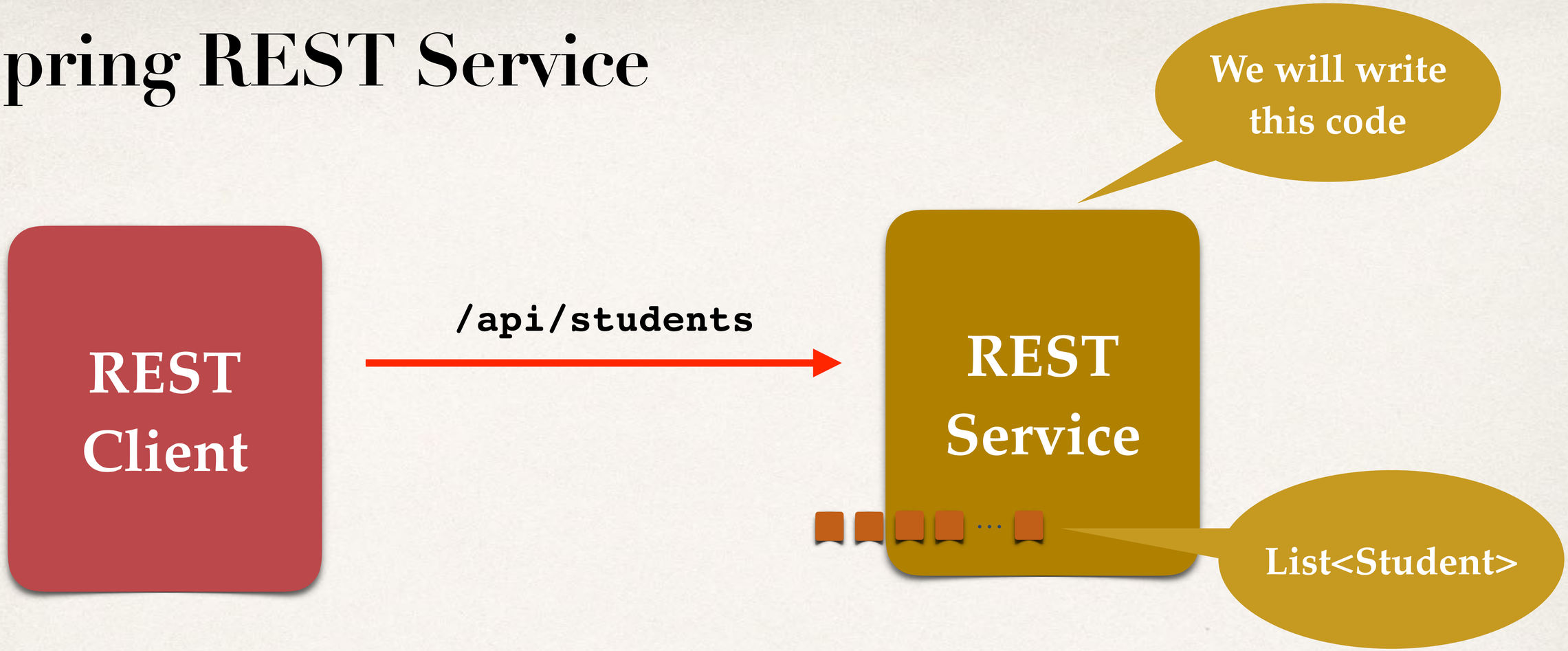
Spring REST Service



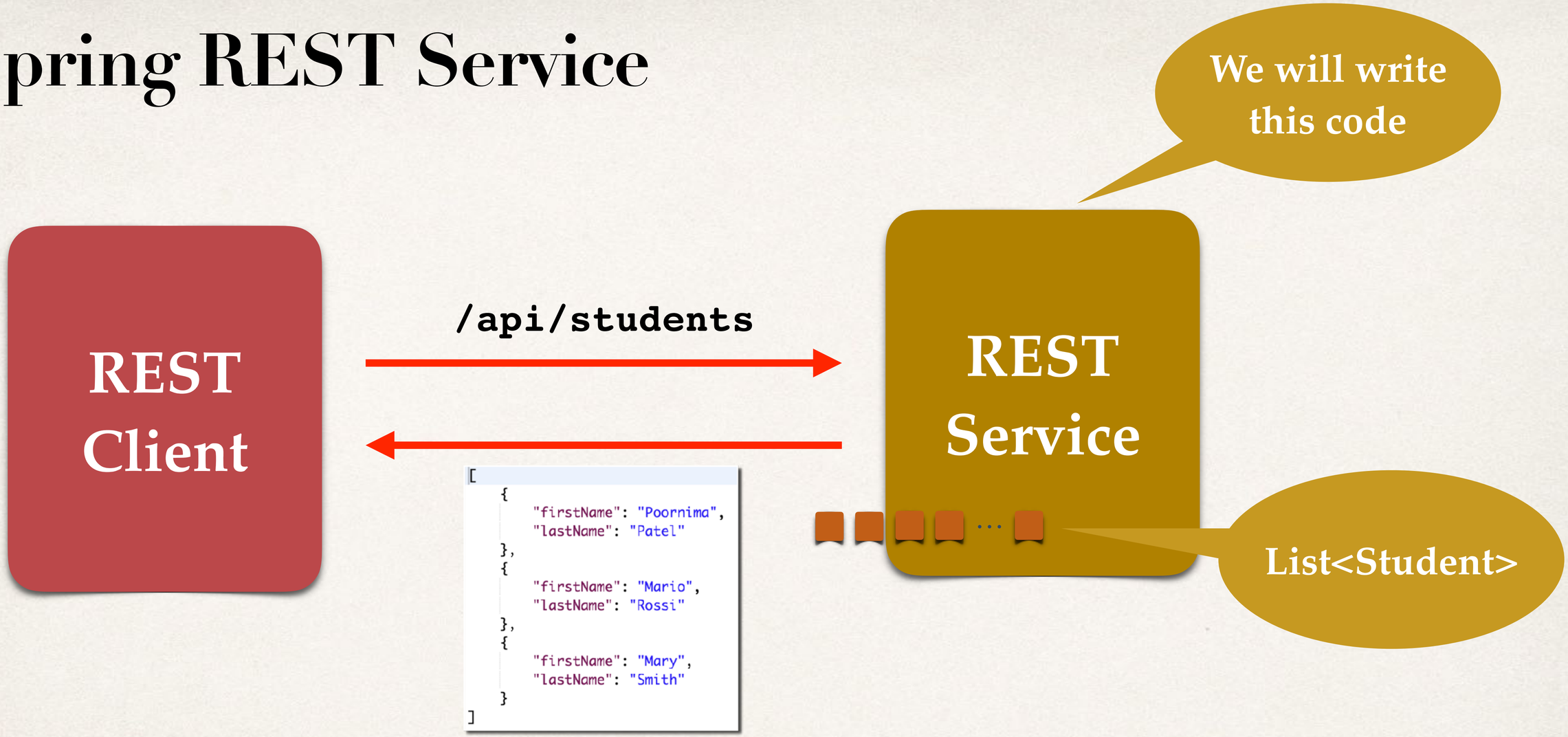
Spring REST Service



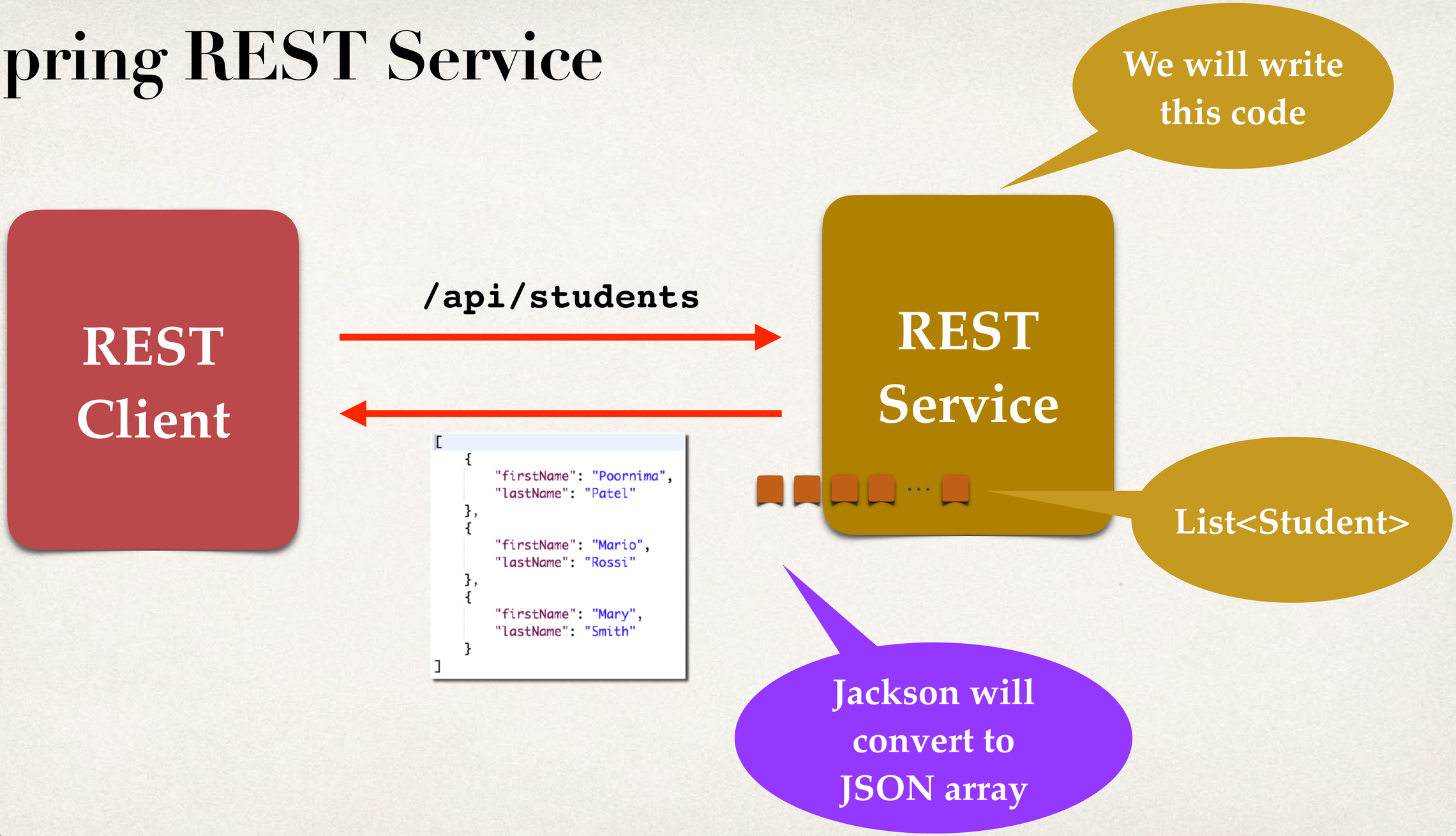
Spring REST Service



Spring REST Service



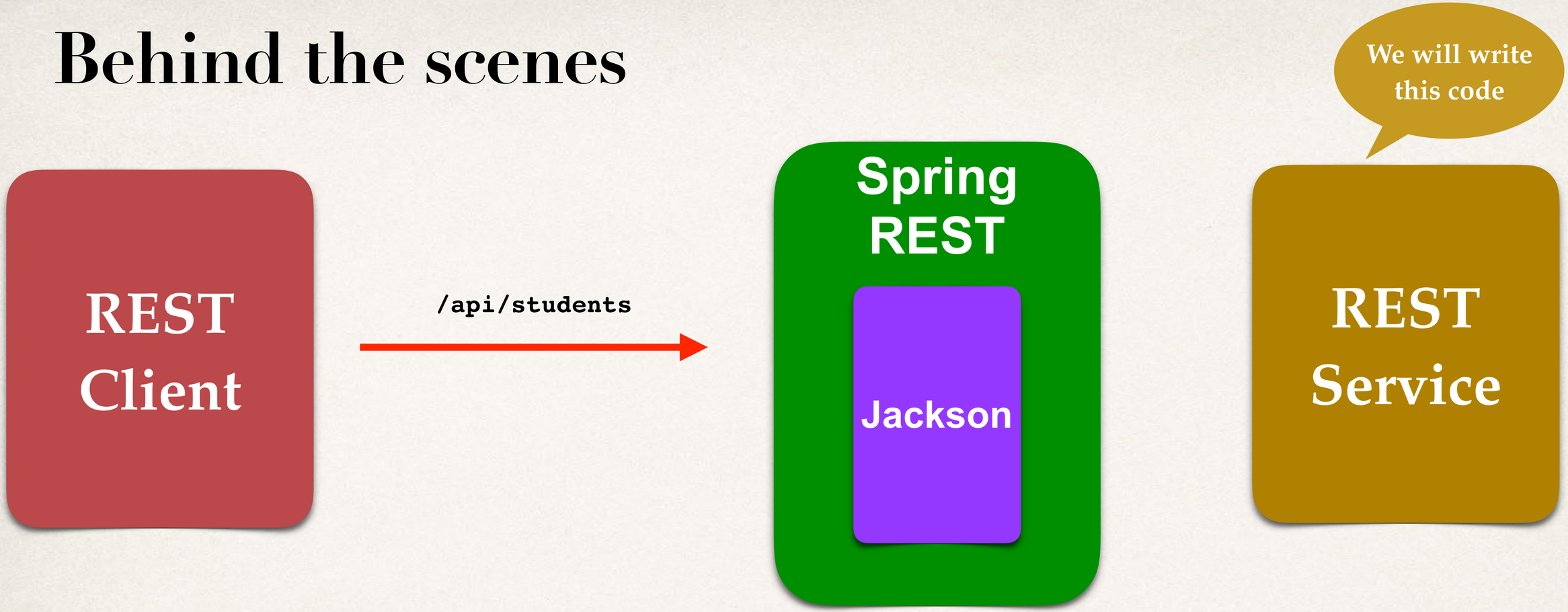
Spring REST Service



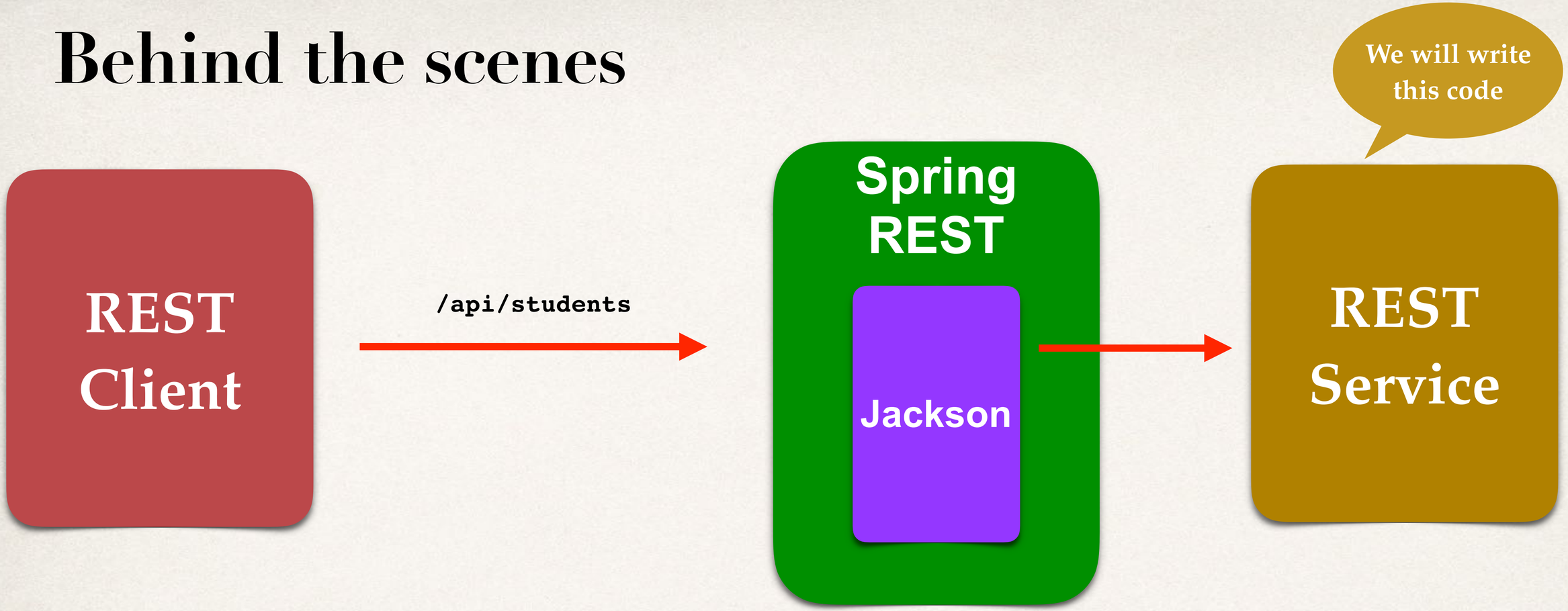
Behind the scenes

Behind the scenes

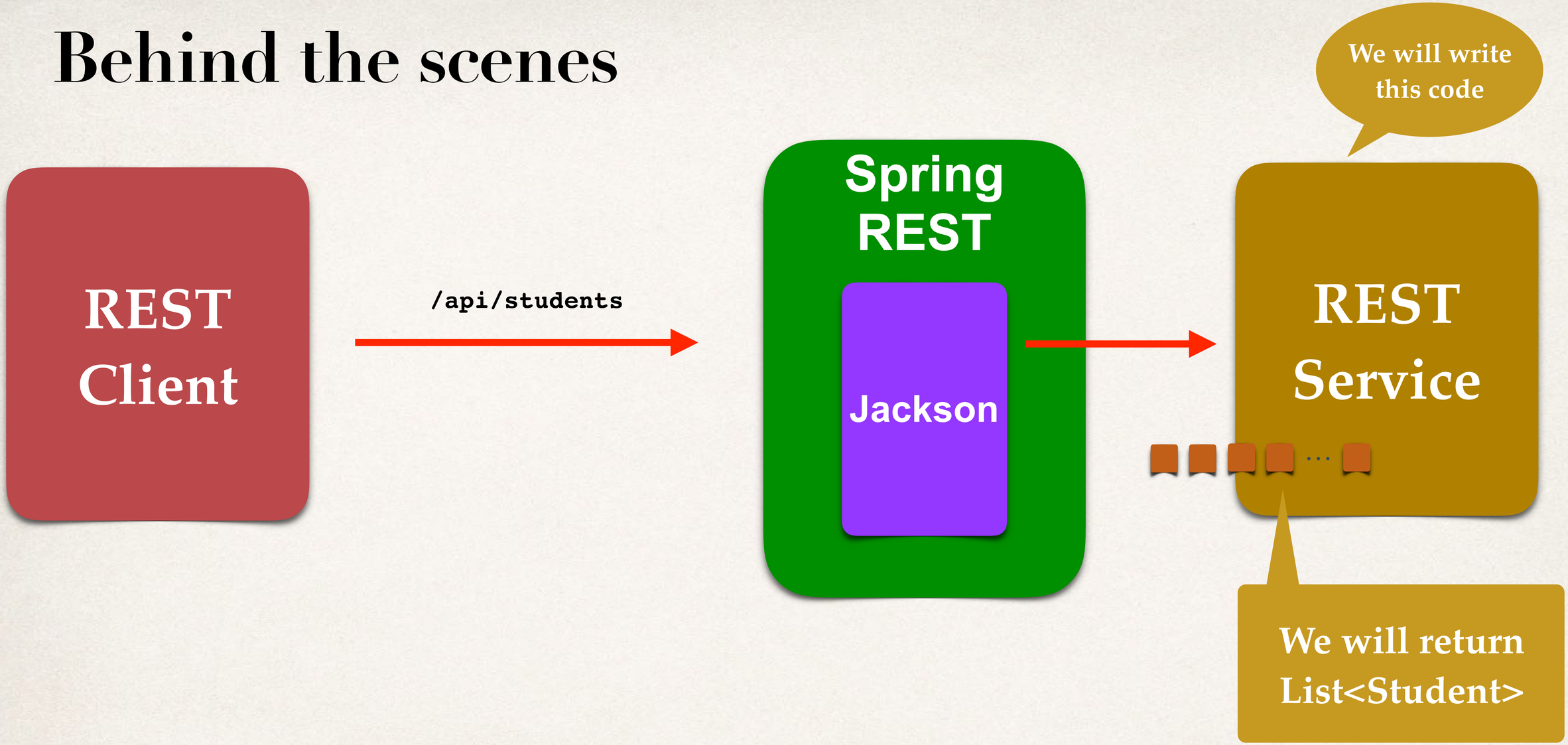
Behind the scenes



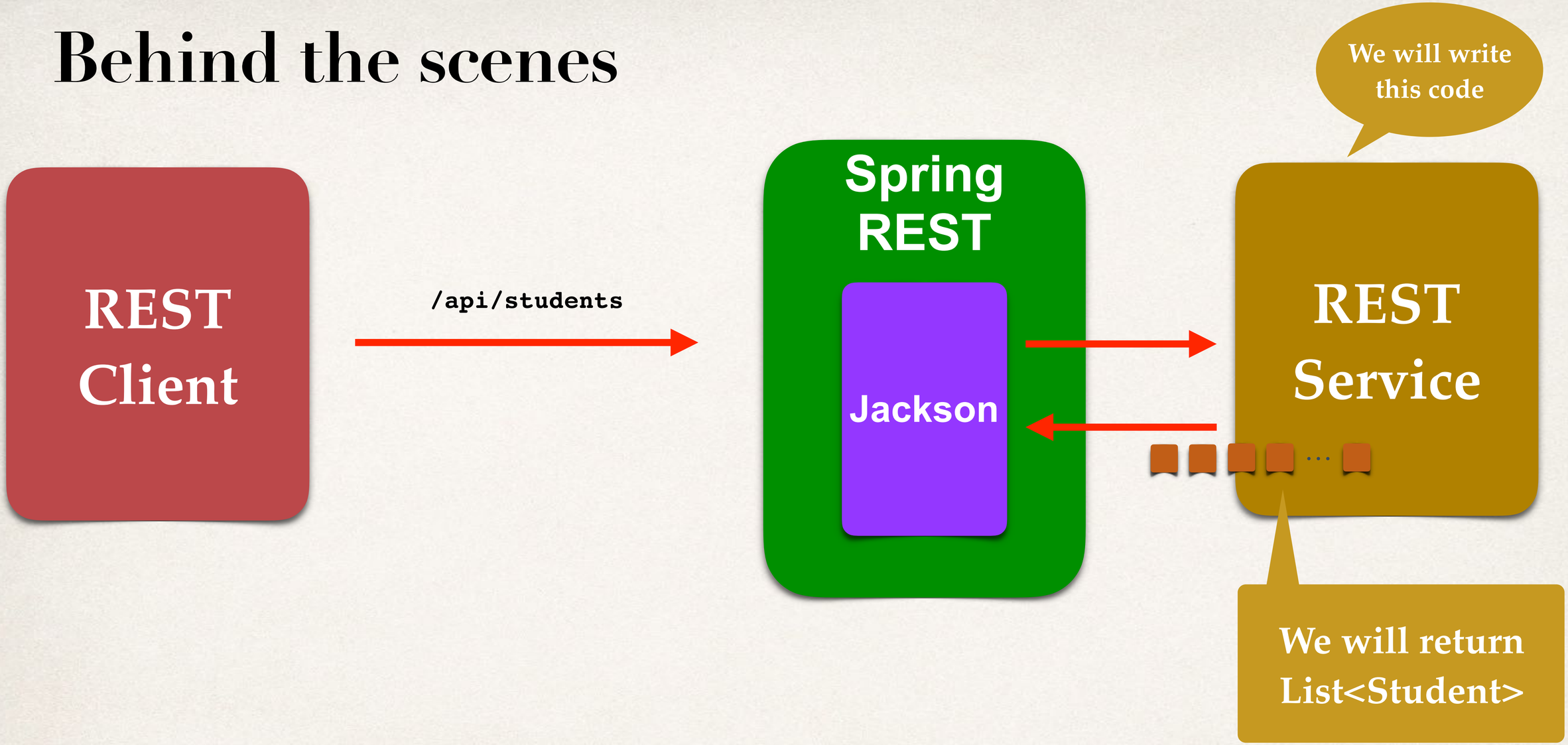
Behind the scenes



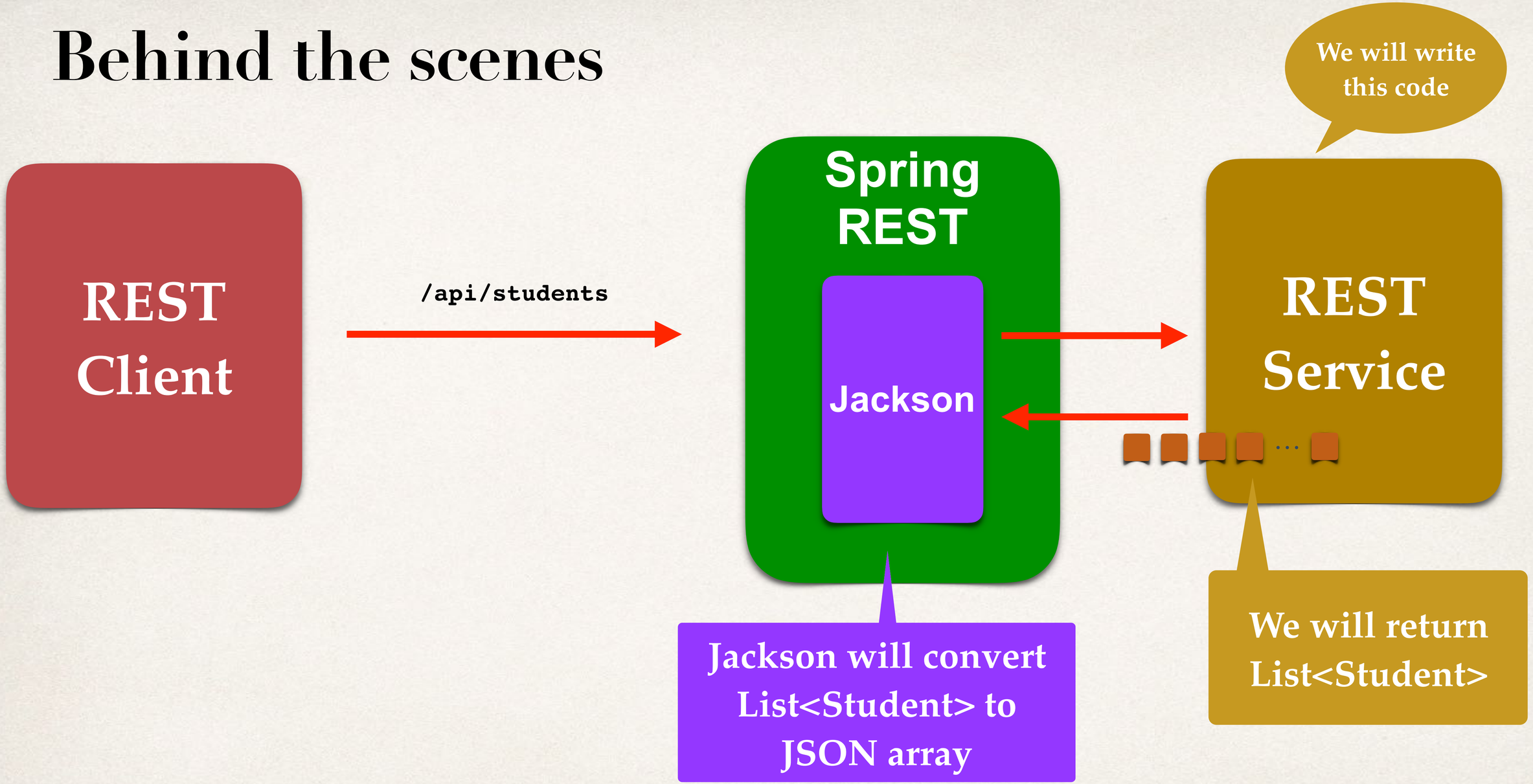
Behind the scenes



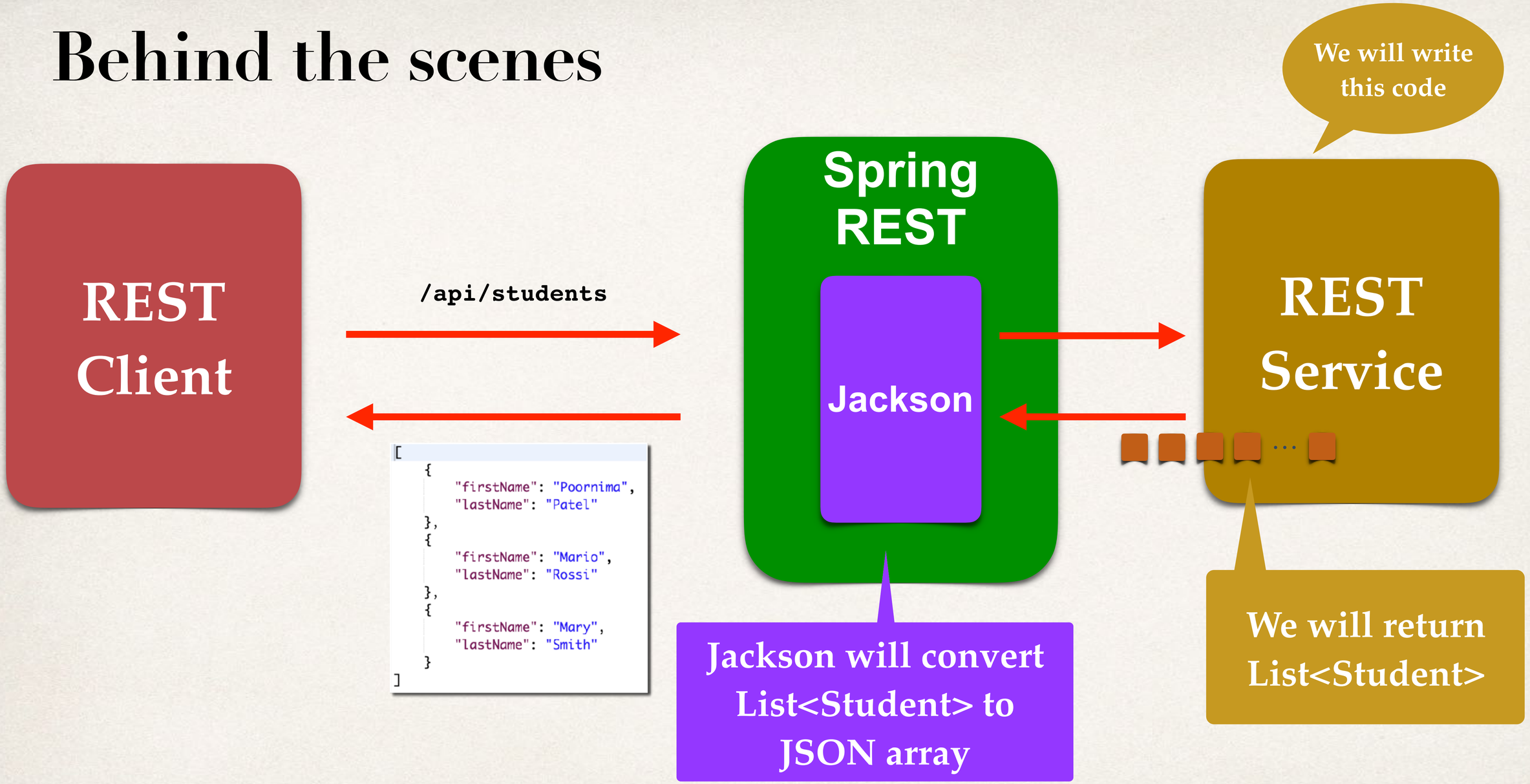
Behind the scenes



Behind the scenes



Behind the scenes



Development Process

Step-By-Step

Development Process

1. Create Java POJO class for **Student**

Development Process

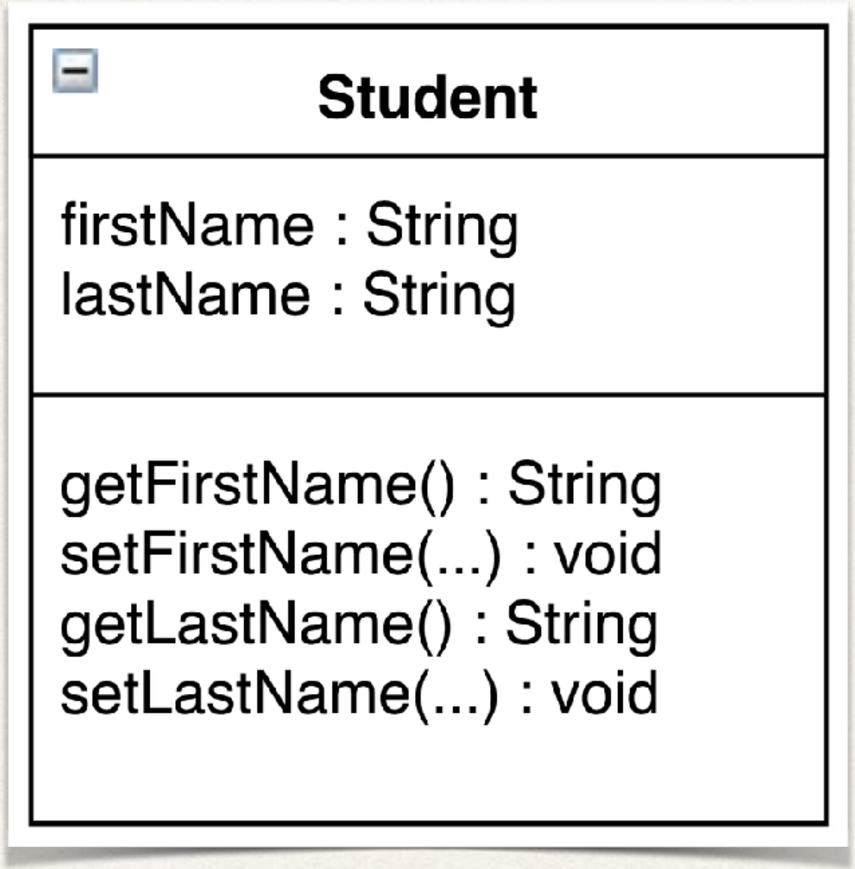
Step-By-Step

1. Create Java POJO class for **Student**

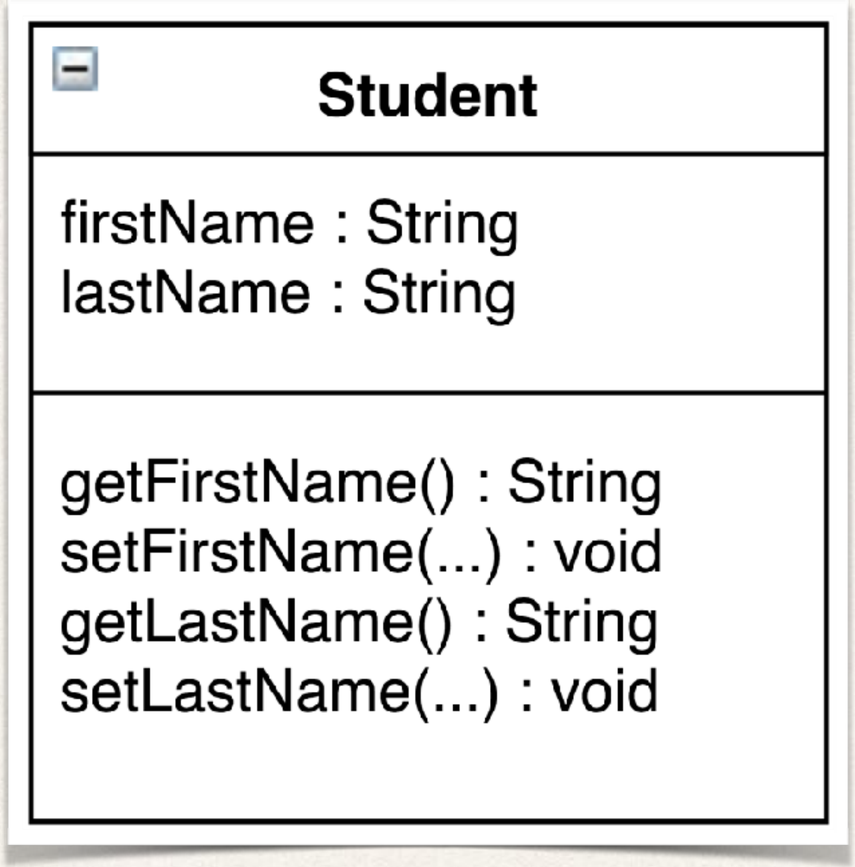
2. Create Spring REST Service using **@RestController**

Step 1: Create Java POJO class for Student

Step 1: Create Java POJO class for Student



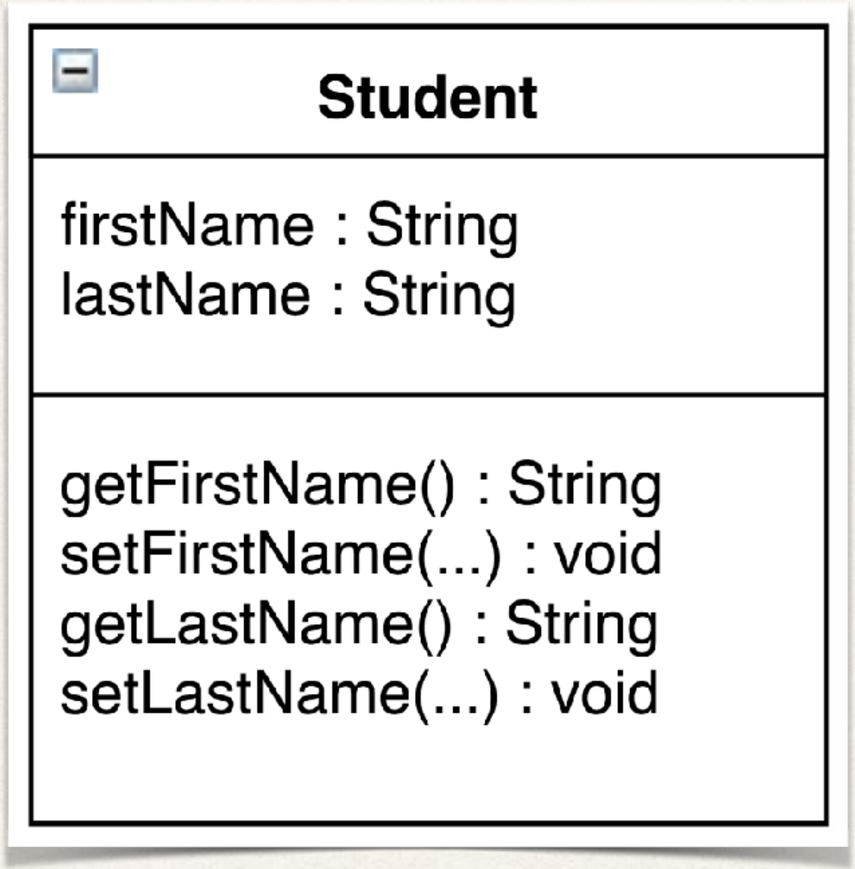
Step 1: Create Java POJO class for Student



File: Student.java

```
public class Student {

    private String firstName;
    private String lastName;

    public Student() {

    }

    public Student(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

}
```

Step 1: Create Java POJO class for Student



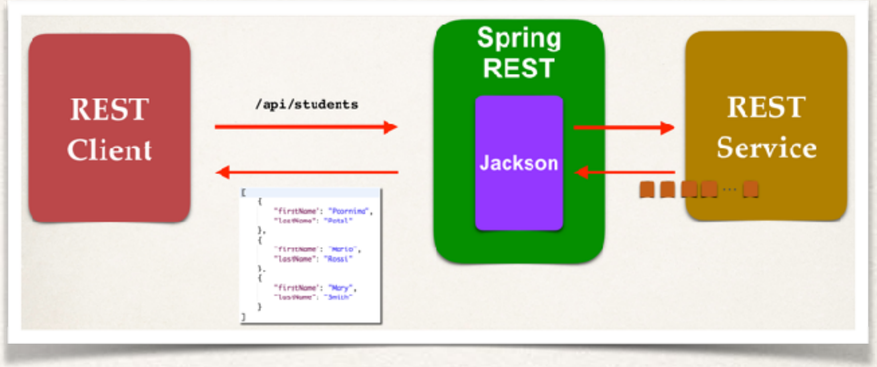
File: Student.java

```
public class Student {

    private String firstName;
    private String lastName;

    public Student() {

    }

    public Student(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

}
```

Fields
Constructors
Getter/Setters

Step 2: Create @RestController

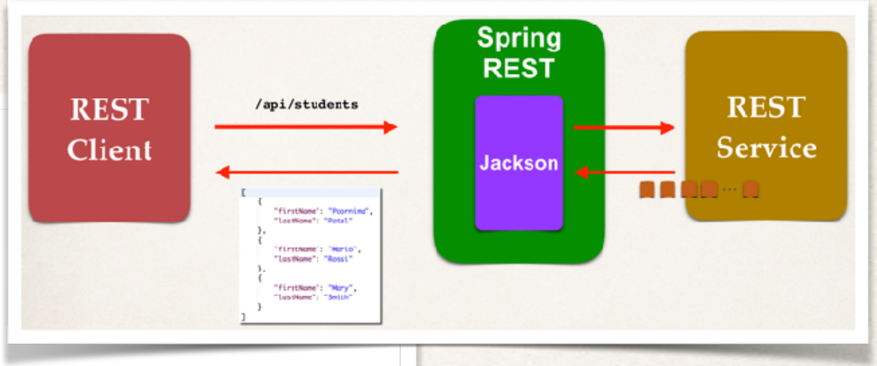
Step 2: Create @RestController



Step 2: Create @RestController

File: StudentRestController.java

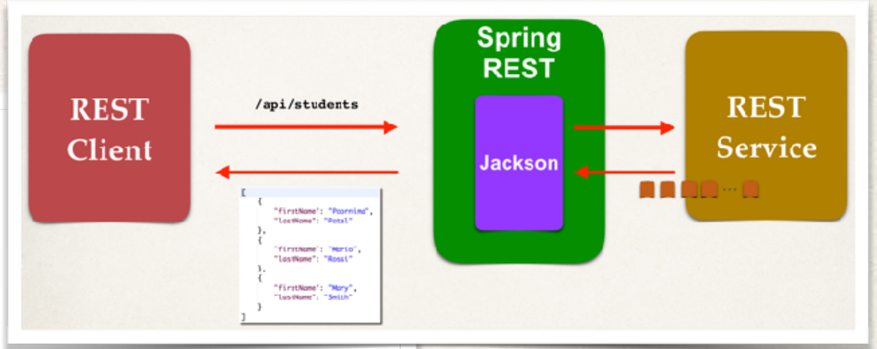
```
@RestController
@RequestMapping("/api")
public class StudentRestController {
```



Step 2: Create @RestController

File: StudentRestController.java

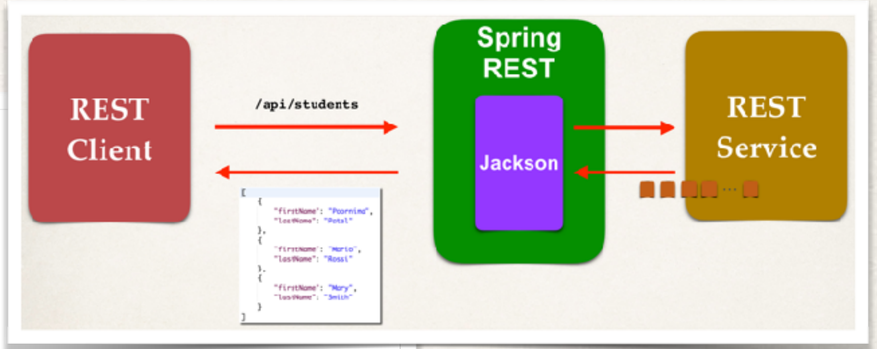
```
@RestController
@RequestMapping("/api")
public class StudentRestController {

   // define endpoint for "/students" - return list of students
```



Step 2: Create @RestController

File: StudentRestController.java

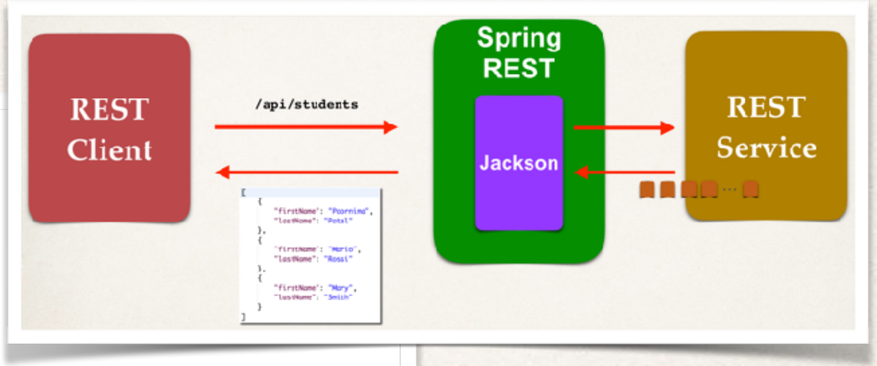
```
@RestController
@RequestMapping("/api")
public class StudentRestController {

   // define endpoint for "/students" - return list of students

   @GetMapping("/students")
   public List<Student> getStudents() {
```



Step 2: Create @RestController

File: StudentRestController.java

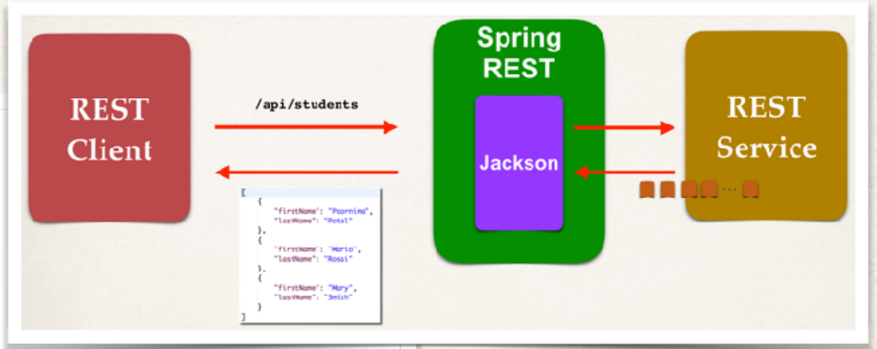
```
@RestController
@RequestMapping("/api")
public class StudentRestController {

  // define endpoint for "/students" - return list of students

  @GetMapping("/students")
  public List<Student> getStudents() {

    List<Student> theStudents = new ArrayList<>();
```



Step 2: Create @RestController

File: StudentRestController.java

```
@RestController
@RequestMapping("/api")
public class StudentRestController {

  // define endpoint for "/students" - return list of students

  @GetMapping("/students")
  public List<Student> getStudents() {

    List<Student> theStudents = new ArrayList<>();

    theStudents.add(new Student("Poornima", "Patel"));
    theStudents.add(new Student("Mario", "Rossi"));
    theStudents.add(new Student("Mary", "Smith"));
```

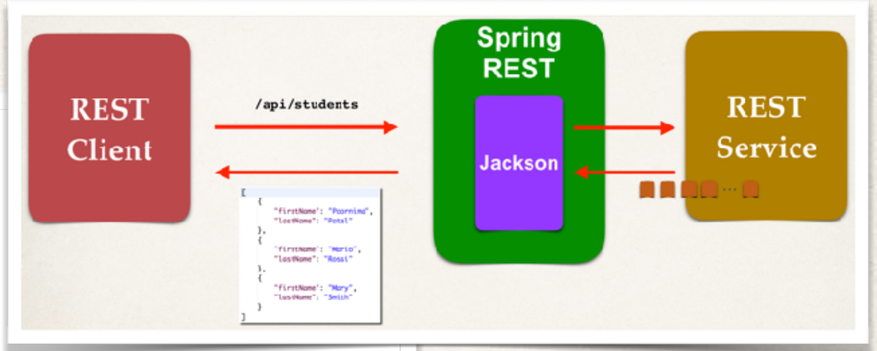


We'll hard code for now … can add DB later …

Step 2: Create @RestController

File: StudentRestController.java

```
@RestController
@RequestMapping("/api")
public class StudentRestController {

  // define endpoint for "/students" - return list of students

  @GetMapping("/students")
  public List<Student> getStudents() {

    List<Student> theStudents = new ArrayList<>();

    theStudents.add(new Student("Poornima", "Patel"));
    theStudents.add(new Student("Mario", "Rossi"));
    theStudents.add(new Student("Mary", "Smith"));

    return theStudents;
  }

}
```



Jackson will convert List<Student> to JSON array