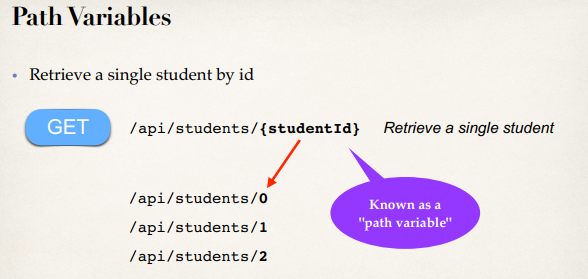
**60. Spring REST - Using @PathVariable for REST Endpoints**

**Path variables**:

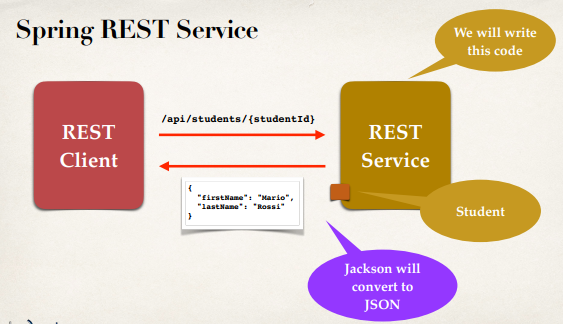
To use the path variable, we will have a new endpoint to retrieve a single student by the id.

**GET** /api/students/{**studentId**} => Retrieve a single student



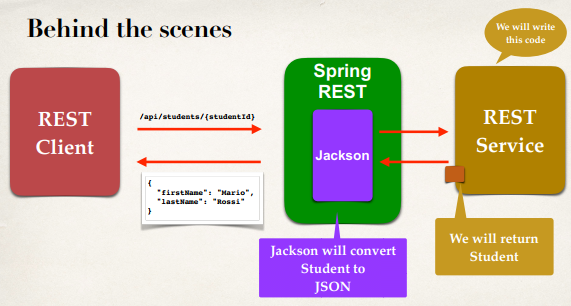
**Spring REST Service**:

* We have to write code for REST service.
* Add a new request mapping for new endpoint
* Retrieve a new student and pass it back.
* Jackson will convert the student to JSON



**Behind the scenes**:

* We have a REST Client and a REST Service.
* Client make a request across "**api/students/{studentId}**"
* After that we have Spring REST along with Jackson
* After that REST server return the related student
* Jackson will convert the student object into JSON



**Development Process (Step-by-Step)**:

1. Add request mapping to Spring REST Service
   1. Bind path variable to method parameter using **@PathVariable**

**Use @PostConstruct to load data**:

Here we will define a field and load the field with data

**Why we use @PostConstruct**:

because when the constructor is called, the bean is not yet initialized - i.e. no dependencies are injected. In the @PostConstruct method the bean is fully initialized and you can use the dependencies.

because this is the contract that guarantees that this method will be invoked only once in the bean lifecycle. It may happen (though unlikely) that a bean is instantiated multiple times by the container in its internal working, but it guarantees that @PostConstruct will be invoked only once.

Step 1: Add Request Mapping:

// define @PostConstruct to load the student data ... only once

@PostConstruct

**private** **void** loadData() {

theStudents = **new** ArrayList<>();

theStudents.add(**new** Student("Ruhul", "Amin"));

theStudents.add(**new** Student("Rezau", "Islam"));

theStudents.add(**new** Student("Osman", "Goni"));

theStudents.add(**new** Student("Kapil", "Dev"));

theStudents.add(**new** Student("Mezbaul", "Islam"));

}

**Step 1: Add Request Mapping**:

// define endpoint for "/students/{studentId}" => return student at index

@GetMapping("/students/{studentId}")

**public** Student getStudent(@PathVariable **int** studentId) {

// just index into the list ... keep it simple now!!!

**return** theStudents.get(studentId);

}

60. Spring REST - Using @PathVariable for REST Endpoints