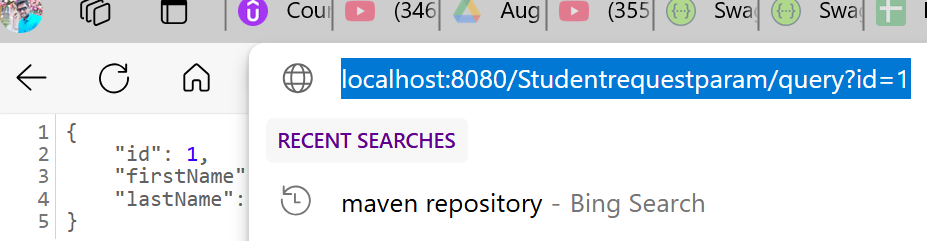
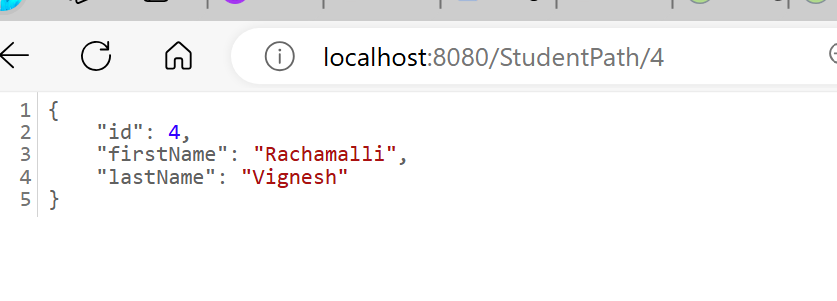
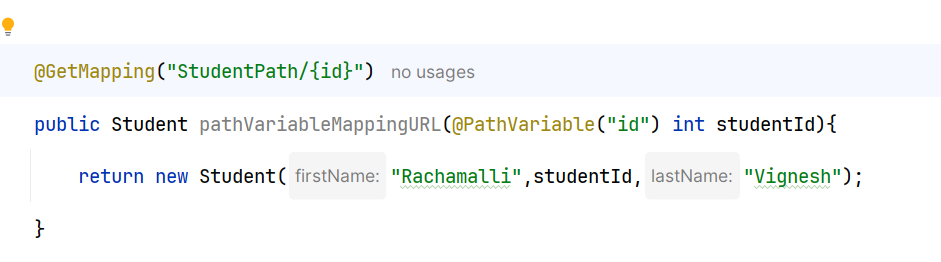
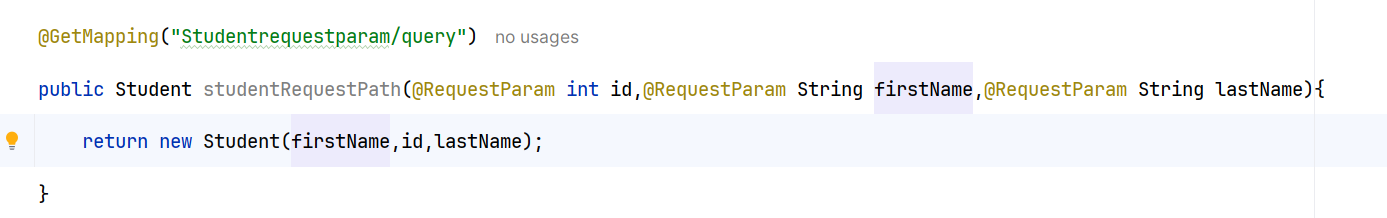
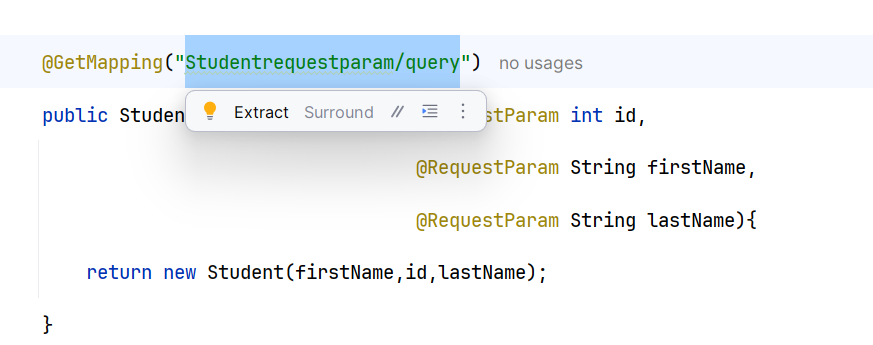
------------------ [30/8/24]---------------------------------  
  
@ResponseBody = This annotation is used to convert into json object notation.  
  
Whenever we want to learn the rest API package we have to give restapi application in the spring boot application.

@Controller , @responsebody annotation to work with the restAPI applications.  
  
@RestController = @Controller + @responsebody annotations.  
  
  
@GetMapping

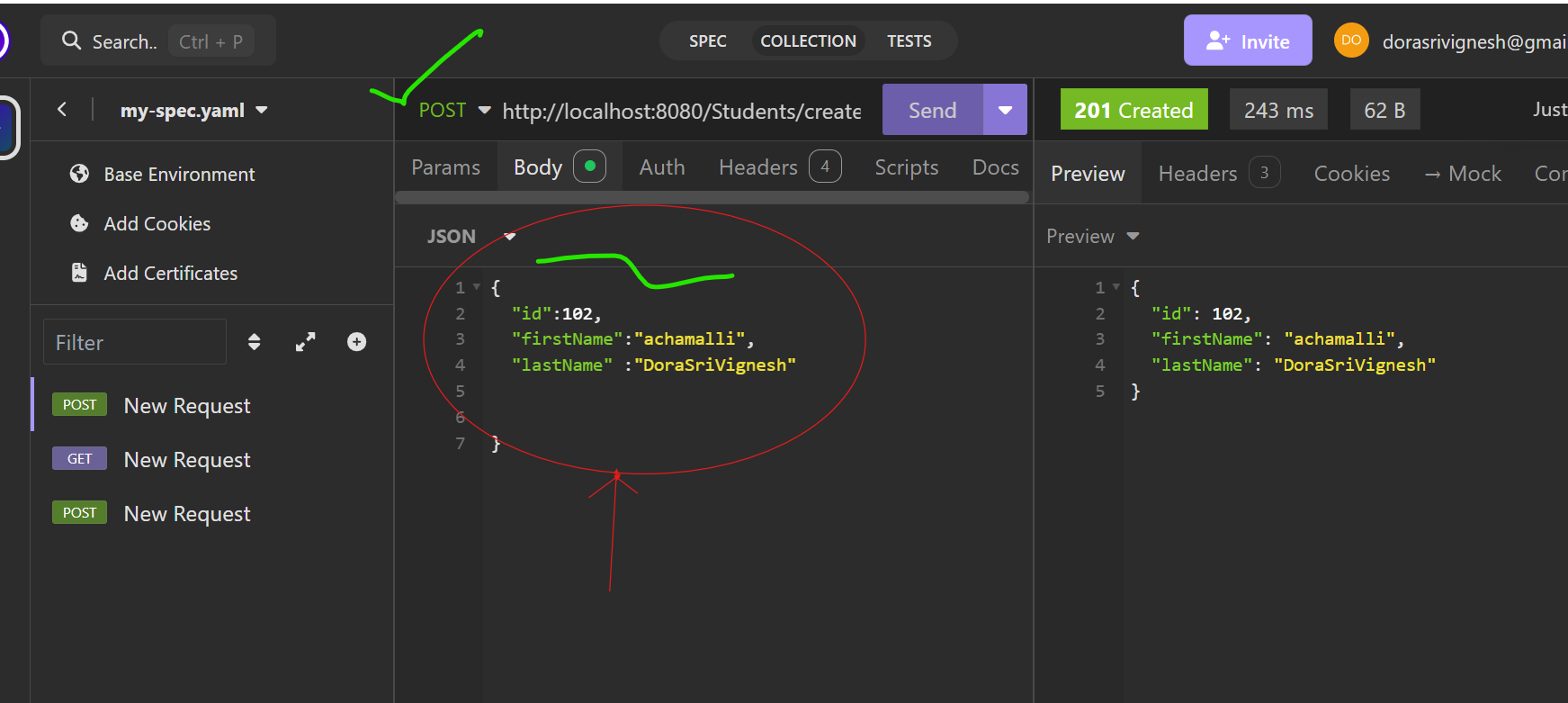
The above annotation is used to handle the HTTP request.  
it is used to map incoming get requests to map the application…  
  
Here the rest API variable should be unique and it should be different.  
  
  
  
  
@reuestparam  
  
This is the annotation used for request mapping for the data.  
--------------- [ 31/8/24] -------------------------------------------  
  
@PathVariable   
This annotation used for mapping the id for our method in the program,  
so with the help of this path variable we can map our values so it will return that value.  
  
  
  
  
  
  
------------------ This is another way of approach -----------  
  
  
  
  
🡪 The difference between @pathvariable and @getmapping is   
  
  
@pathvariable is used to bind uri template into method arguement and method body parameter what we have passed in the value in the url.  
  
------------------ [ 8/9/24] -----------------------------------------  
  
**@queryparam** is the annotation which can extract the data which is given in the value In the request URL.

[localhost:8080/Studentrequestparam/query?&id=104&firstName=Rachammali&&lastName=Vinayaka](http://localhost:8080/Studentrequestparam/query?&id=104&firstName=Rachammali&&lastName=Vinayaka)  
  
  
  
@Requestbody 🡪 This annotation is used to convert the data json object into normal data code from the URL which we are providing for this we are using @postmapping [This annotation is used to post the data] and we can have the information.  
  
  
----------------- [ 9/9/24] ------------------------------------  
  
Http PUT = To update the data.  
  
Post = It is used to create the new resource.  
  
----------------[10/9/24] ---------------------------------  
  
Response Entity :  
  
1. Response Entity represents the whole HTTP response : status code, headers and body . As a result   
we can use it to fully configure to HTTP response.  
  
2. If we want to use it , we have to return it from the endpoint ; Spring take care of the rest.  
  
3. Response Entity is the generic type, consequently , we can use any type as the response body,  
  
package com.dora.springboot\_rest\_api.controller;  
  
import com.dora.springboot\_rest\_api.bean.Student;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@RestController  
@RequestMapping("students")  
public class StudentController {  
  
 @GetMapping("students")  
 public ResponseEntity<Student> getStudent(){  
 Student student=new Student(  
 "Rachamalli",  
 1,  
 "DoraSriVignesh"  
 );  
  
 *// return new ResponseEntity<>( student,HttpStatus.OK); // Here the response entity has a new object instance.  
 // with an object we can able to pass here.  
//return ResponseEntity.ok(student);  
// here in the above 2 lines of code we can write any of the way.* return ResponseEntity.*ok*().header("customer-header","vignesh").body(student);  
  
 }  
  
@GetMapping("students")  
 public ResponseEntity< List<Student>> getStudents(){  
 List<Student> students=new ArrayList<>();  
 students.add(new Student("Rachamalli",1,"Vijaya Durga"));  
 students.add(new Student("koppu",2,"Uma Maheswari"));  
 students.add(new Student("Rachamalli",3,"Mallikarjuna Rao"));  
 return ResponseEntity.*ok*( students);  
 }  
  
  
 @GetMapping("students/{id}/{firstname}/{lastname}")  
 public Student pathVariableMappingURL(@PathVariable("id") int studentId,  
 @PathVariable("firstname") String firstName,  
 @PathVariable("lastname") String lastName  
 ){  
 return new Student(firstName,studentId,lastName);  
 }  
  
  
 @GetMapping("students/query")  
 public ResponseEntity< Student> studentRequestPath(@RequestParam int id,  
 @RequestParam String firstName,  
 @RequestParam String lastName){  
 Student student=new Student(firstName, id, lastName);  
 *// return new Student(firstName,id,lastName); // This is a normal method.  
 // if we want to write the data in the form of HTTP response , below is the way...* return ResponseEntity.*ok*(student);  
 }  
  
*// http:localhost:8080/CreatingStudent* @PostMapping("students/create")  
 @ResponseStatus(HttpStatus.*CREATED*)  
 public ResponseEntity< Student> creatingStudentData(@RequestBody Student studentData){  
  
 System.*out*.println(studentData.getId());  
 System.*out*.println(studentData.getFirstName());  
 System.*out*.println(studentData.getLastName());  
 *// return studentData;* return new ResponseEntity<>(studentData,HttpStatus.*CREATED*);  
  
  
  
 }  
  
  
 *// spring boot REST API that handles HTTP PUT Request - Updating existing resource.* @PutMapping("students/{id}/update")  
  
 public ResponseEntity< Student> updateStudent(@RequestBody Student student, @PathVariable("id") int studentid){  
 System.*out*.println(student.getFirstName());  
 System.*out*.println(student.getLastName());  
  
 return ResponseEntity.*ok*(student);  
 }  
  
  
@DeleteMapping("students/{id}/delete")  
 public ResponseEntity<String > deletingStudentData(@PathVariable("id") int studentId){  
 System.*out*.println(studentId);  
 return ResponseEntity.*ok*( "Student Data sucessfully deleted.....");  
 }  
}

🡪 Whenever we use @ @RequestMapping("students")

We need not specify the students in each of the method  
  
  
  
we can remove students In the @Getmapping

So we use to point out out all the requests by the help of request mapping.  
🡪 This is the brief introduction.  
  
------------------ [ 13/9/24] -------------------------------  
  
@getmapping -> This annotation is used to map HTTP GET request to onto specific handler method.  
  
  
@putmapping : This annotation is used to map HTTP Put request onto specific handler method.  
  
@deletemapping : This annotation is used to map  
 HTTP delete request on to specifier method.

This is all about post mapping and post mapping team,  
  
  
  
  
Here in the above we have to edit the information and we have to change to the Post Request and we can have to edit the data in the JSON format and if we hit the request then we can able to see the data in the proper object data as we have given the data in Response body,  
🡪 This response body data can accept the data and it can be handled the data through our request and it can be handled in the to perform the operation.  
  
🡪 This is all about request body , response body.