Returning a list of students:

package com.flynaut.ListReturn.entity;  
  
public class Student {  
 String firstName;  
 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;  
 }  
  
 @Override  
 public String toString() {  
 return "Student{" +  
 "firstName='" + firstName + '\'' +  
 ", lastName='" + lastName + '\'' +  
 '}';  
 }  
}

package com.flynaut.ListReturn.controller;  
  
import com.flynaut.ListReturn.entity.Student;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@RestController  
@RequestMapping("/api")  
public class StudentRestController {  
 //defining an endpoint for /students - return a list of students  
 @GetMapping("/students")  
 public List<Student> getStudents(){  
 List<Student> theStudents = new ArrayList<>();  
 theStudents.add(new Student("Krishna","Yadav"));  
 theStudents.add(new Student("Gopal","Joshi"));  
 theStudents.add(new Student("Govind","Yadav"));  
 return theStudents;  
  
 // /api/students  
 }  
}

Path Variable

* Retrieve a single student by id

GET – /api/students/{studentId} -> Retrive a single student

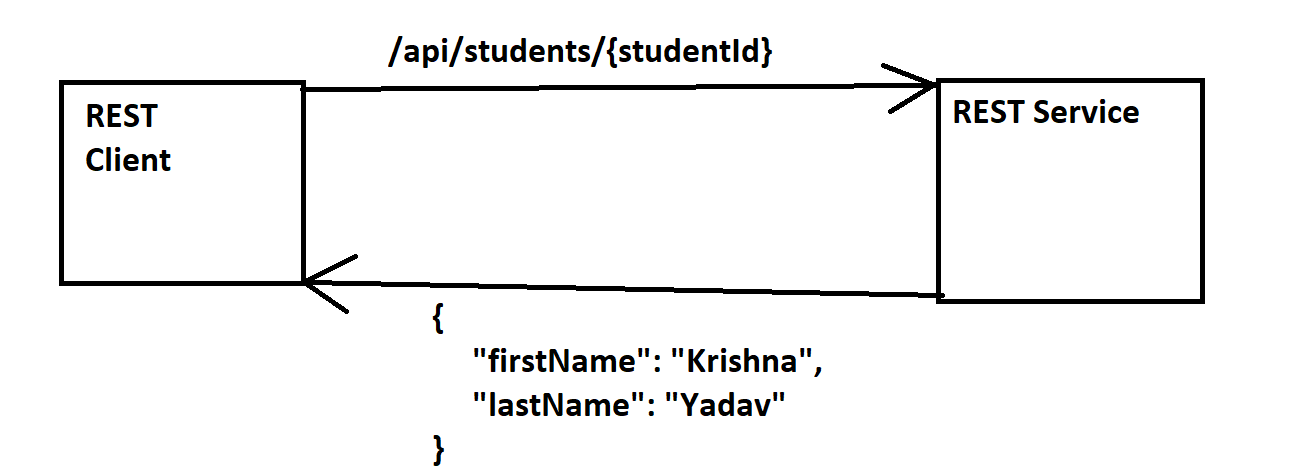
(Path Variable)

/api/students/0

/api/students/1

/api/student/2

Spring REST Service:



Development Process:

Add request mapping to spring REST Service

Bind the path variable to method parameter using @PathVariable

package com.flynaut.ListReturn.controller;  
  
import com.flynaut.ListReturn.entity.Student;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@RestController  
@RequestMapping("/api")  
public class StudentRestController {  
 //defining an endpoint for /students - return a list of students  
 @GetMapping("/students")  
 public List<Student> getStudents(){  
 List<Student> theStudents = new ArrayList<>();  
 theStudents.add(new Student("Krishna","Yadav"));  
 theStudents.add(new Student("Gopal","Joshi"));  
 theStudents.add(new Student("Govind","Yadav"));  
 return theStudents;  
 // /api/students  
 }  
  
 //defining the endpoint "/students/{studentId}" - return student at index  
 @GetMapping("/students/{studentId}")  
 public Student getStudent(@PathVariable int studentId){  
 //populating the list  
 List<Student> theStudents = new ArrayList<>();  
 theStudents.add(new Student("Krishna","Yadav"));  
 theStudents.add(new Student("Gopal","Joshi"));  
 theStudents.add(new Student("Govind","Yadav"));  
  
 return theStudents.get(studentId);  
 }  
}

Spring REST Exception Handling



If we don’t want this page we need to handle the exception

What we want?

* Handle the exception and return error as JSON

Development Process:

1. Create custom error response class
2. Create custom exception class
3. Update the REST service to throw an exception if student not found
4. Add an exception handler method using @ExceptionHandler

1stStep: Create custom error response class

The custom error response class will be sent back to client as JSON

We will define it as java class

package com.flynaut.ListReturn;  
  
public class StudentErrorResponse {  
 private int status;  
 private String message;  
 private long timeStamp;  
  
 public StudentErrorResponse(){}  
  
 public StudentErrorResponse(int status, String message, long timeStamp) {  
 this.status = status;  
 this.message = message;  
 this.timeStamp = timeStamp;  
 }  
  
 public int getStatus() {  
 return status;  
 }  
  
 public void setStatus(int status) {  
 this.status = status;  
 }  
  
 public String getMessage() {  
 return message;  
 }  
  
 public void setMessage(String message) {  
 this.message = message;  
 }  
  
 public long getTimeStamp() {  
 return timeStamp;  
 }  
  
 public void setTimeStamp(long timeStamp) {  
 this.timeStamp = timeStamp;  
 }  
}

2ndStep: Create Custom Exception Class

package com.flynaut.ListReturn;  
  
public class StudentNotFoundException extends RuntimeException{  
 public StudentNotFoundException(String message){  
 super(message); //It will call the super class constructor  
 }  
}

3rdStep: Update the REST service to throw an exception if student not found

package com.flynaut.ListReturn.controller;  
  
import com.flynaut.ListReturn.StudentNotFoundException;  
import com.flynaut.ListReturn.entity.Student;  
import jakarta.annotation.PostConstruct;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@RestController  
@RequestMapping("/api")  
public class StudentRestController {  
 private List<Student> theStudents;  
  
 //define @PostConstruct to load the once  
  
 @PostConstruct  
 public void loadData(){  
 theStudents = new ArrayList<>();  
 theStudents.add(new Student("Krishna","Yadav"));  
 theStudents.add(new Student("Prasad","Jain"));  
 theStudents.add(new Student("Parth","Sharma"));  
 }  
  
  
 //define the endpoint "/students"  
 @GetMapping("/students")  
 public List<Student> getStudents(){  
 return theStudents;  
 }  
  
 //defining the endpoint "students/{studentId}"  
 @GetMapping("/students/{studentId}")  
 public Student getStudent(@PathVariable int studentId){  
 //Check the studentId against the list  
 if((studentId > theStudents.size()) || (studentId<0)){  
 throw new StudentNotFoundException("StudentId not Found -"+ studentId);  
 }  
 return theStudents.get(studentId);  
 }  
  
}

Add an exception handler method using @ExceptionHandler

* Exception handler will return a ResponseEntity
* ResposnseEntity is a wrapper for the HTTP response

package com.flynaut.ListReturn.controller;  
  
import com.flynaut.ListReturn.StudentErrorResponse;  
import com.flynaut.ListReturn.StudentNotFoundException;  
import com.flynaut.ListReturn.entity.Student;  
import jakarta.annotation.PostConstruct;  
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("/api")  
public class StudentRestController {  
 private List<Student> theStudents;  
  
 //define @PostConstruct to load the once  
  
 @PostConstruct  
 public void loadData(){  
 theStudents = new ArrayList<>();  
 theStudents.add(new Student("Krishna","Yadav"));  
 theStudents.add(new Student("Prasad","Jain"));  
 theStudents.add(new Student("Parth","Sharma"));  
 }  
  
 @ExceptionHandler  
 public ResponseEntity<StudentErrorResponse> handleException(StudentNotFoundException exc){  
 StudentErrorResponse error = new StudentErrorResponse();  
 error.setStatus(HttpStatus.*NOT\_FOUND*.value());  
 error.setMessage(exc.getMessage());  
 error.setTimeStamp(System.*currentTimeMillis*());  
 return new ResponseEntity<>(error, HttpStatus.*NOT\_FOUND*);  
 }  
  
 //define the endpoint "/students"  
 @GetMapping("/students")  
 public List<Student> getStudents(){  
 return theStudents;  
 }  
  
 //defining the endpoint "students/{studentId}"  
 @GetMapping("/students/{studentId}")  
 public Student getStudent(@PathVariable int studentId){  
 //Check the studentId against the list  
 if((studentId > theStudents.size()) || (studentId<0)){  
 throw new StudentNotFoundException("StudentId not Found -"+ studentId);  
 }  
 return theStudents.get(studentId);  
 }  
  
}