

Date : 16/03/2021
Spring Boot 9AM
Mr. RAGHU

Exception Handling in REST:
<https://www.youtube.com/watch?v=M-LRfrYHWrk>

ReST : MediaType Annotations

@ResponseBody: return Type : (non-String) -->Obj=> JSON/XML
@RequestBody : Parameter : (non-String) --> JSON/XML => Obj

@RequestBody: This annotation must be applied externally.

=> It should be used as method parameter.

=> It will read Http Request Body (XML/JSON) converts to Object Format

=> Syntax:

@RequestBody ClassName objectName

=> ** EndClient/Consumer must specify one HeaderParam
'Content-Type:_____'

=> To send data from consumer to producer GET request not supported. (GET req never supports Request Body)

=> Even { } is a valid JSON, that indicates object created with default values.

example JSON :

```
{  
  "eid" : 10,  
  "ename" : "ABC",  
  "esal" : 300.2  
}
```

=> If we try to send XML Input using @RequestBody but App is not supporting XML, then FC returns
Http Status - 415 Unsupported MediaType

=> If Output XML (ResponseBody) is not supported then
- 406 Not Acceptable.

=> To Enable XML support,

Step#1 Add XML Dependency in pom.xml file

```
<dependency>  
  <groupId>com.fasterxml.jackson.dataformat</groupId>  
  <artifactId>jackson-dataformat-xml</artifactId>  
</dependency>
```

Step#2 using POSTMAN send data

```
<Employee>  
  <eid>101</eid>  
  <ename>ABC</ename>  
  <esal>250.0</esal>  
</Employee>
```

====code=====

1. Name: SpringBoot2RestInputMediaType
Dep: Web, lombok, Devtools

2. Modle class:

```
package in.nareshit.raghu.model;
```

```
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;
```

```
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Employee {  
  
    private Integer eid;  
    private String ename;  
    private Double esal;  
  
}
```

3. Rest Controller

```
package in.nareshit.raghu.rest;
```

```
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;
```

```
import in.nareshit.raghu.model.Employee;
```

```
@RestController  
@RequestMapping("/employee")  
public class EmployeeRestController {  
  
    @PostMapping("/read")  
    public ResponseEntity<String> readEmp(  
        @RequestBody Employee employee  
    )  
    {  
  
        return new ResponseEntity<String>(  
            employee.toString(),  
            HttpStatus.OK);  
  
    }  
}
```

(POSTMAN SCREEN)

```
-----  
POST http://localhost:8080/employee/read  SEND  
-----
```

Body

```
-----  
(*) raw    [JSON]  
-----
```

```
{
  "eid" : 10,
  "ename" : "ABC",
  "esal" : 300.2
}
```

--Http Request Headers---

Accept:application/xml
Not supported?
Output: 406

Content-Type: application/xml
Not Supported?
Output: 415

Give XML --> JSON
JSON --> XML
JSON --> JSON
XML --> XML

1) Model

```
package in.nareshit.raghu.model;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Employee {
```

```
    private Integer eid;
    private String ename;
    private Double esal;
    private Double hra;
    private Double ta;
}
```

2) Restcontroller

```
package in.nareshit.raghu.rest;
```

```
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
import in.nareshit.raghu.model.Employee;
```

```
@RestController
@RequestMapping("/employee")
public class EmployeeRestController {
```

```

    @PostMapping("/read")
    public ResponseEntity<Employee> readEmp(
        @RequestBody Employee employee
    )
    {
        var sal = employee.getEsal();
        var hra = sal * 10/100.0;
        var ta = sal * 6/100.0;

        employee.setHra(hra);
        employee.setTa(ta);

        return new ResponseEntity<Employee>(
            employee,
            HttpStatus.OK);
    }
}

```

*) At a time one type input/output.

Input: XML

```

<Employee>
    <eid>101</eid><ename>AA</ename><esal>1000.0</esal>
</Employee>

```

Input JSON

```

{
    "eid" : 999, "ename" : "SYED", "esal" : 500.0
}

```

=====

StudentRestController

m1# saveStudent():String

JSON/XML --> @RequestBody --> Student object --> Service/Repo--> save to DB

-> Student saved successfully

m2# getAllStudent()

repo/serv-> findAll(): List<Student> --> return using ResponseEntity
test using POSTMAN.