Date : 15/03/2021
Spring Boot 9AM
Mr. RAGHU

JWT:
https://www.youtube.com/c/NareshIT/search?query=JWT%20raghu

Spring REST JSON : List<Employee>

List<PT> : JSON Format
```
[
  val,
  val,
  val,
]
```

List<T> : JSON Format
```
[
  { },
  { },
  { },
]
```
---

*) default output type is JSON .ie @ResponseBody Converts ReturnType as JSON Data only. XML support is not given by default.

*) To get XML output inplace of JSON,
Step#1 add in pom.xml
```
<dependency>
    <groupId>com.fasterxml.jackson.dataformat</groupId>
    <artifactId>jackson-dataformat-xml</artifactId>
</dependency>
```

Step#2 Provide Header param 'Accept=_____' in request

Q) What is Accept Header Param?
A) It is added in request, but it says what is expected response type.

=>In simple Request is asking Please Accept my output type expected.

ex: Request(Accept=application/xml)
Hey Provider! Please Accept my request that expected output is 'application/xml"

Q) What is the diff b/w Content-Type and Accept?
A)
Content-Type (Request/Response)
It indicates what data exist in Body(either Request or response).

Accept : (Request)
It says what is expected response Content-Type

*)Note:
=> If App is not supports XML then still we are requesting using Accept: application/xml

```
   then FC returns Http Status - 406 Not Acceptable

Q) What is meaning of below Request header?
  Accept = application/xml, */*
  => It is requsting for XML with 1st priority if 'Not Accepted'
    then aay data is fine. Client is ready to take.

  Accept = application/xml, application/json
 => it is requesting for
     XML output as 1st priority, if not supported then
     JSON ouput as 2nd priority, if not supported then
     406 - Not Acceptable.
-----------------------------------------------------------------
Q#1) Request Header Param ?
  Accept = application/xml

Case#1 XML Not Supported
Output: 406 Not Acceptable

Case#2 XML Supported
Output: XML Output (200-OK)

-----------------------------------
Q#2) Request Header Param ?
 Accept = application/xml, application/json

Case#1 XML Not Supported
Output: JSON (200-OK)

Case#2 XML, JSON Both not suuported
Output: 406 Not Acceptable

Q#3) Request Header Param ?
 Accept = application/xml, */*

Case#1 XML, JSON Not supported
Output: Any Format supported by Provider.
   Ex: HTML, TEXT, IMAGE...etc
   (200-OK)

Case#2 JSON only Supported
Output:  JSON (200-OK)

*) Note: if Accept Header Param contains */* then client never gets
      406 from Provider, it will get any one type of response.
==========================code=======================================
name: SpringBoot2XmlOutput
Dep : Spring web, lombok, devtools

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;
}
```

*) RestController

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

import java.util.Arrays;
import java.util.List;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import in.nareshit.raghu.model.Employee;

@RestController
public class EmployeeRestController {


	@GetMapping("/obj")
	public ResponseEntity<Employee> showDataB() {
		Employee body = new Employee(106, "SYED", 600.0);
		return new ResponseEntity<Employee>(body,
HttpStatus.OK);
	}



	@GetMapping("/list")
	public ResponseEntity<List<Employee>> showDataD() {
		List<Employee> body = Arrays.asList(
					new Employee(101, "A", 30.0),
					new Employee(102, "B", 31.0),
					new Employee(103, "C", 32.0),
					new Employee(104, "D", 33.0)
				);

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

	@GetMapping("/listb")
	public ResponseEntity<List<String>> showDataE() {
		List<String> body = Arrays.asList("ONE","TWO","ABC");
		return new ResponseEntity<List<String>>(body,
HttpStatus.OK);
	}
```

```
}

*) Make Request USING POSTMAN

============================================================
Association Mapping(HAS-A):

#ex
*)models
package in.nareshit.raghu.model;

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

@Data
@NoArgsConstructor
@AllArgsConstructor
public class CompanyInfo {

        private Integer cid;
        private String cname;
}
---
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;

}
---
package in.nareshit.raghu.model;

import java.util.List;

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

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Dept {

        private Integer did;
        private String dname;
        private String dcode;
```

```
        private List<Employee> emps;
        private CompanyInfo cob;
}


*) RestController
package in.nareshit.raghu.rest;

import java.util.Arrays;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import in.nareshit.raghu.model.CompanyInfo;
import in.nareshit.raghu.model.Dept;
import in.nareshit.raghu.model.Employee;

@RestController
public class DeptRestController {

        @GetMapping("/dept")
        public ResponseEntity<Dept> showData() {
                Dept body = new Dept(1440, "DEVELOPMENT", "DEV",
                                Arrays.asList(
                                                new Employee(101, "A",
30.0),
                                                new Employee(102, "B",
31.0),
                                                new Employee(103, "C",
32.0),
                                                new Employee(104, "D",
33.0)
                                        ),
                                new CompanyInfo(150, "NIT HYD")
                                );
                return new ResponseEntity<Dept>(body, HttpStatus.OK);
        }
}

*) Req URL:http://localhost:8080/dept
=====================================================================
Task
  Course
    cid
    cname
    cfee

  Student  --<> Course
    sid
    sname
    subs:Set<String>
    marks:List<Integer>
    grades:Map<String,String>
    course:Course
```

=> Define RestController and return Student object in JSON/XML Output

pom.xml

```
<dependency>

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

                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-
web</artifactId>
                </dependency>

                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-devtools</artifactId>
                        <scope>runtime</scope>
                        <optional>true</optional>
                </dependency>
                <dependency>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
                        <optional>true</optional>
                </dependency>
```