# Jersey 2.x (ReST) Content Negotiation - by Mr. RAGHU

## **Producer Application**

```
MediaTypeProducerTest
 ▶ ₱ Deployment Descriptor: <web app>

▲ in.nit.controller

       StudentRestController.java
     Address.java
       Student.java
   > # target/generated-sources/annotations
   > # target/generated-test-sources/test-annotations
 JavaScript Resources
 Deployed Resources
 🛮 🗁 src
   main
     D 🗁 java
     x web.xml

    b test

 target
   pom.xml
```

<artifactId>jersey-hk2</artifactId>

```
<version>2.29</version>
    </dependency>
         <!-- JSON Conversion -->
         <dependency>
    <groupId>org.glassfish.jersey.media</groupId>
    <artifactId>jersey-media-json-jackson</artifactId>
             <version>2.29</version>
    </dependency>
         <!-- XML Conversion -->
    <dependency>
             <groupId>org.glassfish.jaxb
             <artifactId>jaxb-runtime</artifactId>
             <version>2.2.11
    </dependency>
    <dependency>
             <groupId>com.sun.xml.bind
             <artifactId>jaxb-impl</artifactId>
             <version>2.2.11
    </dependency>
    <dependency>
             <groupId>com.sun.xml.bind
             <artifactId>jaxb-core</artifactId>
             <version>2.2.11
    </dependency:
</dependencie
Model classes:
package in.nit.model;
public class Address {
    private String hno;
    private String loc;
    public String getHno() {
         return hno;
```

```
public void setHno(String hno) {
         this.hno = hno;
     public String getLoc() {
         return loc;
    public void setLoc(String loc) {
         this.loc = loc;
    @Override
    public String toString() {
         return "Address [hno=" + hno + "
"]";
    public Address() {
         super();
    public Address(String hno, String loc) {
         super();
         this.hno = hno:
         this.loc = loc;
     }
}
package in.nit.model
import java.util.List;
import java.util.Map;
import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlElementWrapper;
import javax.xml.bind.annotation.XmlRootElement;
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Student {
    private int sid;
    private String sname;
```

```
private double sfee;
@XmlElementWrapper(name = "subjects")
private List<String> subjs;
private Map<String,Integer> marks;
private Address addr;
public Address getAddr() {
    return addr;
public void setAddr(Address addr) {
    this.addr = addr;
public int getSid() {
    return sid;
public void setSid(int sid) {
    this.sid = sid;
public String getSname()
    return sname;
public void setSname(String sname) {
    this.sname = sname;
public double getSfee() {
    return sfee;
public void setSfee(double sfee) {
    this.sfee = sfee;
public List<String> getSubjs() {
   return subjs;
public void setSubjs(List<String> subjs) {
    this.subjs = subjs;
public Map<String, Integer> getMarks() {
    return marks;
public void setMarks(Map<String, Integer> marks) {
    this.marks = marks;
```

```
-by RAGHU SIR [NARESH IT, HYDERABAD, P: 040-2374 6666/ 9000994007 /08]
```

```
public Student() {
         super();
    public Student(int sid, String sname, double sfee,
List<String> subjs, Map<String, Integer> marks, Address
addr) {
         super();
         this.sid = sid;
         this.sname = sname;
         this.sfee = sfee;
         this.subjs = subjs;
         this.marks = marks;
         this.addr = addr;
    @Override
    public String toString() {
         return "Student [sid=
+ ", sfee=" + sfee + ", subjs="
                                 + subjs + ", marks=" +
marks
}
RestController
package in.nit.controller;
import java.util.List;
import java.util.Map;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import in.nit.model.Address;
import in.nit.model.Student;
@Path("/std")
```

```
-by RAGHU SIR [NARESH IT, HYDERABAD, P: 040-2374 6666/ 9000994007 /08]
```

```
public class StudentRestController {
    @GET
    @Path("/data")
    @Produces({"application/json", "application/xml"})
    //@Produces({MediaType.APPLICATION JSON,MediaType.APP
LICATION XML})
    public Student show() {
         return new Student(100, "AA", 3.3,
                   List.of("EN","MAT","TE"),
                   Map.of("E", 100, "R", 25),
                   new Address("8-88",
                   );
}
web.xml
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app 2 3.dtd" >
<web-app>
    <servlet>
         <servlet-name>sample</servlet-name>
         <servlet
class>org.glassfish.jersey.servlet.ServletContainer</servl</pre>
et-class>
         <init-param>
              <param-
name>jersey.config.server.provider.packages
              <param-value>in.nit.controller</param-value>
         </init-param>
     </servlet>
     <servlet-mapping>
         <servlet-name>sample</servlet-name>
         <url-pattern>/rest/*</url-pattern>
     </servlet-mapping>
</web-app>
```

## **Consumer Application**

```
pom.xml
<dependencies>
    <dependency>
              <groupId>org.glassfish.jersey.core
              <artifactId>jersey-client</artifactId>
              <version>2.29</version>
    </dependency>
     <dependency>
         <groupId>org.glassfish.jersey.inject
         <artifactId>jersey-hk2</artifactI</pre>
         <version>2.29</version>
     </dependency>
</dependencies>
Client Code:
package in.nit;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.core.Response;
public class Test
    public static void main(String[] args) {
         try {
              Response resp=ClientBuilder
                        .newClient()
     .target("http://localhost:2019/MediaTypeProducerTest"
              .path("/rest/std/data")
              .request()
              .header("Accept", "application/xml")
              .get();
              System.out.println(resp.getStatus());
```

System.out.println(resp.getStatusInfo());

FB: https://www.facebook.com/groups/thejavatemple/

email: javabyraghu@gmail.com