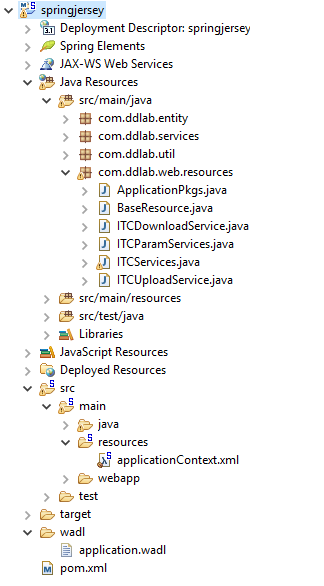
Spring Jersey RESTful webservice -April – 2017

# Project Structure

  
Maven Configuration (pom.xml)

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.ddlab.rnd.jersey</groupId>

<artifactId>springjersey</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>springjersey Maven Webapp</name>

<url>http://maven.apache.org</url>

<properties>

**<jersey.version>2.14</jersey.version>**

**<spring.version>3.2.1.RELEASE</spring.version>**

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

**<dependency>**

**<groupId>javax.ws.rs</groupId>**

**<artifactId>javax.ws.rs-api</artifactId>**

**<version>2.0</version>**

**</dependency>**

**<dependency>**

**<groupId>org.glassfish.jersey.containers</groupId>**

**<artifactId>jersey-container-servlet</artifactId>**

**<version>${jersey.version}</version>**

**</dependency>**

**<dependency>**

**<groupId>org.glassfish.jersey.core</groupId>**

**<artifactId>jersey-client</artifactId>**

**<version>${jersey.version}</version>**

**</dependency>**

**<dependency>**

**<groupId>org.glassfish.jersey.media</groupId>**

**<artifactId>jersey-media-multipart</artifactId>**

**<version>${jersey.version}</version>**

**</dependency>**

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-core-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>org.glassfish.jersey.media</groupId>

<artifactId>jersey-media-json-jackson</artifactId>

<version>2.5.1</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

<!-- Jersey-Spring integration -->

**<dependency>**

**<groupId>org.glassfish.jersey.ext</groupId>**

**<artifactId>jersey-spring3</artifactId>**

**<version>${jersey.version}</version>**

**</dependency>**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-beans</artifactId>**

**<version>${spring.version}</version>**

**</dependency>**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-context</artifactId>**

**<version>${spring.version}</version>**

**</dependency>**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-webmvc</artifactId>**

**<version>${spring.version}</version>**

**</dependency>**

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

<build>

<finalName>springjersey</finalName>

<plugins>

<plugin>

<groupId> org.apache.maven.plugins</groupId>

<artifactId> maven-compiler-plugin</artifactId>

<version>3.3</version>

<configuration>

<source> 1.8</source>

<target> 1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

# Spring Configuration (src/main/resources/applicationContext.xml)

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>

<context:component-scan base-package=*"com.ddlab.service.\*"* />

<bean id=*"userservice"* class=*"com.ddlab.services.UserServiceImpl"* />

</beans>

# Web Application Configuration(webapp/Web-INF/web.xml)

<web-app xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee*

*http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"*

version=*"3.1"*>

<display-name>Spring Jersey Restful example</display-name>

**<context-param>**

**<param-name>contextConfigLocation</param-name>**

**<param-value>classpath:applicationContext.xml</param-value>**

**</context-param>**

**<listener>**

**<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>**

**</listener>**

**<servlet>**

**<servlet-name>com.ddlab.web.resources.ApplicationPkgs</servlet-name>**

**<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>**

**<init-param>**

**<param-name>javax.ws.rs.Application</param-name>**

**<param-value>com.ddlab.web.resources.ApplicationPkgs</param-value>**

**</init-param>**

**<load-on-startup>1</load-on-startup>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>com.ddlab.web.resources.ApplicationPkgs</servlet-name>**

**<url-pattern>/api/\*</url-pattern>**

**</servlet-mapping>**

<welcome-file-list>

<welcome-file>form.html</welcome-file>

</welcome-file-list>

</web-app>

# Html File (form.html)

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Form Page</title>

</head>

<body>

<h1>Upload a File</h1>

<form action=*"api/upload/image"* method=*"post"* enctype=*"multipart/form-data"*>

<p> Select a file : <input type=*"file"* name=*"file"* size=*"45"* /> </p>

<input type=*"submit"* value=*"Upload It"* />

</form>

<br>

<strong>Multiple Files Upload</strong>

<form action=*"api/upload/multiupload"* method=*"post"* enctype=*"multipart/form-data"*>

<p> Select first file : <input type=*"file"* name=*"file1"*> </p>

<p> Select second file : <input type=*"file"* name=*"file2"*> </p>

<p> Select third file : <input type=*"file"* name=*"file3"*> </p>

<input type=*"submit"* value=*"Upload It"* />

</form>

</body>

</html>

# Java (Resource Layer)

## **ApplicationPkgs.java**

**package** com.ddlab.web.resources;

**import** java.util.HashSet;

**import** java.util.Set;

**import** org.glassfish.jersey.jackson.JacksonFeature;

**import** org.glassfish.jersey.media.multipart.MultiPartFeature;

**import** org.glassfish.jersey.server.ResourceConfig;

//To know about WADl : GET http://localhost:8090/springjersey/api/application.wadl

**public** **class** ApplicationPkgs **extends** ResourceConfig {

**public** ApplicationPkgs() {

//**super( ITCServices.class, ITCParamServices.class, ITCUploadService.class,ITCDownloadService.class, ITCHeaderParams.class, JacksonFeature.class, MultiPartFeature.class);**

// OR you can write below

Set<Class<?>> classes = **new** HashSet<Class<?>>();

classes.add(ITCServices.**class**);

classes.add(ITCParamServices.**class**);

classes.add(ITCUploadService.**class**);

classes.add(ITCDownloadService.**class**);

classes.add(ITCHeaderParams.**class**);

classes.add(JacksonFeature.**class**);

classes.add(MultiPartFeature.**class**);

registerClasses(classes);

}

}

## **BaseResource.java**

**package** com.ddlab.web.resources;

**import** java.nio.file.AccessDeniedException;

**import** javax.ws.rs.NotFoundException;

**import** javax.ws.rs.WebApplicationException;

**import** javax.ws.rs.core.Response;

**import** javax.ws.rs.core.Response.Status;

**public** **class** BaseResource {

**protected** WebApplicationException createWebappException(Exception incomingException) {

Status status;

String string = **null**;

**if** (incomingException **instanceof** SecurityException || incomingException **instanceof** AccessDeniedException) {

status = Status.***FORBIDDEN***;

} **else** **if** (incomingException **instanceof** IllegalArgumentException) {

status = Status.***BAD\_REQUEST***;

string=incomingException.getMessage();

} **else** **if** (incomingException **instanceof** NotFoundException) {

status = Status.***NOT\_FOUND***;

string=incomingException.getMessage();

} **else** **if** (incomingException **instanceof** Exception) {

status = Status.***INTERNAL\_SERVER\_ERROR***;

} **else** {

status = Status.***INTERNAL\_SERVER\_ERROR***;

}

**return** **new** **WebApplicationException(Response.*status*(status).entity(string).type("text/plain").build());**

}

}

## **ITCParamServices.java**

**package** com.ddlab.web.resources;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.ws.rs.FormParam;

**import** javax.ws.rs.GET;

**import** javax.ws.rs.MatrixParam;

**import** javax.ws.rs.POST;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.PathParam;

**import** javax.ws.rs.Produces;

**import** javax.ws.rs.QueryParam;

**import** javax.ws.rs.core.Context;

**import** javax.ws.rs.core.MediaType;

**import** javax.ws.rs.core.Response;

**import** com.ddlab.entity.User;

@Path("itcparams")

**public** **class** ITCParamServices **extends** BaseResource {

@Context

HttpServletRequest request;

@Context

ServletConfig servletConfig;

//**GET http://localhost:8090/springjersey/api/itcparams**

//**Status Code : 200 OK**

@GET

@Produces(MediaType.***TEXT\_PLAIN***)

**public** Response getOrganisationName() {

**return Response.*ok*().entity("ITC Infotech, Bangalore, Karnataka").build();**

}

// **GET http://localhost:8090/springjersey/api/itcparams/address**

//Status Code : 200 OK

@Path("/address")

@GET

@Produces(MediaType.***TEXT\_PLAIN***)

**public** Response getAddress() {

String addressString = "ITC Infotech India Limited, 18, Banaswadi Main Rd, Maruthi Sevanagar, Bangalore, 560005";

**return Response.*ok*().entity(addressString).build();**

}

// **~~~~~~~~~~~~~~~~~~~~~~~ @PathParam ~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

//**GET http://localhost:8090/springjersey/api/itcparams/user/1**

//Status Code : 200 OK

//Response : **{"firstName":"Deb","lastName":"Mishra","id":1}**

//In case of id = 0

//**GET http://localhost:8090/springjersey/api/itcparams/user/0**

//Status Code : 400 Bad Request

//Response body : Id cannot be 0, enter a valid id

@Path("/user/{id}")

@GET

@Produces({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

**public** Response getUserById(@PathParam("id") **int** id) {

**if**( id == 0 ) **throw** createWebappException(**new** IllegalArgumentException("Id cannot be 0, enter a valid id"));

User user = **new** User();

user.setFirstName("Deb");

user.setLastName("Mishra");

user.setId(id);

**return** Response.*ok*().entity(user).build();

}

//**GET http://localhost:8090/springjersey/api/itcparams/address/USA**

//**GET http://localhost:8090/springjersey/api/itcparams/address/europe**

//Status Code : 200 OK

@Path("/address/{areaCode}") //USA,EUROPE

@GET

@Produces(MediaType.***TEXT\_PLAIN***)

**public** Response getAddressByCode(@PathParam("areaCode") String areaCode) {

String addressString = **null**;

**try** {

**if** (areaCode.equalsIgnoreCase("USA"))

addressString = "12 North State, Route 17,Suite 303,Paramus,New Jersey,NJ-07652";

**else** **if** (areaCode.equalsIgnoreCase("Europe"))

addressString = "Newell Consulting Oy,P.O. Box 16 , Olari,02211 Espoo, Helsinki";

**else** **if** (areaCode.equalsIgnoreCase("Africa"))

addressString = "Johannesburg,2nd Floor, West Tower,Nelson Mandela Square,Maude Street, Sandton,Johannesburg, 2196";

**else** **if** (areaCode.equalsIgnoreCase("Asia"))

addressString = "ITC Infotech India Limited, 18, Banaswadi Main Rd, Maruthi Sevanagar, Bangalore, 560005";

**else** {

addressString = "No such area code exists for ITC";

}

} **catch** (Exception e) {

**throw createWebappException( new IllegalArgumentException("No such area code exists for ITC"));**

}

**return Response.*ok*().entity(addressString).build();**

}

// **~~~~~~~~~~~~~~~~~~~~~~~ @QueryParam ~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

//**GET http://localhost:8090/springjersey/api/itcparams/regionaladdress/Europe?country=SE**

//Status Code : 200 OK

@Path("/regionaladdress/{areaCode}") //USA,EUROPE

@GET

@Produces(MediaType.***TEXT\_PLAIN***)

**public** Response getAddressByCountry(@PathParam("areaCode") String areaCode, @QueryParam("country") String country) {

String adrsByCountry = **null**;

**try** {

**if** (areaCode.equalsIgnoreCase("USA")

&& country.equalsIgnoreCase("NJ"))

adrsByCountry = "12 North State, Route 17,Paramus,New Jersey,NJ-07652";

**else** **if** (areaCode.equalsIgnoreCase("Europe")

&& country.equalsIgnoreCase("FI"))

adrsByCountry = "Newell Consulting Oy,P.O. Box 16 , Olari,02211 Espoo, Helsinki";

**else** **if** (areaCode.equalsIgnoreCase("Europe")

&& country.equalsIgnoreCase("SE"))

adrsByCountry = "C/o Matrisen AB,Box 22059 , 104 22 Stockholm";

**else** **if** (areaCode.equalsIgnoreCase("Europe")

&& country.equalsIgnoreCase("DK"))

adrsByCountry = "Havnegade 39, 3. sal,1058 Copenhagen K";

**else** **if** (areaCode.equalsIgnoreCase("Asia")

&& country.equalsIgnoreCase("IN"))

adrsByCountry = "ITC Infotech India Limited, 18, Banaswadi Main Rd, Maruthi Sevanagar, Bangalore, 560005";

**else**

adrsByCountry = "No such area code exists for ITC";

} **catch** (Exception e) {

**throw createWebappException( new IllegalArgumentException("No such area code exists for ITC"));**

}

**return Response.*ok*().entity(adrsByCountry).build();**

}

// **~~~~~~~~~~~~~~~~~~~~~~~ @MatrixParam ~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

//**GET http://localhost:8090/springjersey/api/itcparams/itcaddress;country=FI;areacode=europe**

//Status Code : 200 OK

@Path("/itcaddress")

@GET

@Produces(MediaType.***TEXT\_PLAIN***)

**public** Response getITCAddress(@MatrixParam("country") String country, @MatrixParam("areacode") String areaCode) {

**return** getAddressByCountry(areaCode, country);

}

// **~~~~~~~~~~~~~~~~~~~~~~~ @FormParam ~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

//**POST http://localhost:8090/springjersey/api/itcparams/postaladdress**

//**Content-Type : application/x-www-form-urlencoded**

//**Body : country=FI&areacode=Europe**

//Status Code : 200 OK

@Path("/postaladdress")

@POST

@Produces(MediaType.***TEXT\_PLAIN***)

**public** Response postNGetITCAddress(@FormParam("country") String country,@FormParam("areacode") String areaCode) {

**return** getAddressByCountry(areaCode, country);

}

}

## **ITCServices.java**

**package** com.ddlab.web.resources;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.ws.rs.Consumes;

**import** javax.ws.rs.DELETE;

**import** javax.ws.rs.NotFoundException;

**import** javax.ws.rs.POST;

**import** javax.ws.rs.PUT;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.Produces;

**import** javax.ws.rs.core.Context;

**import** javax.ws.rs.core.MediaType;

**import** javax.ws.rs.core.Response;

**import** javax.ws.rs.core.Response.Status;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** com.ddlab.entity.User;

**import** com.ddlab.services.IUserService;

@Path("itc")

**public** **class** ITCServices **extends** BaseResource {

@Context

HttpServletRequest request;

@Context

ServletConfig servletConfig;

@Autowired

**private** IUserService userService;

/\*

\* **POST http://localhost:8090/springjersey/api/itc/createuser**

\* **Content-Type : application/json**

\* Body :

\* {

\* "firstName" : "Deb",

\* "lastName" : "Mishra",

\* "id" : 1

\* }

\*

\* Response Status Code: 201 Created

\*/

@Path("/createuser")

@POST

@Consumes({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

@Produces({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

**public** Response createEmp(User user) {

userService.createUser(user);

**return Response.*status*(Status.*CREATED*).entity("User " + user.getFirstName()+" created successfully...").build();**

}

/\*

\* **PUT http://localhost:8090/springjersey/api/itc/updateuser**

\* Content-Type : application/json

\* Body :

\* {

\* "firstName" : "Deb",

\* "lastName" : "Mishra",

\* "id" : 1

\* }

\*

\* Response Status Code: 200 OK

\*

\* In case of wrong message like id = 0

\* **PUT http://localhost:8090/springjersey/api/itc/updateuser**

\* Content-Type : application/json

\* Body :

\* {

\* "firstName" : "Deb",

\* "lastName" : "Mishra",

\* "id" : 0

\* }

\*

\* Response 404 Not Found

\*/

@Path("/updateuser")

@PUT

@Consumes({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

@Produces({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

**public** Response updateEmp(User user) {

**if**( user.getId() == 0 )

**throw** createWebappException(**new** NotFoundException("No such employee exists..."));

**else**

userService.updateUser(user);

**return** Response.*ok*("User " + user.getFirstName()+" updated successfully...").build();

}

/\*

\* **DELETE http://localhost:8090/springjersey/api/itc/deleteuser**

\* Content-Type : application/json

\* Body :

\* {

\* "firstName" : "Deb",

\* "lastName" : "Mishra",

\* "id" : 1

\* }

\*

\* Response Status Code: 200 OK

\*

\* In case of wrong message like id = 0

\* PUT http://localhost:8090/springjersey/api/itc/deleteuser

\* Content-Type : application/json

\* Body :

\* {

\* "firstName" : "Deb",

\* "lastName" : "Mishra",

\* "id" : 0

\* }

\*

\* Response 404 Not Found

\*/

@Path("/deleteuser")

@DELETE

@Consumes({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

@Produces({ MediaType.***APPLICATION\_XML***, MediaType.***APPLICATION\_JSON*** })

**public** Response deleteEmp(User user) {

**if**( user.getId() == 0 )

**throw** createWebappException(**new** NotFoundException("No such employee exists..."));

**else**

userService.deleteUser(user);

**return** Response.*ok*("User " + user.getFirstName()+" deleted successfully...").build();

}

}

## **ITCUploadService.java**

**package** com.ddlab.web.resources;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.util.List;

**import** java.util.Map;

**import** javax.ws.rs.Consumes;

**import** javax.ws.rs.POST;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.core.MediaType;

**import** javax.ws.rs.core.Response;

**import** org.glassfish.jersey.media.multipart.BodyPart;

**import** org.glassfish.jersey.media.multipart.BodyPartEntity;

**import** org.glassfish.jersey.media.multipart.ContentDisposition;

**import** org.glassfish.jersey.media.multipart.FormDataBodyPart;

**import** org.glassfish.jersey.media.multipart.FormDataContentDisposition;

**import** org.glassfish.jersey.media.multipart.FormDataMultiPart;

**import** org.glassfish.jersey.media.multipart.FormDataParam;

**import** com.ddlab.entity.User;

**import** com.ddlab.util.Constants;

**import** com.ddlab.util.FileUtil;

@Path("upload")

**public** **class** ITCUploadService **extends** BaseResource {

//**POST http://localhost:8090/springjersey/api/upload/image**

//**In case of postman client**

//**Choose form-data, file : c.jpg**

@POST

@Path("/image")

**@Consumes(MediaType.*MULTIPART\_FORM\_DATA*)**

**public** Response **uploadFile(FormDataMultiPart form)** {

FormDataBodyPart filePart = form.getField("file");

ContentDisposition headerOfFilePart = filePart.getContentDisposition();

InputStream fileInputStream = filePart.getValueAs(InputStream.**class**);

String filePath = Constants.***SERVER\_UPLOAD\_LOCATION\_FOLDER*** + headerOfFilePart.getFileName();

// save the file to the server

FileUtil.*saveFile*(fileInputStream, filePath);

String output = "File saved to server location using FormDataMultiPart : " + filePath;

**try** {

fileInputStream.close();

} **catch** (IOException e) {

e.printStackTrace();

}

**return** Response.*status*(200).entity(output).build();

}

/\*

\* Another way to upload file

\* **POST http://localhost:8090/springjersey/api/upload/image1**

\* **In case of postman client, choose form-data file : a.jpg**

\*

\*/

@POST

@Path("/image1")

**@Consumes(MediaType.*MULTIPART\_FORM\_DATA*)**

**public** Response uploadFile(

**@FormDataParam("file") InputStream fileInputStream,**

**@FormDataParam("file") FormDataContentDisposition contentDispositionHeader)** {

String filePath = Constants.***SERVER\_UPLOAD\_LOCATION\_FOLDER***

+ contentDispositionHeader.getFileName();

// save the file to the server

String output = "";

**try** {

FileUtil.*saveFile*(fileInputStream, filePath);

output = "File saved to server location : " + filePath

+ " successfully";

} **catch** (Exception e) {

e.printStackTrace();

output = "Unexpected server exception while uploading the file.";

}

**return** Response.*status*(200).entity(output).build();

}

/\*

\* **Posting image with data**

\* **POST http://localhost:8090/springjersey/api/upload/imageWithData**

\* **In case of postman client choose form-data**

\* **file: a.jpg**

\* **user: { "firstName" : "Deb", "lastName" : "Mishra","id" : 1 }**

\*/

@POST

@Path("/imageWithData")

**@Consumes(MediaType.*MULTIPART\_FORM\_DATA*)**

**public** Response uploadFile(**FormDataMultiPart form, @FormDataParam("user") String user**) {

System.***out***.println("User-------->" + user.toString());

FormDataBodyPart filePart = form.getField("file");

ContentDisposition headerOfFilePart = filePart.getContentDisposition();

InputStream fileInputStream = filePart.getValueAs(InputStream.**class**);

String filePath = Constants.***SERVER\_UPLOAD\_LOCATION\_FOLDER***

+ headerOfFilePart.getFileName();

// save the file to the server

FileUtil.*saveFile*(fileInputStream, filePath);

String output = "File saved to server location using FormDataMultiPart : "

+ filePath;

**return** Response.*status*(200).entity(output).build();

}

/\*

\* **Posting image with data**

\* **POST http://localhost:8090/springjersey/api/upload/imageWithData1**

\* **In case of postman client choose form-data**

\* **file: a.jpg**

\* **user: { "firstName" : "Deb", "lastName" : "Mishra","id" : 1 }**

\*/

@POST

@Path("/imageWithData1")

**@Consumes(MediaType.*MULTIPART\_FORM\_DATA*)**

**public** Response uploadFileWithData(**FormDataMultiPart form**) {

Map<String, List<FormDataBodyPart>> params = form.getFields();

System.***out***.println(params);

String myJson = params.get("user").get(0).getValue();

System.***out***.println("myjson---------->" + myJson);

User user = User.*toUser*(myJson);

System.***out***.println("User-------->" + user.toXML());

FormDataBodyPart filePart = form.getField("file");

ContentDisposition headerOfFilePart = filePart.getContentDisposition();

InputStream fileInputStream = filePart.getValueAs(InputStream.**class**);

String filePath = Constants.***SERVER\_UPLOAD\_LOCATION\_FOLDER***

+ headerOfFilePart.getFileName();

// save the file to the server

FileUtil.*saveFile*(fileInputStream, filePath);

String output = "File saved to server location using FormDataMultiPart : "+ filePath;

**try** {

fileInputStream.close();

} **catch** (IOException e) {

e.printStackTrace();

}

**return** Response.*status*(200).entity(output).build();

}

/\*

\* Posting multiple images

\* POST http://localhost:8090/springjersey/api/upload/multiupload

\* In case of postman client choose form-data

\* file1: a.jpg

\* file2: b.jpg

\* You can also use the html file called form.html

\*/

@POST

@Path("/multiupload")

**@Consumes(MediaType.*MULTIPART\_FORM\_DATA*)**

**public** Response uploadMultipleFile(**FormDataMultiPart form**) {

BodyPartEntity bodyPartEntity;

**for** (BodyPart part : form.getBodyParts()) {

bodyPartEntity = (BodyPartEntity) part.getEntity();

String fileName = part.getContentDisposition().getFileName();

System.***out***

.println("part.getContentDisposition().getFileName()---------->"

+ fileName);

InputStream fileInputStream = bodyPartEntity.getInputStream();

String filePath = Constants.***SERVER\_UPLOAD\_LOCATION\_FOLDER*** + fileName;

FileUtil.*saveFile*(fileInputStream, filePath);

}

**return** Response.*status*(200).entity("All Files uploaded successfully ... ").build();

}

}

## **ITCDownloadService.java**

**package** com.ddlab.web.resources;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.InputStream;

**import** javax.ws.rs.GET;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.Produces;

**import** javax.ws.rs.QueryParam;

**import** javax.ws.rs.core.MediaType;

**import** javax.ws.rs.core.Response;

**import** javax.ws.rs.core.Response.Status;

**import** com.ddlab.util.FileUtil;

@Path("download")

**public** **class** ITCDownloadService **extends** BaseResource {

/\*

\* You can directly hit the url in the browser

\* GET localhost:8090/springjersey/api/download/image

\*/

@Path("/image")

@GET

**@Produces("image/jpg")**

**public** Response getFile() {

**byte**[] docStream = FileUtil.*readContents*("D:/temp/k.jpg");

**return** Response.*ok*(docStream, MediaType.***APPLICATION\_OCTET\_STREAM***)

.header("content-disposition", "attachment; filename = k.jpg").build();

}

/\*

\* You can directly hit the url in the browser

\* GET http://localhost:8090/springjersey/api/download/image1?fileName=k.jpg

\*/

@Path("/image1")

@GET

**@Produces(MediaType.*APPLICATION\_OCTET\_STREAM*)**

**public** Response getFile(@QueryParam("fileName") String fileName) {

// Do not provide path param

Response response = **null**;

String dirPath = "D:/temp";

String filePath = dirPath + File.***separator*** + fileName;

**try** {

**if** (!**new** File(filePath).exists())

**throw** **new** FileNotFoundException();

**byte**[] docStream = FileUtil.*readContents*(dirPath + File.***separator***

+ fileName);

String attachment = "attachment; filename = " + fileName;

response = Response

.*ok*(docStream, MediaType.***APPLICATION\_OCTET\_STREAM***)

.header("content-disposition", attachment).build();

} **catch** (FileNotFoundException e) {

response = Response.*serverError*().status(Status.***BAD\_REQUEST***)

.build();

}

**return** response;

}

/\* You can directly hit the url in the browser

\* GET http://localhost:8090/springjersey/api/download/showImage?fileName=k.jpg

\*/

@Path("/showImage")

@GET

**@Produces("image/jpg")**

**public** Response showImage(@QueryParam("fileName") String fileName) {

Response response = **null**;

String dirPath = "D:/temp";

InputStream inStream = **null**;

String filePath = dirPath + File.***separator*** + fileName;

**try** {

**if** (!**new** File(filePath).exists())

**throw** **new** FileNotFoundException();

inStream = **new** FileInputStream(filePath);

response = Response.*ok*().entity(inStream).build();

} **catch** (FileNotFoundException e) {

e.printStackTrace();

response = Response.*serverError*().status(Status.***BAD\_REQUEST***)

.build();

}

**return** response;

}

}

## **ITCHeaderParams.java**

**package** com.ddlab.web.resources;

**import** javax.ws.rs.GET;

**import** javax.ws.rs.HeaderParam;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.core.Context;

**import** javax.ws.rs.core.HttpHeaders;

**import** javax.ws.rs.core.Response;

@Path("/headers")

**public** **class** ITCHeaderParams {

/\*

\* To know the user-agent header

\* GET http://localhost:8090/springjersey/api/headers/get

\*

\* Response will be

\* In FireFox

\* addUser is called, userAgent : Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0

\*

\* In Chrome

\* addUser is called, userAgent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

\*/

@GET

@Path("/get")

**public** Response addUser(@HeaderParam("user-agent") String userAgent) {

String msg = "addUser is called, userAgent : " + userAgent;

**return** Response.*status*(200).entity(msg).build();

}

/\*

\* To know the user-agent header

\* GET http://localhost:8090/springjersey/api/headers/get

\*

\* Response will be

\* In FireFox

\* addUser is called, userAgent : Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0

\*

\* In Chrome

\* addUser is called, userAgent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

\*

\* You can set your custom header like myHeader: myHeaderValue

\*/

@GET

@Path("/getAll")

**public** Response addUser(@Context HttpHeaders headers) {

//Show All Headers

System.***out***.println("All header details ...\n"+headers.getRequestHeaders());

//It will display

//{host=[localhost:8090], connection=[keep-alive], cache-control=[max-age=0], upgrade-insecure-requests=[1], user-agent=[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36], accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8], accept-encoding=[gzip, deflate, sdch, br], accept-language=[en-US,en;q=0.8]}

**for**(String header : headers.getRequestHeaders().keySet()){

System.***out***.println(header);

//It will display below

//host

//connection

//cache-control

//upgrade-insecure-requests

//user-agent

//accept

//accept-encoding

//accept-language

}

String userAgent = headers.getRequestHeader("user-agent").get(0);

String msg = "addUser is called, userAgent : " + userAgent;

**return** Response.*status*(200).entity(msg).build();

}

}

# Java (Util Layer)

## **FileUtil.java**

**package** com.ddlab.util;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.OutputStream;

**public** **class** FileUtil {

// save uploaded file to a defined location on the server

**public** **static** **void** saveFile(InputStream uploadedInputStream, String serverLocation) {

OutputStream outpuStream = **null**;

**try** {

**int** read = 0;

**byte**[] bytes = **new** **byte**[1024];

outpuStream = **new** FileOutputStream(**new** File(serverLocation));

**while** ((read = uploadedInputStream.read(bytes)) != -1) {

outpuStream.write(bytes, 0, read);

}

} **catch** (IOException e) {

e.printStackTrace();

} **finally** {

**try** {

outpuStream.flush();

outpuStream.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

**public** **static** **byte**[] readContents(String filePath) {

File file = **new** File(filePath);

**byte**[] buffer = **new** **byte**[(**int**) file.length()];

InputStream inStream = **null**;

**try** {

inStream = **new** FileInputStream(file);

inStream.read(buffer);

} **catch** (FileNotFoundException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

} **finally** {

**try** {

**if** (inStream != **null**) inStream.close();

} **catch** (Exception e2) { }

}

**return** buffer;

}

}

## **Constants.java**

**package** com.ddlab.util;

**public** **interface** Constants {

**public** **static** **final** String ***SERVER\_UPLOAD\_LOCATION\_FOLDER*** = "D:/temp/";

**public** **static** **final** String ***DATA\_LOCATION\_FOLDER*** = "D:/temp/data";

}

# Java (Service Layer)

## **IUserService.java (Interface)**

**package** com.ddlab.services;

**import** com.ddlab.entity.User;

**public** **interface** IUserService {

**public** **void** createUser(User user);

**public** **void** updateUser(User user);

**public** **void** deleteUser(User user);

**public** User getUserById(String id);

}

## **UserServiceImpl.java**

**package** com.ddlab.services;

**import** com.ddlab.entity.User;

**public** **class** UserServiceImpl **implements** IUserService {

**public** **void** createUser(User user) {

System.***out***.println("User created successfully...");

}

**public** **void** updateUser(User user) {

System.***out***.println("User information updated successfully...");

}

**public** **void** deleteUser(User user) {

System.***out***.println("User information deleted successfully...");

}

**public** User getUserById(String id) {

**if**( id == **null** ) **throw** **new** NullPointerException("User id can not be blank or empty");

User user = **new** User();

user.setFirstName("Deb");

user.setLastName("Mishra");

user.setId(Integer.*parseInt*(id));

**return** user;

}

}

# Java (Entity Layer)

## **User.java**

**package** com.ddlab.entity;

**import** java.io.IOException;

**import** java.io.StringWriter;

**import** javax.xml.bind.JAXBContext;

**import** javax.xml.bind.JAXBException;

**import** javax.xml.bind.Marshaller;

**import** javax.xml.bind.annotation.XmlAccessType;

**import** javax.xml.bind.annotation.XmlAccessorType;

**import** javax.xml.bind.annotation.XmlElement;

**import** javax.xml.bind.annotation.XmlRootElement;

**import** javax.xml.bind.annotation.XmlType;

**import** org.codehaus.jackson.JsonParseException;

**import** org.codehaus.jackson.annotate.JsonProperty;

**import** org.codehaus.jackson.annotate.JsonPropertyOrder;

**import** org.codehaus.jackson.map.JsonMappingException;

**import** org.codehaus.jackson.map.ObjectMapper;

@XmlRootElement(name = "User")

@XmlAccessorType(XmlAccessType.***FIELD***)

@XmlType(propOrder = { "firstName", "lastName", "id" })

@JsonPropertyOrder(value = { "firstName", "lastName", "id" })

**public** **class** User {

@XmlElement(name = "firstName")

@JsonProperty("firstName")

**private** String firstName;

@XmlElement(name = "lastName")

@JsonProperty("lastName")

**private** String lastName;

@XmlElement(name = "id")

@JsonProperty("id")

**private** **int** id;

**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;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

@Override

**public** String toString() {

**return** "User [firstName=" + firstName + ", lastName=" + lastName + ", id=" + id + "]";

}

**public** String toJSON() {

ObjectMapper mapper = **new** ObjectMapper();

String toJson = **null**;

**try** {

**toJson = mapper.writerWithDefaultPrettyPrinter().writeValueAsString(this);**

} **catch** (IOException e) {

e.printStackTrace();

}

**return** toJson;

}

**public** String toXML() {

String xmlString = "";

**try** {

JAXBContext context = JAXBContext.*newInstance*(**this**.getClass());

Marshaller m = context.createMarshaller();

m.setProperty(Marshaller.***JAXB\_FORMATTED\_OUTPUT***, Boolean.***TRUE***);

StringWriter sw = **new** StringWriter();

m.marshal(**this**, sw);

xmlString = sw.toString();

} **catch** (JAXBException e) {

e.printStackTrace();

}

**return** xmlString;

}

**public** **static** User toUser(String jsonString) {

ObjectMapper mapper = **new** ObjectMapper();

User user = **null**;

**try** {

user = (User) mapper.readValue(jsonString, User.**class**);

} **catch** (JsonParseException e) {

e.printStackTrace();

} **catch** (JsonMappingException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

}

**return** user;

}

// public static void main(String[] args) {

// User user = new User();

// user.setFirstName("Deb");

// user.setLastName("Mishra");

// user.setId(1);

// System.out.println(user.toJSON());

//

// System.out.println(user.toXML());

//

// }

}

# Jersey generated WADL File

## **application.wadl**

<?xml version=*"1.0"* encoding=*"UTF-8"* standalone=*"yes"*?>

<application xmlns=*"http://wadl.dev.java.net/2009/02"*>

<doc xmlns:jersey=*"http://jersey.java.net/"* jersey:generatedBy=*"Jersey: 2.14 2014-12-11 07:22:06"*/>

<doc xmlns:jersey=*"http://jersey.java.net/"* jersey:hint=*"This is simplified WADL with user and core resources only. To get full WADL with extended resources use the query parameter detail. Link: http://localhost:8090/springjersey/api/application.wadl?detail=true"*/>

<grammars>

<include href=*"application.wadl/xsd0.xsd"*>

<doc title=*"Generated"* xml:lang=*"en"*/>

</include>

</grammars>

<resources base=*"http://localhost:8090/springjersey/api/"*>

<resource path=*"download"*>

<resource path=*"/image"*>

<method id=*"getFile"* name=*"GET"*>

<response>

<representation mediaType=*"image/jpg"*/>

</response>

</method>

</resource>

<resource path=*"/showImage"*>

<method id=*"showImage"* name=*"GET"*>

<request>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"fileName"* style=*"query"* type=*"xs:string"*/>

</request>

<response>

<representation mediaType=*"image/jpg"*/>

</response>

</method>

</resource>

<resource path=*"/image1"*>

<method id=*"getFile"* name=*"GET"*>

<request>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"fileName"* style=*"query"* type=*"xs:string"*/>

</request>

<response>

<representation mediaType=*"application/octet-stream"*/>

</response>

</method>

</resource>

</resource>

<resource path=*"itcparams"*>

<method id=*"getOrganisationName"* name=*"GET"*>

<response>

<representation mediaType=*"text/plain"*/>

</response>

</method>

<resource path=*"/itcaddress"*>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"country"* style=*"matrix"* type=*"xs:string"*/>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"areacode"* style=*"matrix"* type=*"xs:string"*/>

<method id=*"getITCAddress"* name=*"GET"*>

<response>

<representation mediaType=*"text/plain"*/>

</response>

</method>

</resource>

<resource path=*"/postaladdress"*>

<method id=*"postNGetITCAddress"* name=*"POST"*>

<request>

<representation mediaType=*"application/x-www-form-urlencoded"*>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"country"* style=*"query"* type=*"xs:string"*/>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"areacode"* style=*"query"* type=*"xs:string"*/>

</representation>

</request>

<response>

<representation mediaType=*"text/plain"*/>

</response>

</method>

</resource>

<resource path=*"/address"*>

<method id=*"getAddress"* name=*"GET"*>

<response>

<representation mediaType=*"text/plain"*/>

</response>

</method>

</resource>

<resource path=*"/address/{areaCode}"*>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"areaCode"* style=*"template"* type=*"xs:string"*/>

<method id=*"getAddressByCode"* name=*"GET"*>

<response>

<representation mediaType=*"text/plain"*/>

</response>

</method>

</resource>

<resource path=*"/user/{id}"*>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"id"* style=*"template"* type=*"xs:int"*/>

<method id=*"getUserById"* name=*"GET"*>

<response>

<representation mediaType=*"application/xml"*/>

<representation mediaType=*"application/json"*/>

</response>

</method>

</resource>

<resource path=*"/regionaladdress/{areaCode}"*>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"areaCode"* style=*"template"* type=*"xs:string"*/>

<method id=*"getAddressByCountry"* name=*"GET"*>

<request>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"country"* style=*"query"* type=*"xs:string"*/>

</request>

<response>

<representation mediaType=*"text/plain"*/>

</response>

</method>

</resource>

</resource>

<resource path=*"upload"*>

<resource path=*"/imageWithData"*>

<method id=*"uploadFile"* name=*"POST"*>

<request>

<representation mediaType=*"multipart/form-data"*/>

</request>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

<resource path=*"/imageWithData1"*>

<method id=*"uploadFileWithData"* name=*"POST"*>

<request>

<representation mediaType=*"multipart/form-data"*/>

</request>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

<resource path=*"/multiupload"*>

<method id=*"uploadMultipleFile"* name=*"POST"*>

<request>

<representation mediaType=*"multipart/form-data"*/>

</request>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

<resource path=*"/image"*>

<method id=*"uploadFile"* name=*"POST"*>

<request>

<representation mediaType=*"multipart/form-data"*/>

</request>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

<resource path=*"/image1"*>

<method id=*"uploadFile"* name=*"POST"*>

<request>

<representation mediaType=*"multipart/form-data"*/>

</request>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

</resource>

<resource path=*"itc"*>

<resource path=*"/deleteuser"*>

<method id=*"deleteEmp"* name=*"DELETE"*>

<request>

<ns2:representation xmlns:ns2=*"http://wadl.dev.java.net/2009/02"* xmlns=*""* element=*"User"* mediaType=*"application/xml"*/>

<ns2:representation xmlns:ns2=*"http://wadl.dev.java.net/2009/02"* xmlns=*""* element=*"User"* mediaType=*"application/json"*/>

</request>

<response>

<representation mediaType=*"application/xml"*/>

<representation mediaType=*"application/json"*/>

</response>

</method>

</resource>

<resource path=*"/updateuser"*>

<method id=*"updateEmp"* name=*"PUT"*>

<request>

<ns2:representation xmlns:ns2=*"http://wadl.dev.java.net/2009/02"* xmlns=*""* element=*"User"* mediaType=*"application/xml"*/>

<ns2:representation xmlns:ns2=*"http://wadl.dev.java.net/2009/02"* xmlns=*""* element=*"User"* mediaType=*"application/json"*/>

</request>

<response>

<representation mediaType=*"application/xml"*/>

<representation mediaType=*"application/json"*/>

</response>

</method>

</resource>

<resource path=*"/createuser"*>

<method id=*"createEmp"* name=*"POST"*>

<request>

<ns2:representation xmlns:ns2=*"http://wadl.dev.java.net/2009/02"* xmlns=*""* element=*"User"* mediaType=*"application/xml"*/>

<ns2:representation xmlns:ns2=*"http://wadl.dev.java.net/2009/02"* xmlns=*""* element=*"User"* mediaType=*"application/json"*/>

</request>

<response>

<representation mediaType=*"application/xml"*/>

<representation mediaType=*"application/json"*/>

</response>

</method>

</resource>

</resource>

<resource path=*"/headers"*>

<resource path=*"/getAll"*>

<method id=*"addUser"* name=*"GET"*>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

<resource path=*"/get"*>

<method id=*"addUser"* name=*"GET"*>

<request>

<param xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* name=*"user-agent"* style=*"header"* type=*"xs:string"*/>

</request>

<response>

<representation mediaType=*"\*/\*"*/>

</response>

</method>

</resource>

</resource>

</resources>

</application>

# Basic Structure of WADL

A sample method is given below.

**@Path("country/{id}")**

public static class CountryResource {

     private CountryService countryService;

     public CountryResource() {

        // init countryService

    }

    @GET

**@Produces(MediaType.APPLICATION\_XML)**

    public Country getCountry(@PathParam("countryId") int countryId) {

        return countryService.getCountry(countryId);

    }

}

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<application xmlns="<http://wadl.dev.java.net/2009/02>">

    <doc xmlns:jersey="<http://jersey.java.net/>" jersey:generatedBy="Jersey: 2.5-SNAPSHOT 2013-12-20 17:14:21"/>

    <grammars/>

    <resources base="<http://localhost:9998/>">

**<resource path="country/{id}">**

            <param xmlns:xs="<http://www.w3.org/2001/XMLSchema>" type="xs:int" style="template" name="countryId"/>

**<method name="GET" id="getCountry">**

**<response>**

**<representation mediaType="application/xml"/>**

**</response>**

**</method>**

        </resource>

    </resources>

</application>