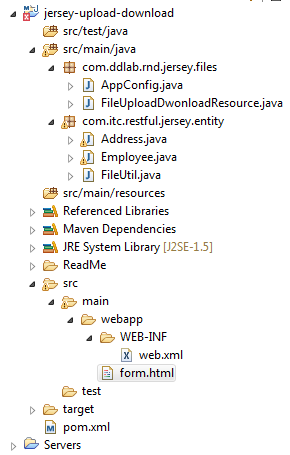
File Upload and Download using Jersey

# **Application 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>jersey-upload-download</groupId>

<artifactId>jersey-upload-download</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>jersey-upload-download Maven Webapp</name>

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

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

**<!—Important Libraries for File Upload and Download -->**

**<dependency>**

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

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

**<version>2.0.1</version>**

**</dependency>**

**<dependency>**

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

**<artifactId>jersey-common</artifactId>**

**<version>2.14</version>**

**</dependency>**

**<dependency>**

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

**<artifactId>jersey-server</artifactId>**

**<version>2.14</version>**

**</dependency>**

**<dependency>**

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

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

**<version>2.14</version>**

**</dependency>**

**<dependency>**

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

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

**<version>2.14</version>**

**</dependency>**

**<dependency>**

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

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

**<version>2.14</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>

<!-- The following is required for object to json -->

<dependency>

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

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

<version>2.14</version>

</dependency>

</dependencies>

<build>

<finalName>jersey-upload-download</finalName>

</build>

</project>

# **Deployment Descriptor (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>jersey-upload-download</display-name>

<servlet>

<servlet-name>com.ddlab.rnd.jersey.files.AppConfig</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.rnd.jersey.files.AppConfig</param-value>

</init-param>

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

</servlet>

<servlet-mapping>

<servlet-name>com.ddlab.rnd.jersey.files.AppConfig</servlet-name>

<url-pattern>/rest/\*</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=*"rest/files/upload"* 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=*"rest/files/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 Files**

## **Entity Files**

### **FileUtil.java**

**package** com.itc.restful.jersey.entity;

**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 {

**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;

}

**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** {

**if**( outpuStream != **null** ) {

outpuStream.flush();

outpuStream.close();

}

} **catch** (Exception e2) {

}

}

}

}

### **Employee.java**

**package** com.itc.restful.jersey.entity;

**import** java.io.IOException;

**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** org.codehaus.jackson.JsonFactory;

**import** org.codehaus.jackson.JsonGenerationException;

**import** org.codehaus.jackson.JsonNode;

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

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

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

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

**import** com.fasterxml.jackson.core.JsonParser;

@XmlRootElement(name = "Emp")

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

**public** **class** Employee {

@XmlElement(name = "Name")

@JsonProperty("Name")

**private** String name;

@XmlElement(name = "Age")

@JsonProperty("Age")

**private** **int** age;

@XmlElement(name = "Email")

@JsonProperty("Email")

**private** String emailId;

@XmlElement(name = "Adrs")

@JsonProperty("Adrs")

**private** Address adrs;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getEmailId() {

**return** emailId;

}

**public** **void** setEmailId(String emailId) {

**this**.emailId = emailId;

}

**public** Address getAdrs() {

**return** adrs;

}

**public** **void** setAdrs(Address adrs) {

**this**.adrs = adrs;

}

**public** String toJSON() {

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

**try** {

**return** mapper.writeValueAsString(**this**);

} **catch** (JsonGenerationException e) {

e.printStackTrace();

} **catch** (JsonMappingException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

}

**return** **null**;

}

**public** **static** Employee toEmp(String jsonString) {

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

Employee emp = **null**;

**try** {

emp = (Employee) mapper.readValue(jsonString, Employee.**class**);

} **catch** (JsonParseException e) {

e.printStackTrace();

} **catch** (JsonMappingException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

}

**return** emp;

}

}

### **Address.java**

**package** com.itc.restful.jersey.entity;

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

@XmlRootElement(name = "Adrs")

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

**public** **class** Address {

@XmlElement(name = "DoorNo")

@JsonProperty("DoorNo")

**private** String doorNo;

@XmlElement(name = "Street")

@JsonProperty("Street")

**private** String streetId;

@XmlElement(name = "City")

@JsonProperty("City")

**private** String city;

@XmlElement(name = "Country")

@JsonProperty("Country")

**private** String country;

**public** String getDoorNo() {

**return** doorNo;

}

**public** **void** setDoorNo(String doorNo) {

**this**.doorNo = doorNo;

}

**public** String getStreetId() {

**return** streetId;

}

**public** **void** setStreetId(String streetId) {

**this**.streetId = streetId;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

}

## **Resource Files**

### **AppConfig.java**

**package** com.ddlab.rnd.jersey.files;

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

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

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

**public** **class** AppConfig **extends** ResourceConfig {

**public** AppConfig() {

//For MultiPart, you have to register MultiPartFeature.class

**super**(FileUploadDwonloadResource.**class**, JacksonFeature.**class**, MultiPartFeature.**class**);

}

}

### **FileUploadDownloadResource.java**

**package** com.ddlab.rnd.jersey.files;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**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.GET;

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

**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** 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.itc.restful.jersey.entity.Employee;

**import** com.itc.restful.jersey.entity.FileUtil;

@Path("/files")

**public** **class** FileUploadDwonloadResource {

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

**@POST**

**@Path("/upload1")**

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

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

**// http://localhost:8080/jersey-upload-download/rest/files/upload1**

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

**ContentDisposition headerOfFilePart = filePart.getContentDisposition();**

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

**String filePath = *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();**

**}**

**// Another way of file uploading**

**@POST**

**@Path("/upload")**

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

**public Response uploadFile(**

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

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

**// http://localhost:8080/jersey-upload-download/rest/files/upload**

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

**+ contentDisposition.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();**

**}**

**@POST**

**@Path("/upload2")**

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

**public Response uploadFile(FormDataMultiPart form,**

**@FormDataParam("emp") String emp) {**

**// http://localhost:8080/jersey-upload-download/rest/files/upload2**

**System.*out*.println("Employee-------->" + emp.toString());**

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

**ContentDisposition headerOfFilePart = filePart.getContentDisposition();**

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

**String filePath = *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();**

**}**

**@POST**

**@Path("/upload3")**

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

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

**// In this case, you extract the emp key from the**

**// http://localhost:8080/jersey-upload-download/rest/files/upload3**

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

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

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

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

**Employee emp = Employee.*toEmp*(myJson);**

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

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

**ContentDisposition headerOfFilePart = filePart.getContentDisposition();**

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

**String filePath = *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();**

**}**

**@POST**

**@Path("/multiupload")**

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

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

**// http://localhost:8080/jersey-upload-download/rest/files/multiupload**

**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 = *SERVER\_UPLOAD\_LOCATION\_FOLDER* + fileName;**

**FileUtil.*saveFile*(fileInputStream, filePath);**

**}**

**return Response.*status*(200)**

**.entity("All Files uploaded successfully ... ").build();**

**}**

**@Path("/download")**

**@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;**

**}**

**@Path("/show/image")**

**@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;**

**}**

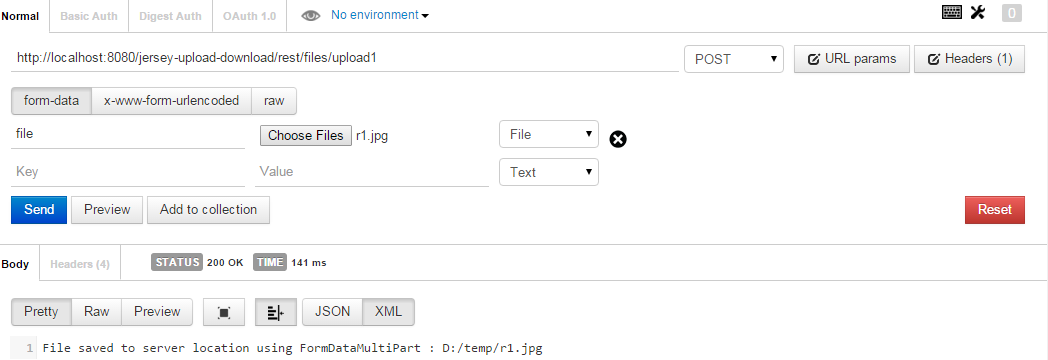
}

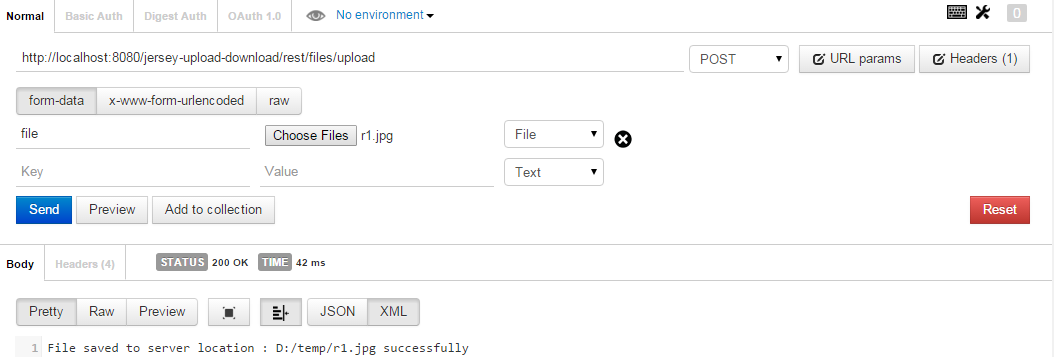
# **How to test**

WADL URL : [**http://localhost:8080/jersey-upload-download/rest/application.wadl**](http://localhost:8080/jersey-upload-download/rest/application.wadl)

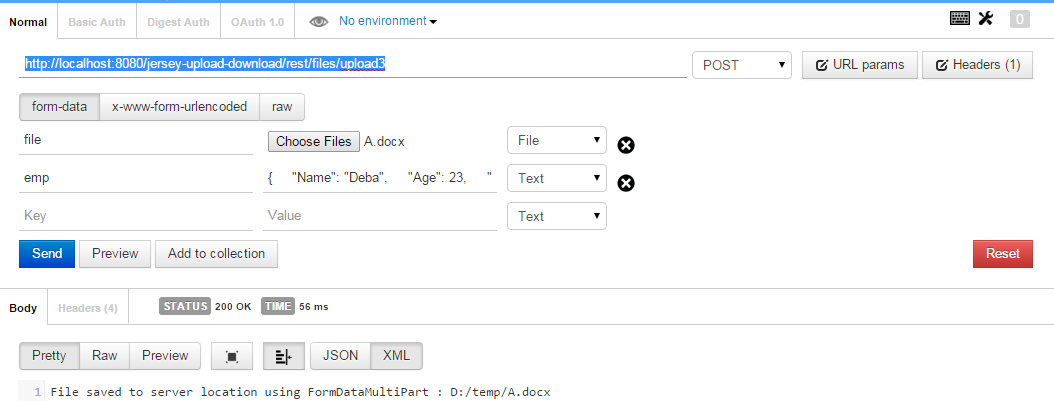
Use only chrome postman client to test the above functionality

To upload a file, see the screenshot

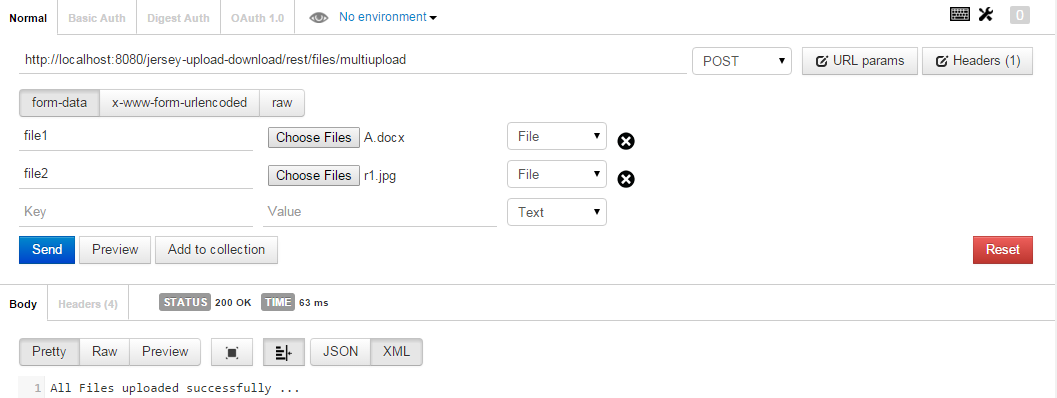




**To upload file with data as String**



**To upload multiple files**

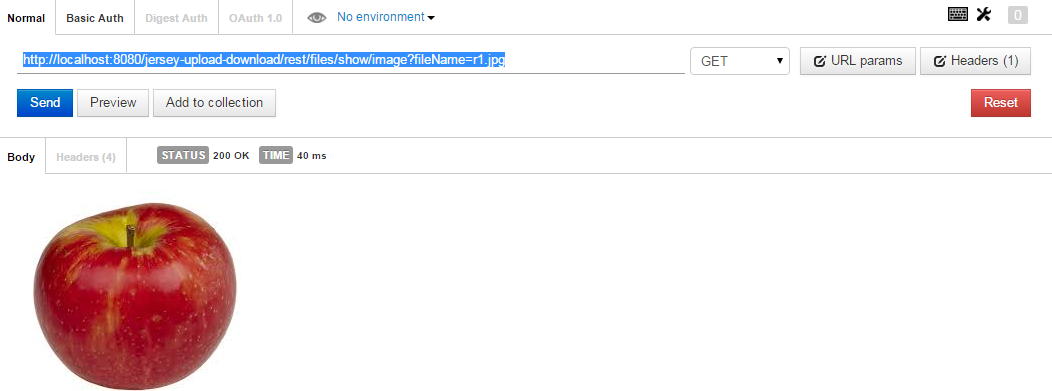


To download a file, use the following url and hit it in the browser. It does not need any rest client.

<http://localhost:8080/jersey-upload-download/rest/files/show/image?fileName=r1.jpg>

To show the image, use the following url and see the screenshot.

**http://localhost:8080/jersey-upload-download/rest/files/show/image?fileName=r1.jpg**



Important points

1. You can not use @PathParam for download a file. You have to use @QueryParam.