Web services are of 2 types:

* SOAP style and RESTstyle

API has no jar and communicate over HTTP over the n/w is webservice

Jax—rs for rest

We need authentication,REQUEST,method(Get,post,put)

SOAP-Simple Object Access Protocol(it is construced above HTTP)

REST-Representational State Transfer Protocol(it uses only HTTp)

REST:

* This is based on ROA-Resource oriented architecture
* Think webservices as resources
* Uniform HTTP codes REST is accessed
  + GET,POST,PUT,DELETE
* Put-modify the order placed
* Delete –to cancel the order

We can do REST using a jersey project.

1.Create a Maven jersey project(jersey webapp)

2.Run it in tomcat server

3.take the value from web.xml url pattern

4.give the url along with /urlpattern/pathname in java file

To create ur own REST

1. Create a class with getters and setter variable(Alient calss)

@XmlRootElement //to specify the XML root element

**public** **class** Alient

{

**private** String name;

**private** **int** points;

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **int** getPoints() {

**return** points;

}

**public** **void** setPoints(**int** points) {

**this**.points = points;

}

}

1. Create a AlienResource class to specify the Rest functionality

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

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

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

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

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

**public** **class** AlienResource

{

@GET //to specify teh get URL

@Produces(MediaType.***APPLICATION\_XML***) //to get XML

**public** Alient getAlien()

{

System.***out***.println("Alien called");

Alient a1=**new** Alient();

a1.setName("Uma");

a1.setPoints(60);

**return** a1;

}

}

1. Run on tomcat server the entire project.output is seen in belowurl

http://localhost:8080/demorest/webresources/Aliens

To run your own list of objects in the URL

**package** com.telusko.demorest;

**import** java.util.Arrays;

**import** java.util.List;

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

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

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

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

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

**public** **class** AlienResource

{

@GET //to specify teh get URL

@Produces(MediaType.***APPLICATION\_XML***) //to get XML

**public** List<Alient> getAlien()

{

System.***out***.println("Alien called");

Alient a1=**new** Alient();

a1.setName("Uma");

a1.setPoints(60);

Alient a2=**new** Alient();

a2.setName("Mahe");

a2.setPoints(70);

List<Alient> l1=Arrays.*asList*(a1,a2);

**return** l1;

}

}

1. Run on tomcat server the entire project.output is seen in belowurl

<http://localhost:8080/demorest/webresources/Aliens>

<alients>

<alient>

<name>Uma</name>

<points>60</points>

</alient>

<alient>

<name>Mahe</name>

<points>70</points>

</alient>

</alients>

To get from the repository

* Create a repository class

**package** com.telusko.demorest;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** AlienRepository {

List<Alient> alien;

**public** AlienRepository() {

alien=**new** ArrayList<Alient>();

Alient a1=**new** Alient();

a1.setId(1);

a1.setName("Uma");

a1.setPoints(60);

Alient a2=**new** Alient();

a2.setId(2);

a2.setName("Mahe");

a2.setPoints(70);

alien.add(a1);

alien.add(a2);

}

**public** List<Alient> getAlien() {

**return** alien;

}

**public** Alient getAlient(**int** id)

{

**for** (Alient alient : alien) {

**if**(alient.getId()==id)

**return** alient;

}

**return** **null**;

}

}

* Change in the resource class a follows

**package** com.telusko.demorest;

**import** java.util.Arrays;

**import** java.util.List;

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

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

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

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

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

**public** **class** AlienResource

{

AlienRepository ale=**new** AlienRepository();

@GET //to specify teh get URL

@Produces(MediaType.***APPLICATION\_XML***) //to get XML

**public** List<Alient> getAlien()

{

System.***out***.println("Alien called");

**return** ale.getAlien();

}

}

1. Run on tomcat server the entire project.output is seen in belowurl

<http://localhost:8080/demorest/webresources/Aliens>/alien1/alien

To enter new value and use post:

**package** com.telusko.demorest;

**import** java.util.Arrays;

**import** java.util.List;

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

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

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

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

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

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

**public** **class** AlienResource

{

AlienRepository ale=**new** AlienRepository();

@GET //to specify teh get URL

@Produces(MediaType.***APPLICATION\_XML***) //to get XML

**public** List<Alient> getAlien()

{

System.***out***.println("Alien called");

**return** ale.getAlien();

}

@POST

@Path("alien")

**public** Alient getali(Alient a1)

{

ale.create(a1);

**return** a1;

}

}

**package** com.telusko.demorest;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** AlienRepository {

List<Alient> alien;

**public** AlienRepository() {

alien=**new** ArrayList<Alient>();

Alient a1=**new** Alient();

a1.setId(1);

a1.setName("Uma");

a1.setPoints(60);

Alient a2=**new** Alient();

a2.setId(2);

a2.setName("Mahe");

a2.setPoints(70);

alien.add(a1);

alien.add(a2);

}

**public** List<Alient> getAlien() {

**return** alien;

}

**public** Alient getAlient(**int** id)

{

**for** (Alient alient : alien) {

**if**(alient.getId()==id)

**return** alient;

}

**return** **null**;

}

**public** **void** create(Alient a1) {

// **TODO** Auto-generated method stub

System.***out***.println(a1);

alien.add(a1);

}

}

URL to be used:

<http://localhost:8080/demorest/webresources/Aliens/alien>

To get a particular value:

URL:

http://localhost:8080/demorest/webresources/Aliens/alien/2

Alien Resource:

package com.telusko.demorest;

import java.util.Arrays;

import java.util.List;

import javax.websocket.server.PathParam;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

public class AlienResource

{

AlienRepository ale=new AlienRepository();

@GET //to specify teh get URL

@Produces(MediaType.APPLICATION\_XML) //to get XML

public List<Alient> getAlien()

{

System.out.println("Alien called");

return ale.getAlien();

}

@GET //to specify teh get URL

@Path("alien/{id}") //it will get the id-we an replace id with any value

@Produces(MediaType.APPLICATION\_XML) //to get XML

public Alient getAlient(@javax.ws.rs.PathParam("id") int id) //from path param teh value is passed

{

//System.out.println("hi");

return ale.getAlient(id);

}

@POST

@Path("alienval")

public Alient getali(Alient a1)

{

ale.create(a1);

return a1;

}

}

In Alien repository:

**public** Alient getAlient(**int** id)

{

**for** (Alient alient : alien) {

**if**(alient.getId()==id)

{

System.***out***.println(alient.getName());

**return** alient;

}

}

**return** **null**;

}

To get in json format:

Change the mediatype as application json

In pom.xml change as

<dependency>

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

<artifactId>jersey-media-moxy</artifactId>

</dependency>

<properties>

<jersey.version>2.26-b03</jersey.version>

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

</properties>

Alien Repository:

package com.telusko.demorest;

import java.util.Arrays;

import java.util.List;

import javax.websocket.server.PathParam;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

public class AlienResource

{

AlienRepository ale=new AlienRepository();

@GET //to specify teh get URL

@Produces(MediaType.APPLICATION\_XML) //to get XML

public List<Alient> getAlien()

{

System.out.println("Alien called");

return ale.getAlien();

}

@GET //to specify teh get URL

@Path("alien/{id}") //it will get the id-we an replace id with any value

@Produces(MediaType.APPLICATION\_JSON) //to get XML

public Alient getAlient(@javax.ws.rs.PathParam("id") int id) //from path param teh value is passed

{

//System.out.println("hi");

return ale.getAlient(id);

}

@POST

@Path("alienval")

public Alient getali(Alient a1)

{

ale.create(a1);

return a1;

}

}

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.telusko</groupId>

<artifactId>demorest</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>demorest</name>

<build>

<finalName>demorest</finalName>

<plugins>

<plugin>

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

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

<version>2.5.1</version>

<inherited>true</inherited>

<configuration>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

</plugins>

</build>

<dependencyManagement>

<dependencies>

<dependency>

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

<artifactId>jersey-bom</artifactId>

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

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<dependencies>

<dependency>

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

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

<!-- use the following artifactId if you don't need servlet 2.x compatibility -->

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

</dependency>

<dependency>

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

<artifactId>jersey-media-moxy</artifactId>

</dependency>

</dependencies>

<properties>

<jersey.version>2.26-b03</jersey.version>

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

</properties>

</project>

<http://localhost:8080/demorest/webapi/Aliens/alien/1>

To get in both xml and json

package com.telusko.demorest;

import java.util.Arrays;

import java.util.List;

import javax.websocket.server.PathParam;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

public class AlienResource

{

AlienRepository ale=new AlienRepository();

@GET //to specify teh get URL

@Produces(MediaType.APPLICATION\_XML) //to get XML

public List<Alient> getAlien()

{

System.out.println("Alien called");

return ale.getAlien();

}

@GET //to specify teh get URL

@Path("alien/{id}") //it will get the id-we an replace id with any value

@Produces({MediaType.APPLICATION\_JSON,MediaType.APPLICATION\_XML}) //to get XML

public Alient getAlient(@javax.ws.rs.PathParam("id") int id) //from path param teh value is passed

{

//System.out.println("hi");

return ale.getAlient(id);

}

@POST

@Path("alienval")

public Alient getali(Alient a1)

{

ale.create(a1);

return a1;

}}

To store in mysql and retieve it:we can use both xml and json

package com.telusko.demorest;

import java.util.Arrays;

import java.util.List;

import javax.websocket.server.PathParam;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import java.sql.\*;

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

public class AlienResource

{

AlienRepository ale=new AlienRepository();

@GET //to specify teh get URL

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

public List<Alient> getAlien()

{

System.out.println("Alien called");

return ale.getAlien();

}

@GET //to specify teh get URL

@Path("alien/{id}") //it will get the id-we an replace id with any value

@Produces({MediaType.APPLICATION\_JSON,MediaType.APPLICATION\_XML}) //to get XML

public Alient getAlient(@javax.ws.rs.PathParam("id") int id) //from path param teh value is passed

{

//System.out.println("hi");

return ale.getAlient(id);

}

@POST

@Path("alienval")

public Alient getali(Alient a1)

{

System.out.println("alien resource'");

ale.create(a1);

return a1;

}

}

Maven:

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.9-rc</version>

</dependency>

When we give in post @Consumes(MediaType.***APPLICATION\_XML***)

It consumes only xml format

To delete using post:

**package** com.telusko.demorest;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** java.util.List;

**import** com.sun.xml.internal.messaging.saaj.packaging.mime.util.QEncoderStream;

**public** **class** AlienRepository {

List<Alient> alien;

Connection con;

**public** AlienRepository() {

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

String URL="jdbc:mysql://localhost:3306/alien";

String Uname ="root";

String passwrd="root";

con=DriverManager.*getConnection*(URL,Uname,passwrd);

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**public** List<Alient> getAlien() {

List<Alient> li=**new** ArrayList<>();

String que="select \* from aliendetails";

**try** {

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(que);

**while**(rs.next())

{

System.***out***.println("inside alien repository get while");

Alient a=**new** Alient();

a.setId(rs.getInt(1));

a.setName(rs.getString(2));

a.setPoints(rs.getInt(3));

li.add(a);

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** li;

}

**public** Alient getAlient(**int** id)

{

Alient a=**new** Alient();

String que="select \* from aliendetails where id="+id;

**try** {

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(que);

**if**(rs.next())

{

a.setId(rs.getInt(1));

a.setName(rs.getString(2));

a.setPoints(rs.getInt(3));

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** a;

}

**public** **void** create(Alient a1) {

// **TODO** Auto-generated method stub

String qu="insert into aliendetails values(?,?,?)";

**try** {

PreparedStatement stmt=con.prepareStatement(qu);

stmt.setInt(1,a1.getId());

stmt.setString(2, a1.getName());

stmt.setInt(3, a1.getPoints());

stmt.executeUpdate();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

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

// **TODO** Auto-generated method stub

String quet="delete from aliendetails where id="+id;

**try** {

PreparedStatement pre=con.prepareStatement(quet);

//pre.setInt(1, id);

pre.executeUpdate();

System.***out***.println("delete");

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

package com.telusko.demorest;

import java.util.Arrays;

import java.util.List;

import javax.websocket.server.PathParam;

import javax.ws.rs.Consumes;

import javax.ws.rs.DELETE;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import java.sql.\*;

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

public class AlienResource

{

AlienRepository ale=new AlienRepository();

@GET //to specify teh get URL

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

public List<Alient> getAlien()

{

System.out.println("Alien called");

return ale.getAlien();

}

@GET //to specify teh get URL

@Path("alien/{id}") //it will get the id-we an replace id with any value

@Produces({MediaType.APPLICATION\_JSON,MediaType.APPLICATION\_XML}) //to get XML

public Alient getAlient(@javax.ws.rs.PathParam("id") int id) //from path param teh value is passed

{

//System.out.println("hi");

return ale.getAlient(id);

}

@POST

@Path("alienval")

@Consumes(MediaType.APPLICATION\_XML)

public Alient getali(Alient a1)

{

System.out.println("alien resource'");

ale.create(a1);

return a1;

}

@DELETE

@Path("alien/{id}")

public Alient killalien(@javax.ws.rs.PathParam("id") int id)

{

Alient a=ale.getAlient(id);

if(a.getId()!=0)

ale.delete(id);

return a;

}

}

Using Put we can update values:

**package** com.telusko.demorest;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** java.util.List;

**import** com.sun.xml.internal.messaging.saaj.packaging.mime.util.QEncoderStream;

**public** **class** AlienRepository {

List<Alient> alien;

Connection con;

**public** AlienRepository() {

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

String URL="jdbc:mysql://localhost:3306/alien";

String Uname ="root";

String passwrd="root";

con=DriverManager.*getConnection*(URL,Uname,passwrd);

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**public** List<Alient> getAlien() {

List<Alient> li=**new** ArrayList<>();

String que="select \* from aliendetails";

**try** {

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(que);

**while**(rs.next())

{

System.***out***.println("inside alien repository get while");

Alient a=**new** Alient();

a.setId(rs.getInt(1));

a.setName(rs.getString(2));

a.setPoints(rs.getInt(3));

li.add(a);

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** li;

}

**public** Alient getAlient(**int** id)

{

Alient a=**new** Alient();

String que="select \* from aliendetails where id="+id;

**try** {

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(que);

**if**(rs.next())

{

a.setId(rs.getInt(1));

a.setName(rs.getString(2));

a.setPoints(rs.getInt(3));

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** a;

}

**public** **void** create(Alient a1) {

// **TODO** Auto-generated method stub

String qu="insert into aliendetails values(?,?,?)";

**try** {

PreparedStatement stmt=con.prepareStatement(qu);

stmt.setInt(1,a1.getId());

stmt.setString(2, a1.getName());

stmt.setInt(3, a1.getPoints());

stmt.executeUpdate();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

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

// **TODO** Auto-generated method stub

String quet="delete from aliendetails where id="+id;

**try** {

PreparedStatement pre=con.prepareStatement(quet);

//pre.setInt(1, id);

pre.executeUpdate();

System.***out***.println("delete");

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**public** **void** update(Alient a1) {

// **TODO** Auto-generated method stub

String qu="update alien.aliendetails set name=? ,points=? where id=?";

**try**{

PreparedStatement stmt=con.prepareStatement(qu);

stmt.setString(1,a1.getName());

stmt.setInt(2, a1.getPoints());

stmt.setInt(3, a1.getId());

System.***out***.println(a1.getName());

stmt.executeUpdate();

System.***out***.println("update");

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

package com.telusko.demorest;

import java.util.Arrays;

import java.util.List;

import javax.websocket.server.PathParam;

import javax.ws.rs.Consumes;

import javax.ws.rs.DELETE;

import javax.ws.rs.GET;

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.MediaType;

import java.sql.\*;

@Path("Aliens") //to specify in the URL path http://localhost:8080/demorest/webresources/Aliens

public class AlienResource

{

AlienRepository ale=new AlienRepository();

@GET //to specify teh get URL

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

public List<Alient> getAlien()

{

System.out.println("Alien called");

return ale.getAlien();

}

@GET //to specify teh get URL

@Path("alien/{id}") //it will get the id-we an replace id with any value

@Produces({MediaType.APPLICATION\_JSON,MediaType.APPLICATION\_XML}) //to get XML

public Alient getAlient(@javax.ws.rs.PathParam("id") int id) //from path param teh value is passed

{

//System.out.println("hi");

return ale.getAlient(id);

}

@POST

@Path("alienval")

public Alient getali(Alient a1)

{

System.out.println("alien resource'");

ale.create(a1);

return a1;

}

@DELETE

@Path("alien/{id}")

public Alient killalien(@javax.ws.rs.PathParam("id") int id)

{

Alient a=ale.getAlient(id);

if(a.getId()!=0)

ale.delete(id);

return a;

}

@PUT

@Path("alienval")

public Alient updateali(Alient a1)

{

System.out.println("alien resource'");

ale.update(a1);

return a1;

}

}

USING SPRING:

**package** com.telusko;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController //to show this class has rest

**public** **class** Alienresource {

@RequestMapping("aliens") //for mPPING

**public** List<Alient> getAlien()

{

List<Alient> al=**new** ArrayList<>();

Alient a1=**new** Alient();

a1.setId(10);

a1.setName("abc");

a1.setPoints(10);

Alient a2=**new** Alient();

a2.setId(10);

a2.setName("abc");

a2.setPoints(10);

al.add(a2);

**return** al;

}

}

C-create-post

R-Retrieve-Get

U-Update-Put

D-Delete-Delete

HTTP request:

* URI
* Headers
* Payload(JSON/XML)

HTTP response:

* Status—200 ok
* Response payload
* String message-Successful or not

To create an entity:POST call is used(headers)

* Get a entity:Get call(URI,Path parameter(define by /) or query parameter(define by?)
* ,headers)

URI:

<http://appi.com/service/accont/1>

<http://appi.com> -server name

service-place holder,account-class name,1-id

404- unavailable

Update an account(PUT)

Create+ update

Status response:

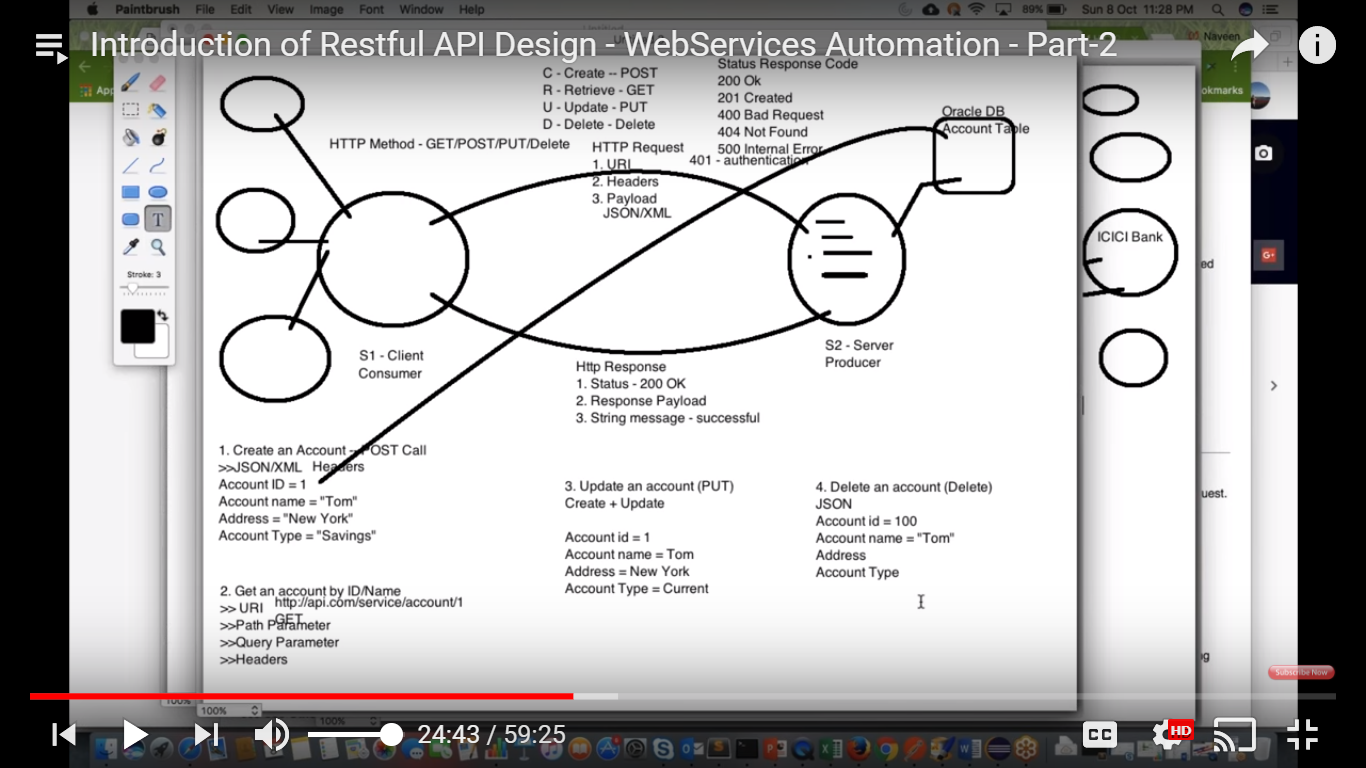
200-ok

201 –created successfully for post

400 -bad request

404 –not found

500 –Internal server error



URL (End point URL)+Service URL(service)=URI

Get response:

Evaluate status,json,headers

spring-boot-starter

