Example

Let's update Example created in [RESTful Web Services - First Application](https://www.tutorialspoint.com/restful/restful_first_application.htm)tutorial to create a Web service which can perform CRUD (Create, Read, Update, Delete) operations. For simplicity, we've used a file I/O to replace Database operations.

Let's update **User.java**, **UserDao.java**, and **UserService.java** files under the com.tutorialspoint package.

*User.java*

package com.tutorialspoint;

import java.io.Serializable;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "user")

public class User implements Serializable {

private static final long serialVersionUID = 1L;

private int id;

private String name;

private String profession;

public User(){}

public User(int id, String name, String profession){

this.id = id;

this.name = name;

this.profession = profession;

}

public int getId() {

return id;

}

@XmlElement

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

@XmlElement

public void setName(String name) {

this.name = name;

}

public String getProfession() {

return profession;

}

@XmlElement

public void setProfession(String profession) {

this.profession = profession;

}

@Override

public boolean equals(Object object){

if(object == null){

return false;

}else if(!(object instanceof User)){

return false;

}else {

User user = (User)object;

if(id == user.getId()

&& name.equals(user.getName())

&& profession.equals(user.getProfession())

){

return true;

}

}

return false;

}

}

*UserDao.java*

package com.tutorialspoint;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.ArrayList;

import java.util.List;

public class UserDao {

public List<User> getAllUsers(){

List<User> userList = null;

try {

File file = new File("Users.dat");

if (!file.exists()) {

User user = new User(1, "Mahesh", "Teacher");

userList = new ArrayList<User>();

userList.add(user);

saveUserList(userList);

}

else{

FileInputStream fis = new FileInputStream(file);

ObjectInputStream ois = new ObjectInputStream(fis);

userList = (List<User>) ois.readObject();

ois.close();

}

} catch (IOException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

return userList;

}

public User getUser(int id){

List<User> users = getAllUsers();

for(User user: users){

if(user.getId() == id){

return user;

}

}

return null;

}

public int addUser(User pUser){

List<User> userList = getAllUsers();

boolean userExists = false;

for(User user: userList){

if(user.getId() == pUser.getId()){

userExists = true;

break;

}

}

if(!userExists){

userList.add(pUser);

saveUserList(userList);

return 1;

}

return 0;

}

public int updateUser(User pUser){

List<User> userList = getAllUsers();

for(User user: userList){

if(user.getId() == pUser.getId()){

int index = userList.indexOf(user);

userList.set(index, pUser);

saveUserList(userList);

return 1;

}

}

return 0;

}

public int deleteUser(int id){

List<User> userList = getAllUsers();

for(User user: userList){

if(user.getId() == id){

int index = userList.indexOf(user);

userList.remove(index);

saveUserList(userList);

return 1;

}

}

return 0;

}

private void saveUserList(List<User> userList){

try {

File file = new File("Users.dat");

FileOutputStream fos;

fos = new FileOutputStream(file);

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(userList);

oos.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

*UserService.java*

package com.tutorialspoint;

import java.io.IOException;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.Consumes;

import javax.ws.rs.DELETE;

import javax.ws.rs.FormParam;

import javax.ws.rs.GET;

import javax.ws.rs.OPTIONS;

import javax.ws.rs.POST;

import javax.ws.rs.PUT;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.Produces;

import javax.ws.rs.core.Context;

import javax.ws.rs.core.MediaType;

@Path("/UserService")

public class UserService {

UserDao userDao = new UserDao();

private static final String SUCCESS\_RESULT="<result>success</result>";

private static final String FAILURE\_RESULT="<result>failure</result>";

@GET

@Path("/users")

@Produces(MediaType.APPLICATION\_XML)

public List<User> getUsers(){

return userDao.getAllUsers();

}

@GET

@Path("/users/{userid}")

@Produces(MediaType.APPLICATION\_XML)

public User getUser(@PathParam("userid") int userid){

return userDao.getUser(userid);

}

@PUT

@Path("/users")

@Produces(MediaType.APPLICATION\_XML)

@Consumes(MediaType.APPLICATION\_FORM\_URLENCODED)

public String createUser(@FormParam("id") int id,

@FormParam("name") String name,

@FormParam("profession") String profession,

@Context HttpServletResponse servletResponse) throws IOException{

User user = new User(id, name, profession);

int result = userDao.addUser(user);

if(result == 1){

return SUCCESS\_RESULT;

}

return FAILURE\_RESULT;

}

@POST

@Path("/users")

@Produces(MediaType.APPLICATION\_XML)

@Consumes(MediaType.APPLICATION\_FORM\_URLENCODED)

public String updateUser(@FormParam("id") int id,

@FormParam("name") String name,

@FormParam("profession") String profession,

@Context HttpServletResponse servletResponse) throws IOException{

User user = new User(id, name, profession);

int result = userDao.updateUser(user);

if(result == 1){

return SUCCESS\_RESULT;

}

return FAILURE\_RESULT;

}

@DELETE

@Path("/users/{userid}")

@Produces(MediaType.APPLICATION\_XML)

public String deleteUser(@PathParam("userid") int userid){

int result = userDao.deleteUser(userid);

if(result == 1){

return SUCCESS\_RESULT;

}

return FAILURE\_RESULT;

}

@OPTIONS

@Path("/users")

@Produces(MediaType.APPLICATION\_XML)

public String getSupportedOperations(){

return "<operations>GET, PUT, POST, DELETE</operations>";

}

}

Now using Eclipse, export your application as a war file and deploy the same in tomcat. To create WAR file using eclipse, follow the option **File -> export -> Web > War File** and finally select project UserManagement and destination folder. To deploy war file in Tomcat, place the UserManagement.war in **Tomcat Installation Directory** > webapps directory and start the Tomcat.

Testing the Web Service

Jersey provides APIs to create a Web Service Client to test web services. We've created a sample test class **WebServiceTester.java** under the com.tutorialspoint package in the same project.

*WebServiceTester.java*

package com.tutorialspoint;

import java.util.List;

import javax.ws.rs.client.Client;

import javax.ws.rs.client.ClientBuilder;

import javax.ws.rs.client.Entity;

import javax.ws.rs.core.Form;

import javax.ws.rs.core.GenericType;

import javax.ws.rs.core.MediaType;

public class WebServiceTester {

private Client client;

private String REST\_SERVICE\_URL = "http://localhost:8080/UserManagement/rest/UserService/users";

private static final String SUCCESS\_RESULT="<result>success</result>";

private static final String PASS = "pass";

private static final String FAIL = "fail";

private void init(){

this.client = ClientBuilder.newClient();

}

public static void main(String[] args){

WebServiceTester tester = new WebServiceTester();

//initialize the tester

tester.init();

//test get all users Web Service Method

tester.testGetAllUsers();

//test get user Web Service Method

tester.testGetUser();

//test update user Web Service Method

tester.testUpdateUser();

//test add user Web Service Method

tester.testAddUser();

//test delete user Web Service Method

tester.testDeleteUser();

}

//Test: Get list of all users

//Test: Check if list is not empty

private void testGetAllUsers(){

GenericType<List<User>> list = new GenericType<List<User>>() {};

List<User> users = client

.target(REST\_SERVICE\_URL)

.request(MediaType.APPLICATION\_XML)

.get(list);

String result = PASS;

if(users.isEmpty()){

result = FAIL;

}

System.out.println("Test case name: testGetAllUsers, Result: " + result );

}

//Test: Get User of id 1

//Test: Check if user is same as sample user

private void testGetUser(){

User sampleUser = new User();

sampleUser.setId(1);

User user = client

.target(REST\_SERVICE\_URL)

.path("/{userid}")

.resolveTemplate("userid", 1)

.request(MediaType.APPLICATION\_XML)

.get(User.class);

String result = FAIL;

if(sampleUser != null && sampleUser.getId() == user.getId()){

result = PASS;

}

System.out.println("Test case name: testGetUser, Result: " + result );

}

//Test: Update User of id 1

//Test: Check if result is success XML.

private void testUpdateUser(){

Form form = new Form();

form.param("id", "1");

form.param("name", "suresh");

form.param("profession", "clerk");

String callResult = client

.target(REST\_SERVICE\_URL)

.request(MediaType.APPLICATION\_XML)

.post(Entity.entity(form,

MediaType.APPLICATION\_FORM\_URLENCODED\_TYPE),

String.class);

String result = PASS;

if(!SUCCESS\_RESULT.equals(callResult)){

result = FAIL;

}

System.out.println("Test case name: testUpdateUser, Result: " + result );

}

//Test: Add User of id 2

//Test: Check if result is success XML.

private void testAddUser(){

Form form = new Form();

form.param("id", "2");

form.param("name", "naresh");

form.param("profession", "clerk");

String callResult = client

.target(REST\_SERVICE\_URL)

.request(MediaType.APPLICATION\_XML)

.put(Entity.entity(form,

MediaType.APPLICATION\_FORM\_URLENCODED\_TYPE),

String.class);

String result = PASS;

if(!SUCCESS\_RESULT.equals(callResult)){

result = FAIL;

}

System.out.println("Test case name: testAddUser, Result: " + result );

}

//Test: Delete User of id 2

//Test: Check if result is success XML.

private void testDeleteUser(){

String callResult = client

.target(REST\_SERVICE\_URL)

.path("/{userid}")

.resolveTemplate("userid", 2)

.request(MediaType.APPLICATION\_XML)

.delete(String.class);

String result = PASS;

if(!SUCCESS\_RESULT.equals(callResult)){

result = FAIL;

}

System.out.println("Test case name: testDeleteUser, Result: " + result );

}

}

Now run the tester using Eclipse. Right click on the file, and follow the option **Run as -> Java Application**. You'll see the following result in Eclipse console:

Test case name: testGetAllUsers, Result: pass

Test case name: testGetUser, Result: pass

Test case name: testUpdateUser, Result: pass

Test case name: testAddUser, Result: pass

Test case name: testDeleteUser, Result: pass