Assignment 5

|  |  |
| --- | --- |
| **Name** | Abhishek Dilip Agashe |
| **Roll No** | 43103 |
| **Batch** | P9 |
| **Problem Statement** | To create a simple web service and write any distributed application to  consume the web service. |

**Code**

**1. SOAP**

*/\**

*\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license*

*\* Click nbfs://nbhost/SystemFileSystem/Templates/WebServices/EjbWebService.java to edit this template*

*\*/*

**package** org.me.stringmanip;

**import** javax.jws.WebService;

**import** javax.jws.WebMethod;

**import** javax.jws.WebParam;

**import** javax.ejb.Stateless;

*/\*\**

*\**

*\* @author tanay*

*\*/*

@WebService(serviceName = "Stringop")

@Stateless()

**public** **class** Stringop {

*/\*\**

*\* Web service operation*

*\*/*

@WebMethod(operationName = "concat")

**public** String concat(@WebParam(name = "a") String a, @WebParam(name = "b") String b) {

*//TODO write your implementation code here:*

String res = a.concat(b);

**return** res;

}

*/\*\**

*\* Web service operation*

*\*/*

@WebMethod(operationName = "length")

**public** **int** length(@WebParam(name = "a") String a) {

*//TODO write your implementation code here:*

**return** a.length();

}

*/\*\**

*\* Web service operation*

*\*/*

@WebMethod(operationName = "upper")

**public** String upper(@WebParam(name = "a") String a) {

*//TODO write your implementation code here:*

**return** a.toUpperCase();

}

*/\*\**

*\* Web service operation*

*\*/*

@WebMethod(operationName = "lower")

**public** String lower(@WebParam(name = "a") String a) {

*//TODO write your implementation code here:*

**return** a.toLowerCase();

}

*/\*\**

*\* Web service operation*

*\*/*

@WebMethod(operationName = "compare")

**public** String compare(@WebParam(name = "a") String a, @WebParam(name = "b") String b) {

*//TODO write your implementation code here:*

**int** comp = a.compareTo(b);

String msg = "";

**if**(comp==0)

{

msg = "Both are equal";

}

**else** if(comp<0)

{

msg = "Second string is lexicographically greater";

}

**else**

{

msg = "First string is lexicographically greater";

}

**return** msg;

}

}

**2. REST**

*/\**

*\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license*

*\* Click nbfs://nbhost/SystemFileSystem/Templates/WebServices/GenericResource.java to edit this template*

*\*/*

**package** org.me.stringopsrest;

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

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

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

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

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

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

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

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

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

*/\*\**

*\* REST Web Service*

*\**

*\* @author tanay*

*\*/*

@Path("generic")

**public** **class** GenericResource {

@Context

**private** UriInfo context;

*/\*\**

*\* Creates a new instance of GenericResource*

*\*/*

**public** GenericResource() {

}

*/\*\**

*\* Retrieves representation of an instance of org.me.stringopsrest.GenericResource*

*\* @return an instance of java.lang.String*

*\*/*

@GET

@Produces(MediaType.APPLICATION\_XML)

**public** String getXml() {

*//TODO return proper representation object*

**return** "<hi>Hello</hi>";

*//new UnsupportedOperationException();*

}

*/\*\**

*\* PUT method for updating or creating an instance of GenericResource*

*\* @param content representation for the resource*

*\*/*

@PUT

@Consumes(MediaType.APPLICATION\_XML)

**public** **void** putXml(String content) {

}

@GET

@Path("/concat/{a},{b}")

**public** String concat(@PathParam("a") String a,@PathParam("b") String b)

{

String res = a.concat(b);

**return** res;

}

@GET

@Path("/compare/{a},{b}")

**public** String compare(@PathParam("a") String a,@PathParam("b") String b)

{

**int** comp = a.compareTo(b);

String msg = "";

**if**(comp==0)

{

msg = "Both are equal";

}

**else** if(comp<0)

{

msg = "Second string is lexicographically greater";

}

**else**

{

msg = "First string is lexicographically greater";

}

**return** msg;

}

@GET

@Path("/length/{a}")

**public** **int** length(@PathParam("a") String a)

{

**return** a.length();

}

@GET

@Path("/upper/{a}")

**public** String upper(@PathParam("a") String a) {

*//TODO write your implementation code here:*

**return** a.toUpperCase();

}

@GET

@Path("/lower/{a}")

**public** String lower(@PathParam("a") String a) {

*//TODO write your implementation code here:*

**return** a.toUpperCase();

}

}

**Output**

























