

Arquitectura e Integración de Aplicaciones Empresariales

Onceava Sesión Despliegue y Consumo de Servicios



Dra. Maricela Bravo
Cubículo H-287-B
mari_clau_18@hotmail.com

Java API for XML Web Services

- ▶ JAX-WS (Java API for XML Web Services) es una interfaz de programación de aplicaciones (API) de Java basada en XML para la creación de servicios web.
- ▶ Es parte de la plataforma Java EE.
- ▶ JAX-WS utiliza anotaciones para simplificar el desarrollo y despliegue de los clientes y puntos finales de servicios web.

Creación de un servicio web sobre una aplicación web en NetBeans

Nuevo Web Service

Pasos

1. Elige el tipo de archivo
2. **Nombre y ubicación**

Nombre y ubicación

Nombre de Clase:

Proyecto:

Ubicación:

Paquete:

Crear Archivo:

Create Web Service from Scratch

Create Web Service from Existing Session Bean

Enterprise Bean:

Implement Web Service as Stateless Session Bean

Proporcionar un nombre de paquete.

Nuevo Web Service

Pasos

1. Elige el tipo de archivo
2. **Nombre y ubicación**

Nombre y ubicación

Nombre de Clase:

Proyecto:

Ubicación:

Paquete:

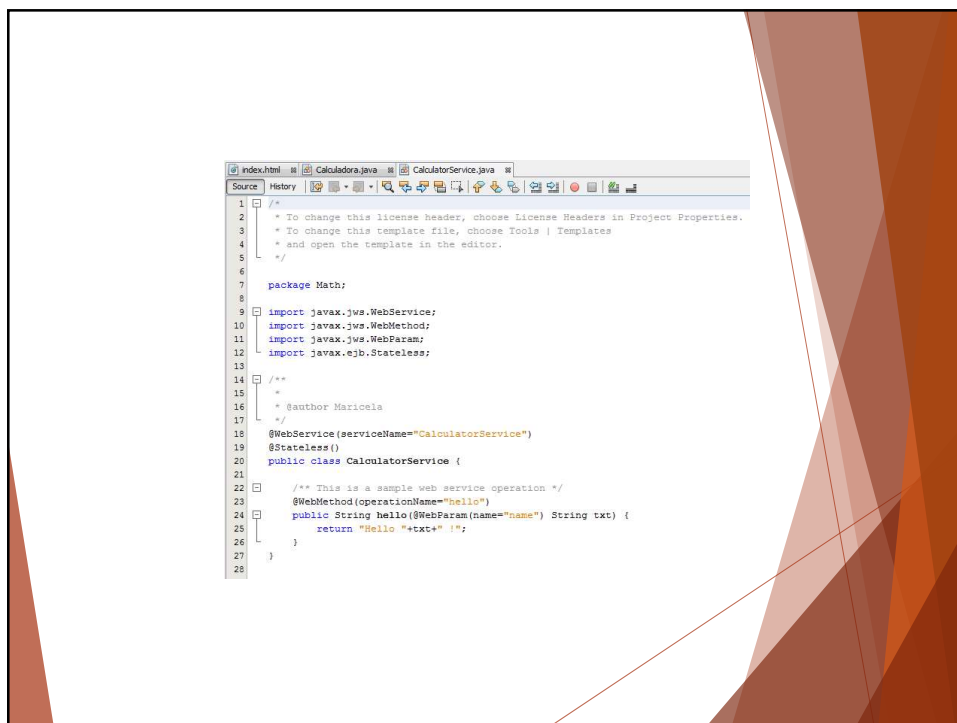
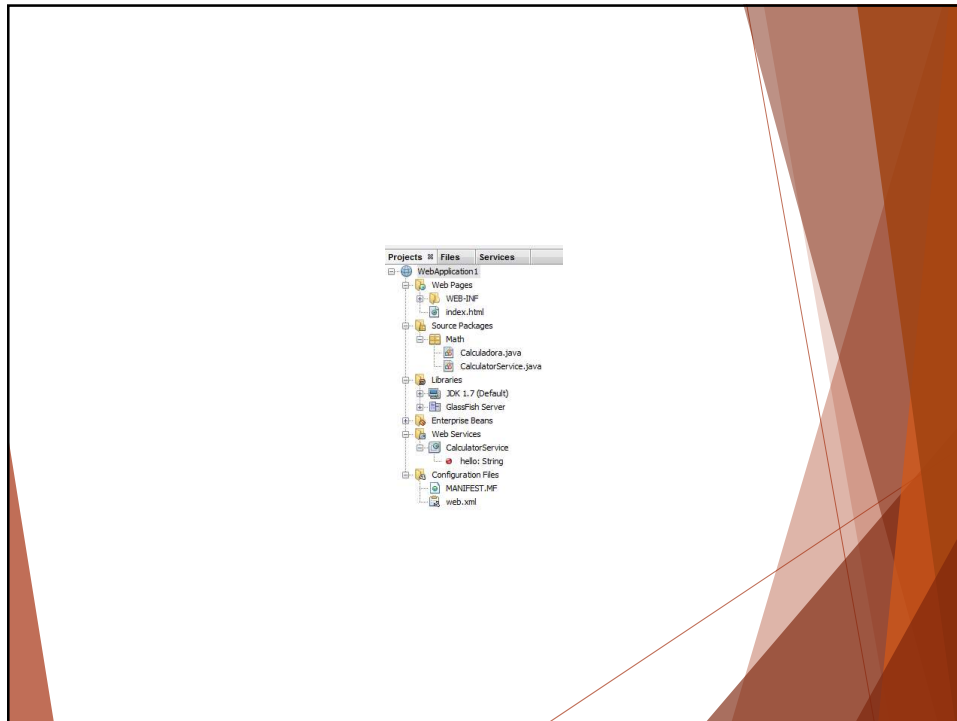
Crear Archivo:

Create Web Service from Scratch

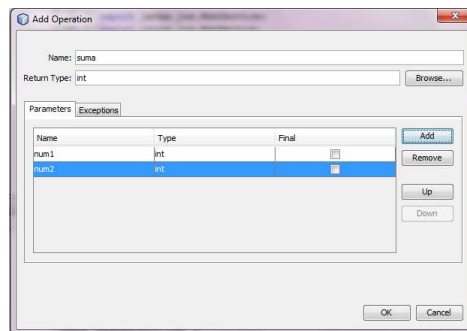
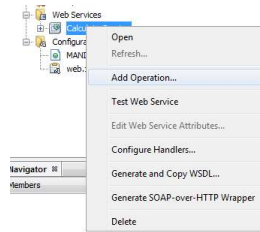
Create Web Service from Existing Session Bean

Enterprise Bean:

Implement Web Service as Stateless Session Bean



Agregar una operación



```
Source History
13
14 /**
15  *
16  * @author Maricela
17  */
18 @WebService(serviceName="CalculatorService")
19 @Stateless()
20 public class CalculatorService {
21
22     /** This is a sample web service operation */
23     @WebMethod(operationName="hello")
24     public String hello(@WebParam(name="name") String txt) {
25         return "Hello "+txt+" !";
26     }
27
28     /**
29      * Web service operation
30      */
31     @WebMethod(operationName = "suma")
32     public int suma(@WebParam(name = "num1") int num1, @WebParam(name = "num2") int num2) {
33         //TODO write your implementation code here:
34         return 0;
35     }
36 }
```

Deploy exitoso de la aplicación Web

```
Salida
Apache Tomcat (1) Log x Apache Tomcat (1) x Apache Tomcat Log x Apache Tomcat x WebApplication2 (run-deploy) x
Copying 1 file to C:\Users\User\Documents\NetBeansProjects\WebApplication2\build\web\WEB-INF\lib
Copying 1 file to C:\Users\User\Documents\NetBeansProjects\WebApplication2\build\web\WEB-INF\lib
lib\resources\resources
Created dir: C:\Users\User\Documents\NetBeansProjects\WebApplication2\build\empty
Created dir: C:\Users\User\Documents\NetBeansProjects\WebApplication2\build\generated-sources\ap-source-output
compile:
compile -> java
In-place deployment at C:\Users\User\Documents\NetBeansProjects\WebApplication2\build\web
deploy-config=file3AN2FC4AN2FUser42FAppData42FLocal42FTemp42FContent222504197859392286.xml;path=/WebApplication2
OK - Desplegada aplicación en trayectoria de contexto /WebApplication2
Start is in progress...
startPath=/WebApplication2
OK - Arrancada aplicación en trayectoria de contexto /WebApplication2
run-deploy:
GENERACIÓN CORRECTA (total time: 6 seconds)
```

Deploy del Web Service

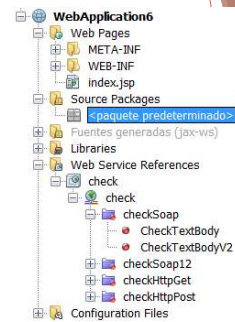


Desarrollo de clientes de servicios web

Pasos

1. Crear una aplicación Web
2. Crear un nuevo Cliente de Servicio Web
3. Especificar el URL del WSDL siguiente

<http://wsf.cdyne.com/SpellChecker/check.asmx?wsdl>



Desarrollo del cliente

- ▶ El cliente consiste de un JSP que le permite al usuario introducir el texto y después se llama a un servlet que invoca el servicio y genera la respuesta.
- ▶ En la carpeta de Web Pages de la aplicación modifica el index.jsp

Codificación del JSP

```

<%%page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<form name="Test" method="post" action="SpellCheckServlet">
<p>Enter the text you want to check:</p>
<p><textarea rows="7" name="TextAreal" cols="40" ID="Textareal"></textarea></p>
<p>
<input type="submit" value="Spell Check" name="spellcheckbutton">
</form>
</body>
</html>

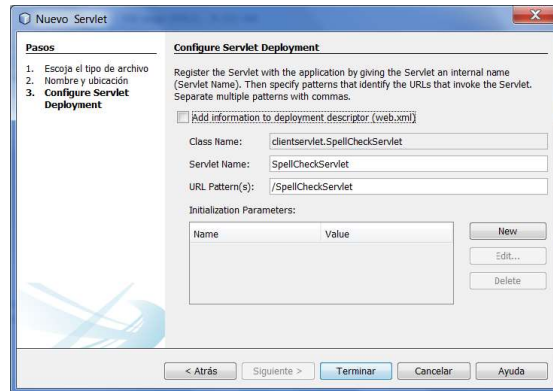
```

Creación y codificación del servlet

The screenshot shows the 'Nuevo Servlet' dialog box with the following configuration:

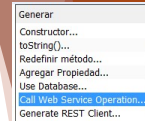
- Pasos:**
 - Escoja el tipo de archivo
 - Nombre y ubicación**
 - Configure Servlet Deployment
- Nombre y ubicación:**
 - Nombre de Clase: SpellCheckServlet
 - Proyecto: WebApplication6
 - Ubicación: Source Packages
 - Paquete: clientservlet
 - Crear Archivo: Projects\WebApplication6\src\java\clientservlet\SpellCheckServlet.java
- Botones:** < Atrás, Siguiete >, Terminar, Cancelar, Ayuda

Configuración del servlet

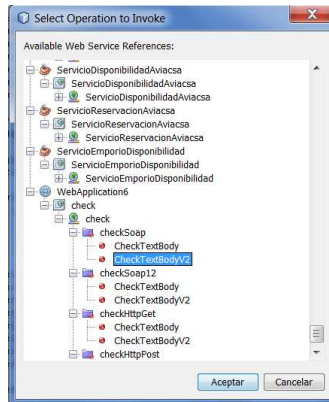


```
@WebServlet(name = "SpellCheckServlet", urlPatterns = {"/SpellCheckServlet"})
public class SpellCheckServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            /*
             * TODO output your page here. You may use following sample code.
             */
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet SpellCheckServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet SpellCheckServlet at " + request.getContextPath() +
" </h1>");
            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
}
```



Seleccionar la operación CheckTextBodyV2



Insertar el siguiente código en el ProcessRequest Method

```
//Get the TextArea from the JSP page
String TextArea1 = request.getParameter("TextArea1");

//Initialize WS operation arguments
java.lang.String bodyText = TextArea1;

//Process result
com.cdyne.ws.DocumentSummary doc = checkTextBodyV2(bodyText);
String allcontent = doc.getBody();

//From the retrieved document summary,
//identify the number of wrongly spelled words:
int no_of_mistakes = doc.getMisspelledWordCount();

//From the retrieved document summary,
//identify the array of wrongly spelled words:
List allwrongwords = doc.getMisspelledWord();
```

```

out.println("<html>");
out.println("<head>");
out.println("<title>Spell Checker Report</title>");
out.println("</head>");
out.println("<body>");
out.println("<h2><font color=red>Spell Checker Report</font></h2>");
out.println("<hr><b>Your text:</b> \'" + allcontent + "\' " + "<p>");

/*For every array of wrong words (one array per wrong word), identify the wrong word,
the number of suggestions, and the array of suggestions. Then display the wrong word and
the number of suggestions and then, for the array of suggestions belonging to the current
wrong word, display each suggestion: */

for (int i = 0; i < allwrongwords.size(); i++) {
    String onewrongword = ((Words) allwrongwords.get(i)).getWord();
    int onewordsuggestioncount = ((Words) allwrongwords.get(i)).getSuggestionCount();
    List allsuggestions = ((Words) allwrongwords.get(i)).getSuggestions();
    out.println("<hr><p><b>Wrong word:</b><font color=red>" + onewrongword +
"</font>");
    out.println("<p><b>" + onewordsuggestioncount + " suggestions:</b><br>");
    for (int k = 0; k < allsuggestions.size(); k++) {
        String onesuggestion = (String) allsuggestions.get(k);
        out.println(onesuggestion);
    }
}

```

```

//Display a line after each array of wrong words:
out.println("<hr>");

//Summarize by providing the number of errors and display them:
out.println("<font color=red><b>Summary:</b> " + no_of_mistakes + "
mistakes (");
for (int i = 0; i < allwrongwords.size(); i++) {
    String onewrongword = ((Words) allwrongwords.get(i)).getWord();
    out.println(onewrongword);
}

out.println(".");
out.println("</font>");
out.println("</body>");
out.println("</html>");

```

Corregir los imports



Ejecutar el proyecto

Enter the text you want to check:

the world is a beotiful place

Spell Check

Spell Checker Report

Your text: "the world is a beotiful place"

Wrong word: world

25 suggestions:

world wield wold would wild weid worlds warlord willed worldly welds wolds wilds veid would

Wrong word: beotiful

21 suggestions:

beautiful beautifully botfly beautifiler bedevil beautifilest bedeviled beautifulness butterfly bedevil butterfly's butterflying bedfellow's

Summary: 2 mistakes (world beotiful).

Ejercicio de repaso

- ▶ Crear un cliente de servicio Web usando la siguiente referencia
- ▶ <http://www.webservices.net/CurrencyConvertor.asmx?wsdl>
- ▶ Crear un JSP para la lectura de parámetros y un servlet para la invocación del servicio, y la generación del response.



Tarea

- ▶ Crear tres clientes (con JSPs o servlets) que invoquen a tres servicios Web públicos.