

Servicios Web

Índice

| | |
|---|----|
| 1 Introducción..... | 2 |
| 2 Creación de servicios web..... | 2 |
| 3 Acceso a servicios web externos..... | 7 |
| 4 Cliente para los servicios web (*)..... | 11 |
| 5 Resumen..... | 12 |

1. Introducción

En esta sesión de integración vamos a ofrecer una serie de funcionalidades de la aplicación como servicios web. Concretamente expondremos como servicio web las operaciones implementadas mediante EJBs. Además, utilizaremos servicios web externos para obtener información extendida sobre los libros de nuestra biblioteca.

Nota:

En esta sesión añadiremos controles Beehive al proyecto, por lo que la aplicación deberá ser desplegada en un dominio que cuente con las librerías necesarias (preparado para BEA Workshop).

2. Creación de servicios web

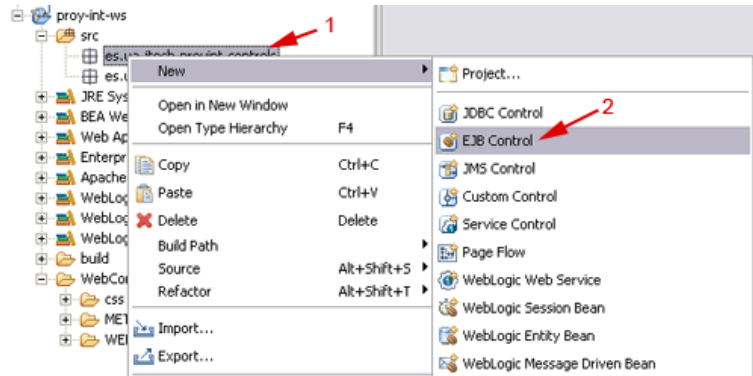
Lo primero que haremos será crear un servicio web de la biblioteca que permita realizar las siguientes operaciones:

- `LibroTO[] listadoDisponibles()`
- `OperacionTO[] listadoPrestamos()`
- `OperacionTO[] listadoReservas()`
- `OperacionTO[] listadoTodosLibros()`
- `String realizaReserva(String login, String password, String socio, String isbn)`

Para añadir este servicio web a la aplicación seguiremos los siguientes pasos:

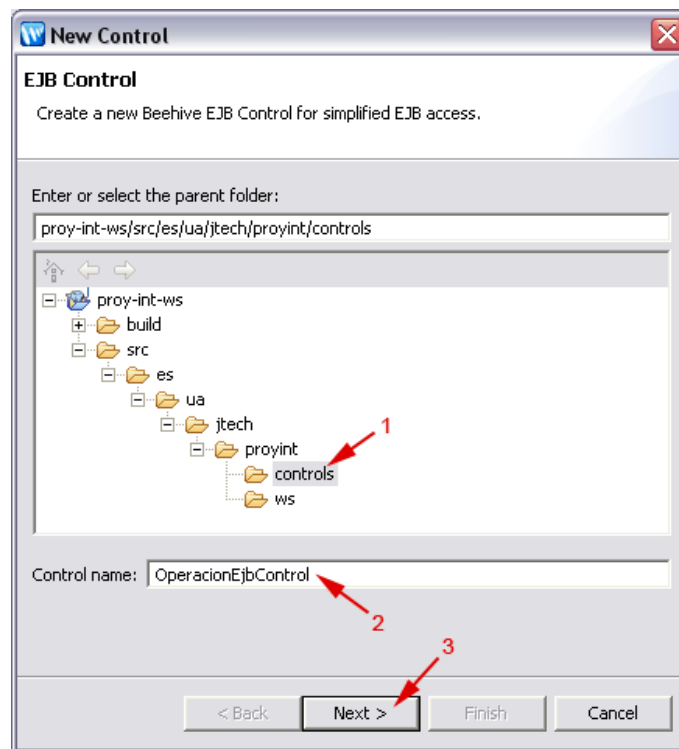
1. Creamos, dentro del proyecto EAR de la aplicación, un proyecto de servicios web (*Web Services Project*). Este proyecto se llamará `proj-int-ws`, y lo crearemos con las opciones por defecto.
2. Dentro de los fuentes del proyecto creamos dos paquetes:
`es.ua.jtech.projint.ws`, donde crearemos la clase que implementará nuestro servicio web, y `es.ua.jtech.projint.controls`, donde crearemos los controles Beehive para acceder a los componentes que necesitemos (EJBs y servicios web externos en nuestro caso).
3. Todas estas operaciones están definidas en el EJB implementado en sesiones anteriores. Para ofrecer las operaciones del EJB desde nuestro servicio web crearemos un *EJB Control* siguiendo los siguientes pasos:
 - Pulsando con el botón derecho sobre el paquete

es.ua.jtech.proyint.controls seleccionamos *New > EJB Control*.



Creación de un EJB Control. Paso 1.

- A la clase que implementará el control le damos como nombre *OperacionEjbControl* y pulsamos el botón *Next*.



Creación de un EJB Control. Paso 2.

- En la siguiente pantalla pulsamos sobre el botón *Browse Application EJBs...* para buscar entre los EJBs definidos dentro de nuestra aplicación (EAR).

New Control

EJB Control

Create a new Beehive EJB Control for simplified EJB access.

This EJB control finds the EJB with this JNDI name or EJB link

☒ JNDI name:

☐ EJB link:

This EJB control uses these interfaces

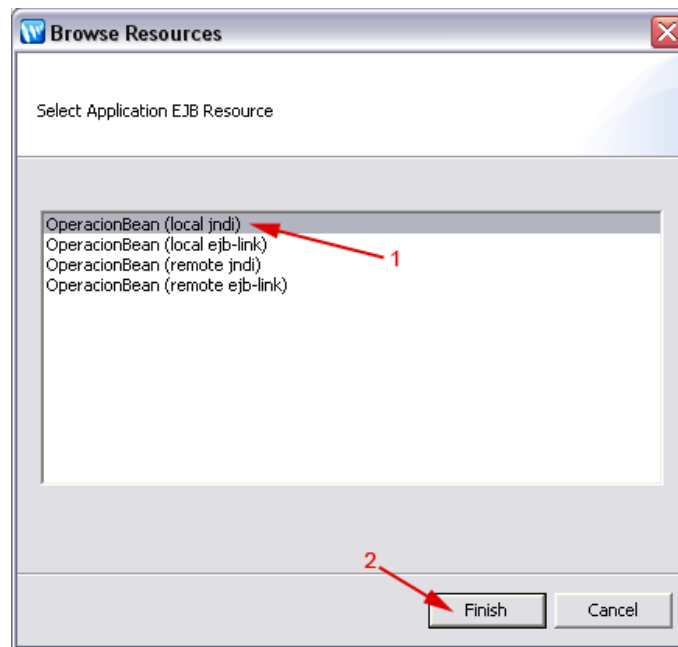
Home interface:

Business interface:

This EJB control is a ☐ Session control ☒ Entity control

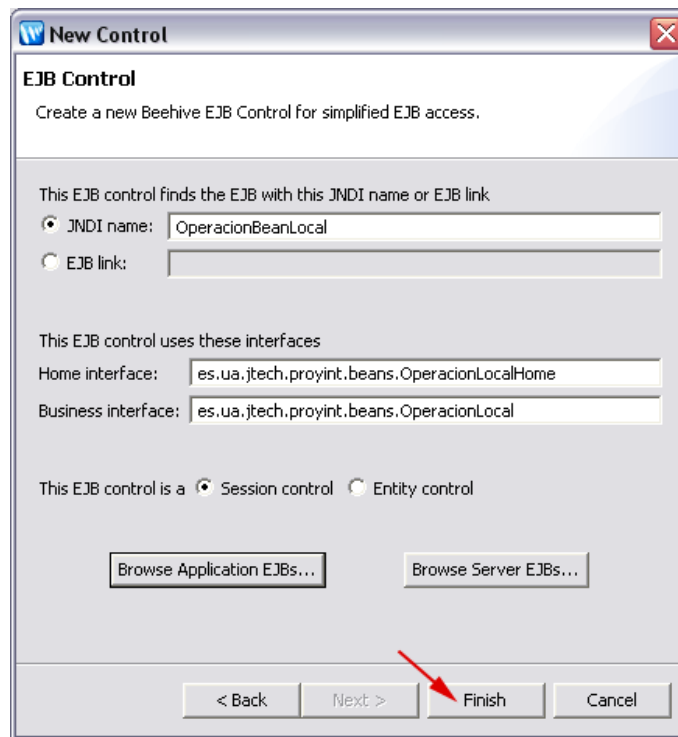
Creación de un EJB Control. Paso 3.

- En la lista veremos el EJB OperacionBean y las posibles formas de acceder a él. Seleccionamos en enlace JNDI a la interfaz local.



Creación de un EJB Control. Paso 4.

- Automáticamente se rellenarán los datos que el control necesita para acceder a dicho EJB. Pulsamos el botón *Finish* para finalizar el asistente y crear el control.



Creación de un EJB Control. Paso 5.

4. Creamos en el paquete `es.ua.jtech.proyint.ws` un nuevo servicio web BibliotecaWS.
5. Añadimos a dicho servicio web una referencia al control `OperacionEjbControl`.
6. Creamos una operación delegada en el servicio web para el método `listadoDisponibles` del control EJB.
7. Por defecto esta operación devuelve el listado como un objeto `List`. Sin embargo, esto nos puede dar problemas, ya que al no quedar expuesto en la interfaz del servicio el tipo de datos propio `LibroTO` no se generarán los serializadores y deserializadores necesarios. Por lo tanto, modificamos el método para que devuelva un array de dichos objetos de la siguiente forma:

```
public LibroTO[] listadoDisponibles() {
    List libros = operacionEjbControl.listadoDisponibles();
    LibroTO[] result = new LibroTO[libros.size()];
    libros.toArray(result);
    return result;
}
```

8. Repetimos el proceso para las operaciones `listadoPrestamos`, `listadoReservas`, y `listadoTodosLibros`.

9. Generamos un método delegado para la operación realizaReserva. Dicha operación debería poder ser invocada únicamente por un usuario con permisos para hacerlo. Por lo tanto tendremos que añadir código para realizar el control de usuarios en este método, para así evitar dejar abierta una operación que no debería ser pública. Añadiremos dos parámetros para indicar el usuario que realiza la petición (login y password) y modificamos el código como se muestra a continuación:

```
public String realizaReserva(String login, String password,
                           String socio, String isbn) {
    try {
        FactoriaBOs fb = FactoriaBOs.getInstance();
        IUsuarioBO iu = fb.getUsuarioBO();

        UsuarioTO usuario = iu.autenticaUsuario(login, password);
        if (usuario != null) {
            operacionEjbControl.realizaReserva(socio, isbn);
            return "Reserva realizada correctamente";
        } else {
            throw new SOAPFaultException(null,
                                         "Login y/o password incorrecto",
                                         "Servicio Web Biblioteca", null);
        }
    } catch (Exception e) {
        throw new SOAPFaultException(null,
                                     "Error al realizar la reserva: " + e.getMessage(),
                                     "Servicio Web Biblioteca", null);
    }
}
```

3. Acceso a servicios web externos

Vamos a añadir a la aplicación la funcionalidad de obtener los detalles de un libro (portada, precio, editorial, fecha de publicación, ...). Para hacer esto recurriremos a los servicios web de Amazon. Dado que los precios que ofrece Amazon están en dólares, utilizaremos el servicio web *Currency Exchange Service* de *XMethods* para obtener el cambio actual y así convertir dichos precios a euros. Seguiremos los siguientes pasos:

1. Añadimos en el proyecto proy-int-comun un nuevo *Transfer Object* que extienda LibroTO para añadir atributos adicionales a los libros:

```
package es.ua.jtech.proyint.to;

public class DetalleLibroTO extends LibroTO {

    private static final long serialVersionUID =
        -5489054672037280657L;

    float precio;
    String imagen;
```

```

int imagenAncho;
int imagenAlto;
String editorial;
String fechaPublicacion;
String urlDetalles;

// Getters y setters
...
}

```

2. Creamos en el paquete `es.ua.jtech.proyint.controls` del proyecto `proy-int-ws` un control (*Service Control*) para acceder al servicio web de Amazon, cuyo documento WSDL se puede encontrar en la siguiente dirección:

<http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl>

Nota:

Durante el proceso de creación del control se debe indicar que se creen los tipos mediante JAX-RPC.

3. Añadimos al servicio web una referencia al control anterior, y una operación `DetalleLibroTO [] buscaLibroAmazon(String isbn)`, que tomará como parámetro el ISBN del libro que buscamos, y nos devolverá un objeto `DetalleLibroTO` con los datos de dicho libro, accediendo para ello al servicio web de Amazon mediante el control referenciado:

```

public DetalleLibroTO [] buscaLibroAmazon(String isbn) {
    ItemLookup itemLookup = new ItemLookup();

    itemLookup.setAWSAccessKeyId("15JCR9XWMPTGV91JC1R2");
    ItemLookupRequest shared = new ItemLookupRequest();

    shared.setSearchIndex("Books");
    shared.setIdType(IdType_Enumeration.isbn);
    shared.setItemId(new String []{ isbn });
    shared.setResponseGroup(new String [] { "Offers", "Medium" });

    itemLookup.setShared(shared);

    ItemLookupResponse response = AWSECommerceServiceControll
        .itemLookup(itemLookup);

    List<DetalleLibroTO> libros = new ArrayList<DetalleLibroTO>();

    Items [] listaResultados = response.getItems();
    if(listaResultados==null) {
        return null;
    }
    for (Items resultado : listaResultados) {

```



```

Item_Element [] listaArticulos = resultado.getItem();
if(listaArticulos==null) {
    return null;
}
for (Item_Element articulo : listaArticulos) {
    ItemAttributes atributos = articulo.getItemAttributes();
    DetalleLibroTO libro = new DetalleLibroTO();
    libro.setIsbn(isbn);
    libro.setTitulo(atributos.getTitle());
    libro.setNumPaginas(atributos.getNumberOfPages().intValue());

    Image imagen = articulo.getMediumImage();
    libro.setImagen(imagen.getURL());
    libro.setImagenAlto(imagen.getHeight().get_value().intValue());
    libro.setImagenAncho(imagen.getWidth().get_value().intValue());

    String [] autores = atributos.getAuthor();
    String listaAutores = "";
    for(String autor: autores) {
        listaAutores = autor + "; ";
    }
    libro.setAutor(listaAutores);

    libro.setPrecio(atributos.getListPrice() != null ? atributos
        .getListPrice().getAmount().floatValue()/100
        : -1);

    libro.setEditorial(atributos.getPublisher());
    libro.setUrlDetalles(articulo.getDetailPageURL());
    libro.setFechaPublicacion(atributos.getPublicationDate());

    libros.add(libro);
}

DetalleLibroTO [] result = new DetalleLibroTO[libros.size()];
libros.toArray(result);

return result;
}

```

4. Creamos en el paquete `es.ua.jtech.proyint.controls` del proyecto `proy-int-ws` un control (*Service Control*) para acceder al servicio web *Currency Exchange Service*, cuyo documento WSDL se puede encontrar en la siguiente dirección:
<http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl>
5. Añadimos una referencia al control anterior en el servicio web y creamos una operación delegada para la operación `getRate` del servicio al que accede. Renombraremos nuestra operación y le daremos como nombre `tasaCambio`.
6. Añadimos a la carpeta `WebContent` del proyecto `proy-int-ws` un JSP llamado `datosLibro.jsp` que muestre los datos de un libro utilizando para ello el servicio

web de Amazon. Deberá tomar como entrada un parámetro isbn con el ISBN del libro buscado. El código fuente de este JSP será el siguiente:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-1">
<title>Biblioteca JTech - Datos del Libro</title>
<link rel="stylesheet" type="text/css" href="css/style.css"
      title="800px style" media="screen,projection" />
</head>
<body>
<div id="wrap">
  <h2>Datos del libro</h2>

  <%
    es.ua.jtech.proyint.ws.BibliotecaWS cliente =
      new es.ua.jtech.proyint.ws.BibliotecaWS();
    es.ua.jtech.proyint.ws.BibliotecaWSClientInitializer
      .initialize(null, cliente);
    es.ua.jtech.proyint.to.DetalleLibroTO [] libros =
      cliente.buscaLibroAmazon(request.getParameter("isbn"));
    float tasaCambio = cliente.tasaCambio("usa", "euro");
    java.text.NumberFormat formato =
      java.text.NumberFormat.getCurrencyInstance();
    if(libros!=null && libros.length>0) {
      %>

      <table>
      <tr>
      <td valign="top">
        "
          height="<%= libros[0].getImagenAlto() %>"/>
      </td>
      <td valign="top">
        <p><strong>ISBN: </strong><%= libros[0].getIsbn() %></p>
        <p><strong>Título: </strong><%= libros[0].getTitulo() %></p>
        <p><strong>Autor: </strong><%= libros[0].getAutor() %></p>
        <p><strong>Editorial: </strong>
          <%= libros[0].getEditorial() %></p>
        <p><strong>Fecha de publicación: </strong>
          <%= libros[0].getFechaPublicacion() %></p>
        <p><strong>Número de páginas: </strong>
          <%= libros[0].getNumPaginas() %></p>
        <p><strong>Precio: </strong>$<%= libros[0].getPrecio() %>
          (<%= formato.format(libros[0].getPrecio()*tasaCambio) %>)
        </p>
        <p><a href="<%= libros[0].getUrlDetalles() %>"
          target="_blank">Más información</a></p>
      </td>
```

```

    </tr>
  </table>

  <% } else { %>
  <p>No se han encontrado datos del libro</p>
  <% } %>

  <a href="javascript: history.back()">Volver atrás</a>
</div>
</body>

```

7. Copiamos la carpeta `css` del proyecto `proy-int-web` a `proy-int-comun`, para así mantener el mismo estilo en el JSP con los datos del libro.
8. Añadir un enlace en la página `jsp/biblio/listadoLibros.jsp` del proyecto `proy-int-web` a la página con los detalles de cada libro. Cada fila de la tabla del listado de libros deberá contener un enlace a la página `datosLibro.jsp` de `proy-int-ws` en el que se le pase como parámetro el ISBN correspondiente al libro de dicha fila:

```

<a href="/proy-int-ws/datosLibro.jsp?isbn=<bean:write name="operacion"
  property="libro.isbn" />">...</a>

```

4. Cliente para los servicios web (*)

Crear un cliente Java con Netbeans para nuestro servicio web de la biblioteca. El programa deberá invocar la operación `listadoDisponibles` y mostrar en la consola el listado de libros.

Nota:

Al generar el cliente del servicio es posible que nos aparezcan numerosos errores de compilación. Esto es debido a un problema al generar los comentarios de las clases que encapsulan los datos que intercambia el servicio (`LibroTO`, `OperacionTO`, y `UsuarioTO`). Weblogic introduce en el WSDL como documentación de estos tipos el Javadoc de sus correspondientes clases. Por su parte `wsimport` genera clases para tratar estos datos en el cliente introduciendo dicha documentación en el comentario, con el siguiente resultado:

```

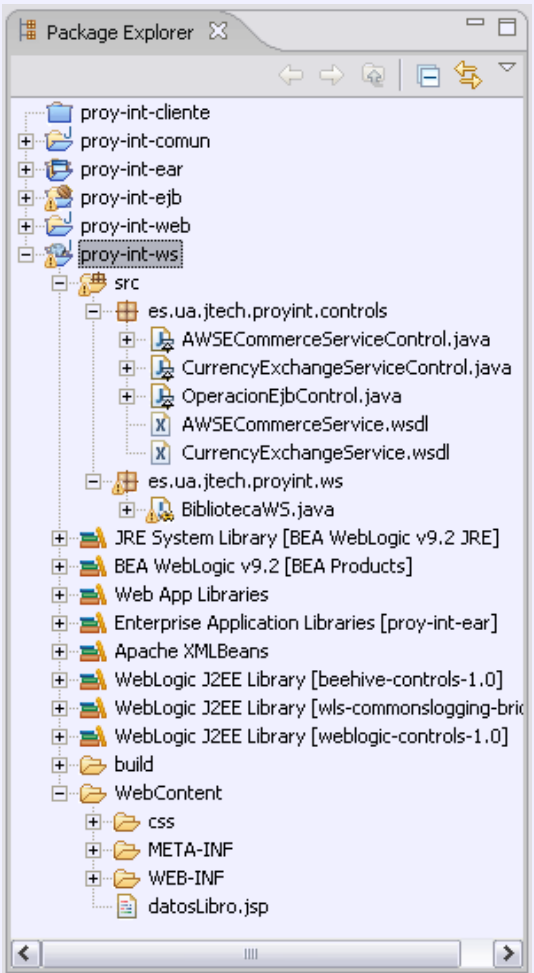
/**
 * /**
 * * Clase que representa un libro. Destacar que la
 * * app no permite 2 ejemplares de un mismo libro
 * * @author $Author: malozano $
 * * @version $Revision: 1.14 $
 * */ <-- No debe cerrar el comentario aquí

```

*

Deberemos eliminar el cierre del comentario interno para evitar problemas de compilación en estas clases. Para hacer esto podemos pinchar sobre los errores de compilación que se muestran en la consola de Netbeans para que nos lleve a la clase afectada, y ahí realizar la corrección.

5. Resumen

| | |
|--|---|
|  <p>Estructura de Proyecto</p> | <p>Se deberán implementar los siguientes componentes dentro del proyecto <code>proy-int-ws</code>:</p> <ul style="list-style-type: none"> • BibliotecaWS: Servicio web de la biblioteca. • OperacionEjbControl: Control para acceder a las operaciones del EJB OperacionBean. • AWSECommerceServiceControl: Control para acceder a los servicios web de Amazon. • CurrencyExchangeServiceControl: Control para acceder al servicio web de cambio de moneda. • datosLibro.jsp: Página JSP que muestra los detalles de un libro dado su ISBN (que se proporciona como parámetro isbn). • css: Se debe copiar la carpeta css del proyecto <code>proy-int-web</code> para mantener el mismo estilo en la página de datos de un libro. <p>Se deberá añadir el siguiente componente al proyecto <code>proy-int-comun</code>:</p> <ul style="list-style-type: none"> • DetalleLibroTO: Extiende LibroTO para añadir información adicional. <p>Se deberá modificar el siguiente componente en el proyecto <code>proy-int-web</code>:</p> <ul style="list-style-type: none"> • <code>jsp/biblio/listadoLibros.jsp</code>: Añadir un enlace en cada fila de la tabla que lleva a la página <code>datosLibro.jsp</code> del proyecto <code>proy-int-ws</code> pasando como parámetro <code>isbn</code> el ISBN del libro seleccionado. |
|--|---|

