Sistemas Web II

Práctica 2

SOAP

Carlos Ambrosio Habela

David Recio Arnés

1.Introducción

La elección del proyecto del documento para trabajar con SOAP en el entorno de desarrollo es sobre un recetario y recetas como en el proyecto anterior. La elección se llevó a cabo por dos integrantes del grupo.

El proyecto se desarrolla en Java, en el entorno de desarrollo *Neatbeans*. Se ha usado también para el correcto desarrollo por parte de los dos integrantes del grupo el control de versiones *Github* para llevar un control y facilitar el reparto de tareas y el seguimiento del proyecto.

2.Resumen

En el caso del proyecto, como no se pide ninguna interfaz gráfica se prefirió decantarse por la terminal de salida sirviendo de un menú simple, para poder enforcase en el tema principal de la práctica.

Para darle un cierto control de la programación se decidió implementar un repositorio como se dijo anteriormente, además de definir una buena estructura, todo ello permitió trabajar paralelamente ya que se tocaban zonas de código diferente, e incluso poder hacer uso del *pair programing*.

El proyecto consta de tres partes diferenciadas o tres proyectos distintos, un proyecto servidor en web y dos proyectos que actúan de clientes, uno en Java y otro en web.

Los clientes le harán peticiones al servidor y este consumirá los recursos y los ofrecerá.

Los ficheros XML necesarios para el correcto funcionamiento del proyrcto se encuentran en la carpeta *files* de los respectivos proyectos, es decir, en el caso por ejemplo del servidor web la carpeta sería (Practica2SOAPClass/files/xml), el funcionamiento es similar con los otros proyectos.

Respecto a la ejecución, solo hay un dato importante, se deben crear o importar las recetas antes de crear un recetario y cuanto este se cree se le adjuntaran automáticamente las recetas creadas o importadas (recetas en caliente).

3.Desarrollo y despliegue

El proyecto cuyo nombre es *Practica2SOAP*, está desarrollado en Java y consta como se dijo anteriormente de tres proyectos separados, dos clientes (web y java) y el servidor. Cada uno de ellos tiene una función y unos paquetes con unas clases diferenciadas de las cuales se hablará más adelante. Primero se hacen las recetas y posteriormente el recetario.

El proyecto *Practica2SOAPClass* es el servidor y consta de tres paquetes: *Funcionalidad, Recurosos y serviciosWeb* con sus respectivas clases.

* El paquete Funcionalidad tiene dos clases Marshalling y ValidarXSD que son idénticos a los de la primera práctica. La clase *Marshalling* realiza el *marshalling* y *Unmarshalling*, que sirve para hacer volcados de objetos a XML o viceversa (*unmarshalling*). Las clases ValidarDTD y ValidarXSD se usan, como su nombre indica para validar los XML correspondientes en la carpeta files y dentro de esta en la subcarpeta con nombre dtd y xsd y el fichero recetario.
* El paquete Recursos tiene dos clases que son Receta y Recetario, que tiene la misma funcionalidad que en la practica1 siendo estas las clases con las que se va a trabajar a lo largo de los proyectos. Estas clases son idénticas a las de la primera práctica.
* El paquete *serviciosWeb* contiene una clase del mismo nombre en el que están todas las funciones necesarias para el desarrollo de la practica como pueden ser añadir Receta y recetario, borrarlas, listar recetas y recetarios, etc.

El proyecto *Practica2SOAPClienteJava* es el servidor Java y consta dos paquetes a parte de los que se generan automáticamente (explicar) que son: *practica2soapclientejava, Funcionalidad* con sus respectivas clases:

* El paquete *practica2soapclientejava*, es como el *launcher* (explicar)
* El paquete *Funcionalidad* tiene dos clases, que son el menú y modelo que se encarga de hacer el trabajo pesado de la aplicación, donde participan las diferentes clases.

El proyecto *Practica2SOAPClienteWeb* es el servidor web y consta de dos paquetes *beans*, en los que hay dos clases que son Receta y Recetario necesarias para las *servlets* de las que se hablará posteriormente. Y el segundo paquete contiene las *servlets* implementadas.

A continuación, se explicarán las clases *servlets* y los respectivos archivos *html* para su correcto funcionamiento.

* *CrearRecetarioServlet:* Esta servlet crea recetarios.
* *CrearRecetaServlet:* Esta servlet crea recetas.
* *ListarRecetarioServlet:* Se obtiene un listado de recetarios disponible.
* *ListarRecetaServlet:* Se obtiene un listado de recetas disponible.
* *RmvReceta:* Borra las recetas seleccionadas.
* *ValidarXSDServlet: Esta servlet comprueba si el archivo XSD es válido.*

Hay que pulsar botón derecho y pinchar en *Deploy (explicar)*

Explicar como se genera el WSDL, como se hace el webService y como se ejecuta todo.