

Release Note

Implementación de un servicio CRUD en Java

Descripción

En la presente práctica se ha realizado un servicio web basado en las operaciones estándares de el protocolo HTTP como son GET, POST, PUT y DELETE.

La práctica está definida en dos partes; una parte con el servicio web que expone los recursos en forma de texto plano, xml y json y otra de un formulario para crear un nuevo recurso utilizando el navegador como cliente web. Para la parte de servicio web se ha utilizado la librería **jersey** junto con la tecnología de los **servlets**, y Apache Tomcat como contenedor de estos para poder redireccionar las peticiones del cliente web al servlet correspondiente que previamente estuviese registrado, en este caso solamente uno. Para la parte del cliente web se ha utilizado el propio navegador para realizar las peticiones de GET y POST.

Como se ha podido comprobar durante todo el desarrollo, eran tres paquetes los que componían el sistema. El paquete *modelo*, el cual incluía la clase que define el dominio del *modelo*. El paquete *singleton*, que contiene una clase que implementa dicho patrón, el cual contiene un contenedor de Car (la clase del modelo). Y el paquete *resources*, donde se desarrolla el conjunto de métodos que se ejecutarán en función del tipo de petición, si es GET o POST.

Tecnologías y herramientas

Son varias las librerías y herramientas empleadas en el desarrollo de la práctica, comenzando por el IDE en el que se ha basado su desarrollo, **Eclipse**, que trae todo un entorno de desarrollo para aplicaciones JEE, Faces Projects, Web Dynamic Module, etc.

Jersey ha sido la librería utilizada para implementar el servicio web en Java.

La librería **JAXB** para la creación automática de ficheros XML a partir de clases Java.

El api **servlet** 3.0 ha sido el que se ha empleado para la implementación de estos módulos.

Para la parte del servidor web, **Apache Tomcat** en las versiones v8.0 y v8.5 han sido las utilizadas en el desarrollo y prueba del proyecto.

Modo de operación

Para poder probar esta práctica, es necesario tener referenciadas todas las librerías y dependencias mencionadas en el apartado anterior. Una vez que se cumple con todos esos requisitos se ha de seguir los siguientes pasos para su ejecución:

1. Ejecutar en modo servidor (Run as Server) el proyecto **com.jersey**.
2. Cuando arranque el servlet se abriría una pagina del navegador web, la parte del cliente, que habrá que añadirle la al final de la ruta **/rest/cars**, de esta forma aparecerá información de todos los recursos que existen en el servicio.
3. Para conocer la cantidad de recursos que existen en el servicio es necesario hacer la petición del recurso **http://localhost:8080/com.jersey/rest/cars/count**.
4. Para conocer los datos de un recurso en concreto hay que indicarle su número de matrícula a la petición del recurso del siguiente modo http://localhost:8080/com.jersey/rest/cars/3245DDF.
5. Para añadir un nuevo recurso hay que ejecutar desde el servidor (Run as Server) el fichero **create_car.html** y rellenar los datos que solicita.

Esquema de los componentes

En la siguiente imagen se muestra el esquema de los componentes que forman el servicio web:

