Guía de desarrollo web

Índice de contenido

Objetivo del documento

Dar una guía de instalación y configuración para el desarrollo de aplicaciones web en JEE, entendiendo estas como el conjunto de APIs incluidas en el denominado ***web profile*** (ver documento https://java.net/downloads/javaee-spec/WebProfile.pdf) entre las que se incluyen:

- Servlet 3.1
- JavaServer Pages (JSP) 2.2
- Expression Language (EL) 3.0
- Debugging Support for Other Languages (JSR-45) 1.0
- Standard Tag Library for JavaServer Pages (JSTL) 1.2
- JavaServer Faces (JSF) 2.2
- Java API for RESTful Web Services (JAX-RS) 2.0
- Common Annotations for the Java Platform (JSR-250) 1.1
- Enterprise JavaBeans (EJB) 3.2 Lite
- Java Transaction API (JTA) 1.2
- Java Persistence API (JPA) 2.1
- Bean Validation 1.1
- Managed Beans 1.0
- Interceptors 1.1
- Contexts and Dependency Injection for the Java EE Platform 1.1
- Dependency Injection for Java 1.0

En nuestro caso particular, se utilizará para el desarrollo de aplicaciones con servlets y JSP. El despliegue de estas aplicaciones en Eclipse se hará de la misma forma que con un proyecto EJB, añadiendo o eliminando (opción *Add and Remove...* en la vista *Servers*).

Desarrollo con Eclipse

Se crea un nuevo proyecto de tipo *Web / **Dynamic Web Project***.

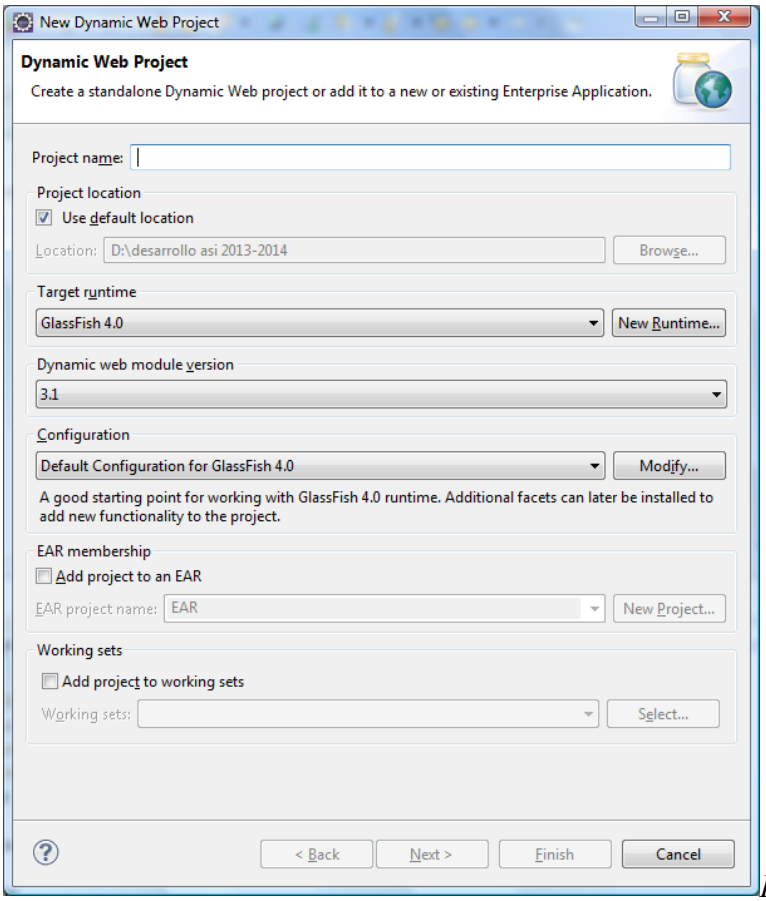

Ilustración 1: Creación de un proyecto web dinámico

El entorno de ejecución es GlassFish y la versión de módulo es 3.1. En la siguiente ventana del asistente se aceptan los directorios por defecto. En la última ventana, aceptamos los directorios por defecto y se marca la casilla para generar un fichero `web.xml`.

En el directorio `src` se alojan los códigos fuente (ficheros `.java` como servlets, bean, helper, etc.) mientras que en la carpeta `webcontent` se alojan los contenidos web como `.html, .js, .css`, etc.

Una vez creado el proyecto, deben revisarse sus *facetas*. Para ello, botón derecho sobre el proyecto, *Properties* y seleccionamos *Project Facets*.

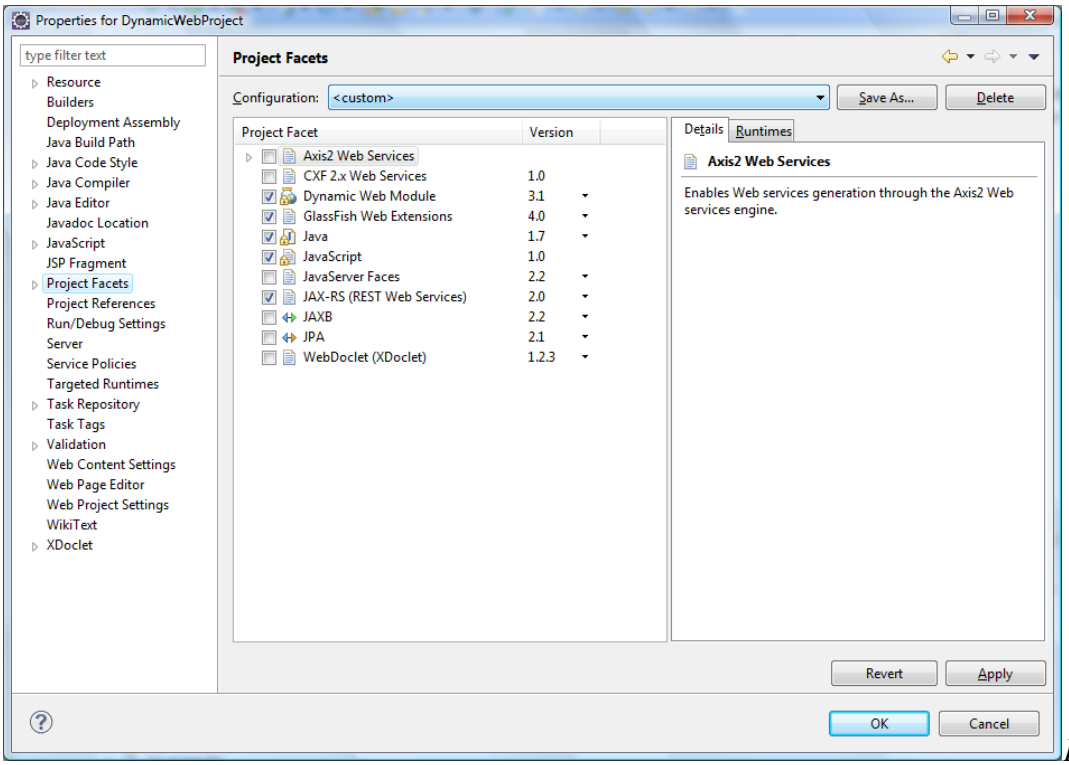

Ilustración 2: Configuración de facetas de un proyecto

En esta ventana se pueden ajustar las propiedades de nuestra aplicación, seleccionando el soporte y versión a distintas facetas del proyecto. Por ejemplo cambiando las versiones de APIs utilizadas en nuestra aplicación.

Desarrollo con Eclipse-Maven

Creamos un proyecto Maven con arquetipo **`webapp-javaee7`**.

El arquetipo podría no estar disponible inicialmente, por lo que seleccionamos *Add Archetype* y en el diálogo siguiente introducimos valores:

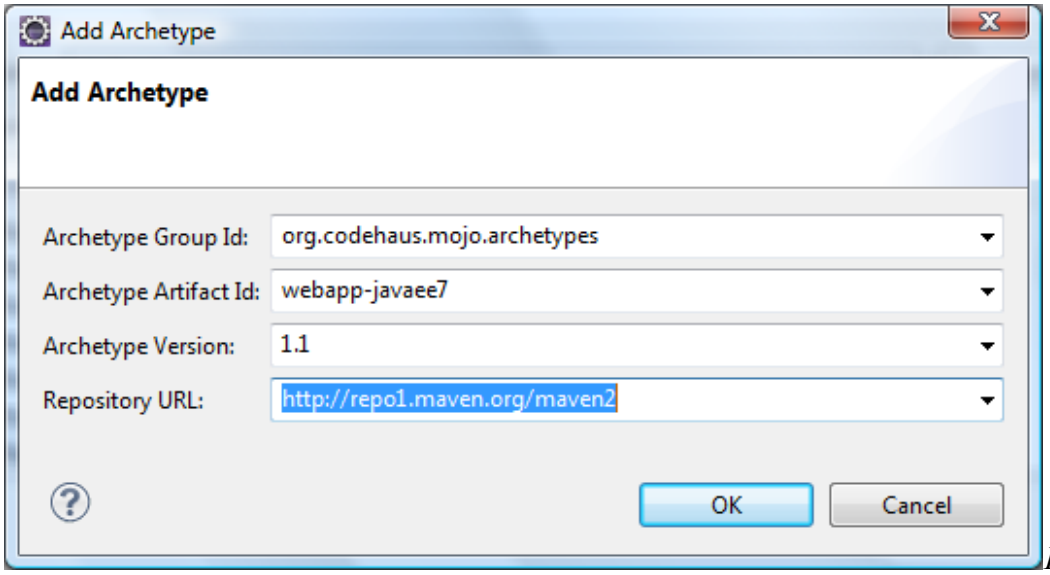

Ilustración 3: Añadir un arquetipo no existente

Con el arquetipo seleccionado introducimos las coordenas de nuestro producto.

En el pom.xml creado hay un bug, presentando el siguiente error:

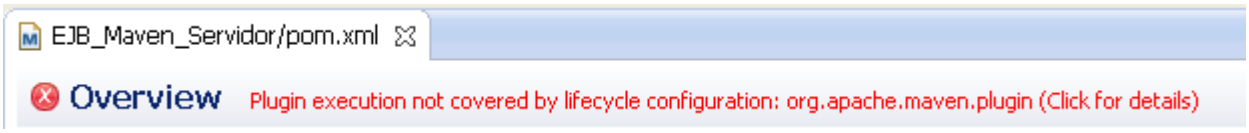

Ilustración 4: Bug en el pom.xml generado.

Hacemos click sobre el errror y seleccionamos la tercera opción *Mark goal copy as ignored in Eclipse preferences* para corregir el bug.

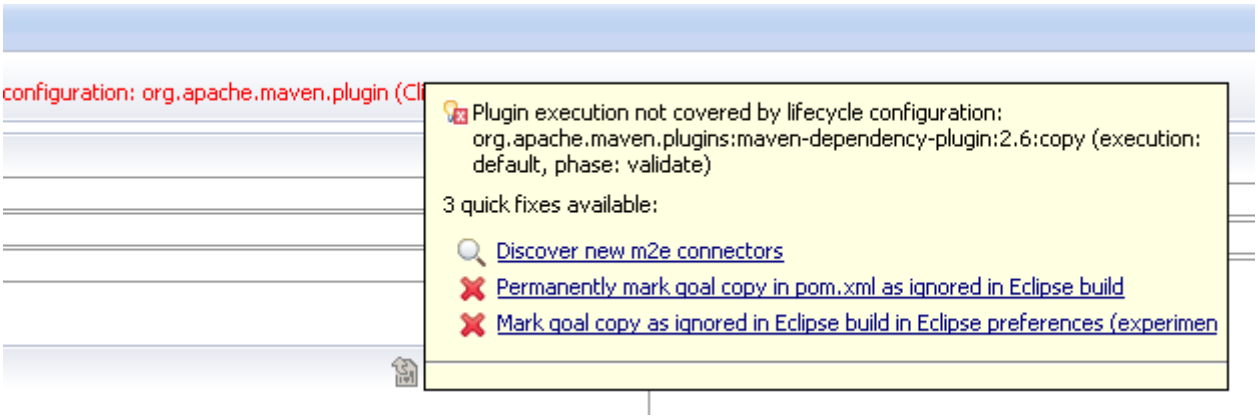

Ilustración 5: Corrección del bug

La estructura de carpetas es ligeramente distinta teniendo:

- `src\main\java` para el desarrollo de ficheros .java.
- `src\main\webapp` para el desarrollo de ficheros web.

Por defecto el arquetipo no crea un fichero `web.xml`. En función del uso o no de anotaciones en el código puede ser necesario añadirlo manualmente.

Como siempre, puede ser necesario actualizar el proyecto maven, ***Maven>Update Project...*** para actualizar cambios.

Licencia

Autor: Raúl Marticorena
Área de Lenguajes y Sistemas Informáticos
Departamento de Ingeniería Informática
Escuela Politécnica Superior
UNIVERSIDAD DE BURGOS