



Desarrollo de plugin de Extensión sobre Liferay

Contenidos

1. Introducción
2. Estructura del proyecto
3. Modificación de la base de datos.



INTRODUCCIÓN

Introducción

- Plugin de muy bajo nivel.
- Permite extender y modificar los comportamientos de Liferay.
- Aplican cambios a los módulos de Liferay
 - ext
 - ext-impl
 - ext-lib
 - ext-service
 - ext-web

Introducción

- Utilizarlos SÓLO en caso de que no sea posible implementar un **Portlet** o un **Hook**.
- El principio de los **Ext**, es la de modificar el comportamiento del portal, para ello, cuando se despliega un **Ext**, se estarán mezclando los ficheros desplegados con los disponibles en el portal, esto permite no modificar las clases del portal de forma directa pudiendo revertir los cambios, aunque esto no se haga fácilmente.



ESTRUCTURA DEL PROYECTO

Estructura del proyecto

- La estructura del proyecto es:
 - **ext-impl/src**: Contiene el fichero **portal-ext.properties** que permite sobrescribir la configuración del JAR "portal-impl.jar" y las clases de código nuevas.
 - **ext-lib/global**: Lugar donde se han de copiar las librerías (JAR) necesarias en el classLoader del Servidor de aplicaciones (TOMCAT).
 - **ext-lib/portal**: Lugar donde se han de copiar las librerías (JAR) necesarias en el classLoader del Portal (Liferay). Necesarias para alguna clase nueva.

Estructura del proyecto

- La estructura del proyecto es:
 - **ext-service/src**: Situar clases que sobrescriban las clases del JAR "portal-service.jar".
 - **ext-web/docroot**: Lugar donde se han de poner ficheros de configuración como **WEB-INF/struts-config-ext.xml** para modificar el comportamiento de los Action de Struts ya configurados en Liferay. También se han de situar aquí los JSP necesarios para personalizar el portal.

Estructura del proyecto

- La estructura del proyecto es:
 - **ext-util-bridges/src**: Situar clases que sobrescriban las clases del JAR "util-bridges.jar".
 - **ext-util-java/src**: Situar clases que sobrescriban las clases del JAR "util-java.jar".
 - **ext-util-taglib/src**: Situar clases que sobrescriban las clases del JAR "util-taglib.jar".



MODIFICACIÓN DE LA BASE DE DATOS

Modificación de la base de datos

- Una de las modificaciones mas habituales a realizar en un **Portal Liferay**, es la de la Base de Datos, ya que por defecto emplea una **Hypersonic**, que no ofrece un gran rendimiento.
- Para poder emplear otra base de datos, será necesario crearla y lanzar sobre ella los scripts que se pueden descargar desde la web

<http://sourceforge.net/projects/lportal/files/Liferay%20Portal/6.2.3%20GA4/liferay-portal-db-upgrade-6.2-ce-ga4-20150416163831865.zip/download>

Modificación de la base de datos

- Veamos un ejemplo para configurar una base de datos Derby

- Descargar Derby

```
http://www.eu.apache.org/dist/db/derby/db-derby-10.12.1.1/db-derby-10.12.1.1-bin.zip
```

- Arrancar el servidor

```
<directorio de instalación>/bin/startNetworkServer.bat
```

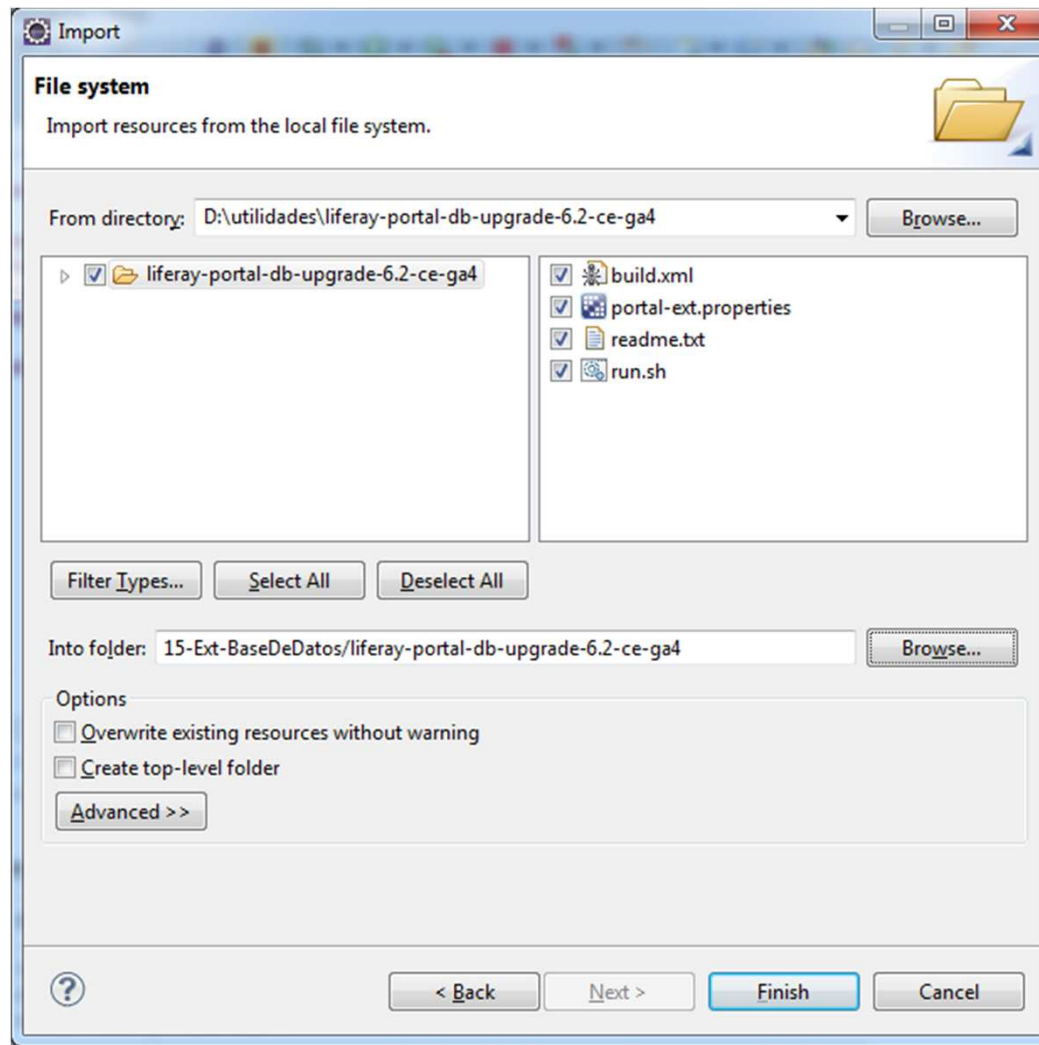
- Hay que ejecutar las consultas de creación de Liferay sobre esa instancia, para ello en Windows se necesita Ant.

Modificación de la base de datos

- Crear un nuevo proyecto de tipo Ext.
- Dentro del principal crear una carpeta que contendrá los ficheros descargados de Liferay con las Queries, algo como, **liferay-portal-db-upgrade-6.2-ce-ga4**.

<boton derecho> -> Import -> File System

Modificación de la base de datos



Modificación de la base de datos

- Se ha de modificar el fichero **portal-ext.properties**, con los datos de conexión de la base de datos nueva

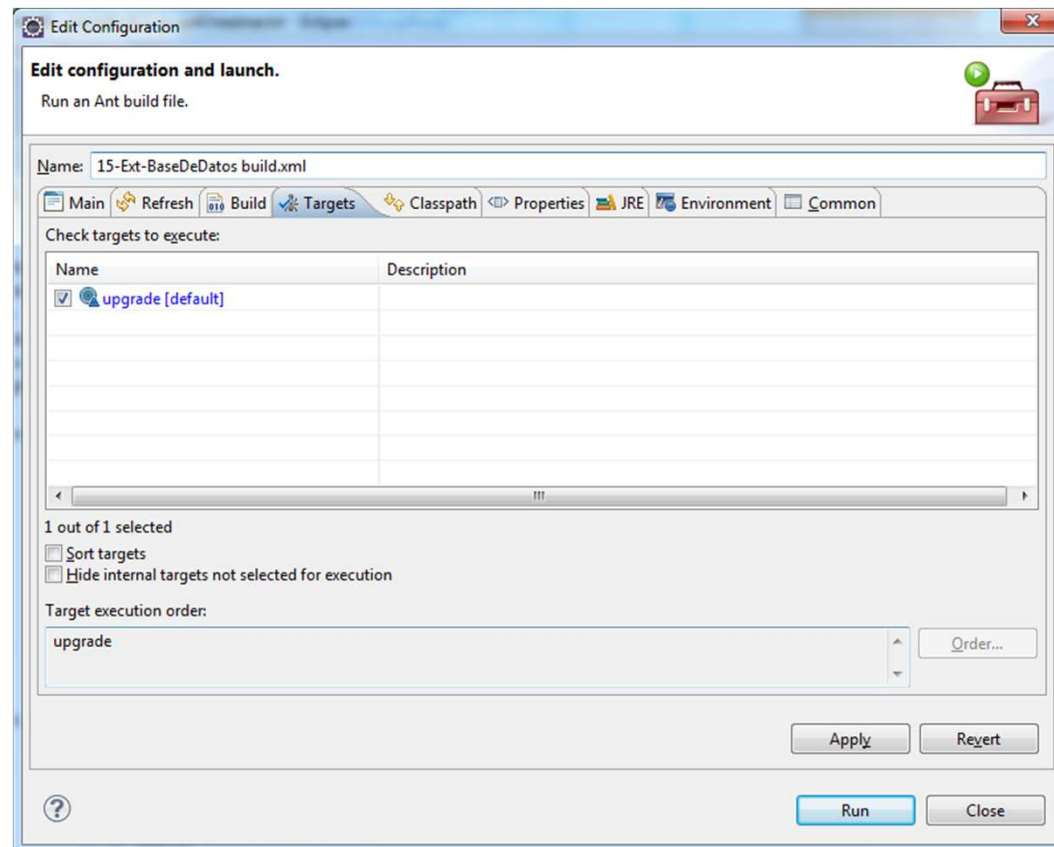
```
jdbc.default.driverClassName=org.apache.derby.jdbc.ClientDriver  
jdbc.default.url=jdbc:derby://localhost:1527/lportal  
jdbc.default.username=admin  
jdbc.default.password=admin
```

- Una vez configurado el fichero se ha de copiar el jar de driver en la carpeta **lib**, en este caso será

```
<directorio de instalación>/lib/derbyclient.jar
```

Modificación de la base de datos

- Por último lanzar la tarea **ant** con eclipse, siendo la tarea a lanzar **upgrade**



Modificación de la base de datos

- Una vez creado el esquema de Liferay en la nueva base de datos, se ha de configurar el proyecto de Ext para que cambie dicha configuración, para ello, se ha de incluir un fichero **portal-ext.properties**, en el proyecto *-**ext-impl**, con

```
jdbc.default.driverClassName=org.apache.derby.jdbc.ClientDriver  
jdbc.default.url=jdbc:derby://localhost:1527/lportal  
jdbc.default.username=admin  
jdbc.default.password=admin
```

Modificación de la base de datos

- Y en el **pom.xml** del proyecto ***-ext-lib-portal**

```
<dependencies>
  <dependency>
    <groupId>org.apache.derby</groupId>
    <artifactId>derbyclient</artifactId>
    <version>10.11.1.1</version>
  </dependency>
</dependencies>
```



DEPLOY DE EXT

Desplegar Plugin Ext

- Los plugin se desplegaran desde una tarea Maven del proyecto Maven principal, que desplegara todos los proyecto que incluye.

```
mvn liferay:deploy
```



UNDEPLOY DE EXT

Undeploy de plugin Ext

- Para quitar un plugin tipo Ext de Liferay, habrá que borrar los siguientes ficheros de Liferay.

Undeploy de plugin Ext

- Suponiendo que el plugin se llama **ejemplo**.
 - /webapps/ejemplo-ext
 - /lib/ext/ext-ejemplo-ext-service.jar
 - /webapps/ROOT/WEB-INF/lib/ext-ejemplo-ext-util-bridges.jar
 - /webapps/ROOT/WEB-INF/lib/ext-ejemplo-ext-util-taglib.jar
 - /webapps/ROOT/WEB-INF/lib/ext-ejemplo-ext-util-java.jar
 - /webapps/ROOT/WEB-INF/lib/ext-ejemplo-ext-impl.jar
 - /webapps/ROOT/WEB-INF/ext-ejemplo-ext.xml
 - /webapps/ROOT/WEB-INF/classes/portal-ext.properties



HOOK VS EXT

Hook vs EXT

- Los **hook**, son un plugin que interceptan cierta funcionalidad del portal, permitiendo la alteración del comportamiento.
- Los **ext**, modifican físicamente los ficheros del portal.
- Pueden existir desplegados múltiples **hook**, pero solo un **ext**.