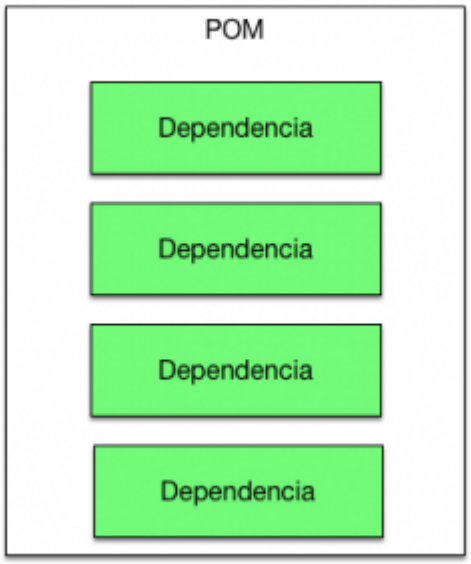
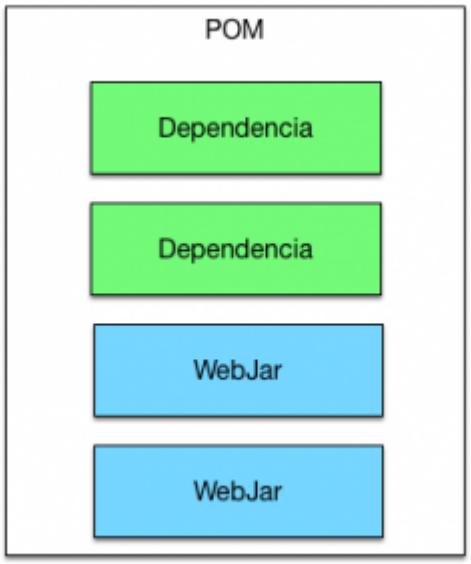
¿Qué son los WebJar?. Habitualmente estamos acostumbrados a trabajar con aplicaciones Java complejas que incluyen un grupo de dependencias amplio. Maven es una de las tecnologías que más usamos para resolver las dependencias que declaramos en el pom.xml.



Ahora bien las cosas están cambiando y aunque seguimos necesitando jars en nuestra aplicación, cada día tenemos más dependencias de librerías de JavaScript. Casos típicos son jQuery , React o Angular . El problema que tenemos con Maven es que no se trata de ficheros jars sino de ficheros JavaScript. Por lo tanto la gestión no es tan inmediata.

Usando un WebJar

¿Qué es un WebJar? un WebJar es un fichero jar que contiene librerías de JavaScript que podemos usar en nuestros proyectos Java. La particularidad que tienen es que se pueden gestionar como dependencias de Maven o del gestor que deseemos.



Vamos a construir una aplicación Web Java que use el WebJar de de jQuery. El primer paso es ir a la página oficial de WebJars y buscar la dependencia que necesitamos.

```
<dependency>
 <groupId>org.webjars</groupId>
 <artifactId>jquery</artifactId>
 <version>2.2.3
```

Una vez disponemos de nuestra dependencia la añadimos a nuestro fichero POM. En este caso al ser una aplicación web sencilla solo usaremos dos dependencias , nuestro WebJar y la de Servlets API.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
```

```
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.arquitecturajava</groupId>
<artifactId>WebJars</artifactId>
<version>1</version>
<properties>
<failOnMissingWebXml>false</failOnMissingWebXml>
</properties>

<packaging>war</packaging>
<dependencies>
<dependency>
<groupId>org.webjars</groupId>
<artifactId>jquery</artifactId>
<version>2.2.3</version>
</dependency>
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>javax.servlet-api</artifactId>
<version>3.1.0</version>
</dependency>

</dependencies>
</project>
```

Al añadir el WebJar a nuestra aplicación este se publicará como un recurso estático en una URL. Creamos un JSP y mapeamos la ruta al script.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

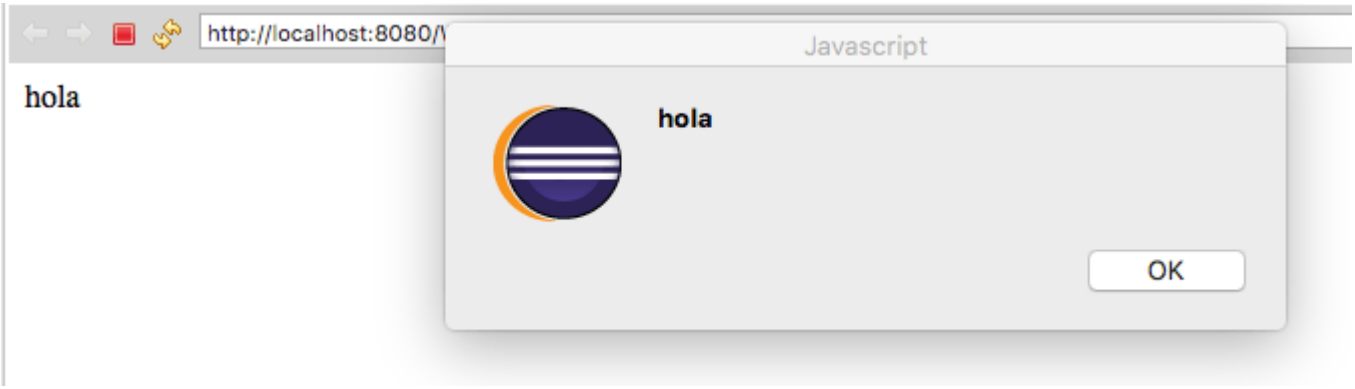
```
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src='webjars/jquery/2.2.3/jquery.js'>
</script>
<script type="text/javascript">
$(document).ready(function() {

alert("hola");
})

</script>
</head>
<body>
hola

</body>
</html>
```

Ejecutamos y e l navegador responde mostrando una alerta por pantalla.



Acabamos de usar nuestro primer WebJar para añadir jQuery como dependencia al proyecto.

Otros artículos relacionados : Introducción a Maven , Maven y Dependencias