配置

### 依赖的配置要在pom.xml文件中写

**配置依赖根节点，所有依赖在这里写**

<dependencies>

</dependencies>

例如写一个servlet-api的依赖

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

</dependencies>

例如Spring的依赖

<dependency>

**Jar包工厂坐标**

<groupId>org.springframework</groupId>

**Jar包名**

<artifactId>spring-core</artifactId>

**版本号**

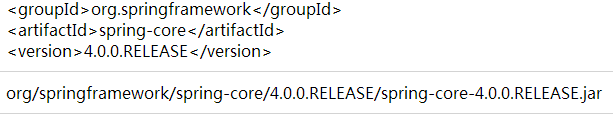
<version>4.0.0.RELEASE</version>

**存在范围**

<scope>compile</scope>

</dependency>

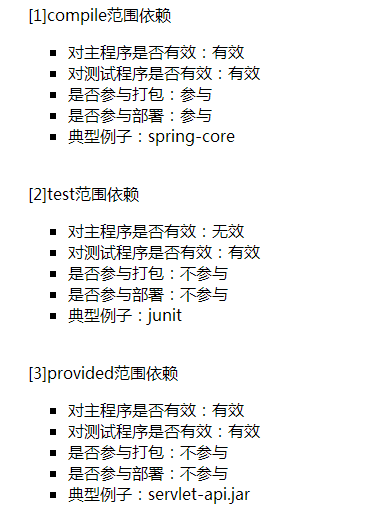
Maven工厂坐标和仓库路径的关系，用一个例子来说明



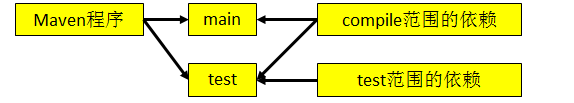
### 2． <scope>标签的值

有三个常用取值

compile，test，provided



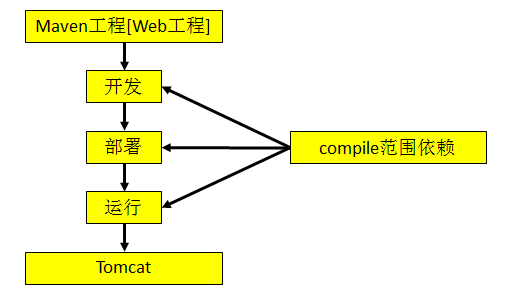
对比compile和test



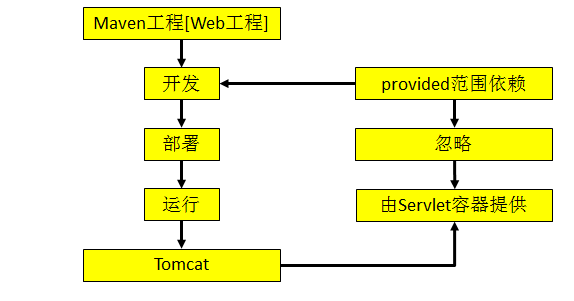
test可以依赖compile，反之不行

对比compile和provided

compile 例如Spring框架的jar包



provided 例如servlet-api的jar包



**注意：**<**scope>的值不能乱写，像servlet-api必须写proxided，因为只在开发的时候需要它，运行时使由Tomcat提供，如果被部署的话，会和Tomcat本身的包冲突**

依赖的特性

### 依赖的传递

依赖一个有依赖的包,就会继承它的依赖，Maven会自动将其依赖的包做映射

只有compile的才能传递

### 依赖的排除

可以通过排除，割断一个包的依赖 代码：

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>4.0.0.RELEASE</version>

<scope>compile</scope>

<exclusions>

<exclusion>

<artifactId>commons-logging</artifactId>

<groupId>commons-logging</groupId>

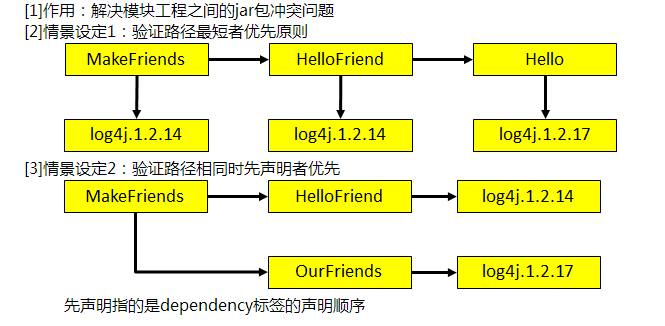
</exclusion>

</exclusions>

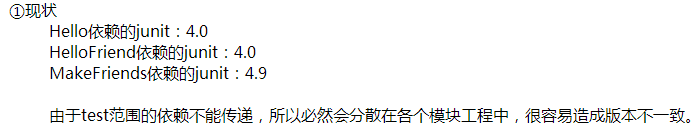
</dependency>

这样就割断了commons-logging包的传递性，在当前工程，以及依赖当前的工程里就都不会出现这个包了

### 依赖的原则

****

继承





**以父工程统一设定为准，同时也便于修改**











聚合

