

依赖的范围，使用scope表示

在我们添加依赖jar包的代码到pom文件的依赖属性中时，通常会在依赖坐标之下添加一个<scope></scope>,其中填写关键词表示这个依赖作用的范围。

--scope中可填写的值有：

compile、test、provide三个

--依赖添加scope，表示有使用范围，什么是使用范围？

就是表示这个依赖在maven构建项目的哪个阶段起作用

--maven构建项目包括：清理、编译、测试、打包、安装、部署这几个过程阶段

如下单元测试junit依赖写法：

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.11</version>
  <scope>test</scope>
</dependency>
```

--依赖范围是test：表示在项目构建过程中只在测试阶段使用该jar

什么意思呢？

就是说我们在pom文件中的依赖有这样包含test范围的，在项目构建的编译主程序时是没有这个依赖的，但是在编译测试程序或者测试阶段，这个依赖才参与进来。

非常明显的就是我们在主程序中写好方法后，写个主方法，然后调用junit中Assert类中的比较方法Assert.assertEquals(30,res);

就会发现需要导包，但是junit还得自己添加。

而在测试程序使用上面方法只需要导入相应的junit的Assert类就可以了，不需要导junit的jar包

--当依赖范围为compile时：表示编译、测试、打包、安装、部署这些过程都可以使用这个jar包

--默认不写范围scope时就是compile

--当依赖范围为provide时：

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
  <scope>provided</scope>
</dependency>
```

--表示在项目构建到打包和安装阶段该jar包不起作用，上面的servlet的jar包，我们打包web项目为war文件，部署到Tomcat服务器中，而Tomcat的lib目录是有servlet规范的jar包的，启动服务器，我们的web项目运行所需要的servlet依赖用的是服务器中的，所以打包时就没必要将servlet的jar包封装到war文件中，所以需要provide，表示以后会有提供者。

如果是compile的话，打包后的文件中WEB-INF目录下lib就是这些依赖的jar包存放位置

4.1 依赖的范围

依赖的范围：compile、test、provided，默认采用 compile

	compile	test	provided
对主程序是否有效	是	否	是
对测试程序是否有效	是	是	是
是否参与打包	是	否	否
是否参与部署	是	否	否