

classpath: 和 classpath*: 的区别

概念解释及使用场景：

classpath是指WEB-INF文件夹下的classes目录。

通常我们一般使用这种写法实在web.xml中，比如spring加载bean的上下文时，如下：

```
<!--系统自动加载文件-->
<!--这里使用的是classpath*的形式-->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath*/spring-context-*.xml</param-value>
</context-param>
<!--配置spring的context监听器 -->
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener
</listener-class>
</listener>
```

经过如上的写法，可能会认为这个就是web.xml固有的写法，其实不是，这种写法是spring的写法，与web.xml无关。可以通过spring的方法使用这种方式进行路径的读取。

classpath和classpath*区别：

classpath：只会到你的class路径中查找文件。

classpath*：不仅包含class路径，还包括jar文件中（class路径）进行查找。

注意：用classpath*：需要遍历所有的classpath，所以加载速度是很慢的；因此，在规划的时候，应该尽可能规划好资源文件所在的路径，尽量避免使用classpath*。

classpath*的使用：

当项目中有多个classpath路径，并同时加载多个classpath路径下（此种情况多数不会遇到）的文件，*就发挥了作用，如果不加*，则表示仅仅加载第一个classpath路径。

一些使用技巧：

1、从上面使用的场景看，可以在路径上使用通配符*进行模糊查找。比如：

```
<param-value>classpath:**/applicationContext-*.xml</param-value>
```

2、“**/”表示的是任意目录；“**/applicationContext-*.xml”表示任意目录下的以“applicationContext-”开头的XML文件。

3、程序部署到tomcat后，src目录下的配置文件会和class文件一样，自动copy到应用的WEB-INF/classes目录下；classpath:与classpath*的区别在于，前者只会从第一个classpath中加载，而后者会从所有的classpath中加载。

4、如果要加载的资源，不在当前ClassLoader的路径里，那么用classpath:前缀是找不到的，这种情况下就需要使用classpath*:前缀。

5、在多个classpath中存在同名资源，都需要加载时，那么用classpath:只会加载第一个，这种情况下也需要用classpath*:前缀。