servlet 两种配置方法详解

1、web.xml 中 Servlet 的注解

```
<servlet>
        <!-- servlet 的内部名称，自定义 -->
        <servlet-name>DemoAction</servlet-name>
        <!-- servlet 的类全名：包名+类名 -->
        <servlet-class>com.uplooking.controller.DemoAction</servlet-class>
        <load-on-startup>1</load-on-startup>
</servlet>
<!-- servlet 的映射配置 -->
<servlet-mapping>
        <!-- servlet 的内部名称，一定要和上面的内部名称保持一致 -->
        <servlet-name>DemoAction</servlet-name>
        <!-- servlet 的映射路径（访问 serclet 的名称 -->
        <url-pattern>/DemoAction</url-pattern>
</servlet-mapping>
```

1、首先，从浏览器中发送请求，是从当前工程中的路径与 servlet-mapping 标签中的 url-pattern 的标签值进行匹配。

2、根据这个映射值，找到 servlet-mapping 标签中的 servlet-name 的值与 servlet 标签中的 servlet-name 进行匹

3、匹配到以后，找到 servlet 标签中的 servlet-class 标签中对应 servlet 类的 src 文件夹下的全路径。

4、从而调用并执行相应的 servlet 类。

注意：servlet-mapping 标签中的 servlet-name 的值与 servlet 标签中的 servlet-name 必须相同。

<init-param>初始化参数

```
<init-param>
            <param-name>abc</param-name>
            <param-value>123</param-value>
</init-param>
```

一个 Servlet 可以配置一个或多个初始化参数。

在应用程序中，可以使用 Servlet 的 getInitParameter(String param)来读取初始化 param 对应的参数；若要读取所有的初始化参数名称，则可以使用 getInitParameterNames()方法获得所有的参数名称，类型为枚举(Enumeration)。

```
//获取所有初始化参数
Enumeration<String> strs=this.getInitParameterNames();
while(strs.hasMoreElements()) {
        str=strs.nextElement();
```

```
        System.out.println(str+"     "+this.getInitParameter(str));
         //-->abc 123
        //-->aaa 111
    }
```

这些初始化参数也可以由 ServletConfig 对象获取，Servlet 提供 getServletConfig()方法提供 ServletConfig 对象。由 ServletConfig 获取初始化参数和由 Servlet 获取初始化参数的方法是一样的。

```
//两种调用 getInitParameter 的情况，视情况而定
//this 是指 ServletConfig 的对象 config
String str=this.getInitParameter("abc");  //第一种

String str=config.getInitParameter("abc");//第二种
```

初始化参数的一个有趣应用是进行单个文件的访问加密，原理是将用户名和密码写入初始化参数中，这样的好处是简单、方便，缺点是不灵活，安全性也不高，适用于临时性的措施。

<context-param>上下文参数

```
<context-param>
        <param-name>root</param-name>
        <param-value>123</param-value>
 </context-param>
```

获取 context-param 需要使用 ServletContext 对象。ServletContext 对象可以通过在 Servlet 中的 getServletConfig().getServletContext()方法获得。得到 ServletContext 对象后，使用 getInitParameter(String param)方法获取名为 param 的参数值，通过 getInitParameterNames()获取所有的 context-param 名称。

```
ServletContext context=this.getServletContext();
String root=context.getInitParameter("root");
System.out.println("root="+root);
```

2、注解访问 servlet

1.只需在对应的 servlet 类中添加 servlet 注解即可，从浏览器发送请求时，是用当前“工程”下的路径，会去对应 servlet 类的上面寻找是否存在对应 url 名称的@webServlet 注解，存在的话，调用并执行对应的 servlet 类。

```
1 @WebServlet("/DemoAction")
2 public class DemoAction  extends HttpServlet{
3 }
```

总结：
共同点：注解访问 servlet 和 web 配置文件访问 servlet 都能完成对 servlet 的访问。

注解访问 servlet：
优点：代码少，可读性强，易于理解。
缺点：如果大量使用 servlet 注解，servlet 类文件数量过多，不便于查找和修改。

web 配置文件访问 servlet：
优点：集中管理各 servlet 类路径的映射路径，便于修改和管理。
缺点：代码多，可读性不强，不易于理解。

注意：有时候在服务上运行的 web 程序数据不能及时更新，记得重启服务器或者清除浏览器缓存。