

1. Servlet:

Java类必须符合一定的规范:

- a. 必须继承 `java.servlet.http.HttpServlet`
- b. 重写其中的 `doGet()` 和 `doPost()` 方法
`doGet()`: 接受并处理所有的get提交方式的请求
`DoPost()`: 接受并处理所有的post提交方式的请求

2. Servlet要想请求，必须配置

Servlet2.5: web.xml

```
<servlet>
    <servlet-name>WelcomeServlet</servlet-name>
    <servlet-
class>java.com.jiang.jsp.WelcomeServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>WelcomeServlet</servlet-name>
    <url-pattern>/WelcomeServlet</url-pattern>
</servlet-mapping>
```

Servlet3.0: @WebServlet

```
@WebServlet(value="/WelcomeServlet2")
```

3. /表示(重要)

web.xml 中 (项目的根路径): `http://localhost:8888/webapp`

jsp 中: http (服务器的根路径): <http://localhost:8888/>

4. Servlet的生命周期: 5个阶段

加载

初始化: `init()`, 该方法会在Servlet被加载并实例化 以后执行

服务: `service()` → `doGet()` `doPost()`

销毁: `destroy()` Servlet被系统回收时执行

卸载

`init()`:

a. (默认) 第一次访问Servlet时会被执行(只执行这一次)

b. 可以修改为 Tomcat启动时自动执行

```
<servlet>
```

```
    <servlet-name>WelcomeServlet</servlet-name>
```

```
    <servlet-
```

```
class>java.com.jiang.jsp.WelcomeServlet</servlet-
```

```
class>
```

```
    <load-on-startup>1</load-on-startup>
```

```
</servlet>
```

```
@WebServlet(value="/WelcomeServlet2", loadOnStartup=1)
```

`service()`: `doGet()` `doPost()` 调用几次, 执行几次

`destroy()`: 关闭tomcat服务器时, 执行一次

5. Servlet API:

由两个软件包组成: 对应于HTTP协议的软件包, 对应于除了HTTP协议以外的其他软件包, 即

Servlet API可以适用于 任何 通信协议

`javax.servlet.http` (我们现在学的)

6. servlet的继承关系

三个接口:

Servlet : `init()`, `service`, `destory`

ServletConfig

ServletConfig `getServletContext()`: 获取Servlet上下文对象 返回值创建 application

`getInitParameter(String name)`:在当前Servlet范围内,获取名为name的参数值(初始化参数)

`Serializable`

-->GenericServlet-->HttpServlet-->自建Servlet

a. ServletContext中的常用方法(application)

`getRealPath()`:绝对路径

`getContextPath()`:相对路径

`getAttribute()`, `setAttribute()`

`String getInitParameter(String name)`;在当前Web容器范围内,获取名为name的参数值(初始化参数)

HttpServletRequest中的方法与request相同

HttpServletResponse中的方法与response相同

7. 在Servlet里获得其他内置对象

`out:PrintWriter out=response.getWriter();`

`session:request.getSession()`

`application request.getSession().getServletContext()`