

JavaWeb 核心之 Servlet

教学导航

教学目标	案例一、完成用户登录功能 案例二、记录成功登录系统的人次
教学方法	

一、Servlet 简介

1. 什么是 Servlet

Servlet 运行在服务端的 Java 小程序，是 sun 公司提供一套规范（接口），用来处理客户端请求、响应给浏览器的动态资源。但 servlet 的实质就是 java 代码，通过 java 的 API 动态的向客户端输出内容

servlet 规范：包含三个技术点

- 1) servlet 技术
- 2) filter 技术---过滤器
- 3) listener 技术---监听器

2. Servlet 快速入门

实现步骤：

- 1) 创建类实现 Servlet 接口
- 2) 覆盖尚未实现的方法---service 方法
- 3) 在 web.xml 进行 servlet 的配置

但在实际开发中，我们不会直接去实现 Servlet 接口，因为那样需要覆盖的方法太多，我们一般创建类继承 HttpServlet

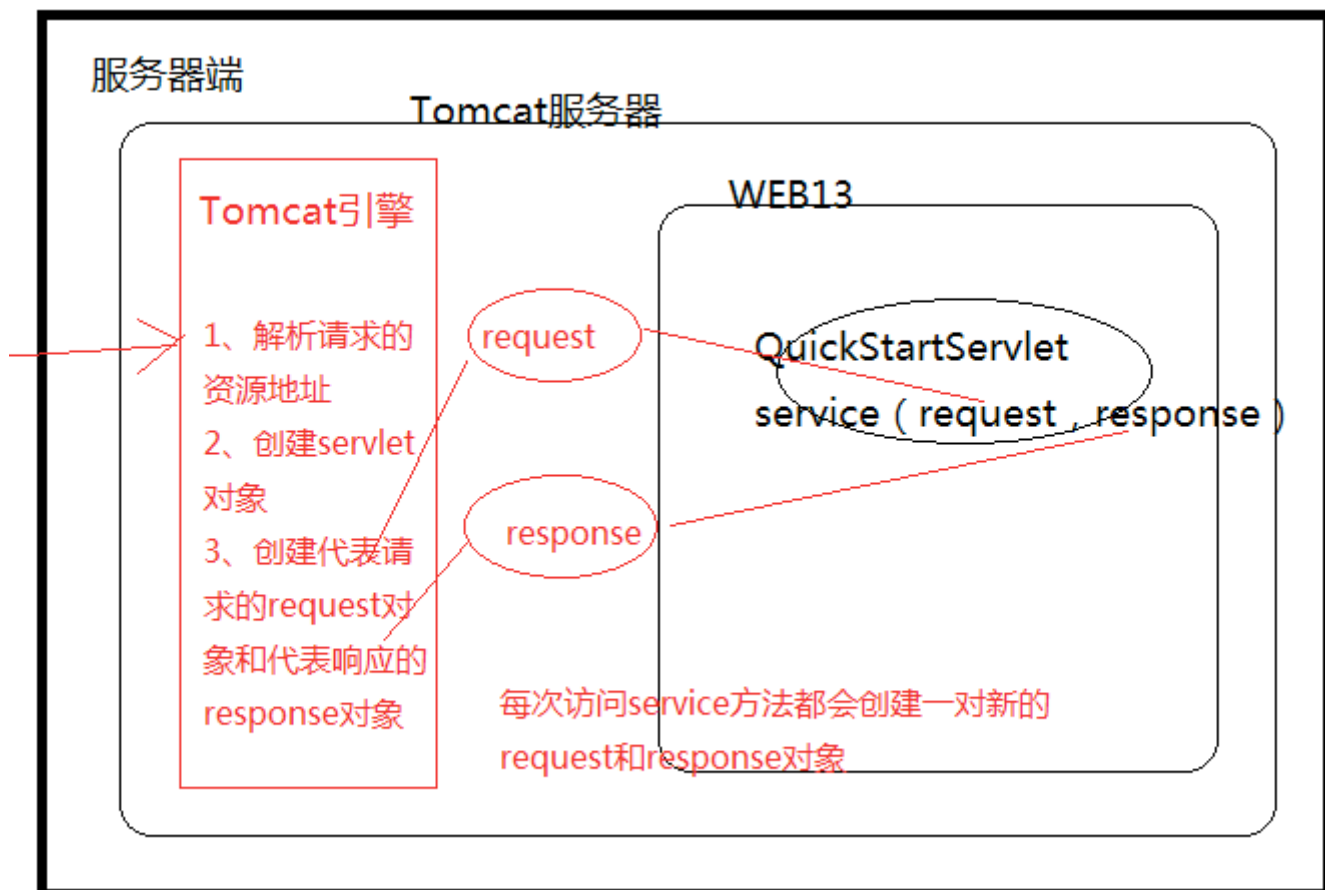
实现步骤：

- 1) 创建类继承 HttpServlet 类
- 2) 覆盖 doGet 和 doPost



3) 在 web.xml 中进行 servlet 的配置

画图描述整个访问过程：



二、Servlet 的 API (生命周期)

(1) Servlet 接口中的方法

1) init(ServletConfig config)

何时执行：servlet 对象创建的时候执行

ServletConfig ： 代表的是该 servlet 对象的配置信息

2) service (ServletRequest request,ServletResponse response)

何时执行：每次请求都会执行

ServletRequest ：代表请求 认为 ServletRequest 内部封装的是
http 请求的信息

ServletResponse ：代表响应 认为要封装的是响应的信息

3) destroy()

何时执行：servlet 销毁的时候执行

(2) HttpServlet 类的方法

1) init()

2) doGet(HttpServletRequest request,HttpServletResponse response)

3) doPost(HttpServletRequest request,HttpServletResponse response)

4) destroy()

(3) Servlet 的生命周期 (面试题)

1) Servlet 何时创建

默认第一次访问 servlet 时创建该对象

2) Servlet 何时销毁

服务器关闭 servlet 就销毁了

3) 每次访问必然执行的方法

service(ServletRequest req, ServletResponse res)方法



问题：对 XXXServlet 进行了 10 次访问，init()，destory()，service()，doGet()，doPost() 一共执行多少次？request 对象创建几个？response 创建几个？

三、Servlet 的配置

1. 基本配置

```
<!-- servlet的类的配置 -->
<servlet>
    <servlet-name>abc</servlet-name>
    <servlet-class>com.ithema.servlet.QuickStratServlet</se
    <init-param>
        <param-name>url</param-name>
        <param-value>jdbc:mysql:///mydb</param-value>
    </init-param>
</servlet>
<!-- servlet的虚拟路径的配置 -->
<servlet-mapping>
    <servlet-name>abc</servlet-name>
    <url-pattern>/quickStratServlet</url-pattern>
</servlet-mapping>
```

其中 url-pattern 的配置方式：

1) 完全匹配 访问的资源与配置的资源完全相同才能访问到

```
<url-pattern>/quickStratServlet</url-pattern>
```

2) 目录匹配 格式：/虚拟的目录../ * 代表任意

```
<url-pattern>/aaa/bbb/ccc/*</url-pattern>
```

3) 扩展名匹配 格式：*.扩展名

```
<url-pattern>*.abcd</url-pattern>
```

注意：第二种与第三种不要混用 /aaa/bbb/*.abcd (错误的)

2. 服务器启动实例化 Servlet 配置

Servlet 的何时创建：**默认**第一次访问时创建

为什么是默认？

当在servlet的配置时 加上一个配置 <load-on-startup> servlet对象在服务器启动时就创建

3. 缺省 Servlet

可以将 url-pattern 配置一个/，代表该 servlet 是缺省的 servlet

什么是缺省的 servlet？

当你访问资源地址所有的 servlet 都不匹配时，缺省的 servlet 负责处理

其实，web 应用中所有的资源的响应都是 servlet 负责，包括静态资源

4. 欢迎页面

```
<welcome-file-list>
  <welcome-file>1.html</welcome-file>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
```

四、ServletContext 对象

1. 什么是 ServletContext 对象

ServletContext 代表是一个 web 应用的环境（上下文）对象，ServletContext 对象内部封装是该 web 应用的信息，ServletContext 对象一个 web 应用只有一个

问题：一个 web 应用有几个 servlet 对象？----多个

ServletContext 对象的生命周期？

创建：该 web 应用被加载（服务器启动或发布 web 应用（前提，服务器启动状态））

销毁：web 应用被卸载（服务器关闭，移除该 web 应用）

2. 怎样获得 ServletContext 对象

1) ServletContext servletContext = config.getServletContext();

2) ServletContext servletContext = this.getServletContext();

3. ServletContext 的作用

(1) 获得 web 应用全局的初始化参数

web.xml 中配置初始化参数

```
<!-- 配置全局的初始化参数 -->
<context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
</context-param>
```

通过 context 对象获得参数

```
//获得ServletContext对象
ServletContext context = getServletContext();
//1、获得初始化参数
String initParameter = context.getInitParameter("driver");
System.out.println(initParameter);
```

(2) 获得 web 应用中任何资源的绝对路径 (**重要 重要 重要**)

方法：String path = context.getRealPath(相对于该 web 应用的相对地址);

(3) ServletContext 是一个域对象 (**重要 重要 重要**)

什么是域对象？什么是域？

存储数据的区域就是域对象

ServletContext 域对象的作用范围：整个 web 应（所有的 web 资源都可以随意向 servletcontext 域中存取数据，数据可以共享）

域对象的通用的方法：

setAttribute(String name, Object obj);

getAttribute(String name);

removeAttribute(String name);