

传智播客
《JavaWeb 程序开发入门》
教学设计

课程名称： JavaWeb 程序开发入门

授课年级： 2014 年级

授课学期： 2014 学年第一学期

教师姓名： 某某老师

2014 年 09 月 09 日

课题名称	第4章 Servlet 技术	计划学时	6 课时
内容分析	随着 Web 应用业务需求的增多，动态 Web 资源的开发变得越来越重要，为此 Sun 公司提供了 Servlet 技术。在 Servlet 技术中，有许多知识点，本课程将针对 Servlet 的生命周期、HttpServlet、Servlet 虚拟路径的映射进行详细讲解，并结合程序学习如何使用 Servlet 技术开发 web 应用		
教学目标及基本要求	要求学生熟悉 ServletConfig 和 ServletContext 接口的用法，熟练使用 Eclipse 工具开发 Servlet 程序，掌握 Servlet 的生命周期、HttpServlet 类中的方法和 Servlet 虚拟路径映射的配置		
重点及措施	教学重点：Servlet 的生命周期、HttpServlet 类、Servlet 虚拟路径的映射等		
难点及措施	教学难点：Servlet 生命周期中方法的调用时机和次数、自定义 Servlet 实现 HttpServlet 需要重写 doGet() 和 doPost() 方法、Servlet 虚拟路径的映射。		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	<p style="text-align: center;">第一课时 (Servlet 的概念、实现第一个 Servlet 程序)</p> <p>🚩 Servlet 的概念</p> <p>✧ Servlet 的产生</p> <p>随着 Web 应用业务需求的增多，动态 Web 资源的开发变得越来越重要。很多公司都提供了开发动态 Web 资源的相关技术，SUN 公司基于 Java 的动态 Web 资源开发技术，推出了 Servlet 和 JSP 两种技术。</p> <p>✧ 什么是 Servlet</p> <p>Servlet (Server Applet)，全称 Java Servlet。是用 Java 编写的服务器端程序，其主要功能在于交互式地浏览和修改数据，生成动态 Web 内容。</p> <p>✧ Servlet 接口和 API</p> <p>针对 Servlet 技术的开发，SUN 公司提供了一系列接口和类，其中最重要的是 javax.servlet.Servlet 接口。在 Servlet 接口中，定义了 5 个抽象方法，其中 init()、service()、destroy() 方法是 Servlet 生命周期方法，getServletInfo() 方法用于返回 Servlet 的相关信息。getServletConfig() 方法用于返回 ServletConfig 对象，该对象包含 Servlet 的初始化信息。</p> <p>🚩 实现第一个 Servlet 程序</p> <p>第一步：编写一个 HelloWorldServlet 类，使其继承 GenericServlet 类。实现 service() 方法，在 service() 方法中，获取 PrintWriter 对象向页面输出“Hello World!”；</p> <p>第二步：将 Tomcat 目录下的 servlet-api.jar 包所在路径添加到 classpath 中，使用 javac 编译 HelloWorldServlet.java，生成 HelloWorldServlet.class。</p> <p>第三步：在 Tomcat/webapps/ 目录下创建一个工程 chapter04，然后在 chapter04 目录下创建/WEB-INF/classes 目录，将 HelloWorldServlet.class 拷贝到 classes 目录下。</p>		

第四步：在 WEB-INF 目录下，创建一个 web.xml 文件，其编写方式可以参考 Tomcat/webapps/examples/WEB-INF 子目录下的 web.xml 文件。

第五步：启动 Tomcat 服务器，访问相应的 url，页面刷新返回“Hello World!”。

第二课时

(Servlet 的生命周期)

🚩 Servlet 的生命周期

在 Java 中，任何对象都有生命周期，Servlet 也不例外。按照功能的不同，可以分为初始化阶段、运行阶段、销毁阶段。

◇ 初始化阶段

当第一次请求访问 Servlet 程序时，Servlet 容器会解析请求，创建 Servlet 实例对象，并调用 init() 方法实现 Servlet 初始化工作。但是，init() 方法在 Servlet 整个生命周期内，只会被调用一次。

◇ 运行阶段

在 Servlet 的整个生命周期内，对于 Servlet 的每一次访问请求，Servlet 容器都会调用一次 Servlet 的 service() 方法，并且创建新的 ServletRequest 和 ServletResponse 对象，也就是说，service() 方法在 Servlet 的整个生命周期中会被调用多次。这个阶段是 Servlet 生命周期中最重要的阶段。

◇ 销毁阶段

当 Servlet 对象从 Servlet 容器中移除时，Servlet 容器会调用 Servlet 的 destroy() 方法，以便让 Servlet 对象释放它所占用的资源。在 Servlet 的整个生命周期中，destroy() 方法也只被调用一次。

第三课时

(HttpServletRequest、使用 Eclipse 开发 Servlet、Servlet 虚拟路径映射)

🚩 HttpServletRequest

◇ 什么是 HttpServletRequest

在 Servlet 接口中，提供了一个抽象类 javax.servlet.http.HttpServletRequest，它是 GenericServlet 的子类，专门用于创建应用 HTTP 协议的 Servlet。

◇ HttpServletRequest 的作用

通过分析 HttpServletRequest 的源代码片段，发现 HttpServletRequest 主要有两大功能，第一是根据用户请求方式的不同，定义相应的 doXxx() 方法处理用户请求；第二是通过 service() 方法将 HTTP 请求和响应分别转为 HttpServletRequest 和 HttpServletResponse 类型的对象。

◇ 总结

由于 HttpServletRequest 类在重写的 service() 方法时，为每一种 HTTP 请求方式都定义了对应的 doXxx() 方法，因此，当定义的类继承 HttpServletRequest 后，只需根据请求方式，重写对应的 doXxx() 方法即可，而不需要重写 service() 方法。

🚩 使用 Eclipse 工具开发 Servlet

◇ 为什么使用 Eclipse 开发 Servlet

当没有使用开发工具时，开发一个 Servlet 程序，需要导 jar 包、编译、部署、修改 web.xml 文件等步骤，是相当的繁琐的。而使用 Eclipse 开发不仅会自动编译 Servlet、自动创建 web.xml 文件，还会为 Servlet 虚拟路径的映射大大

提高了 Servlet 程序的开发，所以在实际开发中，通常都会使用 Eclipse 工具完成 Servlet 的开发。

✧ 使用 Eclipse 工具开发 Servlet

第一步：新建 web 工程 chapter04

第二步：创建 TestServlet 程序，在程序中添加一些代码，具体如下：

```
protected void doGet(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    this.doPost(request, response);
}

protected void doPost(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.print("this servlet is created by eclipse");
    out.close();
}
```

第三步：部署和访问 Servlet，在 Eclipse 中安装 Tomcat 服务器插件，将 chapter04 工程部署到 tomcat 服务器中，并启动服务器。打开浏览器，访问 <http://localhost:8080/chapter04/TestServlet>，观察页面显示结果。

🚩 Servlet 虚拟路径映射

✧ 虚拟路径

在 web.xml 文件中，一个 <servlet-mapping> 元素用于映射一个 Servlet 的对外访问路径，该路径也被称为虚拟路径。浏览器是通过访问虚拟路径来访问 Servlet 的。

✧ Servlet 的多重映射

Servlet 的多重映射指的是同一个 Servlet 可以被映射成多个虚拟路径，这种情况下客户端可以通过多个虚拟路径实现对 Servlet 的访问。

Servlet 多重映射的实现方式有两种，具体如下：

第一种方式：

```
<servlet>
    <servlet-name>TestServlet01</servlet-name>
    <servlet-class>
        cn.itcast.chapter04.servlet.TestServlet01
    </servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name> TestServlet01</servlet-name>
    <url-pattern>/TestServlet01</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>TestServlet01</servlet-name>
    <url-pattern>/Test01</url-pattern>
```

```
</servlet-mapping>
```

第二种方式:

```
<servlet>
    <servlet-name>TestServlet01</servlet-name>
    <servlet-class>
        cn.itcast.chapter04.servlet.TestServlet01
    </servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>TestServlet01</servlet-name>
    <url-pattern>/TestServlet01</url-pattern>
    <url-pattern>/Test02</url-pattern>
</servlet-mapping>
```

◇ Servlet 映射路径中使用通配符

有时候，我们希望某个目录下的所有路径都可以访问同一个 Servlet，这时，可以在 Servlet 映射的路径中使用通配符“*”。通配符的格式只有两种，具体如下：

- 1) 格式为“*.扩展名”，例如“*.do”匹配以“.do”结尾的所有 URL 地址。
- 2) 格式为“/*”，例如“/abc/*”匹配以“/abc”开始的所有 URL 地址。

需要注意的是，这两种通配符的格式不能混合使用，并且当客户端访问一个 Servlet 时，如果请求的 URL 地址能够匹配多个虚拟路径，那么，Tomcat 将采取最具体匹配原则查找与请求 URL 最接近的虚拟映射路径。

◇ 缺省 Servlet

如果在 web.xml 中某个 Servlet 的映射路径仅仅是一个正斜线 (/)，那么这个 Servlet 就是当前 Web 应用的缺省 Servlet。

第四课时

(ServletConfig、ServletContext)

🚩 ServletConfig

◇ ServletConfig 的产生

Servlet 容器初始化一个 servlet 对象时，会为这个 servlet 对象创建一个 servletConfig 对象。servletConfig 对象中包含了 servlet 的初始化参数信息。

◇ ServletConfig 的特性

Servlet 容器在调用 servlet 对象的 init(ServletConfig config)方法时，会把 servletConfig 对象当做参数传递给 servlet 对象。Init(ServletConfig config)方法会使得当前 servlet 对象与 servletConfig 对象建立关联关系。

◇ ServletConfig 接口中方法的作用

- 1、根据指定参数名获取匹配的初始化参数值。
- 2、获取一个包含所有初始化参数对象 Enumeration。
- 3、获取 Servlet 上下文对象。
- 4、获取 Servlet 名称的信息。

◇ ServletConfig 的获取方式

ServletConfig 对象可以通过 HttpSession.getServletConfig()方法获取。

🚩 ServletContext

◇ ServletContext 的产生

当 Servlet 容器启动时，会为每个 Web 应用创建一个唯一的 ServletContext 对象。

◇ ServletContext 的特性

ServletContext 是 servlet 与 servlet 容器之间的直接通信的接口，每个 web 应用有唯一的 servletContext 对象。同一个 web 应用的所有 servlet 对象共享一个 servletContext，servlet 对象可以通过它来访问容器中的各种资源。

◇ ServletContext 接口中方法的作用

- 1、获取 Web 应用程序的初始化参数
- 2、实现多个 Servlet 对象共享数据
- 3、读取 Web 应用下的资源文件

◇ ServletContext 的获取方式

ServletContext 对象可以通过 HttpSession.getServletContext() 方法和 javax.servlet.ServletConfig.getServletContext() 方法获取。

第五课时

(总结，上机练习)

- 1、总结本章内容
- 2、通过题库发放相关测试题，检查学生掌握情况。
- 3、上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为 Servlet 接口）

请按照以下要求设计一个 Servlet01 类。

要求如下：

- 1) 创建 javaweb 工程 chapter04，创建包路径 cn.itcast.servlet。
- 2) 自定义 Servlet01 类继承 GenericServlet 类，实现 service() 方法，在 service() 方法中获取输出流对象，并输出一段话到页面。
- 3) 配置 web.xml 信息，在 tomcat 服务器中部署 chapter04 工程。
- 4) 启动服务，使用浏览器访问 Servlet01 映射的 url 地址。

第六课时

(上机练习)

上机一：（考察知识点为 Servlet 生命周期）

请按照以下要求修改上机一中的 Servlet01 类。

要求如下：

- 1) 定义一个 HelloWorldServlet 类，其中定义了 init()、destroy() 方法。
- 2) 在 service()、init()、destroy() 三个方法中分别添加输出语句。
- 3) 配置 web.xml 信息，在 tomcat 服务器中部署 chapter04 工程。
- 4) 启动 Tomcat 服务器，使用浏览器访问 Servlet01 映射的 url 地址，观察控制台输出内容。
- 5) 关闭服务，观察控制台输出内容。

	<p>上机二：（考察知识点为 <code>HttpServlet</code> 的 <code>doGet</code> 和 <code>doPost</code> 方法）</p> <p>请按照以下要求设计一个 <code>Servlet02</code> 类。</p> <p>要求如下：</p> <ol style="list-style-type: none"> 1) 在上机一中的包路径下创建 <code>Servlet02</code> 类，继承 <code>HttpServlet</code> 类。重写其 <code>doGet()</code>和 <code>doPost()</code>方法。 2) 在 <code>chapter04</code> 工程目录下中，创建一个 <code>login</code> 登陆表单页面，采用 <code>post</code> 方式提交给 <code>Servlet02</code> 处理。 3) 配置 <code>web.xml</code> 信息，在 <code>tomcat</code> 服务器中部署 <code>chapter04</code> 工程。 4) 启动服务，使用浏览器访问 <code>Servlet02</code> 映射的 <code>url</code> 地址。 5) 接着，使用浏览器访问 <code>login</code> 页面，提交表单。
<p>思考题 和习题</p>	<p>见教材配套的习题</p>
<p>教 学 后 记</p>	