**doGet和doPost**

|  |
| --- |
| 首先你要明确servlet的生命周期和HTTP协议.  Serlvet接口只定义了一个服务方法就是service，而HttpServlet类实现 了该方法并且要求调用下列的方法之一：  **doGet**：处理GET请求  doPost：处理POST请求  当发出客户端请求的时候，调用service 方法并传递一个请求和响应对象。Servlet首先判断该请求是GET 操作还是POST 操作。然后它调用下面的一个方法：**doGet** 或 doPost。如果请求是GET就调用**doGet**方法，如果请求是POST就调用doPost方法。**doGet**和doPost都接受请求(HttpServletRequest)和响应(HttpServletResponse)。  get和post这是http协议的两种方法，另外还有head, delete等  这两种方法有本质的区别，get只有一个流，参数附加在url后，大小个数有严格限制且只能是字符串。post的参数是通过另外的流传递的，不通过url，所以可以很大，也可以传递二进制数据，如文件的上传。  在servlet开发中，以**doGet**()和doPost()分别处理get和post方法。  另外还有一个doService(), 它是一个调度方法，当一个请求发生时，首先执行doService(),不管是get还是post。在HttpServlet这个基类中实现了一个角度，首先判断是请求时get还是post,如果是get就调用**doGet**(), 如果是post就调用doPost()。你也可以直接过载doService()方法，这样你可以不管是get还是post。都会执行这个方法。   service()是在javax.servlet.Servlet接口中定义的, 在 javax.servlet.GenericServlet 中实现了这个接口, 而 **doGet**/doPost 则是在 javax.servlet.http.HttpServlet 中实现的, javax.servlet.http.HttpServlet 是 javax.servlet.GenericServlet 的子类. 所有可以这样理解, 其实所有的请求均首先由 service() 进行处理, 而在 javax.servlet.http.HttpServlet 的 service() 方法中, 主要做的事情就是判断请求类型是 Get 还是 Post, 然后调用对应的 **doGet**/doPost 执行.   **doGet**：处理GET请求 doPost：处理POST请求 doPut：处理PUT请求 doDelete：处理DELETE请求 doHead：处理HEAD请求 doOptions：处理OPTIONS请求 doTrace：处理TRACE请求 通常情况下，在开发基于HTTP的servlet时，开发者只需要关心**doGet**和doPost方法，其它的方法需要开发者非常的熟悉HTTP编程，因此这些方法被认为是高级方法。 而通常情况下，我们实现的servlet都是从HttpServlet扩展而来。 doPut和doDelete方法允许开发者支持HTTP/1.1的对应特性； doHead是一个已经实现的方法，它将执行**doGet**但是仅仅向客户端返回**doGet**应该向客户端返回的头部的内容； doOptions方法自动的返回servlet所直接支持的HTTP方法信息； doTrace方法返回TRACE请求中的所有头部信息。 对于那些仅仅支持HTTP/1.0的容器而言，只有**doGet**, doHead 和 doPost方法被使用，因为HTTP/1  service()是在javax.servlet.Servlet接口中定义的, 在 javax.servlet.GenericServlet 中实现了这个接口, 而 **doGet**/doPost 则是在 javax.servlet.http.HttpServlet 中实现的, javax.servlet.http.HttpServlet 是 javax.servlet.GenericServlet 的子类. 所有可以这样理解, 其实所有的请求均首先由 service() 进行处理, 而在 javax.servlet.http.HttpServlet 的 service() 方法中, 主要做的事情就是判断请求类型是 Get 还是 Post, 然后调用对应的 **doGet**/doPost 执行,**doGet**在地址栏中显示请求的内容，doPost隐藏. 其时说来很简单，在servlet中doPost方法里还是调用了**doGet**方法，所以在创建servlet时可以不要doPost方法，但在做大型项目涉及密码的传送时doPost方法会更安全些，通常情况下二者没什么区别。  继一下:  下面主要介绍javax.servlet.http提供的HTTP Servlet应用编程接口。   HTTP Servlet 使用一个 HTML 表格来发送和接收数据。要创建一个 HTTP Servlet，请扩展 HttpServlet 类，该类是用专门的方法来处理 HTML 表格的 GenericServlet 的一个子类。 HTML 表单是由和标记定义的。表单中典型地包含输入字段(如文本输入字段、复选框、单选按钮和选择列表)和用于提交数据的按钮。当提交信息时，它们还指定服务器应执行哪一个Servlet(或其它的程序)。 HttpServlet 类包含 init()、destroy()、service() 等方法。其中 init() 和 destroy() 方法是继承的。   (1) init() 方法   在 Servlet 的生命期中，仅执行一次 init() 方法。它是在服务器装入 Servlet 时执行的。 可以配置服务器，以在启动服务器或客户机首次访问 Servlet 时装入 Servlet。 无论有多少客户机访问 Servlet，都不会重复执行 init() 。   缺省的 init() 方法通常是符合要求的，但也可以用定制 init() 方法来覆盖它，典型的是管理服务器端资源。 例如，可能编写一个定制 init() 来只用于一次装入 GIF 图像，改进 Servlet 返回 GIF 图像和含有多个客户机请求的性能。另一个示例是初始化数据库连接。缺省的 init() 方法设置了 Servlet 的初始化参数，并用它的 ServletConfig 对象参数来启动配置， 因此所有覆盖 init() 方法的 Servlet 应调用 super.init() 以确保仍然执行这些任务。在调用 service() 方法之前，应确保已完成了 init() 方法。   (2) service() 方法   ervice() 方法是 Servlet 的核心。每当一个客户请求一个HttpServlet 对象，该对象的service() 方法就要被调用，而且传递给这个方法一个"请求"(ServletRequest)对象和一个"响应"(ServletResponse)对象作为参数。在 HttpServlet 中已存在 service() 方法。缺省的服务功能是调用与 HTTP 请求的方法相应的 do 功能。例如， 如果 HTTP 请求方法为 GET，则缺省情况下就调用 **doGet**() 。Servlet 应该为 Servlet 支持的 HTTP 方法覆盖 do 功能。因为 HttpServlet.service() 方法会检查请求方法是否调用了适当的处理方法，不必要覆盖 service() 方法。只需覆盖相应的 do 方法就可以了。   当一个客户通过HTML 表单发出一个HTTP POST请求时，doPost()方法被调用。与POST请求相关的参数作为一个单独的HTTP 请求从浏览器发送到服务器。当需要修改服务器端的数据时，应该使用doPost()方法。   当一个客户通过HTML 表单发出一个HTTP GET请求或直接请求一个URL时，**doGet**()方法被调用。与GET请求相关的参数添加到URL的后面，并与这个请求一起发送。当不会修改服务器端的数据时，应该使用**doGet**()方法。   Servlet的响应可以是下列几种类型：   一个输出流，浏览器根据它的内容类型(如text/HTML)进行解释。   一个HTTP错误响应, 重定向到另一个URL、servlet、JSP。   (3) destroy() 方法   destroy() 方法仅执行一次，即在服务器停止且卸装Servlet 时执行该方法。典型的，将 Servlet 作为服务器进程的一部分来关闭。缺省的 destroy() 方法通常是符合要求的，但也可以覆盖它，典型的是管理服务器端资源。例如，如果 Servlet 在运行时会累计统计数据，则可以编写一个 destroy() 方法，该方法用于在未装入 Servlet 时将统计数字保存在文件中。另一个示例是关闭数据库连接。   当服务器卸装 Servlet 时，将在所有 service() 方法调用完成后，或在指定的时间间隔过后调用 destroy() 方法。一个Servlet 在运行service() 方法时可能会产生其它的线程，因此请确认在调用 destroy() 方法时，这些线程已终止或完成。   (4) GetServletConfig()方法   GetServletConfig()方法返回一个 ServletConfig 对象，该对象用来返回初始化参数和ServletContext。ServletContext 接口提供有关servlet 的环境信息。   (5) GetServletInfo()方法   GetServletInfo()方法是一个可选的方法，它提供有关servlet 的信息，如作者、版本、版权。   当服务器调用sevlet 的Service()、**doGet**()和doPost()这三个方法时，均需要 "请求"和"响应"对象作为参数。"请求"对象提供有关请求的信息，而"响应"对象提供了一个将响应信息返回给浏览器的一个通信途径。   javax.servlet 软件包中的相关类为ServletResponse和ServletRequest，而javax.servlet.http 软件包中的相关类为HttpServletRequest 和 HttpServletResponse。   Servlet 通过这些对象与服务器通信并最终与客户机通信。Servlet 能通过调用"请求"对象的方法获知客户机环境，服务器环境的信息和所有由客户机提供的信息。Servlet 可以调用"响应"对象的方法发送响应，该响应是准备发回客户机的。   <!-- Standard Action Servlet Configuration (with debugging) -->   <servlet>  <servlet-name>log4j-init</servlet-name>  <servlet-class>com.neusoft.ehr.common.Log4jInit</servlet-class>  <init-param>  <param-name>log4j-init-file</param-name>  <param-value>/WEB-INF/classes/property/log4j.properties</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet> |