Servlet生命周期

1. 构造方法 ： 创建servlet对象时调用，默认首次访问servlet时创建servlet对象，只调用一次
2. Init方法：创建servlet对象时调用
3. Service方法：每次发出请求时调用
4. Destory方法：销毁servlet对象调用

Servlet自动加载

首次创建servlet对象会比较慢，可以在web.xml文件添加 <load-on-startup>1</load-on-startup>来自动加载servlet对象。里面的数字越大，优先级越低

Init方法分为有参数的和无参数的，其中有参数的是专供tomcat服务器调用的，如果需要自己编写初始化方法，重写无参数方法即可

Servlet对象时单实例多线程的，所以当servlet中service方法访问同一个成员变量时会产生线程安全问题，及类似买到同一张票，所以需要使用线程锁，尽量不要使用成员变量，如果不得已使用的话，要尽量缩小同步代码块的范围，避免因同步导致的并发效率降低

Ep：

Public void doGet(HttpServletRequrst request , HttpServletResponse response){

Throws ServletException , IOException{

synchronize(ServletDemo.class){ //锁对象建议使用servlet本身的字节码对象

、、、可能出现线程安全问题的代码块

}

}

}

Servlet学习

HttpServletRequest 请求对象，获取请求信息

HttpServletResponse 响应对象，设置响应对象

ServletConfig servlet配置对象

ServletContext servlet上下文对象

ServletConfig对象 ： 用于加载servlet的初始化参数，一个web应用中可以存在多个ServletConfig对象（一个servlet对应一个servletConfig对象）

Servlet配置对象

方法

getInitParameter(java.lang.string name) 根据参数名获取参数值

getInitParameterNames() 获取所有参数

getServletContext() 得到servlet上下文对象

GetServletName() 得到servlet名称

配置格式

<init-param>

<param-name>path</param-name>

<param-value>c:/a.txt</param-value>

</init-param>

如何获取servletConfig对象，在继承HTTPServlet类中直接使用this.getServletConfig()即可

ServletContext对象 ： servlet上下文对象，表示一个当前的web应用环境，一个web应用只有一个ServletContext对象

创建和得到对象

创建时机 ： 加载web应用时创建该对象

得到对象 ： 从ServletConfig的getServletContext()方法得到

方法

getContextPath() 得到当前web的应用路径

getInitParameter(name) 得到web应用的初始化参数

getInitParameterNames()

setAttribute() 域对象有关的方法

getAttribute

removeAttribute

getRequestDispather(path) 转发

getRealPath(path) 得到web应用的资源文件

getResponseAsStream(path)