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

配置一个servlet映射路径

//先加一个servlet

<servlet>

//一个servlet名称，用于和servlet-mapping匹配

<servlet-name></servlet-name>

//servlet映射类 包名+完整类名

<servlet-class></servlet-class>

</servlet>

//再加一个servlet-mapping

<servlet-mapping>

<servlet-name></servlet-name>

//用于浏览器访问路径

<url-pattern></url-pattern>

</servlet-mapping>

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,写在servlet配置里，属于局部参数

<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

获取方法

ServletContext context = this.getServletContext();

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

web的应用路径就是部署到tomcat服务器上运行的web应用名称

如何修改web的应用路径

右键项目名称->proterties->MyEclipse->Web-> 修改web Content-root

getInitParameter(name) 得到web应用的初始化参数,根据名称获取值

getInitParameterNames() 得到web应用的所有初始化参数

配置web应用参数 ， 写在web-app第一层，servlet之外，属于全局参数

<context-param>

<param-name></param-name>

<param-value></param-value>

</context-param>

域对象：作用是保存和获取数据，可以再不同的动态资源之间共享数据，

域对象有4种

HttpServletRequest

ServletContext

HttpSession

PageContext

域对象有关的方法

setAttribute(name , object) 保存数据

getAttribute(name) 获取数据

removeAttribute(name) 删除数据

getRequestDispather(path) 转发 , 转发只能转发到当前项目以内的资源

完整的转发写法

ServletContext context = this.getServletContext();

RequestDispatcher rd = context.getRequestDispatcher(path);

rd.forward(request , response);

在javaweb项目中， ./ 代表tomcat/bin目录。所以在web项目中一般不使用这种相对路径写法

以下是web项目获取路径的方法，path是web应用的相对路径，src中的资源文件会编译复制到web目录下的WEB-INF/classes/下

getRealPath(path) 得到web应用的资源文件

getResponseAsStream(path)