|  |
| --- |
| Servlet 课程:  Servlet(Server applet)是Java Servlet的简称，称为小服务程序或服务的连接器，用JAVA编写服务器程序。主要用于交互式地浏览和修改数据，生成动态的web内容.  Servlet 是JAVAEE的一种规范.  服务器软件:tomcat/jboss,weblogic等服务器  Tomcat 软件可以帮我们对接http协议的请求(做些通用的处理),然后将请求转发到我们的servlet进行处理，返回客户端。  http协议  tcp/ip  Servlet是JAVA语言实现的接口，-----实现servlet接口。  mysql  Tomcat服务器  IE浏览器  Servlet 处理  响应 |
| 定义:  Servlet是一种服务器端的JAVA应用程序，具有独立平台和协议的特性,可以生成动态的web页面。  担当客户请求，与服务器响应的中间件  Tomcat 软件 9.x/8.x/7.x/6.x  目录结构:  Bin目录 tomcat 应用程序 启动应用程序 startup.bat批处理文件启动  Logs 目录: 记录tomcat软件日志信息.  Conf 目录: 配置服务器的配置相关的信息的.(端口，发布站点)  Libs目录 tomcat下的相关的资源jar包  Temp临时文件  Webapps目录 部署项目  Work 工作目录 .class文件  Tomcat集成eclipse开发工具:  集成之后真正使用的tomcat位置  F:\iotek46\.metadata\.plugins\org.eclipse.wst.server.core\tmp0  Servlet 的创建方式:--servlet本质就是一个类(多线程),生成一个动态的网页.  支持http超文本协议:   1. 继承Servlet 接口   操作流程:  <1>新建一个Servlet类继承Servlet接口  <2>实现servlet 里service()方法  <3>配置web.xml文件，配置servlet  <4>部署（发布）项目加载tomcat wtpwebapps目录下.  <5>最后浏览器测试…   1. 获取表单的信息: 2. 继承HttpServlet   GET请求 (1) form 的属性method=”get”  (2)直接从地址请求  (3)a href=”test.do”  Post请求 form method=”post”   1. servlet的生命周期 2. 第一次请求的时候，servlet实例化 3. init方法:用于servlet对象初始化信息的. 只执行一次 4. doget循环执行，（不停地执行的,请求）/服务 5. destory销毁执行. 服务器停止.   备注: servlet 一旦实例化，驻留在内存，(服务器停止才会释放内存.) |
| Servlet API   1. Request 对象: (ServletRequest--HttpServletRequest) :获取请求信息 2. Config对象—ServletConfig: 获取servlet 初始化的参数信息.   servletConfigA  ServletA 服务名  ServletConfigB  ServletB 服务名  每一个servlet中的config对象只能访问自身的servlet的初始参数信息. |
| 1. application 简称---ServletContext对象(全局对象---上下文对象)   只要服务器不停止,此对象一直存在的.  服务器  ServletContext对象  ServletB  ServletC  ServletA   1. session(HttpSession):会话---用于保存客户端与服务器端的交互信息.   Tomcat 服务器  Session.do  Session2对象  820ED58580A2C3FC59C921D3EA2D52FA  Session1对象  6DFC71DB0DD4324AFB590FA9AE071FDC  Session3对象  浏览器  浏览器  浏览器 火狐  Session对象的生命周期(属于服务器端---内存)  一般用于保存少量的信息(用户登录的用户信息)  生命周期设置:  1)web.xml配置文件设置:默认是30分钟  <session-config>  <session-timeout>30</session-timeout>  </session-config>  2)硬编码:session.setMaxInactiveInterval()方法. 单位是秒. 10秒   1. cookie 存放在客户端的对象(浏览器)，不占用服务器资源，存放在浏览器。   不安全，  Cookie浏览器可以设置禁用  服务器  Servlet ---cookie创建对象  response  浏览器  Cookie对象的生命周期:硬编码  存/取问题   1. 七天免登录 2. 历史浏览商品记录 3. response   Dbutils 工具类:封装jdbc操作的CRUD  JSP Java Server Page是一种动态页面的技术.  JSP的构成:  Jsp=html+java  JSP的本质就是一个servlet.  JSP的特点: ---负责页面显示.   1. 内容与显示逻辑分离.   Servlet 的长处就是流程控制，JSP的长处生成显示动态页面。 |
| JSP页面的脚本元素:  三种类型  1)表达式 <%=%>  2)script <%%>  3)声明: <%! %>  类的成员:  属性,方法,内部类  Class A{  属性  方法  内部类  }  JSP容器:  把JSP页面生成源代码.java文件—编译.class  JSP执行的流程:  执行servlet  \_jspService()  服务器  用户请求 |
| 指令是发送给JSP编译器的信息，告诉JSP编译器如何处理JSP页面，它不直接生成输出  JSP的三种类型的指令:   1. page指令   语法: <%@page {attribute=value}%>  language="java" 在页面中嵌入的脚本的语言.  contentType="text/html; charset=UTF-8"  response.setContentType("text/html; charset=UTF-8");    pageEncoding="UTF-8" JSP页面的字符编码  session="true" 是否参与会话 默认值是true  errorPage="error.jsp" 页面发生异常，跳转到error.jsp  isErrorPage="true" 开启exception隐式对象 ,默认是false  buffer="8kb" 页面初始输出的JspWriter处理内容的输出缓冲区的大小。  autoFlush="true" 缓存内容是否被自动刷新.  isELIgnored="false" 会运算EL表达式 true，不会当作EL表达式处理.   1. include指令 :包含   静态include指令:  先包含进来，然后一起编译生成一个.class文件.  动态include 标签  org.apache.jasper.runtime.JspRuntimeLibrary.include(request, response, "includeC.jsp", out, false);  先分开编译，再动态包含进来。   1. taglib指令 引用标签库   <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  .jsp的注释:  <!-- -->  <%-- --%> jsp脚本注释  forward 标签:属于服务器端跳转  跳转的二种方式:   1. 客户端跳转(重定向)   <1>URL地址是会发生改变的  <2>不能共享request对象的(二次请求,跳转了二次)  <3>支持跨域跳转   1. 服务器跳转(内部跳转)   <1> URL地址不会发生改变  <2> 共享request对象的数据 (一次请求范围有效)  <3>不支持跨域  解决乱码问题.  前提：所有的编码要统一.   1. 输出乱码   response输出乱码  response.setContentType("text/html;charset=utf-8");  response.setCharacterEncoding(“utf-8”);   1. 获取请求的参数的乱码 request 2. post提交 请求的乱码   解决方案:  request.setCharacterEncoding("utf-8");   1. get提交 ---默认是iso-8859-1   <a href=”aa.do?name=小白”></a>  解决方案一:  String user=new String(req.getParameter("username").getBytes("iso-8859-1"),"utf-8");  解决方案二:  设置tomcat 的server.xml  <Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443" URIEncoding="UTF-8"/>  Weblogic  jboss |
| Javabean动作标签:  javaBean本质就是一个类,  遵循一定的设计原则的任何JAVA类都可以是JAVABEAN组件.   1. 可序列化 2. 无参数的构造方法 3. 类的私有属性 4. 通过公共方法来获取属性和设置属性   Javabean的生命周期(作用域)  Scope属性:  Page 默认值 当前页面有效  Request 一次请求  Session 会话范围  Application 全局 |
| JSP九大内置对象（隐式对象）   1. request 2. response 3. session 4. config 5. application 6. out 对象是一个输出流，用来向客户端输出数据的 7. page 对象代表JSP本身.准确地说JSP编译之后servlet类。 8. exception 9. pageContext 为我们提供了所有JSP程序执行过程中所需要的属性及方法.   换句话说：这个对象可以获取其它八大内置对象  EL表达式语言 Expression Language  方便获取对象的数据  ${EL表达式语言}  数据类型 true/false  数值，字符串 null值  ${5+2 }  ${'5'+'2' }  ${'容'+'儿'} 错误的  EL的隐式对象  <%  pageContext.setAttribute("user", "小燕子");  request.setAttribute("user", "小白");  session.setAttribute("user","小青");  application.setAttribute("user", "德华");    %>  <%=pageContext.getAttribute("user") %>    ${pageScope.user}/${user} <!-- 默认会取作用域范围小的 -->  ${requestScope.user }/${sessionScope.user }/${applicationScope.user }  Jstl ===JSP standard Tag Library （jsp标准标签库） 重点  导入二个JAR包  Stardard.jar  Jstl.jar  拷贝web-inf/lib目录下  分层(CRUD增删改查)/ 分页  MVC===Model View Controller:分层:  Model 模型 ---javabean 表示应用程序的核心，比如数据库记录列表  View 视图 ---jsp 显示数据(数据库记录)  Controller 控制器---servlet 处理输入/输出,交互的部分.  控制用户的输入信息/并向模型发送数据.  Dao (Data ACCESS Object) 数据访问对象----与数据库进行打交道的.  Servcie层 ---服务层.业务逻辑的判断…(业务发生变化，只要改变服务层，DAO不会改变的。分工明确)  一个servlet 处理多个请求?  上传组件:fileupload组件.  作业:上传商品信息 /图片的路径保存到数据库  ${goods.imagepath}  11月5日开始预约答辩  下次课讲 监听器，过滤器--- json –ajax-jquery   1. 过滤器filter   过滤器处于请求与请求的资源之间.  服务器  Servlet  Jsp  html  request  客户端  Second之前  Second之后  First 之前  First 之后  过滤器的应用: 权限检查,过滤非法文字.乱码处理.日志等等.  过滤器：拦截一组请求资源.  1）创建过滤器:继承filter接口  <filter>  <filter-name>first</filter-name>  <filter-class>com.iotek.filters.FirstFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>first</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <url-pattern>/\*</url-pattern> 所有的资源都要通过过虑器  <url-pattern>\*.jsp</url-pattern>只拦截所有jsp页面  <url-pattern>/目录名/\*</url-pattern> 拦截某目录名下的所有资源。  <url-pattern>/servlet/\*</url-pattern>  2）Filter的生命周期:  Filter实例化  Init（）只执行一次  destroy()方法   1. 过滤链(至少二个过滤器以上，构成过滤链)   存在执行流程问题?哪个过滤器先，后执行?  与 filter-mapper 配置顺序有关.  监听器:listener  有三种监听器:   1. ServletContextListener 监听application   ServletContextAttributeListener 监听对象属性   1. HttpSessionListener 监听session 2. ServletRequestListener 监听request对象   统计访问IP的次数:  监听器:全局对象保存所有的IP与次数;  192.168.1.20 1 2  AJAX = Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。  最大优点:局部刷新  Ajax的应用:地图，搜索框,注册信息(验证检查 用户名)  Ajax的核心对象  XMLHttpRequest:异步发送请求/接收服务器返回的数据  Ajax的交互流程  服务器  servlet  DB  Js: xhr  处理:  响应  Html元素  级联 省 市 ajax+json完成 |