1. Request.getParameter有关的必考，考名字函数和函数参数V
2. public void init(ServletConfig config){}主键初始化V
3. String value = getInitParamater(“rate”)考参数，配置信息V
4. HttpServlet HttpConfig HttpContext V

HttpServletRequest HttpServletResponse ServletRequest ServletResponse

1. xml区分大小写V
2. 元素内容为空的时候与<b/>等价V
3. xml不解析空格和回车，所以内容依旧为空V
4. 考点HttpServlet（HttpServletRequest HttpServletResonse）V

service(ServletRequest,servletResponse);

service(HttpServletRequest, HttpServletResponse);

String method = request.getMethod();

if("POST".equals(method)){doPost();}

else if("GET".equals(method)){doGet(request,response);}

1. ResquestDispatcher的forward和include两个函数都考V

区别：转发/这个主键包含另外一个主键的操作输出输入等，结束之后返回到原来主键

include：

response.getWriter().println("test");

RequestDispatcher rd = request.getRequestDispatcher("/b");

rd.include(request,response);

response.getWriter().println("over");

输出结果为 text hello(/b输出的) over

forward:

把请求转发给其他主键，但是结束后不返回原来主键操作

rd.forward(request,response);

之后不应该再有response.getWriter().println();

但转发请求到那个主键不能用close函数，谁开的谁关

如果使用forward，不允许在原来转发的函数中输出内容,所以输出的是 hello，而不应该是text hello

1. HttpSession的share和data这两个函数必考V

getAttribute(String name) Object setAttribute(String name,Object 0)

1. 考getCookies()返回值是cookies()V
2. 考cookies这个类是什么，名值对V
3. get post put delete HTTP获取协议的方法，共七个，考V
4. pw.print(“\r\n”)告诉服务器header结束了V
5. web.xml必须得会V
6. 会话实现的三种办法，必考：cookie，url rewriting，hidden fieldV
7. 实现getSession，必考：Cookies[] request.Cookies()，request.getRequestURL()，request.getRequestURI()V
8. Service(){request.getSession().invalidate();}让这个Session失效V
9. 考过滤器的九个作用FilterV
10. 必考doFilter(Servlet)函数V
11. request.getRequestDispatcher(“/login.html”).forward(request,response)把请求源派发到登陆页面V
12. doFilter做法：V

public void doFilter(request,response,chain)throws ServletException,IOException {

long start = System.currentTimeMillis();//预处理，处理request对象

chain.doFilter(request,response);//如果写这个则把这个过滤器往后传

long end = System.currentTimeMillis();//后处理，处理response对象

System.out.println("cost:" + end-start);

config.getServletContext().log(""+(end-start));}

1. <?xml version?>V

<web-app>

<listener>

<listener-class>MyApplistener</listener-class>

</listener>

<filter>

<filter-name>f1</filter-name>

<filter-class>MyFilter<.filter-class>

<init-param>

<param-name>software</param-name>

<param-value> /WEB-INF/software</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>f1</filter-name>

<servlet0-name>s1</servlet-name>

<dispatcher>REQUEST</dispatcher>//必须是客户浏览器发过来的请求,默认是request，可以有多个dispatcher，但会优先匹配url-pattern，再匹配servlet-name

<dispatcher>INCLUDE</dispatcher>

<dispatcher>ERROR</dispatcher>

<url-pattern> </url-pattern>

</filter-mapping>

<filter-mapping>

</filter-mapping>

<servlet>

<servlet-name>s1</servlet-name>

<servlet-class>com.abc.servlet.MyServlet</servlet-class>

<init-param>

<param-name>abc</param-name>

<param-value>8</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name> </servlet-name>

<servlet-pattern> </servlet-pattern>

</servlet-mapping>

</web-app>

1. UTF-8 汉字是3个字节，常用的字符是1个字节，有字节顺序V

Unicode 汉字是两个字节，没有字节顺序，传输的时候可能是四个字节

如果UNICODE字符由2个字节表示，则编码成UTF-8很可能需要3个字节。而如果UNICODE字符由4个字节表示，则编码成UTF-8可能需要6个字节

1. request.setCharacterEncoding("GBK");//只对body中的内容有效，但不处理header中的数据V
2. ServletContextEvent对应ServletContextListener接口V
3. contextInitialized()应用程序初始化V
4. 部署V

<listener>

<listener-class>MyApplistener</listener-class>

</listener>

1. HttpSessionBindingListener HttpSessionAttributeListenerV
2. jar cvf hello.war -C c:\abc\hello 打包把这些文件归档V
3. Session写法V!

NumberGuessBean bean = null;

HttpSession session = request.getSession();

Object o = session.getAttribute("bean");

if(o==null) {

bean = new NumberGuessBean();

session.setAttribute("bean:,bean);

}else {

bean = (NumberGuessBean) o;

}

String guess = reguest.getParameter("guess");

if(guess!=null){

bean.setGuess(guess);

1. PageContext pageContext 重点看一下它的函数V
2. param values重点V
3. Exception exception只能在errorpage中使用V
4. <jsp:forward page=”/a.jsp/”> <jsp:include page=”/b.jsp/”>动态包含V
5. <%@ include file=/”b.jsp”/>静态包含（在转换文件的时候）V
6. <jsp:useBan id ="a" class ="A" scope="page|request|session|application/> a （1）函数局部变量名；（2）作用域关联名称V
7. <jsp:getProperty name="a" property="name" value="zs"/> name()函数，参数为zsV
8. <jsp:getProperty name="numguess" property="guess" param ="guess"/> 获取参数guess写入guess中，属性名首字母是小写V
9. PageContext的API函数重点V
10. <tei-class>MyTagExtraInfo</tei-class>考这个，在编译时运行V
11. <%@ taglib uri="/WEB\_INF/tlds/time.tld" prefix="t" %> uri填空题V
12. Void invoke(Writer out)考试会出V
13. <%@ taglib tagdir="/WEB-INF/tags" prefix="a"%> 考试考这个tagdir="/WEB-INF/tags"不用写.tldV
14. public class NumberGuessServlet extends HttpServlet {V

public void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

//考试

NumberGuessBean bean = null;

HttpSession session = request.getSession();

Object o = session.getAttribute("bean");

if(o==null) {

bean = new NumberGuessBean();

session.setAttribute("bean",bean);

}

else {

bean = (NumberGuessBean) o;

}

String guess = request.getParameter("guess");

if(!(guess==null)|| guess.length()==0) {

bean.setGuess(guess);

}

//选择视图，考试

if(bean.getSuccess()) { //success.jsp

request.getRequestDispatcher("/success.jsp").forward(request,response);

bean.reset();

}

else if(bean.getNumGuesses()==0) { //welcome.jsp

request.getRequestDispatcher("/welcome.jsp").forward(request,response);

}

else { //guess.jsp

request.getRequestDispatcher("/guess.jsp").forward(request,response);

}

}

}

1. <jsp:useBean id="bean" class"num.NumberGuessBean" scope="session" />输出bean对象 次数，考试；属性名首字母都小写propertyV
2. <http://8080/abc/xyz/a.jsp上下文地址不一定是abc，考>V

<% response.sendRedirect %> 这个与<jsp:forward>是服务器处理请求的，网页地址不会变的区别，考

1. ServletContext（难）,ServeletConfig （就两个函数）考试各考一道题，API函数V
2. public void contextInitialized(ServletContextEvent sce) {V

DataSource ds = ;

sce.getServletContext().setAttribute("ds",ds);

Connection con = ds.getConnection();//阻塞机制，会等待，考

con.close();//释放回链接池}

1. import javax.sql.\*; jdbc datasource扩展包,考试题 javax.sql.\*V
2. public void test() jdbc直连,不可取，采用链接池V
3. 考DataSource出现在配置文件里V

DataSource ds = new BasicDataSource(); //DriverManagerDataSource();

1. JSTL 考V
2. 对主键的读写操作V
3. out set 流程控制 标签V
4. if标记和when考试V
5. structs 1和structs 2的区别：V

（1）必须继承Action控制器的数据封装在 FormBean

（2）函数的参数 返回值 都比较灵活 取消了 FromBean 将ActionForm 和Action 合并在一起 都封装在Action中

1. 将所有的请求都由spring处理，包括静态的资源文件V

缺点：效率不高

优点：可以建立虚拟的路径 可以将资源放在其他的文件夹，建立路径的映射

1. 对象的创建运算都是由spring来完成的V
2. spring的容器先于servlet启动V