

J2EE 期末复习参考

往年考题解析

1. JSP out的隐含对象

| 隐含对象 | 类型 |
|-------------|---------------------|
| out | JspWriter |
| config | ServletConfig |
| page | JSP.this |
| pageContext | PageContext |
| exception | throwable |
| request | HttpServletRequest |
| response | HttpServletResponse |
| application | ServletContext |
| session | HttpSession |

2. EL（表达式语言）

记住，表达式语言中的隐含对象除了 pageContext 的类型是 PageContext 外，其余类型皆是 Map。

3. PageContext类：

PageContext 扩展了 JspContext，提供有关何时在 Servlet 环境中使用 JSP 技术的有用上下文信息。

PageContext 实例提供对与某个 JSP 页面关联的所有名称空间的访问，隐式对象是自动添加到 pageContext 的。

在PageContext类中的get对象基本都是其隐含对象，而RequestDispatcher必然不会被包含其中。
RequestDispatcher应该由Servlet对象来创建和取得。

| All Methods | Instance Methods | Abstract Methods | Concrete Methods |
|--------------------------|------------------|--|---|
| Modifier and Type | | Method and Description | |
| abstract void | | forward(String relativeUriPath) | This method is used to re-direct, or "forward" the current ServletRequest and ServletResponse to another active component in the application. |
| ErrorData | | getErrorData() | Provides convenient access to error information. |
| abstract Exception | | getException() | The current value of the exception object (an Exception). |
| abstract Object | | getPage() | The current value of the page object (In a Servlet environment, this is an instance of javax.servlet.Servlet). |
| abstract ServletRequest | | getRequest() | The current value of the request object (a ServletRequest). |
| abstract ServletResponse | | getResponse() | The current value of the response object (a ServletResponse). |
| abstract ServletConfig | | getServletConfig() | The ServletConfig instance. |
| abstract ServletContext | | getServletContext() | The ServletContext instance. |
| abstract HttpSession | | getSession() | The current value of the session object (an HttpSession). |
| abstract void | | handlePageException(Exception e) | This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP. |
| abstract void | | handlePageException(Throwable t) | This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP. |
| abstract void | | include(String relativeUriPath) | Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread. |
| abstract void | | include(String relativeUriPath, boolean flush) | Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread. |
| abstract void | | initialize(Servlet servlet, ServletRequest request, ServletResponse response, String errorPageURL, boolean needsSession, int bufferSize, boolean autoFlush) | The initialize method is called to initialize an uninitialized PageContext so that it may be used by a JSP Implementation class to service an incoming request and response within it's _jspService() method. |
| BodyContent | | pushBody() | Return a new BodyContent object, save the current "out" JspWriter, and update the value of the "out" attribute in the page scope attribute namespace of the PageContext. |
| abstract void | | release() | This method shall "reset" the internal state of a PageContext, releasing all internal references, and preparing the PageContext for potential reuse by a later invocation of initialize(). |

4. ServletConfig类：

这是servlet的配置类，负责在 servlet 容器初始化时向 servlet 传递信息。

Method Summary

| All Methods | Instance Methods | Abstract Methods |
|---------------------|--|------------------|
| Modifier and Type | Method and Description | |
| String | getInitParameter(String name) Gets the value of the initialization parameter with the given name. | |
| Enumeration<String> | getInitParameterNames() Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters. | |
| ServletContext | getServletContext() Returns a reference to the ServletContext in which the caller is executing. | |
| String | getServletName() Returns the name of this servlet instance. | |

有一个小技巧，就是servletconfig类只能获取初始化参数（InitParameter）而不能获取参数（parameter），这是由于servletconfig只是负责初始化servlet而并不会对servlet内的参数。参数这种东西只会在请求中出现。初始化尚未完成何来请求？

5. servlet类：

Method Summary

| All Methods | Instance Methods | Abstract Methods |
|----------------------|--|------------------|
| Modifier and Type | Method and Description | |
| void | destroy() Called by the servlet container to indicate to a servlet that the servlet is being taken out of service. | |
| ServletConfig | getServletConfig() Returns a ServletConfig object, which contains initialization and startup parameters for this servlet. | |
| String | getServletInfo() Returns information about the servlet, such as author, version, and copyright. | |
| void | init(ServletConfig config) Called by the servlet container to indicate to a servlet that the servlet is being placed into service. | |
| void | service(ServletRequest req, ServletResponse res) Called by the servlet container to allow the servlet to respond to a request. | |

servlet中的Service函数可以抛出 Servlet Exception 当 servlet 出现异常时；由于这个服务可能会涉及到输入输出，所以IOException是必然会有，用来处理输入输出时可能遇到的问题。

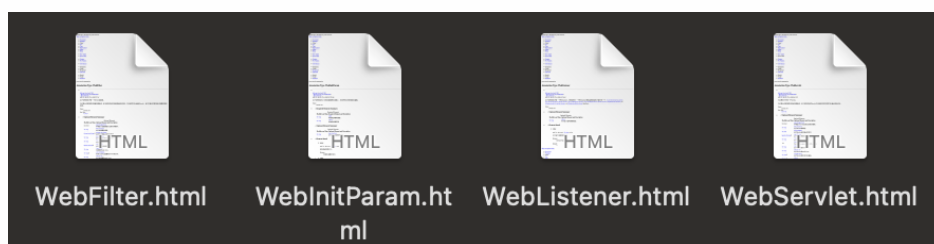
6. WebServlet类：

本注释用于声明一个Servlet。本注释会在部署的时候由容器处理，这个注释对应的Servlet可以通过注释声明的URL地址进行访问。

| Optional Element Summary | |
|--------------------------|--|
| Optional Elements | |
| Modifier and Type | Optional Element and Description |
| boolean | asyncSupported 声明这个servlet是否支持异步操作。 |
| String | description 这个Servlet的描述。 |
| String | displayName 这个Servlet向外展示的名字。 |
| WebInitParam[] | initParams 这个Servlet的初始化参数。 |
| String | largeIcon 这个Servlet的large-icon。 |
| int | loadOnStartup The load-on-startup order of the servlet |
| String | name 这个Servlet的名字。 |
| String | smallIcon 这个Servlet的small-icon。 |
| String[] | urlPatterns 这个Servlet的所有URL地址。 |
| String[] | value 这个Servlet的所有URL地址。 |

你不看上图也能推测到，URLPattern必然是一个String类型的，它的复数形式就是String数组。

7. Annotation:



你需要了解的Annotation只有这四个：

1. WebFilter是用来这个注释用来声明一个Servlet过滤器，会在部署的时候被容器处理，和本注释对应的过滤器会被应用到一个特定的URL地址和Servlet，也可以通过注释指定过滤器的派发类型。
2. WebInitParam用在Servlet和过滤器的实现类上，用来声明它们的初始化参数。
3. WebListener用来声明一个WebListener。
4. 最后一个第六题已经讲了。

8. HttpServletResponse类:

拓展 ServletResponse 接又来提供 HTTP 在发送响应时的功能。它可以通过特定的方法来访问 HTTP 头部和 cookies。

它的方法无非是操作HTTP协议报文：

1. 访问cookie
2. 访问数据报头（和它相关的操作）
3. 访问报头（和它相关的操作）
4. 设置状态码

9. HttpServletRequest类:

这个类是用来给 HTTP 提供请求信息的，也是对ServletRequest的拓展。因为这个类封装了请求信息，所以几乎就没有set方法，可以通过这个类获取这些请求信息，如URI上下文（URL和查询字符串）、Cookies、HTTP报头相关、HTTP方法（DOGHPPPT），以及请求所在会话的相关信息（session）。

| Method and Description | |
|---|---------------------|
| authenticate(HttpServletResponse response) 使用 ServletContext 的容器登录机制来认证发送请求的用户。 | boolean |
| changeSessionId() 改变与本请求相关的会话的id, 返回新的id号。 | String |
| getAuthType() 返回搭建本 servlet 的认证方案类型。 | String |
| getContextPath() 返回请求 URI 中指明请求上下文的那部分。 | String |
| getCookies() 返回随本请求发送的包含所有 Cookie 对象的数组。 | Cookie[] |
| getDateHeader(String name) 返回一个Date对象的long类型的值, 该值对应一个指定名字的请求头部。 | long |
| getHeader(String name) 返回 String类型的指定请求头部的值。 | String |
| getHeaderNames() 返回这个请求包含的所有报头名字的集合。 | Enumeration<String> |
| getHeaders(String name) 返回一个指定的请求报头对应的所有值的集合 (所有值都是String 类型) | Enumeration<String> |
| getIntHeader(String name) 返回一个int类型的请求报头的值。 | int |
| getMethod() 返回生成该请求用到的 HTTP 方法名字, 例如 GET, POST, 或者 PUT. | String |
| getQueryString() 返回查询字符串 (在该请求 URL 路径部分的后面)。 | String |
| getRequestedSessionId() 返回由客户端所确定的会话ID。 | String |
| getRequestURI() 返回请求 URL 中从协议名字到查询字符串的部分 (这部分出现在 HTTP 请求的第一行)。 | String |
| getRequestURL() 重构客户端发送请求的 URL。 | StringBuffer |
| getServletPath() 返回请求 URL 中调用本 servlet 的那一部分。 | String |
| getSession() 返回和本请求关联的目前正在使用的会话, 如果该请求没有创建会话, 那么就帮它创建一个。 | HttpSession |
| getSession(boolean create) 返回和本请求相关联的 HttpSession 对象, 如果目前还没有创建会话并且 create为 true, 返回一个新的会话。 | HttpSession |
| isRequestedSessionIdFromCookie() 检查被请求的会话ID是否被服务器以 HTTP cookie 的形式传送。 | boolean |
| isRequestedSessionIdFromURL() 检查被请求的会话ID是否被服务器作为请求 URL 的一部分。 | boolean |
| isRequestedSessionIdValid() 检查被请求的会话ID是否仍然有效。 | boolean |

那为什么请求URL要想得到需要返回的是Stringbuffer类型而不是String类型的? 因为实际上并没有完整现成的URL, 而是需要通过调用该函数根据已有的信息如URI和查询字符串合成一个URL, 这就需要缓冲区来存储可能需要的数据。

0. JSP的三大元素:

众所周知, JSP只需要理解它的三大元素就可以了:

| 元素 | 内容 | 表示 |
|------|--|---|
| 脚本元素 | 声明(用于定义在其它脚本元素中可以使用的变量、方法或类) | <%! ... %> |
| | 脚本段(在一个脚本段定义的变量为局部变量) | <% Java代码块 %> |
| | 表达式(java 语言中完整的表达式) | <%= Java 表达式 %> |
| 指令元素 | Page指令 (用于设置 JSP页面的属性) | <% @ page 属性1="属性1" ... %> |
| | Include指令 (用于在 jsp 页面中静态包含一个文件) | <%@ include file = "文件名" %> |
| | taglib 指令 (将标签库描述符文件引入到该 JSP页面中, 利用前缀名去引用) | <%@ taglib uri = "标签库表述符文件" prefix = "前缀" %> |
| | > | 用于实例化 JavaBean, 或者定位一个已经存在的 JavaBean实例, 并把实例的引用赋给一个变量 |
| 动作元素 | <jsp:setProperty> | 使用 Bean 中的 setXXX() 方法设置 JavaBean的简单属性和索引属性。 |
| | <jsp:forward> | 允许在运行时将当前的请求转发给一个静态的资源。 |
| | <jsp:include> | 用于在当前页面中包含静态和动态的资源,一旦被包含的页面执行完毕,请求被转向到页面中继续执行。 |
| | <jsp:getProperty> | 用来访问一个 Bean 的属性, 并把属性的值转化成一个 String, 然后发送到输出流中。 |

1.

填空代码解析