Java™ Portlet Specifica

Versio

版权

在线版本由神州超工作室发布.

1. Creative Commons BY-ND-NC

序：Beta 0.1

我们现在处于Beta发布阶段。也许你会问这代表了什么，其实这只是说从现在开始到正式1.0版本发布， 本书结构不会又太大的变化了。以后可能会有新的章节加入，也可能不会有。目前，我们关注于当前内容的清晰和完整， 时我们也会注意限制本书的大小和范围。

我们已经获得了关于本书的大量反馈，请不要停止给我们意见和建议，我们将非常感激，请发送至surpass_li@tom.com[1]。要了解本书的最近更新，关于英文版你可以关注该博客：http://jcp.org/en/jsr/detail?id=286；关于中文版你可以关注该博客：http://surpass-li.javaeye.com。

本书是开源的，这意味着你可以直接浏览该书的源码，并贡献你的一份力量：

- 中文版源码地址：http://github.com/surpass/GoldChamberlain

[1] mailto:surpass_li@tom.com

第 1 章 PLT.1 Preface 前言

。。。

1.1. PLT.1.1 Additional Sources

。。。

1.2. PLT.1.2 Who Should Read This Specification

。。。

1.3. PLT.1.3 API Reference API参考

。。。

1.4. 其它Java平台规范

The following Java API specifications are referenced throughout this specification: • Java 2 Platform, Enterprise Edition, v1.4 (J2EE™) • Java Servlet™, v2.4 • JavaServer Pages™, v2.0 (JSP™) • The Java™ Architecture for XML Binding (JAXB) 2.0 These specifications may be found at the Java 2 Platform Enterprise Edition website: http://java.sun.com/j2ee/. PLT.1.5 Other Important References

1.5. Other Important References

。。。

1.6. PLT.1.6 Terminology 术语

。。。

1.7. Providing Feedback

。。。

1.8. Acknowledgements V 2.0

。。。

1.9. Acknowledgements V1.0

。。。

第 2 章 Overview

。。。

2.1. PLT.2.1 What is a Portal?

。。。

2.2. PLT.2.2 What is a Portlet?

。。。

2.3. PLT.2.3 What is a Portlet Container?

。。。

2.4. PLT.2.4 An Example

。。。

2.5. PLT.2.5 Compatibility

。。。

2.6. PLT.2.6 Major changes introduced with V 2.0

。。。

2.6.1. PLT.2.6.1 Clarifications that may make V1.0 Portlets Non-compliant

。。。

2.6.2. PLT.2.6.2 Changes to the Programming Model

。。。

2.6.3. PLT.2.6.3 List of all Changes in the Specification

。。。

2.6.4. PLT.2.6.3 List of all Changes in the Specification

。。。

2.7. PLT.2.7 Relationship with Java 2 Platform, Standard and Enterprise Edition

。。。

第 3 章 PLT.3 Relationship with the Servlet Specification

。。。

3.1. PLT.3.1 Bridging from Portlets to Servlets/JSPs

。。。

3.2. PLT.3.2 Using Servlet Application Lifecycle Events

。。。

3.3. PLT.3.3 Relationship Between the Servlet Container and the Portlet Container

。。。

第 4 章 Portlet Concepts Porlet概念

。。。

4.1. PLT.4.1 Portlets

。。。

4.2. PLT.4.2 Embedding Portlets as Elements of a Portal Page

。。。

4.2.1. PLT.4.2.1 Portal Page Creation

。。。

4.2.2. PLT.4.2.2 Portal Page Request Sequence

。。。

4.3. PLT.4.3 Portlets and Web Frameworks

。。。

第 5 章 PLT.5 The Portlet Interface and Additional Life Cycle Interfaces（Porlet 接口和附加的生命周期接口）

。。。

5.1. PLT.5.1 Number of Portlet Instances实例化数量

。。。

5.2. PLT.5.2 Portlet Life Cycle

。。。

5.2.1. PLT.5.2.1 Loading and Instantiation装载和实例化

。。。

5.2.2. PLT.5.2.2 Initialization初使化

。。。

5.2.3. PLT.5.2.3 End of Service服务结束

。。。

5.3. PLT.5.3 Portlet Customization Levels 自定义

。。。

5.3.1. PLT.5.3.1 Portlet Definition and Portlet Entity 定义和实体

。。。

5.3.2. PLT.5.3.2 Portlet Window 窗口

。。。

5.4. PLT.5.4 Request Handling

。。。

5.4.1. PLT.5.4.1 Action Request

。。。

5.4.2. PLT.5.4.2 Event Request

。。。

5.4.3. PLT.5.4.3 Render Request

。。。

5.4.4. PLT.5.4.4 Resource Request

。。。

5.4.5. PLT.5.4.5 GenericPortlet

。。。

5.4.6. PLT.5.4.6 Multithreading Issues During Request Handling

。。。

5.4.7. PLT.5.4.7 Exceptions During Request Handling

。。。

5.4.8. PLT.5.4.8 Thread Safety线程安全

。。。

第 6 章 PLT.6 Portlet Config

。。。

6.1. PLT.6.1 Initialization Parameters初使化参数

。。。

6.2. PLT.6.2 Portlet Resource Bundle 资源绑定

。。。

6.3. PLT.6.3 Default Event Namespace默认事件命名空间

。。。

6.4. PLT.6.4 Public Render Parameter Names

。。。

6.5. PLT.6.5 Publishing Event QNames

。。。

6.6. PLT.6.6 Processing Event QNames

。。。

6.7. PLT.6.7 Supported Locales

。。。

6.8. Supported Container Runtime Options

。。。

第 7 章 PLT.7 Portlet URLs

。。。

7.1. PLT.7.1 Portlet URLs

。。。

7.1.1. PLT.7.1.1 BaseURL interface

。。。

7.1.2. PLT.7.1.2 Including a Portlet Mode or a Window State

。。。

7.1.3. PLT.7.1.3 Portlet URL security

。。。

7.2. PLT.7.2 Portlet URL listeners

。。。

7.2.1. PLT.7.2.1 PortletURLGenerationListener Interface

。。。

7.2.2. PLT.7.2.2 Registering Portlet URL Listeners

。。。

第 8 章 PLT.8 Portlet Modes（Porlet 模式）

。。。

8.1. PLT.8.1 VIEW Portlet Mode 查看模式

。。。

8.2. PLT.8.2 EDIT Portlet Mode 编辑模式

。。。

8.3. PLT.8.3 HELP Portlet Mode （Portlet帮助模式）

。。。

8.4. PLT.8.4 Custom Portlet Modes

。。。

8.5. PLT.8.5 GenericPortlet Render Handling

。。。

8.6. PLT.8.6 Defining Portlet Modes Support

。。。

8.7. PLT.8.7 Setting next possible Portlet Modes

。。。

第 9 章 PLT.9 Window States 窗口状态

一个 portlet 可以根据视窗状态来决定在一个页面里该占多少大小。当调用一个 portlet 时， portlet-container 需要告诉该 portlet 目前的视窗状态。 此时 portlet 可以根据视窗状态 来决定它该对多少信息来作处理。 在处理请求的过程中，portlet 可以通过程序的方式来改 变视窗状态。

Portlet规范定义三种窗口状态： NORMAL， MAXMIZED， MINIMIZED。 WindowState 这个类定义了这三个状态的常量值。

9.1. PLT.9.1 NORMAL Window State

NORMAL状态窗口表示，portlet可以与其它portlet共享一个页面。 它可以显示在有受限的显示功能的设备上。 因此在这此视窗状态下，portlet必须限制其大小。

9.2. PLT.9.2 MAXIMIZED Window State 最大化窗口

MAXIMIZED状态窗口表示，一个portlet可以占一个portal页面或者一个portlet有比其它portlet更多的空间。 在这种状态下,portle可以产生更丰富的内容。

9.3. PLT.9.3 MINIMIZED Window State 最小化窗口

当一个portlet在MINIMIZED状态下时， portlet必须只产生最小化的显示， 或者是不输出任何内容。

9.4. PLT.9.4 Custom（自定义） Window States

Portal的提供者可以定义定制的视窗状态

Portlets只能使用 portal 所定义的窗口状态。 Portlets必须在其部署描述文件中定义其可以使用的窗口状态。 在部署的时候，部署描述文件中所定义的定制化的视窗状态必须符合portal执行所支持的窗口状态。

否则， 在此状态下 portlet 是不会被调用的。 假设要做到 half_page 这个视窗状态时，

要如何定义，以下例子说明在部署描述文件中该如何定义所支持的视窗状态： If a custom window state defined in the deployment descriptor is not mapped to a custom window state provided by the portal, portlets must not be invoked in that window state. for example, the deployment descriptor for a portlet

application containing portlets that use a custom half_page window state would have the following definition:

```
<portlet-app>
...
<custom-window-state>
10 <description>Occupies 50% of the portal page</description>
<window-state>half_page</window-state>
</custom-window-state>
...
</portlet-app>
```

9.5. PLT.9.5 Defining Window State Support.定认窗口状态支持

。。。

第 10 章 PLT.10 Portlet Context

The PortletContext interface defines a portlet's view of the portlet application within which the portlet is running. Using the PortletContext object, a portlet can log events, obtain portlet application resources, application and portlet runtime options and set and store attributes that other portlets and servlets in the portlet application can access.

10.1. PLT.10.1 Scope of the Portlet Context

。。。

10.2. Portlet Context functionality（Portlet下下文功能）

。。。

10.3. PLT.10.3 Relationship with the Servlet Context

...

10.3.1. PLT.10.3.1 Correspondence(通信) between ServletContext and PortletContext methods

下列方法ServletContext提供，PortletContext也应该提供相同的功能：
getAttribute,getAttributeNames, getInitParameter, getInitParameterNames, getMimeType, getRealPath, getResource, getResourcePaths, getResourceAsStream, log,removeAttribute and setAttribute.

10.4. PLT.10.4 Portlet Container Runtime Options

The portlet can define additional runtime behavior in the portlet.xml on either the portlet application level or the portlet level with the container-runtime-option element. Runtime options that are defined on the application level should be applied to all portlets in the portlet application. Runtime options that are defined on the portlet level should be applied for this portlet only and override any runtime options defined on the application level with the same name.

Container runtime options besides the javax.portlet.actionScopedRequestAttributes option are optional to support

by the portlet container and the portlet can find out which container runtime options are supported by the portlet container running the portlet via the method getContainerRuntimeOptions on the PortletContext.

The getContainerRuntimeOptions method returns an enumeration of type String containing the keys of all container runtime options that the current portlet container supports.

10.4.1. PLT.10.4.1 Runtime Option javax.portlet.escapeXml

...

10.4.2. PLT.10.4.2 Runtime Option javax.portlet.renderHeaders

...

10.4.3. PLT.10.4.3 Runtime Option javax.portlet.servletDefaultSessionScope

...

10.4.4. PLT.10.4.4 Runtime Option javax.portlet.actionScopedRequestAttributes

...

第 11 章 PLT.11 Portlet Requests

...

11.1. PLT.11.1 PortletRequest Interface

...

11.1.1. PLT.11.1.1 Request Parameters请求参数

...

11.1.2. PLT.11.1.2 Public Render Parameters

...

11.1.3. PLT.11.1.3 Extra Request Parameters扩展请求参数

...

11.1.4. PLT.11.1.4 Request Attributes

...

11.1.5. PLT.11.1.5 Request Properties

...

11.1.6. PLT.11.1.6 Request Context Path请求上下文路径

...

11.1.7. PLT.11.1.7 Security Attributes安全属性

...

11.1.8. PLT.11.1.8 Response Content Types响应内容类型

...

11.1.9. PLT.11.1.9 Internationalization国际化

...

第 12 章 Portlet Responses

...

12.1. PLT.12.1 PortletResponse Interface

...

12.1.1. PLT.12.1.1 Response Properties Response属性

...

12.1.2. PLT.12.1.2 Encoding of URLs URLs编码

...

12.1.3. PLT.12.1.3 命名空间

...

12.1.4. PLT.12.1.4 Setting Cookies设置Cookies

...

12.2. PLT.12.2 StateAwareResponse Interface

...

12.2.1. PLT.12.2.1 Render Parameters

...

12.2.2. PLT.12.2.2 Portlet Modes and Window State Changes Portlet模式与窗口状态转换

...

12.2.3. PLT.12.2.3 Publishing Events公共事件

...

12.3. PLT.12.3 ActionResponse Interface

...

12.3.1. PLT.12.3.1 Redirections

...

12.4. PLT.12.4 EventResponse Interface

...

12.5. PLT.12.5 MimeResponse Interface

...

12.5.1. PLT.12.5.1 Content Type内容类型

...

12.5.2. PLT.12.5.2 Output Stream and Writer Objects

...

12.5.3. PLT.12.5.3 Access to Response Headers响应头的使用

...

12.5.4. PLT.12.5.4 Setting Markup Head Elements

...

12.5.5. PLT.12.5.5 Buffering缓冲

...

12.5.6. PLT.12.5.6 Predefined MimeResponse Properties

...

12.6. PLT.12.6 RenderResponse Interface

...

第 13 章 Resource Serving资源服务

…

13.1. PLT.13.1 ResourceServingPortlet Interface

…

13.2. PLT.13.2 Access to Render Parameters, Portlet Mode, and Window State

…

13.3. PLT.13.3 Access to Request and Response Headers

…

13.4. PLT.13.4 Getting the HTTP Method

…

13.5. PLT.13.5 Access to the Resource ID

…

13.6. PLT.13.6 Resource URLs

…

13.7. PLT.13.7 资源缓存

…

13.8. PLT.13.8 Generic Portlet Support

…

第 14 章 PLT.14 Serving Fragments through Portlets

...

14.1. PLT.14.1 Serving Fragments via serveResource Method

...

第 15 章 PLT.15 Coordination between portlets

...

15.1. PLT.15.1 Public Render Parameters

...

15.2. PLT.15.2 Portlet 事件

...

15.2.1. PLT.15.2.1 EventPortlet 接口

...

15.2.2. PLT.15.2.2 Receiving Events接收事件

...

15.2.3. PLT.15.2.3 Sending Events发送事件

...

15.2.4. PLT.15.2.4 Event declaration事件声明

...

15.2.5. PLT.15.2.5 Event processing事件处理

...

15.2.6. PLT.15.2.6 Exceptions during event processing

...

15.2.7. PLT.15.2.7 GenericPortlet support（支持）

...

15.3. PLT.15.3 Predefined Container Events

...

第 16 章 PLT.16 Portal Context

...

16.1. PLT.16.1 Support for Markup Head Elements

Portals should indicate if they support the MimeResponse property MimeResponse.MARKUP_HEAD_ELEMENT (value: "javax.portlet.markup.head.element") by providing the PortalContext.HTML_HEAD_ELEMENT_SUPPORT (value: "javax.portlet.markup.head.element.support") property on the PortalContext. A non-null value of MARKUP_HEAD_ELEMENT_SUPPORT indicates that the portal application supports the MARKUP_HEAD_ELEMENT property.

第 17 章 PLT.17 Portlet Preferences

...

17.1. PLT.17.1 PortletPreferences 接口

...

17.2. PLT.17.2 Preference Attributes 作用域

...

17.3. PLT.17.3 Preference Attributes 定义

...

17.3.1. PLT.17.3.1 本地化Preference属性

...

17.4. PLT.17.4 验证Preference values

...

第 18 章 PLT.18 Sessions

18.7. PLT.18.7 Session Timeouts超时设定

...

18.8. PLT.18.8 最后一次访问时间

...

18.9. PLT.18.9 Important Session Semantics语义

...

第 19 章 PLT.19 分发请求到Servlets 和 JSPs

...

19.1. PLT.19.1 Obtaining a PortletRequestDispatcher

...

19.1.1. PLT.19.1.1 Query Strings in Request Dispatcher Paths

...

19.2. PLT.19.2 Using a Request Dispatcher

...

19.3. PLT.19.3 包含方法

...

19.3.1. PLT.19.3.1 包含请求参数

...

19.3.2. PLT.19.3.2 包含请求属性

...

19.3.3. PLT.19.3.3 Request and Response Objects for Included Servlets 或 JSPs from within the Action and Event processing Methods

...

19.3.4. PLT.19.3.4 Request and Response Objects for Included Servlets或JSPs from within the Render Method

...

19.3.5. PLT.19.3.5 Request and Response Objects for Included Servlets 或 JSPs from within the ServeResource Method

...

19.3.6. PLT.19.3.6 不同的Included请求分发的比较

...

19.3.7. PLT.19.3.7 错误处理

...

19.3.8. PLT.19.3.8 Path and Query Information in Included 或 Forwarded Servlets

...

19.4. PLT.19.4 转向方法

...

19.4.1. PLT.19.4.1 查询字符串

...

19.4.2. PLT.19.4.2 Forwarded 请求参数

...

19.4.3. PLT.19.4.3 Request and Response Objects for Forwarded Servlets或JSPs from within the Action and Event processing Methods

...

19.4.4. PLT.19.4.4 Request and Response Objects for Forwarded Servlets或JSPs from within the Render Method

...

19.4.5. PLT.19.4.5 Request and Response Objects for Forwarded Servlets或JSPs from within the ServeResource Method

...

19.4.6. 不同的Forwards请求分发的比

...

19.5. PLT.19.5 Servlet过滤器和请求分发

...

19.6. PLT.19.6 Session作用域内 Include/Forwarded默认行为转换

...

第 20 章 PLT.20 Portlet过滤器

...

20.1. PLT.20.1 什么是portlet过滤器?

...

20.2. PLT.20.2 主要的概念

...

20.2.1. 过滤器生命周期

...

20.2.2. 请求与回应的包装

> Central to the notion of filtering is the concept of wrapping a request or response in order that it can override behavior to perform a filtering task. In this model, the developer has the ability to override existing methods on the request and response objects. The portlet should not add additional methods to the wrapper as further downstream wrappers may not honor these. In order to support this style of filter the container must support the following requirement. When a filter invokes the doFilter method on the portlet container’s filter chain implementation, the container must ensure that the request and response object that it passes to the next component in the filter chain, or to the target portlet if the filter was the last in the chain, is the same object that was passed into the doFilter method by the calling filter or one of the above mentioned wrappers. ccxcvi

20.2.3. 过滤器环境

...

20.2.4. Portlet应用中配置一个过滤器

...

20.2.5. 定义一个过滤器目录的方法

...

第 21 章 用户信息

…

21.1. PLT.21.1 Defining User Attributes

…

21.2. 访问用户属性

…

21.3. PLT.21.3 Important Note on User Information

…

第 22 章 缓存

…

22.1. Expiration Cache呼出缓存

…

22.2. 确认缓存

…

第 23 章 Portlet应用

...

23.1. 与Web应用的关系

...

23.2. 与PortletContext的关系

...

23.3. Porlet应用原理

...

23.4. 目录结构

...

23.5. Porlet应用类装载

...

23.6. Porlet应用Archive文件

...

23.7. Porlet应用部暑描述

...

23.8. Porlet应用更换

...

23.9. 错误处理

...

23.10. Porlet应用环境

...

第 24 章 安全

...

24.1. 介绍

...

24.2. 角色

...

24.3. PLT.24.3 Programmatic Security

...

24.4. PLT.24.4 Specifying Security Constraints

...

24.5. Propagation of Security Identity in EJBTM Calls

...

第 25 章 打包和部暑描述

...

25.1. PLT.25.1 Portlet and Web Application Deployment Descriptor

...

25.2. 打包

...

25.2.1. 实例目录结构

...

25.2.2. PLT.25.2.2 Version Information

...

25.3. Portlet部暑描述原理

...

25.4. PLT.25.4 Rules for processing the Portlet Deployment Descriptor

...

25.5. Portlet部暑描述

...

25.6. PLT.25.6 Pictures of the structure of a Deployment Descriptor

...

25.7. Uniqueness of Deployment Descriptor Values

…

25.8. PLT.25.8 Localization 本地化

…

25.9. 部署描述例子

…

25.10. 资源绑定

…

25.11. 资源绑定例子

…

第 26 章 PLT.26 Portlet标签库

第 27 章 PLT.27 Leveraging JAXB for Event payloads

...

第 28 章 PLT.28 Technology Compatibility Kit Requirements

...

28.1. PLT.28.1 TCK Test Components

...

28.2. PLT.28.2 TCK Requirements

...

28.2.1. PLT.28.2.1 Declarative configuration of the portal page for a TCK test

...

28.2.2. PLT.28.2.2 Programmatic configuration of the portal page for a test

...

28.2.3. PLT.28.2.3 Test Portlets Content

...

28.2.4. PLT.28.2.4 Test Cases that Require User Identity

...

附录 A. PLT.A Custom Portlet Modes

...

附录 B. PLT.B Markup Fragments

...

附录 C. PLT.C CSS Style Definitions

...

附录 D. PLT.D User Information Attribute Names

...

附录 E. PLT.E Deployment Descriptor Version 1.0

...

附录 F. PLT.F TCK Assertions

...