



## JSR ポートレットを使用した外部システムとの統合の開発

サービス カタログ (Service Catalog) の Portal Designer ソリューションは、JSR ポートレットを通じて外部アプリケーションと統合するための高機能なプラットフォームです。ポータルフロントエンドでは、フレームワークに Apache Pluto 1.1 ライブラリを使用します。Java Portlet Specification (JSR168, JSR286) 規格を満たす API を使用して作成されたポートレットは、サービス カタログ (Service Catalog) と共に導入できます。導入すると、Portal Designer に [サードパーティのポートレット (Third-Party Portlets)] と表示され、ポータル ページに追加できるようになります。JSR ポートレットその他のコンテンツを Portal Manager ソリューションでメンテナンスする方法の詳細については、『[Cisco Prime サービス カタログ \(Service Catalog\) Designer Guide](#)』を参照してください。

この章では、Portal Designer ソリューション用に JSR ポートレットを作成および導入する際のガイドラインをいくつか示します。章全体を通じて、「MyJSR」というサンプル ポートレットを使用して説明します。ポートレットは、Spring 3.0 Annotation ベースのコントローラおよび Sencha の Ext JS (ポータル フロントエンドの JavaScript フレームワーク) を使用して作成します。

### ポートレットの構造とパッケージ化

ポートレット ファイルは JSR 168 または 286 仕様に従って、使用するアプリケーション サーバに適した Web アプリケーション (war) ファイルの形式でパッケージ化する必要があります。一般的なポートレット war ファイルには、サーブレット、リソース バンドル、イメージ、html、jsp、css ファイルなどが含まれます。

### JBoss アプリケーション サーバ

「MyJSR.war」という単純なポートレットの構造を示します。

1. css  
MyJSR.css
2. images  
<Custom Images that the Portlet needs can be placed here>
3. js  
MyJSRCreatePersonView.js  
MyJSREdit.js  
MyJSRHelp.js  
MyJSRView.js

```

4. WEB-INF
  classes
    com
      myjsr
        MyJSRController.class
    config
      spring
        MyJSRApplicationContext.xml
      jsrportlet.properties
      log4j.properties
  jsp
    MyJSREdit.jsp
    MyJSRHelp.jsp
    MyJSRView_listperson.jsp
    MyJSRView_updateperson.jsp
  lib
    newscale_appclient.jar
    newscale_core.jar
    cxf-2.2.12.jar
    cxf-rt-transport-http-2.2.12.jar
    pluto-portal-driver-2.0.2.jar
    org.springframework.aop-3.1.0.RELEASE.jar
    org.springframework.asm-3.1.0.RELEASE.jar
    org.springframework.aspects-3.1.0.RELEASE.jar
    org.springframework.beans-3.1.0.RELEASE.jar
    org.springframework.context-3.1.0.RELEASE.jar
    org.springframework.context.support-3.1.0.RELEASE.jar
    org.springframework.core-3.1.0.RELEASE.jar
    org.springframework.expression-3.1.0.RELEASE.jar
    org.springframework.instrument-3.1.0.RELEASE.jar
    org.springframework.instrument.tomcat-3.1.0.RELEASE.jar
    org.springframework.jdbc-3.1.0.RELEASE.jar
    org.springframework.jms-3.1.0.RELEASE.jar
    org.springframework.orm-3.1.0.RELEASE.jar
    org.springframework.oxm-3.1.0.RELEASE.jar
    org.springframework.test-3.1.0.RELEASE.jar
    org.springframework.transaction-3.1.0.RELEASE.jar
    org.springframework.web-3.1.0.RELEASE.jar
    org.springframework.web.portlet-3.1.0.RELEASE.jar
    org.springframework.web.servlet-3.1.0.RELEASE.jar
    org.springframework.web.struts-3.1.0.RELEASE.jar
  tld
    c.tld
    pluto.tld
    portlet.tld
    portlet_2_0.tld
    portlet-el.tld
    portlet-el_2_0.tld
  portlet.xml
  web.xml
  jboss-deployment-structure.xml

```

このサンプル ポートレットでは、ポートレットが REST API を呼び出してサービス カタログ (Service Catalog) からデータを取得するため、lib フォルダに nsAPI java クライアント (newscale\_appclient.jar) が含まれています。nsAPI java クライアントが使用する pluto ライブラリおよびその他のライブラリが lib フォルダに含まれている必要があります。さらに JBoss モジュールの依存関係を記述するために、jboss-deployment-structure.xml が含まれます。

ポートレット関連の設定を指定するために、追加の記述子 (portlet.xml) が必要です。

## Weblogic アプリケーション サーバ

WebLogic の JSR ポートレット war の構造は似ていますが、サービス カタログ (Service Catalog) アプリケーションの外に導入するため、ポートレットが使用するすべてのライブラリを含める必要があります。

nsAPI java クライアントを呼び出し、ポートレットの共通設定を読み取るには、次のようないくつかのライブラリが必要です。ここでは、ポートレットが JSP の JSTL タグおよび Spring ポートレット MVC を使用するため、いくつかのライブラリが必要です。

1. css
  - MyJSR.css
2. images
  - <Custom Images that the Portlet needs can be placed here>
3. js
  - MyJSRCreatePersonView.js
  - MyJSREdit.js
  - MyJSRHelp.js
  - MyJSRView.js
4. WEB-INF
  - classes
    - com
      - myjsr
        - MyJSRController.class
    - config
      - spring
        - MyJSRApplicationContext.xml
    - jsrportlet.properties
    - log4j.properties
  - jsp
    - MyJSREdit.jsp
    - MyJSRHelp.jsp
    - MyJSRView\_listperson.jsp
    - MyJSRView\_updateperson.jsp
  - lib
    - newscale\_appclient.jar
    - newscale\_core.jar
    - newscale\_udkernel.jar
    - newscale\_compbeans.jar
    - newscale\_conf.jar
    - castor-0.9.5.4.jar
    - commons-beanutils-1.8.3.jar
    - commons-httpclient-3.1.jar
    - commons-logging-1.0.4.jar
    - cxf-2.2.12.jar
    - ezmorph-1.0.4.jar
    - json-lib-2.2.2-jdk13.jar
    - jsr311-api-1.0.jar
    - neethi-2.0.4.jar
    - oscache-2.4.jar
    - standard.jar
    - wsdl4j-1.6.1.jar
    - XmlSchema-1.4.6.jar
    - commons-collections-3.2.1.jar
    - commons-lang-2.4.jar
    - neethi-2.0.4.jar
    - castor-0.9.5.4.jar
    - jstl.jar
    - org.springframework.aop-3.1.0.RELEASE.jar
    - org.springframework.asm-3.1.0.RELEASE.jar
    - org.springframework.aspects-3.1.0.RELEASE.jar
    - org.springframework.beans-3.1.0.RELEASE.jar
    - org.springframework.context-3.1.0.RELEASE.jar

```

org.springframework.context.support-3.1.0.RELEASE.jar
org.springframework.core-3.1.0.RELEASE.jar
org.springframework.expression-3.1.0.RELEASE.jar
org.springframework.instrument-3.1.0.RELEASE.jar
org.springframework.instrument.tomcat-3.1.0.RELEASE.jar
org.springframework.jdbc-3.1.0.RELEASE.jar
org.springframework.jms-3.1.0.RELEASE.jar
org.springframework.spring-library-3.1.0.RELEASE.libd
org.springframework.test-3.1.0.RELEASE.jar
org.springframework.transaction-3.1.0.RELEASE.jar
org.springframework.web-3.1.0.RELEASE.jar
org.springframework.web.portlet-3.1.0.RELEASE.jar
org.springframework.web.servlet-3.1.0.RELEASE.jar
org.springframework.web.struts-3.1.0.RELEASE.jar
tld
  c.tld
  pluto.tld
  portlet.tld
  portlet_2_0.tld
  portlet-el.tld
  portlet-el_2_0.tld
portlet.xml
web.xml

```

## 依存ライブラリ

JSR ポートレットの war ファイルに含める必要があるライブラリ セットは、アプリケーションサーバの RequestCenter 配置済みアプリケーションまたはサービス カタログ (Service Catalog) インストーラ イメージで入手できます。

- pluto-taglib-2.0.2.jar: 製品イメージの「preinstall」フォルダ下
- 他のすべてのファイル: RequestCenter.war/WEB-INF/lib

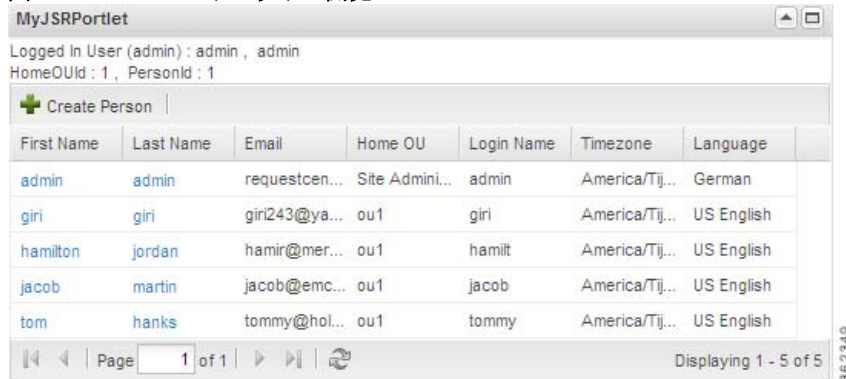
## ポートレットの開発

一般的な JSR ポートレットでは、表示 (View)、編集 (Edit)、ヘルプ (Help) の 3 つのレンダリングモードをサポートする必要があります。さらに、通常 (Normal)、最小化 (Minimized)、最大化 (Maximized) のウィンドウ状態をサポートする必要があります。

以下に示す MyJSR ポータルの例では、2 つの機能をサポートするユーザ インターフェイスが提供されます。

1. サービス カタログ (Service Catalog) ユーザをグリッドで一覧表示します。

図 7-1 ポートレットの開発



2. ユーザを追加または更新できます。

図 7-2 ユーザの追加 (Add User)

The screenshot shows a web application window titled "My JSR Portlet" with a sub-header "Person Details - Add". The form contains several input fields for user information:

- First Name:
- Last Name:
- Login Name:
- Email:
- Home OU:
- Language:
- Timezone:

Below these fields are two tabs: "Address" (selected) and "Contacts". Under the "Address" tab, there are two more input fields:

- Business:
- Home:

At the bottom right of the form, there are "Save" and "Cancel" buttons. A vertical ID "362350" is visible on the right side of the window.

この要件を満たすために、次に示すサンプル コードには、次の高水準操作が含まれています。

- nsAPI java クライアントを使用して、サービス カタログ (Service Catalog) ユーザを取得する
- ユーザの詳細を JSON 形式でユーザ インターフェイスに返す
- Ext JS グリッドで、ブラウザにユーザのリストをレンダリングする
- Ext JS で設計した形式で、ユーザの詳細を表示または入力する
- nsAPI java クライアントを使用して、サービス カタログ (Service Catalog) リポジトリのユーザの詳細を追加または更新する

## MyJSR.css

ポートレットのコード カスタム スタイルをここで設計できます。

以降で、MyJSR.war 内のコンポーネントそれぞれの内容を見ていきます。

## MyJSRCreatePersonView.js

Ext JS を使用して、ユーザを作成するためのフォームを表示するコード例。

```
/* Code custom JavaScript for the portlet here */

Ext.onReady(function() {
  var tab2 = new Ext.FormPanel({
    id : 'personEditForm',
    labelAlign : 'top',
    title : 'Person Details - Add',
    bodyStyle : 'padding:5px',
    width : 600,
    renderTo : MyJSREditDiv,
    items : [{
      layout : 'column',
      border : false,
      items : [{
        columnWidth : .5,
        layout : 'form',
        border : false,
        items : [{
          xtype : 'textfield',
          fieldLabel : 'First Name',
          value : personListObj.firstName,
          name : 'firstName',
          anchor : '95%'
        }, {
          xtype : 'textfield',
          fieldLabel : 'Login Name',
          value : personListObj.login,
          name : 'login',
          anchor : '95%'
        }, {
          xtype : 'textfield',
          fieldLabel : 'Home OU',
          name : 'homeOrganizationalUnitName',
          value : personListObj.homeOrganizationalUnitName,
          anchor : '95%'
        }, {
          xtype : 'textfield',
          fieldLabel : 'Timezone',
          name : 'timeZoneName',
          value : personListObj.timeZoneName,
          anchor : '95%'
        }
      ]
    }, {
      columnWidth : .5,
      layout : 'form',
      border : false,
      items : [{
        xtype : 'textfield',
        fieldLabel : 'Last Name',
        name : 'lastName',
        value : personListObj.lastName,
```

```

        anchor : '95%'
    }, {
        xtype : 'textfield',
        fieldLabel : 'Email',
        name : 'email',
        value : personListObj.email,
        vtype : 'email',
        anchor : '95%'
    }, {
        xtype : 'textfield',
        fieldLabel : 'Language',
        name : 'languageName',
        value : personListObj.languageName,
        anchor : '95%'
    }
    ]
    ], {
        xtype : 'tabpanel',
        plain : true,
        activeTab : 0,
        height : 235,
        defaults : {
            bodyStyle : 'padding:10px'
        },
        items : [{
            title : 'Address',
            layout : 'form',
            defaults : {
                width : 230
            },
            defaultType : 'textfield',
            items : [{
                fieldLabel : 'Business',
                name : 'businessAddress',
                disabled : true
            }, {
                fieldLabel : 'Home',
                name : 'homeAddress',
                disabled : true
            }
        ]
    }, {
        title : 'Contacts',
        layout : 'form',
        defaults : {
            width : 230
        },
        defaultType : 'textfield',
        items : [{
            fieldLabel : 'Business',
            name : 'businessPhone',
            disabled : true
        }, {
            fieldLabel : 'Home',
            name : 'homePhone',
            disabled : true
        }, {
            fieldLabel : 'Mobile',
            name : 'mobilePhone',
            disabled : true
        }, {
            fieldLabel : 'Fax',
            name : 'faxNumber',
            disabled : true
        }
    ]
    ]

```

```

        ]],
    }, {
        text : 'Save',
        handler : function() {
            Ext.getCmp("personEditForm").getForm().submit({
                url : addPersonActionUrl,
                params : {},
                success : function(form, action) {
                    var responseObj = Ext.util.JSON.decode(action.response.responseText);
                    if(responseObj.success == "true")
                        Ext.Msg.alert('Success', responseObj.successMsg);
                    else
                        Ext.Msg.alert('Error', responseObj.errorMsg);
                }
            });
        }
    }, {
        text : 'Cancel',
        handler : function() {
            window.location=viewPersonUrl;
        }
    }
    ]]);
});

```

## MyJSREdit.js

この JavaScript を使用すると、ポートレット編集モード用のカスタム コードを追加できます。

```

/* Code custom JavaScript for the Portlet here */

Ext.onReady(function() {

});

```

## MyJSRHelp.js

この JavaScript を使用すると、ポートレット編集モード用のカスタム コードを追加できます。

```

/* Code custom JavaScript for the Portlet here */

Ext.onReady(function() {

});

```



## MyJSRView.js

Ext JS グリッドにユーザを表示する JavaScript の例。

```
/* Code custom JavaScript for the Portlet here */
Ext.onReady(function() {
// Demonstrates how to getUser info from Java Script and set it to div
varLogin=document.getElementById('MyJSRLoginNameDiv');
    Login.innerHTML=nsAPP_CurrentUserLoginName;
varFirstName=document.getElementById('MyJSRFirstNameDiv');
    FirstName.innerHTML=nsAPP_CurrentUserFirstName;
varLastName=document.getElementById('MyJSRLastNameDiv');
    LastName.innerHTML=nsAPP_CurrentUserLastName;
varHomeOU=document.getElementById('MyJSRHomeOUDiv');
    HomeOU.innerHTML=nsAPP_CurrentUserHomeOuId;
var PersonID=document.getElementById('MyJSRPersonIDDiv');
    PersonID.innerHTML=nsAPP_CurrentUserId;

var pid = portletId.substr(pidPrefix.length);
if (Ext.getCmp(pid).height && Ext.getCmp(pid).height >= 29) {
var gridHeight = Ext.getCmp(pid).height - 29;
    }

var gridStore = new Ext.data.JsonStore({
proxy : new Ext.data.HttpProxy({
url : pagingUrl,
timeout : connectionTimeout
}),
autoLoad: {params:{start: 0, limit: defaultRecordSize}},
root: 'rows',
totalProperty: 'results',
fields : [{
name : 'firstName',
type : 'string'
}, {
name : 'lastName',
type : 'string'
}, {
name : 'email',
type : 'string'
}, {
name : 'homeOrganizationalUnitName',
type : 'string'
}, {
name : 'login',
type : 'string'
}, {
name : 'timeZoneName',
type : 'string'
}, {
name : 'languageName',
type : 'string'
}, {
name : 'businessPhone',
type : 'string'
}, {
name : 'homePhone',
type : 'string'
}, {
name : 'mobilePhone',
type : 'string'
}, {
```

```

name : 'faxNumber',
type : 'string'
    }, {
name : 'businessAddress',
type : 'string'
    }, {
name : 'homeAddress',
type : 'string'
    }
    ]
    });
gridStore.load();

var expander = new Ext.ux.grid.RowExpander({
tpl : new Ext.Template(
'<h2 class="title">Address</h2><table>',
'<tr><td width=400><b>Business</b> {businessAddress}</td>',
'<td width=400><b>Home</b>{homeAddress}</td></tr></table>',
'<h2 class="title">Contact</h2><table>',
'<tr><td width=400><b>Business</b> {businessPhone}</td>',
'<td width=400><b>Home</b> {homePhone}</td></tr>',
'<tr><td width=400><b>Mobile</b> {mobilePhone}</td>',
'<td width=400><b>Fax</b> {faxNumber}</td></tr></table>'
    );
});

var gridColModel = new Ext.grid.ColumnModel({
defaults : {
sortable : true,
autoWidth : true
    },
columns : [{
header : "First Name",
dataIndex : 'firstName'
    }, {
header : "Last Name",
dataIndex : 'lastName'
    }, {
header : "Email",
dataIndex : 'email'
    }, {
header : "Home OU",
dataIndex : 'homeOrganizationalUnitName'
    }, {
header : "Login Name",
dataIndex : 'login'
    }, {
header : "Timezone",
dataIndex : 'timeZoneName'
    }, {
header : "Language",
dataIndex : 'languageName'
    }
    ]
    });

var gridConfig = {
renderTo : MyJSREditDiv,
width : "100%",
layout : 'fit',
store : gridStore,
cm : gridColModel,
loadMask: true,
autoWidth : true,
plugins : expander,
tbar : [{
text : 'Create Person',

```

```

iconCls : 'add',
handler : function() {
window.location=createNewPersonActionUrl;
        }
        },'-'],
bbar : new Ext.PagingToolbar({
pageSize : defaultRecordSize,
store : gridStore,
displayInfo : true,
params:{
start: 0,
limit: defaultRecordSize
        }
    })
    });

if ('maximized' == portletWindowState) {
gridConfig.height = document.documentElement.clientHeight - 188;
    } elseif ('normal' == portletWindowState) {
var viewConfig = {
forceFit : true
    };
gridConfig.viewConfig = viewConfig;

if (gridHeight && gridHeight > -1) {
gridConfig.height = gridHeight;
    } else {
gridConfig.autoHeight = true;
    }
    }

var grid = new Ext.grid.GridPanel(gridConfig);
});

```

## portlet.xml

ポートレット仕様の例。portlet-class は、init-params のペア contextConfigLocation と nsContentPortlet とともに設定する必要があります(nsContentPortlet は、常に「false」に設定します)。

```

<?xmlversion="1.0"encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.

See the License for the specific language governing permissions and
limitations under the License.
-->
<portlet-app
xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd"
version="1.0"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd
                    http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd">
<portlet>
<description>MyJSR Description</description>
<portlet-name>nsMyJSR</portlet-name>
<display-name>My JSR Portlet</display-name>
<portlet-class>org.springframework.web.portlet.DispatcherPortlet</portlet-class>
<init-param>
<name>contextConfigLocation</name>
<value>/WEB-INF/classes/config/spring/MyJSRApplicationContext.xml</value>
</init-param>
<init-param>
<name>nsContentPortlet</name>
<value>>false</value>
</init-param>
<expiration-cache>-1</expiration-cache>
<supports>
<mime-type>text/html</mime-type>
<portlet-mode>VIEW</portlet-mode>
<portlet-mode>EDIT</portlet-mode>
<portlet-mode>HELP</portlet-mode>
</supports>
<portlet-info>
<title>My JSR Portlet</title>
</portlet-info>
</portlet>
</portlet-app>

```

## web.xml

サーブレットによる展開記述子の例。ポータルサーバ(この例では Apache Pluto)でサーブレットのマッピングが必要です。

```

<?xmlversion="1.0"encoding="UTF-8"?>
<!DOCTYPEweb-appPUBLIC"-Sun Microsystems, Inc. DTD Web Application
2.3EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<display-name>My JSR Portlet Application</display-name>
<description>My JSR Portlet</description>

<!-- Resources bundle base class -->
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>
    /WEB-INF/classes/config/spring/MyJSRApplicationContext.xml
</param-value>
</context-param>

<context-param>
<param-name>parameter-name</param-name>
<param-value>parameter-value</param-value>
</context-param>

<servlet>
<servlet-name>ViewRendererServlet</servlet-name>
<servlet-class>
    org.springframework.web.servlet.ViewRendererServlet
</servlet-class>
</servlet>

```

```
<servlet>
<servlet-name>MyJSR</servlet-name>
<servlet-class>org.apache.pluto.container.driver.PortletServlet</servlet-class>
<init-param>
<param-name>portlet-name</param-name>
<param-value>MyJSR</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
<servlet-name>ViewRendererServlet</servlet-name>
<url-pattern>/WEB-INF/servlet/view</url-pattern>
</servlet-mapping>

<servlet-mapping>
<servlet-name>MyJSR</servlet-name>
<url-pattern>/PlutoInvoker/nsMyJSR</url-pattern>
</servlet-mapping>

<!-- Declare Tag libraries that are used in which are going to use in JSP pages-->
<taglib>
<taglib-uri>http://portals.apache.org/pluto</taglib-uri>
<taglib-location>/WEB-INF/tld/pluto.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/portlet_2_0</taglib-uri>
<taglib-location>/WEB-INF/tld/portlet_2_0.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>/WEB-INF/tld/c.tld</taglib-uri>
<taglib-location>/WEB-INF/tld/c.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/portlet</taglib-uri>
<taglib-location>/WEB-INF/tld/portlet.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://portals.apache.org/pluto/portlet-el</taglib-uri>
<taglib-location>/WEB-INF/tld/portlet-el.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://portals.apache.org/pluto/portlet-el_2_0</taglib-uri>
<taglib-location>/WEB-INF/tld/portlet-el_2_0.tld</taglib-location>
</taglib>

</web-app>
```

## MyJSREdit.jsp

ポートレット編集モード用の JSP。

```

<%
/**
 * Copyright (c) 2012, Cisco Systems, Inc. All rights reserved.
 */
%>

<%@tagliburi="http://java.sun.com/portlet"prefix="portlet"%>
<%@taglibprefix="portlet2"uri="http://java.sun.com/portlet_2_0"%>
<%@taglibprefix="c"uri="/WEB-INF/tld/c.tld"%>

<% String contextPath = request.getContextPath(); %>

<!-- This is for IE -->
<scripttype="text/javascript">
if(document.createStyleSheet) {
document.createStyleSheet('<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>');
}
else {
var styles = "@import url('<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>');";
var newSS=document.createElement('link');
newSS.rel='stylesheet';
newSS.href='data:text/css,'+escape(styles);
document.getElementsByTagName("head")[0].appendChild(newSS);
}
</script>

<!-- This is foFirefox -->
<linkrel="stylesheet"type="text/css"href="<%= response.encodeURL(contextPath +
"/css/MyJSR.css") %>"></link>

<script>
var head = document.getElementsByTagName('head')[0];
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = '<%= response.encodeURL(contextPath + "/js/MyJSREdit.js") %>';
head.appendChild(script);
</script>

<!-- Write your JSP Code for Portlet Edit here -->
<c:iftest="${portletWindowState == 'NORMAL' or portletWindowState == 'normal'}">
Portlet Mode = <c:outvalue='${portletMode}'/>
Portlet Window State = <c:outvalue='${portletWindowState}'/>
</c:if>

<c:iftest="${portletWindowState == 'MINIMIZED' or portletWindowState == 'minimized'}">
Portlet Mode = <c:outvalue='${portletMode}'/>
Portlet Window State = <c:outvalue='${portletWindowState}'/>
</c:if>

<c:iftest="${portletWindowState == 'MAXIMIZED' or portletWindowState == 'maximized'}">
Portlet Mode = <c:outvalue='${portletMode}'/>
Portlet Window State = <c:outvalue='${portletWindowState}'/>
</c:if>

<divid="MyJSREditDiv-<portlet:namespace/>"class="x-grid-mso"></div>

```

```

<script>
var MyJSREditDiv = 'MyJSREditDiv-<portlet:namespace/>';
var addPersonActionUrl = '<portlet2:resourceURL id="addPersonData" escapeXml="false" />';
var personListObj = Ext.util.JSON.decode('<c:out value="{PersonData}"
escapeXml="false"/>');
</script>

```

## MyJSRHelp.jsp

ポートレット ヘルプ モード用の JSP。

```

<%
/**
 * Copyright (c) 2012, Cisco Systems, Inc. All rights reserved.
 */
%>

<%@taglib uri="http://java.sun.com/portlet" prefix="portlet"%>
<%@taglib prefix="c" uri="/WEB-INF/tld/c.tld"%>

<% String contextPath = request.getContextPath(); %>

<!-- This is for IE -->
<script type="text/javascript">
if(document.createStyleSheet) {
document.createStyleSheet("<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>");
}
else {
var styles = "@import url('<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>')";
var newSS=document.createElement('link');
newSS.rel='stylesheet';
newSS.href='data:text/css,'+escape(styles);
document.getElementsByTagName("head")[0].appendChild(newSS);
}
</script>

<!-- This is foFirefox -->
<link rel="stylesheet" type="text/css" href="<%= response.encodeURL(contextPath +
"/css/MyJSR.css") %>"></link>

<script>
var head = document.getElementsByTagName('head')[0];
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = '<%= response.encodeURL(contextPath + "/js/MyJSRHelp.js") %>';
head.appendChild(script);
</script>

<!-- Write your JSP Code for Portlet Help here -->
<c:if test="${portletWindowState == 'NORMAL' or portletWindowState == 'normal'}">
Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' />
</c:if>

<c:if test="${portletWindowState == 'MINIMIZED' or portletWindowState == 'minimized'}">
Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' />
</c:if>

<c:if test="${portletWindowState == 'MAXIMIZED' or portletWindowState == 'maximized'}">
Portlet Mode = <c:out value='${portletMode}' />

```

```

Portlet Window State = <c:outvalue='${portletWindowState}'/>
</c:if>

<divid="MyJSRHelpDiv-<portlet:namespace/>"class="x-grid-mso"></div>

<script>
var MyJSRHelpDiv = 'MyJSRHelpDiv-<portlet:namespace/>';
</script>

```

## MyJSRView\_listperson.jsp

ポートレット表示モードの JSP コード。

```

<%
/**
 * Copyright (c) 2012, Cisco Systems, Inc. All rights reserved.
 */
%>

<%@tagliburi="http://java.sun.com/portlet"prefix="portlet"%>
<%@taglibprefix="portlet2"uri="http://java.sun.com/portlet_2_0"%>
<%@taglibprefix="c"uri="/WEB-INF/tld/c.tld"%>

<% String contextPath = request.getContextPath(); %>

<!-- This is for IE -->
<scripttype="text/javascript">
if(document.createStyleSheet) {
document.createStyleSheet('<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>');
}
else {
var styles = "@import url('<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>');";
var newSS=document.createElement('link');
newSS.rel='stylesheet';
newSS.href='data:text/css,'+escape(styles);
document.getElementsByTagName("head")[0].appendChild(newSS);
}
var portletId = 'portlet-container-<c:out value="${portlet}" />';
var pidPrefix = "portlet-container-";
var portletWindowState = "<c:out value='${portletWindowState}' />";
var portletMode = "<c:out value='${portletMode}' />";
var defaultRecordSize = <c:out value='${defaultRecordSize}' />;
var nsappPersonViewDiv = 'nsappPersonViewDiv-<portlet:namespace/>';
var pagingUrl = '<portlet2:resourceURL escapeXml="false" />';
var connectionTimeout = <c:out value='${connectionTimeout}' />;
</script>

<!-- This is foFirefox -->
<linkrel="stylesheet"type="text/css"href="<%= response.encodeURL(contextPath +
"/css/MyJSR.css") %>"></link>

<script>
var head = document.getElementsByTagName('head')[0];
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = '<%= response.encodeURL(contextPath + "/js/MyJSRView.js") %>';
head.appendChild(script);
</script>

<!-- Write your JSP Code for Portlet Edit here -->
<c:iftest="${portletWindowState == 'NORMAL' or portletWindowState == 'normal'}">

```



```

<!--PortletMode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' /> -->
</c:if>

<c:iftest="${portletWindowState == 'MINIMIZED' or portletWindowState == 'minimized'}">
<!--Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' /> -->
</c:if>

<c:iftest="${portletWindowState == 'MAXIMIZED' or portletWindowState == 'maximized'}">
<!--Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' />-->
</c:if>

<!--Un-comment this to get User Info from the Portlet Controller that sets user info in
model
Logged In User (<c:out value="${userName}" escapeXml="false"/>): <c:out
value="${firstName}" escapeXml="false"/>&nbsp;&nbsp;&nbsp;<c:out value="${lastName}"
escapeXml="false"/>&nbsp;&nbsp;&nbsp;<c:out value="${PersonId}"
escapeXml="false"/>&nbsp;&nbsp;&nbsp;<c:out value="${HomeOUIId}" escapeXml="false"/>
-->

<!--Write declare divs to display user info -->
<div>Logged In User (<div id="MyJSRLoginNameDiv"
style="text-align:right;display:inline;width:100%;"><span
style="padding-left:20px;"></span></div>&nbsp;&nbsp;&nbsp;<div id="MyJSRFirstNameDiv"
style="text-align:right;display:inline;width:100%;"><span></span></div>&nbsp;&nbsp;&nbsp;, &nbsp;&nbsp;&nbsp;
<div id="MyJSRLastNameDiv" style="text-align:right;display:inline;width:100%;"><span
style="padding-left:20px;"></span></div>
</div>
<div>HomeOUIId&nbsp;&nbsp;&nbsp;:&nbsp;&nbsp;&nbsp;<div id="MyJSRHomeOUIDiv"
style="text-align:right;display:inline;width:100%;"><span
style="padding-left:20px;"></span></div>&nbsp;&nbsp;&nbsp;, &nbsp;&nbsp;&nbsp;
PersonId&nbsp;&nbsp;&nbsp;:&nbsp;&nbsp;&nbsp;<div id="MyJSRPortletPersonIDDiv"
style="text-align:right;display:inline;width:100%;"><span
style="padding-left:20px;"></span>
</div>

<div id="MyJSREditDiv-<portlet:namespace/>" class="x-grid-mso"></div>

<script>
var MyJSREditDiv = 'MyJSREditDiv-<portlet:namespace/>';
var createNewPersonActionUrl = '<portlet:renderURL><portlet:param name="formAction"
value="createNewPerson" /></portlet:renderURL>';
var addPersonActionUrl = '<portlet2:resourceURL id="addPersonData" escapeXml="false" />';
var personListObj = Ext.util.JSON.decode('<c:out value="${PersonData}"
escapeXml="false"/>');
</script>

```

## MyJSRView\_updateperson.jsp

ユーザ更新操作のデモ用 JSP コードの例。

```

<%
/**
 * Copyright (c) 2012, Cisco Systems, Inc. All rights reserved.
 */
%>

<%@taglib uri="http://java.sun.com/portlet" prefix="portlet"%>
<%@taglib prefix="portlet2" uri="http://java.sun.com/portlet_2_0"%>

```

```

<%@taglibprefix="c"uri="/WEB-INF/tld/c.tld"%>

<%String contextPath = request.getContextPath(); %>

<!-- This is foFirefox -->
<linkrel="stylesheet" type="text/css" href="<%= response.encodeURL(contextPath +
"/css/MyJSR.css") %>"></link>

<!-- This is for IE -->
<script type="text/javascript">
if(document.createStyleSheet) {
document.createStyleSheet("<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>");
}
else {
var styles = "@import url('<%= response.encodeURL(contextPath + "/css/MyJSR.css") %>');";
var newSS=document.createElement('link');
newSS.rel='stylesheet';
newSS.href='data:text/css,'+escape(styles);
document.getElementsByTagName("head")[0].appendChild(newSS);
}
}
</script>

<script>
var head = document.getElementsByTagName('head')[0];
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = '<%= response.encodeURL(contextPath + "/js/MyJSRCreatePersonView.js") %>';
head.appendChild(script);
</script>

<!-- Write your JSP Code for Portlet View here -->
<c:if test="${portletWindowState == 'NORMAL' or portletWindowState == 'normal'}">
<!--Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' /> -->
</c:if>

<c:if test="${portletWindowState == 'MINIMIZED' or portletWindowState == 'minimized'}">
<!--Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' /> -->
</c:if>

<c:if test="${portletWindowState == 'MAXIMIZED' or portletWindowState == 'maximized'}">
<!--Portlet Mode = <c:out value='${portletMode}' />
Portlet Window State = <c:out value='${portletWindowState}' /> -->
</c:if>

<div id="MyJSREditDiv"<portlet:namespace/>" class="x-grid-mso"></div>
<script>
var MyJSREditDiv = 'MyJSREditDiv<portlet:namespace/>';
var addPersonActionUrl = '<portlet2:resourceURL id="addPersonData" escapeXml="false" />';
var viewPersonUrl = '<portlet:renderURL></portlet:renderURL>';
var personListObj = Ext.util.JSON.decode('<c:out value="${PersonData}"
escapeXml="false" />');
</script>

```

## MyJSRController.java

java ポートレット コントローラを作成する一般的な手順は、次のとおりです。

- 
- 手順 1 3つのポートレット モード(表示(View)、編集(Edit)、ヘルプ(Help))のハンドラ コードを記述します。
  - 手順 2 3つのポートレット ビュー(通常(Normal)、最小化(Minimized)、最大化(Maximized))のハンドラ コードを記述します。
  - 手順 3 サービス カタログ(Service Catalog) エンティティを処理/表示する JSR ポートレットでは、nsAPI クライアントを使用してポートレット コントローラに関連する REST API を呼び出すことができます。
    - a. nsAPI クライアント API の参照を取得します。
    - b. nsAPI クライアントを呼び出して、必要なサービス カタログ(Service Catalog) エンティティのインスタンスのリストを取得します。
    - c. オプションで、現在ログインしているユーザの詳細(ユーザ ID、名、姓など)を取得します。
  - 手順 4 インスタンスをグリッドまたはその他の形式でレンダリングします(この例では、Ext JS グリッドの nsAPI でページングをする方法も示します)。
- 

```
package com.myjsr;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Properties;

import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;
import javax.portlet.ResourceRequest;
import javax.portlet.ResourceResponse;

import net.sf.json.JSON;
import net.sf.json.JSONSerializer;
import javax.portlet.ActionRequest;
import javax.servlet.http.HttpSession;
import javax.portlet.PortletURL;
import org.apache.commons.collections.map.MultiValueMap;
import org.apache.commons.lang.StringEscapeUtils;
import org.apache.commons.lang.StringUtils;
import org.springframework.ui.Model;
import org.springframework.web.portlet.ModelAndView;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.portlet.bind.annotation.ResourceMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.newscale.comps.conf.domain.AppParamUtil;
import com.newscale.nsapi.directory.person.Person;
import com.newscale.nsapi.directory.person.PersonList;
import com.newscale.nsapiclient.NSApiClient;
import com.newscale.nsapiclient.NSApiClientConstants;
```

```

import com.newscale.nsapiclient.NSApiClientFactory;
import com.newscale.portlets.GenericNewScaleSpringPortletBase;

/**
 *MyJSRController
 */
public class MyJSRController extends GenericNewScaleSpringPortletBase {
    private static final String configPropsFile = "jsrportlet.properties";
    private static final String viewPageList = "MyJSRView_listperson";
    private static final String viewPageUpdate = "MyJSRView_updateperson";
    private static final String editPage = "MyJSREdit";
    private static final String helpPage = "MyJSRHelp";
    private NSApiClient nsApiClient = getNSApiClient();

    public NSApiClient getNsApiClient() {
        return nsApiClient;
    }

    public void setNsApiClient(NSApiClient nsApiClient) {
        this.nsApiClient = nsApiClient;
    }

    public String viewNormal(RenderRequest request, RenderResponse response, Model model) {
        try {
            super.viewNormal(request, response, model);
            getLoginUsername(request, model);
        } catch (Exception e) {
            e.printStackTrace();
        }

        return doView(request, response, model);
    }

    public String viewMinimized(RenderRequest request, RenderResponse response, Model model) {
        try {
            super.viewMinimized(request, response, model);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return viewPageList;
    }

    public String viewMaximized(RenderRequest request, RenderResponse response, Model model) {
        try {
            super.viewMaximized(request, response, model);
            getLoginUsername(request, model);
        } catch (Exception e) {
            e.printStackTrace();
        }

        return doView(request, response, model);
    }

    private void getLoginUsername(RenderRequest request, Model model) {
        Properties properties = getConfigProperties(configPropsFile);
        nsApiClient.login(properties.getProperty("BASE_URL"),
            request.getPortletSession().getId());
        // Get Currently Logged-in user from nsAPI client
        Person persons = nsApiClient.getDirectory().getCurrentUser();

        // Set user info into model so that JSP can access it
        model.addAttribute("PersonID", persons.getPersonId());
        model.addAttribute("HomeOUId", persons.getHomeOrganizationalUnitId());
    }
}

```

```

        model.addAttribute("firstName", persons.getFirstName());
        model.addAttribute("lastName", persons.getLastName());
        model.addAttribute("userName", persons.getLogin());
    }

    private String doView(RenderRequest request, RenderResponse response, Model model) {
    try {
        Properties properties = getConfigProperties(configPropsFile);
        nsApiClient.login(properties.getProperty("BASE_URL"),
            request.getPortletSession().getId());

        int defaultRecordSize = AppParamUtil.getInstance().getMaxMaxPagingSizeInNSApi();
        model.addAttribute("defaultRecordSize", "" + defaultRecordSize);
        int connectionTimeOut = 0;
        if (AppParamUtil.getInstance().isParamExists((AppParamUtil.SESSION_TIMEOUT)) {
            connectionTimeOut =
                AppParamUtil.getInstance().getIntegerParam(AppParamUtil.SESSION_TIMEOUT);
        }
        if (connectionTimeOut < 1) {
            connectionTimeOut = 20;
        }
        model.addAttribute("connectionTimeOut", "" + connectionTimeOut * 1000 * 60);

        String formAction = request.getParameter("formAction");
        String personIdStr = request.getParameter("personId");

        if (null != formAction && formAction.equals("createNewPerson") ) {
            showAddPersonPage(request, response, model);
            return viewPageUpdate;
        } elseif (null != personIdStr) {
            editPerson(request, response, model, new Integer(personIdStr).intValue());
            return viewPageUpdate;
        } else {
            return viewPageList;
        }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @RequestMapping("VIEW")
    @RequestMapping
    public ModelAndView doPagingOrSavePerson(ResourceRequest request, ResourceResponse
        response, Person person)
        throws Exception {
        String instanceName= getInstanceName(request.getWindowID());
        Save Button is clicked while adding person
        if ("createNewPerson".equals(request.getParameter("formAction"))) || null !=
            request.getParameter("personId")) {
            addPersonData(person ,request, response );
        } elseif (null != request.getParameter("start") &&null !=
            request.getParameter("limit")) { Paging
            int startInt = Integer.parseInt(request.getParameter("start")) + 1; extjs sends 1 less
            than what nsAPI wants
            int limit = Integer.parseInt(request.getParameter("limit")) + 1; extjs sends 1 less than
            what nsAPI wants
            try {
                if (request.getWindowState().equals(request.getWindowState().NORMAL)) {
                    doPagingInternal(request, response, instanceName, 1, startInt, limit);
                }
                if (request.getWindowState().equals(request.getWindowState().MAXIMIZED)) {
                    doPagingInternal(request, response, instanceName, 2, startInt, limit);
                }
            }
        }
    }

```

```

        } catch (Exception e) {
e.printStackTrace();
        }
    }
returnnull;
}

private void doPagingInternal(ResourceRequest request, ResourceResponse response, String
portletInst,
int windowStateInt, int start, int limit) throws Exception {
    Map<String, Object> jsonMap = new HashMap<String, Object>();
    List recordList = new ArrayList();
int totalCount = 1;
    String editPersonUrl = request.getParameter("editPersonUrl");
    MultiValueMap paramsmap = new MultiValueMap();
    paramsmap.put(NSApiClientConstants.QUERYPARAM_START_ROW, "" + start);
    paramsmap.put(NSApiClientConstants.QUERYPARAM_RECORD_SIZE, "" + limit);
    PersonList personList = nsApiClient.getDirectory().getPeople(paramsmap);
if(personList.getPeople() != null) {
for(Iterator iterator = personList.getPeople().iterator(); iterator.hasNext();) {
    Person portalPerson = (Person) iterator.next();
portalPerson.setPersonURL(StringEscapeUtils.escapeXml(portalPerson.getPersonURL()));
    PortletURL editPersonURL = response.createRenderURL();
editPersonURL.setParameter("personId", "" + portalPerson.getPersonId());
    String firstNameUrl = "<a href='" + editPersonURL.toString() + "'>" +
portalPerson.getFirstName() + "</a>";
    String lastNameUrl = "<a href='" + editPersonURL.toString() + "'>" +
portalPerson.getLastName() + "</a>";
portalPerson.setFirstName(firstNameUrl);
portalPerson.setLastName(lastNameUrl);
recordList.add(portalPerson);
    }
jsonMap.put("success", "true");
jsonMap.put("results", personList.getTotalCount());
jsonMap.put("rows", recordList);
    JSON json = (JSON) JSONSerializer.toJSON(jsonMap);
    String jsonStr = json.toString();

response.setContentType("text/plain");
response.getPortletOutputStream().write(jsonStr.getBytes());
response.getPortletOutputStream().flush();
    }
}

private ModelAndView addPersonData(@ModelAttribute("personData") Person person,
ResourceRequest request, ResourceResponse response) throws Exception {
    Add Person from Form Data in Request
    Map jsonMap = newHashMap();
try {
Person Updateperson = nsApiClient.getDirectory().updatePerson(person);
jsonMap.put("success", "true");
jsonMap.put("successMsg", "Person Added/Updated Successfully");
jsonMap.put("rows", Updateperson);
    JSON json = (JSON) JSONSerializer.toJSON(jsonMap);
    String jsonStr = json.toString();

response.setContentType("text/plain");
response.getPortletOutputStream().write(jsonStr.getBytes());
response.getPortletOutputStream().flush();
    } catch (Exception e) {
e.printStackTrace();
jsonMap.put("success", "false");
jsonMap.put("errorMsg", "Person Add/Update Failed : " + e.getMessage());

```

```
        JSON json2 = (JSON) JSONSerializer.toJSON(jsonMap);
        String jsonStr2 = json2.toString();

        response.setContentType("text/plain");
        response.getPortletOutputStream().write(jsonStr2.getBytes());
        response.getPortletOutputStream().flush();
    }

    return null;
}

private void editPerson(RenderRequest request, RenderResponse response, Model model, int
personId) {
    int person = personId;
    Map<String, Object> jsonMap = new HashMap<String, Object>();
    try {
        Person persons = nsApiClient.getDirectory().getPersonById(person);
        persons.setPersonURL(StringEscapeUtils.escapeXml(persons.getPersonURL()));
        JSON json = (JSON) JSONSerializer.toJSON(persons);
        String jsonStr = json.toString();
        model.addAttribute("PersonData", jsonStr);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private String showAddPersonPage(RenderRequest request, RenderResponse response, Model
model) {
    Person DummyPerson = new Person();
    DummyPerson.setPersonURL(StringEscapeUtils.escapeXml(DummyPerson.getPersonURL()));
    JSON json1 = (JSON) JSONSerializer.toJSON(DummyPerson);
    String jsonStr1 = json1.toString();
    model.addAttribute("PersonData", jsonStr1);

    return viewPageUpdate;
}

public String editNormal(RenderRequest request, RenderResponse response, Model model) {
    try {
        super.editNormal(request, response, model);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return editPage;
}

public String editMinimized(RenderRequest request, RenderResponse response, Model model) {
    try {
        super.editMinimized(request, response, model);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return editPage;
}

public String editMaximized(RenderRequest request, RenderResponse response, Model model) {
    try {
        super.editMaximized(request, response, model);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return editPage;
}
```

```

public String helpNormal(RenderRequest request, RenderResponse response, Model model) {
    try {
        super.helpNormal(request, response, model);
    } catch(Exception e){
        e.printStackTrace();
    }
    return helpPage;
}

public String helpMinimized(RenderRequest request, RenderResponse response, Model model) {
    try {
        super.helpMinimized(request, response, model);
    } catch(Exception e){
        e.printStackTrace();
    }
    return helpPage;
}

public String helpMaximized(RenderRequest request, RenderResponse response, Model model) {
    try {
        super.helpMaximized(request, response, model);
    } catch(Exception e){
        e.printStackTrace();
    }
    return helpPage;
}

private NSApiClient getNSApiClient() {
    return NSApiClientFactory.getInstance();
}
}

```

## MyJSRApplicationContext.xml

ポートレット用の Spring アプリケーション コンテキスト XML。

```

<?xmlversion="1.0"encoding="UTF-8"?>
<beansxmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xmlns:p="http://www.springframework.org/s
chema/p"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-2.5.xsd">

    <beanid="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <propertyname="cache"value="true"/>
    <propertyname="viewClass"
value="org.springframework.web.servlet.view.JstlView"/>
    <propertyname="prefix"value="/WEB-INF/jsp"/>
    <propertyname="suffix"value=".jsp"/>
    </bean>
    <context:annotation-config/>
    <bean
class="org.springframework.web.portlet.mvc.annotation.DefaultAnnotationHandlerMapping">
    <propertyname="interceptors">
    <bean

```



```

class="org.springframework.web.portlet.handler.ParameterMappingInterceptor"/>
</property>
</bean>
<beanid="MyJSRController"class="com.myjsr.MyJSRController">
</bean>
</beans>

```

## jsrportlet.properties

nsAPI で使用する MyJSRerver の URL。



(注)

クラスタ環境でポートレットがサービス カタログ (Service Catalog) アプリケーション URL を参照する場合は、URL を「http://localhost:<port>/RequestCenter」として指定します (<port> はクラスタの各ノードで使用するポート番号)。つまり、URL は「http:<host\_name>/RequestCenter」として指定しないでください (<host\_name> は Web サーバまたはクラスタのいずれかのホストのコンピュータ名)。

```

#(Port number and host has to changed as per the application server).
BASE_URL=http://localhost:8088/RequestCenter

```

## Log4j.properties

```

log4j.rootCategory=INFO, CONSOLE

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%d{ABSOLUTE}%-5p[%c{1}]:%L)%m%n

```

## jboss-deployment-structure.xml

```

<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="javax.portlet" slot="main" export="true"/>
      <module name="org.apache.pluto.container.om" export="true"/>
      <module name="org.apache.pluto.container.driver" export="true"/>
      <module name="org.apache.pluto.tags" export="true"/>
    </dependencies>
  </deployment>
</jboss-deployment-structure>

```

## JSR ポートレットコントローラのコンパイル

ポートレットコントローラをコンパイルするときは、従属ライブラリをクラスパスに含めます。すべてのライブラリのリストについては、「[ポートレットの構造とパッケージ化](#)」を参照してください。

## ポートレットの導入

導入手順は、使用するアプリケーションサーバによって異なります。一般的な注意点として、JSR ポートレットは通常の Web アプリケーションのように、アプリケーションサーバの管理コンソールを使用して導入できます。

JSR ポートレットを導入する方法、および導入後にポータル ページでポートレットを使用する方法の詳細については、『[Cisco Prime サービス カタログ \(Service Catalog\) Designer Guide](#)』を参照してください。