

# Лекция 21

# Портлеты

Портлеты - представляют собой специальный тип сервлета, и они используют JSP-страницы для отображения пользовательского интерфейса.

Portlet API расширяет и является подклассом Servlet API, и это означает, что портлеты могут делать то же, что и сервлеты с некоторыми изменениями при исполнении и дополнительными возможностями.

Наиболее существенное изменение исполнения в том, как обслуживаются запросы: сервлеты обрабатывают запросы "doGet" и "doPost", которые преобразуются в http GET и POST запросы, в то время как портлеты обрабатывают запросы "doView" и "doEdit", которые приходят не напрямую от Web-браузера, а от порталного сервера.

## Возможности портлетов:

- встроенная поддержка автоматического использования различных JSP- страниц для различных пользовательских устройств, таких как настольные компьютеры, Palm-компьютеры с ограниченными Web-браузерами, PDA и мобильные телефоны;
- назначать права пользователям групп на использование портлетов.

В случае отсутствия оных они даже не будут видеть портлеты;

- создание сохраняемых между сессиями пользовательских настроек;
- публикация в виде Web-сервиса;
- разделение сложных приложений на задачи, когда группа тесно связанных задач равняется одному портлету;
- добавление новых функций к приложению;
- хорошая совместимость с брандмауэрами (firewalls), так как портлеты используют стандартные Web-протоколы для получения и отображения информации;
- одноразовая установка и настройка портлета для пользователей.

# Сходства и различия сервлетов и портлетов

Сходства между сервлетами и портлетами:

- относятся к J2EE Web-компонентам;
- управляются контейнерами;
- генерируют динамическое Web-содержимое при помощи запросов и ответов

Различия между сервлетами и портлетами:

- портлеты генерируют часть документа, в то время как сервлеты генерируют его полностью;

- за счёт того, что операции кодирования URL выполняются на стороне сервера, пользователь не может обратиться к нему напрямую, зная имя портлета: портлет - часть страницы, поэтому знания одного URL мало;
- портлеты имеют несколько иную схему управления запросами, которые делятся на запросы выполнения действий и запросы генерирования содержимого;
- портлеты придерживаются стандартного набора состояний, которые определяют их контекст работы и правила генерации содержимого.

# Жизненный цикл

Как и у сервлетов, жизненный цикл портлетов управляется контейнером, и у него есть метод **init()**, который используется для инициализации всех данных, необходимых для корректной работы портлета (создание ресурсов, конфигурирование и т.д.).

Метод **init()** в качестве параметра принимает объект, который реализует интерфейс **PortletConfig**, и этот объект предоставляет необходимые для инициализации параметры.

Он может быть использован для получения ссылки на объект, реализующий интерфейс **PortletContext**.



При создании портлета доступа к окружающему коду, например к контейнеру, нет, поэтому код внутри портлета не может оценить, насколько портлет доступен извне.

Метод **destroy()** предоставляет возможность для произведения очистки ресурсов, которые были востребованы и инициализированы методом **init()**.

Этот метод аналогичен методу **destroy()** в сервлетах и вызывается один раз: когда контейнер выгружает портлет.

# Состояния

Состояния портлетов – это часть порталльной модели отображения.

Состояния позволяют портлету отображать различные «виды» в зависимости от ситуации.

Есть четыре состояния портлета:

- **View** – основное состояние портлета;
- **Help** – если портлет обеспечивает состояние помощи;
- **Edit** – редактирование параметров портлета с сохранением результатов для этого конкретного пользователя;
- **Configure** – конфигурирование портлета с сохранением результатов для всех пользователей, права к состояниям никак не относятся.

Рассмотрим пример простого портлета.

Класс **SimplePortlet**.

Этот класс, собственно, и является основным классом портлета.

Все, что здесь происходит – это инициализация портлета в методе **init(PortletConfig config)**, за каждое из представлений портлета отвечают методы **doEdit()**, **doView()**, **doHelp()**.

В методе **processAction()** производится обработка событий портлета (в данном случае этот метод будет вызван при подтверждении пользователем желания отредактировать настройки портлета).

```
package aaa;  
import java.io.IOException;  
import javax.portlet.ActionRequest;  
import javax.portlet.ActionResponse;  
import javax.portlet.GenericPortlet;  
import javax.portlet.PortletConfig;  
import javax.portlet.PortletContext;  
import javax.portlet.PortletException;  
import javax.portlet.PortletMode;  
import javax.portlet.PortletPreferences;  
import javax.portlet.PortletRequest;  
import javax.portlet.PortletRequestDispatcher;  
import javax.portlet.RenderRequest;  
import javax.portlet.RenderResponse;
```

```
public class SimplePortlet extends GenericPortlet {  
    private static final String EDIT_PAGE_PARAM  
                                = "Edit-Page";  
    private static final String HELP_PAGE_PARAM  
                                = "Help-Page";  
    private static final String VIEW_PAGE_PARAM  
                                = "View-Page";  
    public void init(PortletConfig config) throws  
                    PortletException {  
        super.init(config); // инициализация  
    }  
}
```

*// метод, отвечающий за представление страницы*

*редактирования*

```
public void doEdit(RenderRequest request,  
                    RenderResponse response)  
                    throws PortletException,IOException {  
  
// получение контекста портлета  
PortletContext context = getPortletContext();  
  
setRequestAttributes(request); // установка атрибутов  
// получение диспатчера для включения ресурсов в response  
PortletRequestDispatcher rd = context.  
    getRequestDispatcher(getInitParameter(EDIT_PAGE_PARAM));  
  
rd.include(request, response); // включение содержимого ресурса  
}
```

```
// метод, отвечающий за представление страницы помощи  
public void doHelp(RenderRequest request,  
                    RenderResponse response)  
    throws PortletException, IOException {  
// получение контекста портлета  
    PortletContext context = getPortletContext();  
    setRequestAttributes(request); // установка атрибута  
  
// получение диспатчера для включения ресурсов в response  
    PortletRequestDispatcher rd = context.  
        getRequestDispatcher(getInitParameter(HELP_PAGE_PARAM));  
    rd.include(request, response); // включение содержимого ресурса  
}
```

```
// метод, отвечающий за представление страницы просмотра  
public void doView(RenderRequest request,  
                    RenderResponse response)  
                    throws PortletException, IOException {  
// получение контекста портлета  
PortletContext context = getPortletContext();  
  
setRequestAttributes(request); // устанавливаем атрибуты  
// получение диспатчера для включения ресурсов в response  
PortletRequestDispatcher rd = context.  
    getRequestDispatcher(getInitParameter(VIEW_PAGE_PARAM));  
rd.include(request, response); // включение содержимого ресурса  
}
```



*// вызывается контейнером для обработки событий*

```
public void processAction(ActionRequest request,  
                          ActionResponse response)  
    throws PortletException, IOException {  
    PortletMode mode = request.getPortletMode(); // получение состояния  
    PortletPreferences preferences = request.getPreferences(); //настройки  
    if (mode.equals(PortletMode.VIEW)) {  
        // сохранение настроек  
        preferences.setValue("firstName", request.getParameter("firstName"));  
        preferences.setValue("lastName", request.getParameter("lastName"));  
        preferences.setValue("address", request.getParameter("address"));  
        preferences.setValue("telephone", request.getParameter("telephone"));  
        preferences.store();  
    }  
}
```

*// для установки атрибутов*

```
private void setRequestAttributes(PortletRequest request) {  
    PortletPreferences preferences = request.getPreferences();  
    request.setAttribute("firstName",  
        preferences.getValue("firstName", "undefined"));  
    request.setAttribute("lastName",  
        preferences.getValue("lastName", "undefined"));  
    request.setAttribute("address",  
        preferences.getValue("address", "undefined"));  
    request.setAttribute("telephone",  
        preferences.getValue("telephone", "undefined"));  
    request.setAttribute("portletName", getPortletName());  
}  
}
```

Согласно спецификации для развертывания портлета необходим файл **portlet.xml**.

В данном случае этот файл имеет вид:

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns=
"http://java.sun.com/xml/ns/portlet/portlet-
app_1_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">
```

```
<portlet id="simple-application">
```

```
<!-- описание -->
```

```
<description>Portlet Application</description>
```

```
<!-- название -->
```

```
<portlet-name>SimplePortlet</portlet-name>
```

```
<!-- отображаемое имя -->
```

```
<display-name>Simple Portlet</display-name>
```

```
<!-- класс портлета -->
```

```
<portlet-class>
```

```
aaa.SimplePortlet
```

```
</portlet-class>
```

*<!-- параметры инициализации -->*

**<init-param>**

**<name>View-Page</name>**

**<value>/WEB-INF/jsp/simple-view.jsp</value>**

**</init-param>**

**<init-param>**

**<name>Help-Page</name>**

**<value>/WEB-INF/jsp/simple-help.jsp</value>**

**</init-param>**

**<init-param>**

**<name>Edit-Page</name>**

**<value>/WEB-INF/jsp/simple-edit.jsp</value>**

**</init-param>**

*<!-- количество секунд, на которое  
кешируется портлет. -1 значит cache  
не истекает -->*

**<expiration-cache>-1</expiration-cache>**

*<!-- поддерживаемые режимы -->*

**<supports>**

**<mime-type>text/html</mime-type>**

**<portlet-mode>view</portlet-mode>**

**<portlet-mode>help</portlet-mode>**

**<portlet-mode>edit</portlet-mode>**

**</supports>**

*<!-- локализация -->*

**<supported-locale>en</supported-locale>**

*<!-- текстовые ресурсы -->*

**<resource-bundle>**

**com.learning.portlet.SimplePortlet**

**</resource-bundle>**

*<!-- информация -->*

**<portlet-info>**

*<!-- заголовок -->*

**<title>Portlet Application</title>**

**<short-title>Portlet</short-title>**

*<!-- ключевые слова -->*

**<keywords>portlet</keywords>**

**</portlet-info>**

**</portlet>**

**</portlet-app>**

Файл **web.xml** является дескриптором web-приложения, поскольку портлет-приложение является и web-приложением.

Данный файл имеет вид:

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee  
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"  
    version="2.4">  
  <display-name>Simple</display-name>  
</web-app>
```



Рассмотрим соответствующие jsp страницы

simple-edit.jsp

```
<%@ taglib prefix="c"
        uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt"
        uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@ taglib prefix="portlet"
        uri="http://java.sun.com/portlet"%>
```

*//Данный тег определяет три объекта  
renderRequest, renderResponse и portletConfig для  
использования в “портлетных” jsp страницах*

```
<portlet:defineObjects/>
```

*//связываем действие с модой view*

```
<portlet:actionURL var="editAction"  
                    portletMode="view"/>
```

```
<c:set var="inputTextStyle" value="border: none;  
                                width:150px; color: #000000;"/>
```

```
<c:set var="inputSubmitStyle" value=  
                                "border:none; width: 75px;"/>
```

```
<c:set var="labelStyle" value="text-align: right;  
                                color:#000000; white-space: nowrap;"/>
```

*//Подключаем внешний ресурс*

```
<fmt:setBundle basename="aaa.SimplePortlet"/>
<form action="{editAction}" method="post">
  <table style="border: none;">
    <tbody>
      <tr>
        <td style="{labelStyle}">
          <fmt:message key="portlet.label.firstname"/>:
        </td>
        <td nowrap="nowrap">
          <input type="text" name="firstName"
            value="{firstName}" style="{inputTextStyle}"/>
        </td>
      </tr>
    </tbody>
  </table>
</form>
```

```
<tr>
  <td style="{labelStyle}">
    <fmt:message key="portlet.label.lastname"/>:
  </td>
  <td nowrap="nowrap">
    <input type="text" name="lastName" value="{lastName}"
      style="{inputTextStyle}"/>
  </td>
</tr>
<tr>
  <td style="{labelStyle}">
    <fmt:message key="portlet.label.address"/>:
  </td>
  <td nowrap="nowrap">
    <input type="text" name="address" value="{address}"
      style="{inputTextStyle}"/>
  </td>
</tr>
```

```
<tr>
  <td style="{labelStyle}">
    <fmt:message key="portlet.label.telephone"/>:
  </td>
  <td nowrap="nowrap">
    <input type="text" name="telephone" value="{telephone}"
      style="{inputTextStyle}"/>
  </td>
</tr>
<tr>
  <td colspan="2" align="right">
    <button type="submit" style="{inputSubmitStyle}">
      <fmt:message key="portlet.button.submit"/>
    </button>
  </td>
</tr>
</tbody>
</table>
</form>
```

sample-help.jsp

**<%@ taglib prefix="fmt"**

**uri="http://java.sun.com/jsp/jstl/fmt"%>**

**<%@ taglib prefix="portlet"**

**uri="http://java.sun.com/portlet"%>**

**<portlet:defineObjects/>**

**<fmt:setBundle basename="aaa.SimplePortlet"/>**

**<table>**

**<tbody>**



```
<tr>
```

```
  <td align="center">
```

```
    <portlet:renderURL var="viewUrl" portletMode="view"/>
```

```
    <a href="{viewUrl}" style="text-decoration: none;">
```

```
      <fmt:message key="portlet.viewpage.link"/>.
```

```
    </a>
```

```
  </td>
```

```
  <td align="center">
```

```
    //связываем render с модой edit
```

```
    <portlet:renderURL var="editUrl" portletMode="edit"/>
```

```
    <a href="{editUrl}" style="text-decoration: none;">
```

```
      <fmt:message key="portlet.editpage.link"/>.
```

```
    </a>
```

```
  </td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```



simple-view.jsp

```
<%@ taglib prefix="c"
```

```
    uri="http://java.sun.com/jsp/jstl/core"%>
```

```
<%@ taglib prefix="fmt"
```

```
    uri="http://java.sun.com/jsp/jstl/fmt"%>
```

```
<%@ taglib prefix="portlet"
```

```
    uri="http://java.sun.com/portlet"%>
```

```
<portlet:defineObjects/>
```

```
<c:set var="labelStyle" value="text-align: right;
```

```
    color:#000000; white-space: nowrap;"/>
```

```
<c:set var="textStyle" value="text-align: left;
```

```
    color: #000000; white-space: nowrap;"/>
```

```
<fmt:setBundle basename=  
    "com.learning.portlet.SamplePortlet"/>
```

```
<table>
```

```
<tbody style="background-color: ">
```

```
<tr>
```

```
<td style="{labelStyle}">
```

```
<fmt:message key="portlet.label.firstname"/>:
```

```
</td>
```

```
<td style="{textStyle}">{firstName}</td>
```

```
</tr>
```

```
<tr>
  <td style="{labelStyle}">
    <fmt:message key="portlet.label.lastname"/>:
  </td>
  <td style="{textStyle}">{lastName}</td>
</tr>
```

```
<tr>
  <td style="{labelStyle}">
    <fmt:message key="portlet.label.address"/>:
  </td>
  <td style="{textStyle}">{address}</td>
</tr>
```

```
<tr>
```

```
  <td style="{labelStyle}">
```

```
    <fmt:message key="portlet.label.telephone"/>:
```

```
  </td>
```

```
  <td style="{textStyle}">{telephone}</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

файл свойств для английского языка :

*SamplePortlet\_en.txt*

**portlet.label.firstname** = First name

**portlet.label.lastname** = Last name

**portlet.label.address** = Address

**portlet.label.telephone** = Telephone number

**portlet.button.submit** = Store

**portlet.name.message** = This is the simple help page  
of the portlet

**<span style="font-weight: bolder"> {0} </span>.**

**portlet.storeduser.message** = Stored user is

**<span style="font-weight: bolder"> {0} {1}</span>.**

**portlet.viewpage.link** = View page

**portlet.editpage.link** = Edit page

файл свойств для русского языка:

*SamplePortlet\_ru.txt*

**portlet.label.firstname** = Имя

**portlet.label.lastname** = Фамилия

**portlet.label.address** = Адрес

**portlet.label.telephone** = Номер телефона

**portlet.button.submit** = Сохранить

**portlet.name.message** = Страница помощи портлета  
**<span style="font-weight: bolder">{0}</span>.**

**portlet.storeduser.message** = Сохраненный

пользователь

**<span style="font-weight: bolder"> {0} {1}</span>.**

**portlet.viewpage.link** = Смотреть

**portlet.editpage.link** = Редактировать

Рассмотрим развертывание приложения с использованием порталного сервера jetspeed

1. Создать директорию temp(имя не имеет значение)

2. Создать в ней поддиректорию src

3. В src создать структуру каталогов

src->config ->(web.xml; portlet.xml)

src-> java ->SimplePortlet.java

src->jsp->(simple-view.jsp; simple-help.jsp;  
simple-edit.jsp)

src-> lib ->(portlet-api\_2.0\_spec-1.0.jar; jstl.jar)

src->message->aaa->(SimplePortlet\_ru.txt;  
SimplePortlet\_en.txt)

src-> tld->(portlet\_2\_0.tld; portlet.tld; fmt.tld)

src->build.xml

Стоит заметить, что файлы portlet\_2\_0.tld; portlet.tld можно найти в пакете

portlet-api\_2.0\_spec-1.0.jar, а fmt.tld в пакете jstl.jar

Файл build.xml имеет вид:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project name="SamplePortlet" default="war" basedir=".">
```

```
<property name="src.dir" value="src"/>
```

```
<property name="jsp.dir" value="jsp"/>
```

```
<property name="java.dir" value="java"/>
```

```
<property name="message.dir" value="message"/>
```

```
<property name="config.dir" value="config"/>
```

```
<property name="lib.dir" value="lib"/>
```

```
<property name="tld.dir" value="tld"/>
```



```
<property name="src.lib.dir" value="\${src.dir}/\${lib.dir}"/>
<property name="src.java.dir" value="\${src.dir}/\${java.dir}"/>
<property name="src.jsp.dir" value="\${src.dir}/\${jsp.dir}"/>
<property name="src.message.dir"
           value="\${src.dir}/\${message.dir}"/>
<property name="src.config.dir" value="\${src.dir}/\${config.dir}"/>
<property name="src.tld.dir" value="\${src.dir}/\${tld.dir}"/>
<property name="webapp.dir" value="WEB-INF"/>
<property name="result.dir" value="result"/>
<property name="classes.dir" value="classes"/>
<property name="project.name" value="SimplePortlet"/>
<property name="localization.encoding" value="Cp1251"/>
<property name="localization.src.dir"
           value="\${src.message.dir}/aaa"/>
<property name="localization.result.dir"
           value="\${result.dir}/\${classes.dir}/aaa"/>
<property name="localization.ext" value=".properties"/>
```

```
<target name="compile" description="Compiles all source files of the
portlet.">
  <mkdir dir="${result.dir}/${classes.dir}"/>
    <javac srcdir="${src.java.dir}" destdir= "${result.dir}/${classes.dir}"
          optimize="yes" debug="no">
      <classpath>
        <fileset dir="${src.lib.dir}">
          <include name="**/*.jar"/>
        </fileset>
      </classpath>
    </javac>

    <copy todir="${result.dir}/${classes.dir}">
      <fileset dir="${src.message.dir}" excludes="**/*.txt"/>
    </copy>
    <native2ascii src="${localization.src.dir}" dest="${localization.result.dir}"
                  includes="**/*.txt" encoding="${localization.encoding}"
                  ext="${localization.ext}"/>
  </target>
```

```
<target name="war" depends="compile"
    description="Creates .war file of the portlet.">
<war destfile="${result.dir}/${project.name}.war"
    webxml="${src.config.dir}/web.xml" compress="on">
    <lib dir="${src.lib.dir}">
        <exclude name="**/portlet-api_2.0_spec-1.0.jar"/>
    </lib>
    <classes dir="${result.dir}/${classes.dir}"/>

    <zipfileset dir="${src.config.dir}"
                prefix="${webapp.dir}">
        <include name="**/*.*/>
        <exclude name="**/web.xml"/>
    </zipfileset>
```

```
<zipfileset dir="${src.jsp.dir}"  
            prefix="${webapp.dir}/${jsp.dir}"/>  
<zipfileset dir="${src.tld.dir}"  
            prefix="${webapp.dir}/${tld.dir}"/>
```

```
</war>
```

```
</target>
```

```
</project>
```

4. Запустить из директории temp ant.

5. Установить jetspeed, для этого необходимо скачать jetspeed-installer-2.2.2.jar

Затем запустить java -jar jetspeed-installer-2.2.2.jar

(перед запуском инсталлятора должна быть установлена база данных)

6. Перекопировать файл из директории

temp -> result -> SimplePortlet.war

в директорию

\$jetspeed\_root\_directory->webapps->jetspeed

->WEB-INF -> deploy

7. Запустить сервер jetspeed

\$jetspeed\_root\_directory->bin->startup.bat

8. Загрузить стартовую страницу

<http://localhost:8080/jetspeed/>

9. Зайти как администратор

обычно

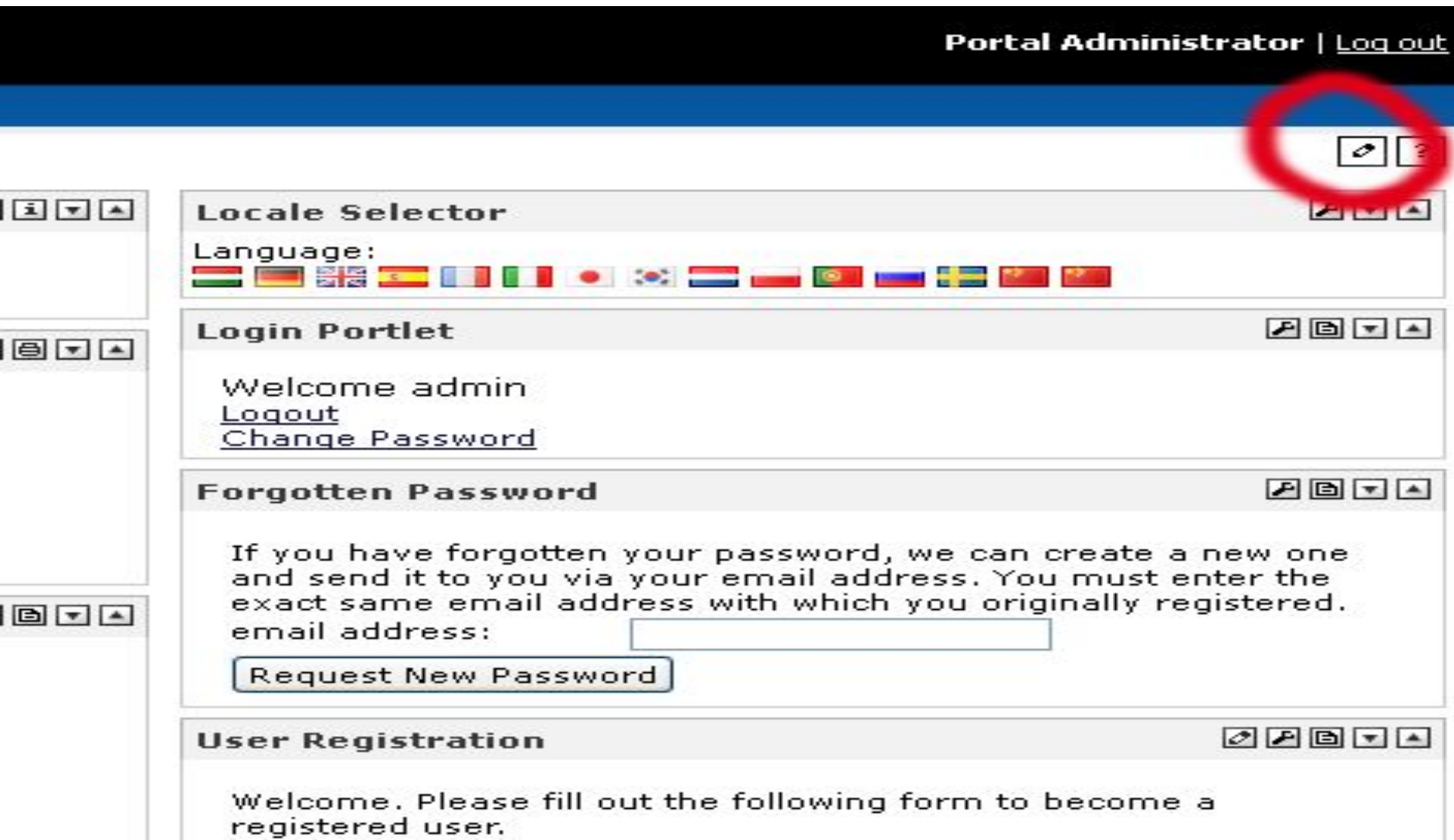
Login: admin

Password: admin

# 10. Добавить портлет

Для этого необходимо щелкнуть на кнопке в верхнем правом углу

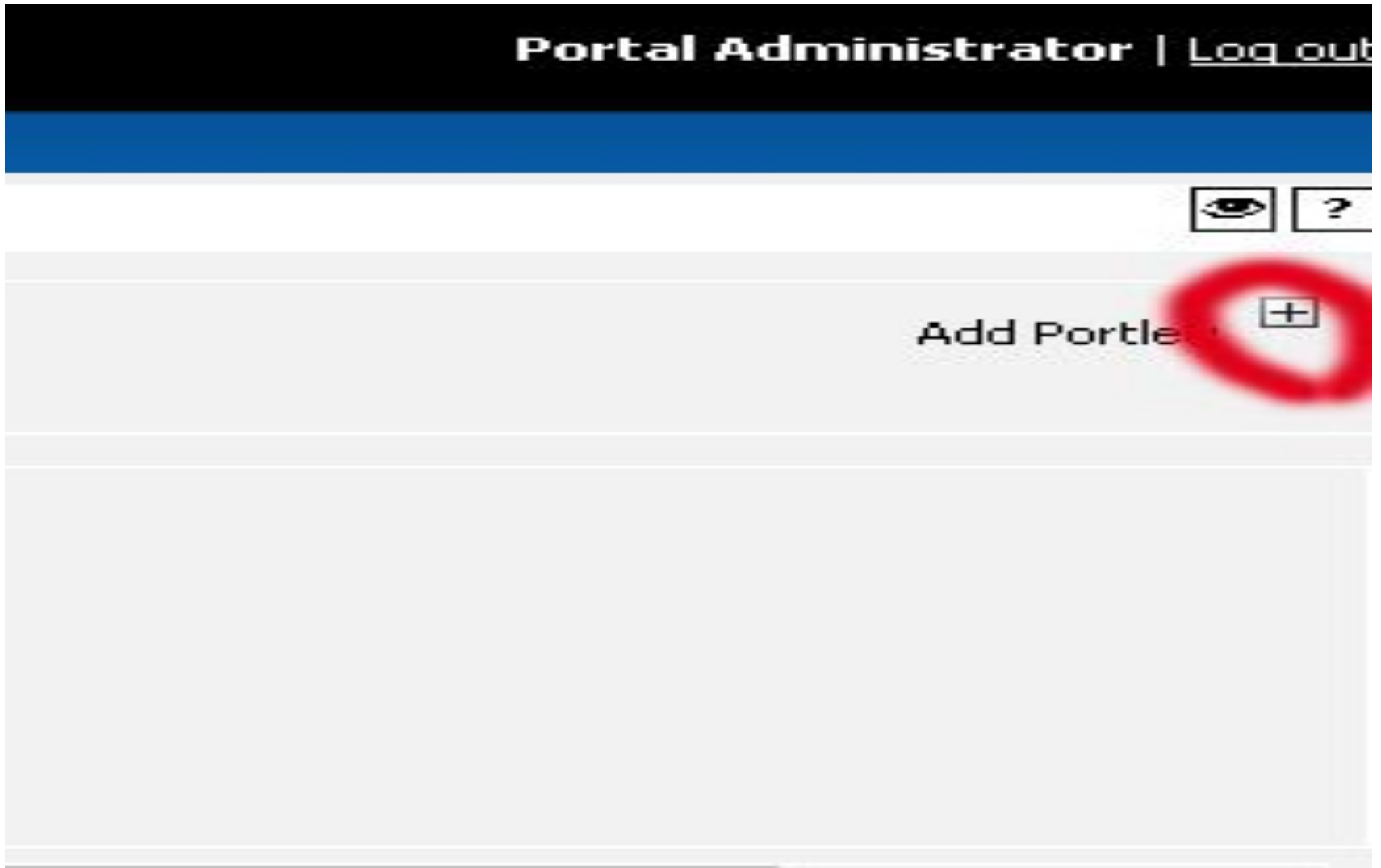
Portal Administrator | [Log out](#)



The screenshot shows the Portal Administrator interface. At the top right, there is a navigation bar with the text "Portal Administrator | [Log out](#)". Below this, there is a blue header bar. In the top right corner of the main content area, there is a red circle highlighting a button with a pencil icon and a question mark icon. Below the header, there are several portlets:

- Locale Selector**: A portlet for selecting the language. It includes a "Language:" label and a row of flags representing various languages: Hungarian, German, British English, Spanish, French, Italian, Japanese, Korean, Dutch, Polish, Portuguese, Russian, Swedish, Chinese (Simplified), and Chinese (Traditional). There are also icons for edit, save, and expand in the top right corner.
- Login Portlet**: A portlet for user login. It displays "Welcome admin" and provides links for [Logout](#) and [Change Password](#). It has icons for edit, save, and expand in the top right corner.
- Forgotten Password**: A portlet for password recovery. It contains the text: "If you have forgotten your password, we can create a new one and send it to you via your email address. You must enter the exact same email address with which you originally registered. email address:" followed by an input field and a "Request New Password" button. It has icons for edit, save, and expand in the top right corner.
- User Registration**: A portlet for user registration. It contains the text: "Welcome. Please fill out the following form to become a registered user." It has icons for edit, save, and expand in the top right corner.

Затем снова нажать на кнопку “+”, рядом с надписью Add Portlet



Затем в окне поиска ввести название SimplePortlet и после этого добавить портлет и нажать на кнопку в правом верхнем углу с изображением глаза. Как результат появится портлет:

The screenshot displays the Jetspeed web application interface. The top navigation bar includes links for "Demo", "Welcome to Jetspeed 2", "About", "SSO Demo", "RSS Feeds", and "SimplePortlets". The main content area is titled "Guest Space >> Welcome to Jetspeed 2".

On the left side, there are several menu sections:

- Users**: Jetspeed Administration, Public Space
- Pages**: Jetspeed Administration
- Filed Pages**: SSO Demo
- Additional Links**: Jetspeed 2 Home Page, Jetspeed 2 Wiki, Apache Software Foundation, Apache Portals, Portlet

The main content area contains three portlets:

- Portlet Application**: This is the simple help page of Stored user is **ssss ggggg**. View page. Edit page. (A red circle highlights the top-right corner of this portlet, containing icons for edit, info, dropdown, and up/down arrows.)
- Pick a Number**: Pick a Number Guess Count Game. You have made 0 guess thus far. Enter a number between 1 and 100. Ready to start a New game. (Input field with "0" and "Guess" button). Help Edit Max Min Normal Redirect Test
- Role Security Test**: User roles. User: admin. isUserInRole: (Info, edit, dropdown, up/down arrows icons)



Другой способ добавления портлета – это  
отредактировать файл **default-page.psmi**

находящийся в директории

`$jetspeed_root_directory->pages`

добавив следующее:

```
<fragment id="db-200" type="portlet"  
    name="SimplePortlet::SimplePortlet">  
<property name="row" value="0"></property>  
<property name="column" value="0"></property>  
</fragment>
```