

Лабораторная работа №5

Основы каркаса JSF

Цель

Изучить базовые возможности каркаса JavaServer Faces для разработки веб-приложений (организация интерфейса пользователя, обработка событий, навигация) и получить навыки реализации веб-приложений с использованием данного каркаса.

Задание

1. Разработать веб-приложение, состоящее из двух страниц: одна страница должна обеспечивать ввод данных, а вторая – отображение результатов их обработки и возврат на первую страницу.
2. Данные должны быть трех типов: строковые, числовые и дата/время.
3. Для сбора и обработки данных должен использоваться Managed Bean контекста запроса.
4. Обработка данных заключается в вычислении значений некоторых свойств Managed Bean по введенным данным.
5. В приложении должен также использоваться Managed Bean контекста сессии (например, для подсчета количества обращений к первой странице).
6. В приложении должна быть предусмотрена проверка (валидация) входных данных: строковых - на соответствие шаблону, числовых – на принадлежность диапазону. В случае ошибки валидации на странице должно выводиться соответствующее сообщение.

Порядок выполнения работы

1. Создайте в среде разработки NetBeans новый проект типа **Web / Web Application**. Введите в поле **Project Name** название проекта. Остальные настройки рекомендуется выполнить так, как показано на рис. 1. Нажмите кнопку **Next**. Установите флажок для каркаса JavaServer Faces (рис. 2). Нажмите кнопку **Finish**.
2. Создайте новый класс Managed Bean (файл типа **JavaServer Faces / JSF Managed Bean**) контекста сессии. Укажите имя класса в поле Class Name, пакет в поле Package, выберите в списке Scope контекст сессии (*session*). Нажмите кнопку **Finish**. При этом помимо создания класса Managed Bean в конфигурационный файл JSF (*faces-config.xml*) добавляется определение Managed Bean. Откройте файл и изучите структуру определения Managed Bean.
3. Добавьте в созданный класс Managed Bean необходимые свойства (поля и соответствующие get/set-методы).
4. Создайте новый класс Managed Bean (файл типа **JavaServer Faces / JSF Managed Bean**) для обработки запроса. Укажите имя класса в поле Class Name, пакет в поле Package, выберите в списке Scope контекст запроса (*request*). Нажмите кнопку **Finish**.
5. Добавьте в класс Managed Bean, созданный в п.4, свойства (поля и соответствующие get/set-методы) и action-метод, который будет вызываться для обработки действия пользователя. Action-метод не имеет параметров и возвращает значение типа String – строку, которая используется для определения следующей страницы по правилам навигации. Также добавьте в класс свойство, в качестве типа которого выступает класс Managed Bean контекста сессии (созданный в п.2).

New Web Application

Steps

1. Choose Project
- 2. Name and Location**
3. Frameworks

Name and Location

Project Name: Clients

Project Location: E:\dev\workspaces\nb6 Browse...

Project Folder: E:\dev\workspaces\nb6\Clients

Add to Enterprise Application: <None>

Server: Apache Tomcat 6.0.14 Add...

Java EE Version: Java EE 5

Context Path: /Clients

☒ Set as Main Project

< Back Next > Finish Cancel Help

Рис. 1. Настройка нового проекта веб-приложения

New Web Application

Steps

1. Choose Project
2. Name and Location
- 3. Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

☐ Visual Web JavaServer Faces

☒ JavaServer Faces

☐ Struts 1.2.9

JavaServer Faces Configuration

Configuration Libraries

JSF Servlet Name: Faces Servlet

Servlet URL Mapping: /faces/*

☒ Validate XML ☐ Verify Objects

< Back Next > Finish Cancel Help

Рис. 2. Настройка использования каркаса JSF

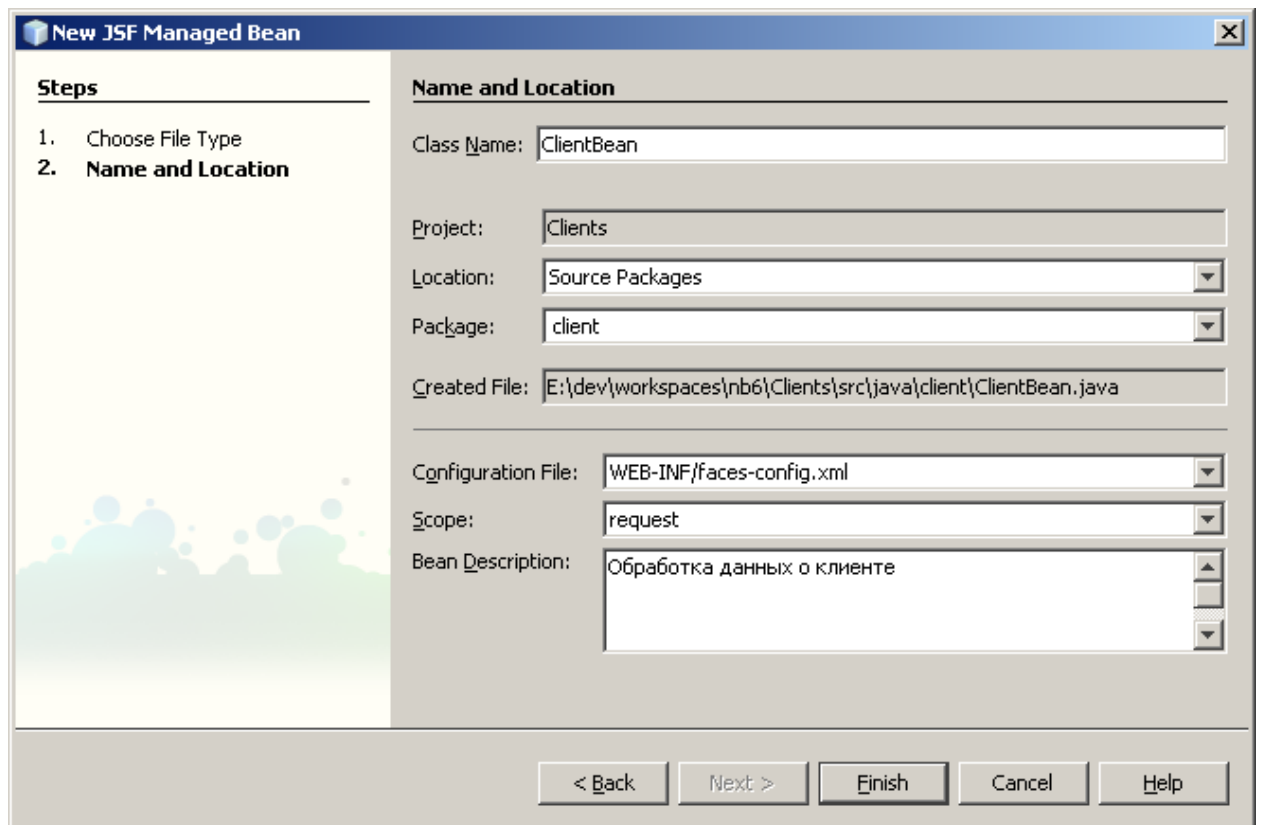


Рис. 3. Создание класса Managed Bean

6. Откройте файл `faces-config.xml` и добавьте инициализацию свойств для Managed Bean контекста запроса. Пример инициализации ссылки на Managed Bean контекста сессии (инициализируется свойство с названием `sessionBean`, для инициализации используется отложенное EL-выражение):

```
<managed-bean>
  <managed-bean-name>ClientBean</managed-bean-name>
  <managed-bean-class>client.ClientBean</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
  <managed-property>
    <property-name>sessionBean</property-name>
    <value>#{ClientSessionBean}</value>
  </managed-property>
</managed-bean>
```

7. Создайте JSP-страницы. Не забудьте добавить в каждую страницу следующие директивы `taglib` для подключения базовых действий JSF и стандартного набора рендереров JSF:


```
<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
```
8. Добавьте в созданные страницы корневой элемент `<f:view>`, форму `<h:form>`, поля ввода и вывода данных (`<h:inputText>` и `<h:outputText>`), кнопку для выполнения действия (`<h:commandButton>` или `<h:commandLink>`). Установите обязательные атрибуты для данных действий.
9. Добавьте для полей ввода данных действия валидации (например, `<f:validateLongRange>` для проверки целых чисел на вхождение в диапазон). Для поля ввода даты добавьте конвертер (действие `<f:convertDateTime>`).
10. Свяжите между собой страницы с помощью закладки PageFlow редактора файла `faces-config.xml` (рис. 4).

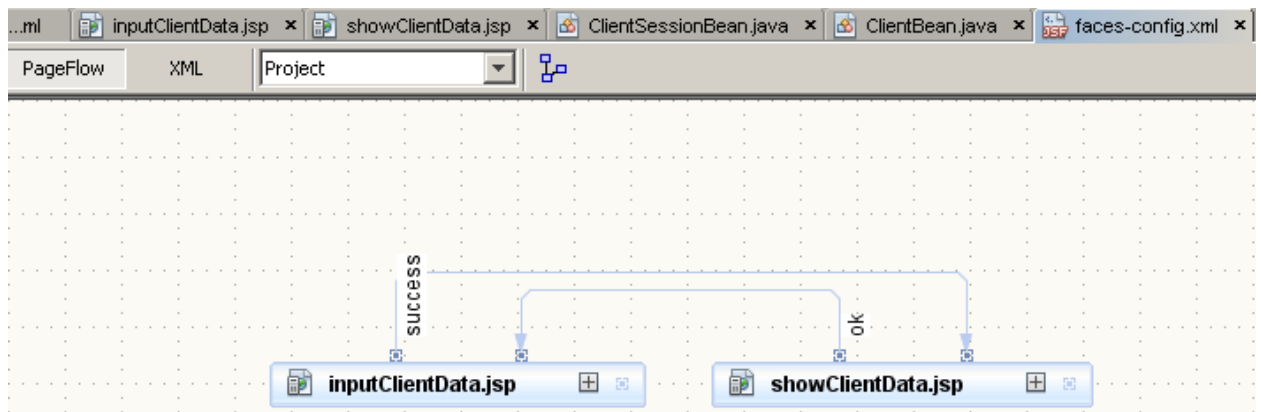


Рис. 4. Редактор правил навигации

11. Запустите веб-приложение, выбрав в контекстном меню проекта пункт **Run Project**. Убедитесь в правильном функционировании созданных страниц. Выполните несколько запросов, в том числе с неправильными данными. Просмотрите в окне HTTP Monitor обработанные запросы.