

Фильтры. JSF.

№ урока: 8 **Курс:** Основы Java EE

Средства обучения: IntelliJ Idea

Обзор, цель и назначение урока

Знакомство с фильтрами. Использование фильтров. Создание пользовательских интерфейсов с помощью JSF. Создание CDI биннов.

Изучив материал данного занятия, учащийся сможет:

- Научиться использовать фильтры.
- Создавать пользовательские интерфейсы со значениями объектов (биннов).
- Обработать формы с использованием объектов (биннов).
- Создавать интерфейсы с помощью Facelets.

Содержание урока

1. Введение в фильтры.
2. Начало работы с фильтрами.
3. Использование фильтра при регистрации.
4. Введение в JSF.
5. Создание бина.
6. Рассмотрение Facelets.
7. Получение списка биннов.

Резюме

- **Фильтры** в Java позволяют вмешаться в обработку запроса как до передачи запроса в jsp-страницу, так и после.
Типичное применение: фильтрация запросов на входе, аутентификация пользователя или фильтрация результатов работы страницы jsp.
Фильтры проектируются так, чтобы быть максимально независимыми от других частей приложения и от других фильтров. Т.е. следует в фильтры выносить наиболее общие функции по обработке.
Фильтры можно инициализировать в web.xml определенными параметрами при помощи web.xml.
- **JavaServer Faces (JSF)** — это Java спецификация для построения компонентно-ориентированных пользовательских интерфейсов для веб-приложений, написанный на Java. Он служит для того, чтобы облегчать разработку пользовательских интерфейсов для Java EE-приложений. В отличие от прочих MVC-фреймворков, которые управляются запросами, подход JSF основывается на использовании компонентов. Состояние компонентов пользовательского интерфейса сохраняется, когда пользователь запрашивает новую страницу и затем восстанавливается, если запрос повторяется. Для отображения данных обычно используется JSP, Facelets, но JSF можно приспособить и под другие технологии, например XUL.
Технология **JavaServer Faces** включает:
Набор API для представления компонентов пользовательского интерфейса (UI) и управления их состоянием, обработкой событий и валидацией вводимой информации, определения навигации, а также поддержку интернационализации (i18n) и доступности (accessibility).
Специальная библиотека JSP тегов для выражения интерфейса JSF на JSP странице. В JSF 2.0 в качестве обработчика представления используется технология Facelets которая пришла на замену JSP.

Созданная быть гибкой, технология JavaServer Faces усиливает существующие стандартные концепции пользовательского интерфейса (UI) и концепции Web-уровня без привязки разработчика к конкретному языку разметки, протоколу или клиентскому устройству. Классы компонентов пользовательского интерфейса, поставляемые вместе с технологией JavaServer Faces, содержат функциональность компонент, а не специфичное для клиента отображение, открывая тем самым возможность рендеринга JSF-компонент на различных клиентских устройствах. Совмещая функциональность компонент интерфейса пользователя со специальными рендерерами, разработчики могут конструировать специальные теги для заданного клиентского устройства. В качестве удобства технология JSF предоставляет специфичный рендерер и специальную библиотеку JSP-тегов для рендеринга на HTML-клиенте, позволяя разработчикам приложений на J2EE платформе использовать технологию JSF в своих приложениях.

- **JavaBeans** — классы в языке Java, написанные по определённым правилам. Они используются для объединения нескольких объектов в один (англ. bean — фасоль, кофейное зерно) для удобной передачи данных.

Спецификация Sun Microsystems определяет JavaBeans как повторно используемые программные компоненты, которыми можно управлять, используя графические конструкторы и средства IDE.

JavaBeans обеспечивают основу для многократно используемых, встраиваемых и модульных компонентов ПО. Компоненты JavaBeans могут принимать различные формы, но наиболее широко они применяются в элементах графического пользовательского интерфейса. Одна из целей создания JavaBeans — взаимодействие с похожими компонентными структурами. Например, Windows-программа, при наличии соответствующего моста или объекта-обёртки, может использовать компонент JavaBeans так, будто бы он является компонентом COM или ActiveX.

Чтобы класс мог работать как bean, он должен соответствовать определённым соглашениям об именах методов, конструкторе и поведении. Эти соглашения дают возможность создания инструментов, которые могут использовать, замещать и соединять JavaBeans.

Правила описания гласят:

- Класс должен иметь конструктор без параметров, с модификатором доступа `public`. Такой конструктор позволяет инструментам создать объект без дополнительных сложностей с параметрами.
- Свойства класса должны быть доступны через `get`, `set` и другие методы (так называемые методы доступа), которые должны подчиняться стандартному соглашению об именах. Это легко позволяет инструментам автоматически определять и обновлять содержание bean'ов. Многие инструменты даже имеют специализированные редакторы для различных типов свойств.
- Класс должен быть сериализуем. Это даёт возможность надёжно сохранять, хранить и восстанавливать состояние bean независимым от платформы и виртуальной машины способом.

Класс должен иметь переопределённые методы `equals()`, `hashCode()` и `toString()`.

Так как требования в основном изложены в виде соглашения, а не интерфейса, некоторые разработчики рассматривают JavaBeans, как Plain Old Java Objects, которые следуют определённым правилам именования.

- **Facelets** — открытый веб-фреймворк, распространяемый под лицензией Apache license. Технология управления представлением для JavaServer Faces (JSF), альтернатива технологии JavaServer Pages (JSP). Фреймворк требует для функционирования валидные XML документы. Это означает, что веб-страницы должны быть созданы с использованием языка разметки XHTML. Facelets поддерживает все компоненты JSF и создаёт собственное дерево компонент, отражая **view** (см. MVC) для JSF-приложения. В Facelets полностью устранена широко известная проблема смешанного контента, присущая JavaServer Pages.

Закрепление материала

- Что такое фильтры и зачем они нужны?
- Что такое JSF?

- Что такое бин и зачем он нужен?
- Зачем применять Facelets в JSF?

Дополнительное задание

Задание

Сделать форму регистрации клиента. При успешной регистрации на странице отображать всех зарегистрированных пользователей (имя, возраст, телефон).

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

К проекту с предыдущих уроков добавить фильтр для перехвата регистрации клиента, в случае, если клиент существует, перенаправить на домашнюю страницу и вывести сообщение, что клиент существует.

Рекомендуемые ресурсы

JSF

<http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

Facelets

<http://www.oracle.com/technetwork/articles/java/facelets-454361.html>

Filter

<http://www.oracle.com/technetwork/java/filters-137243.html>