**JSF пример welcome**

**http://java-online.ru/jsf-welcome.xhtml**

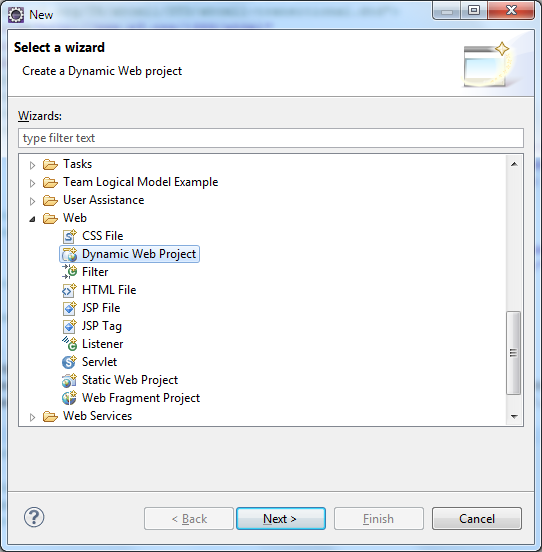
В данной статье рассматривается создание простейшего WEB-приложения с использованием фреймворка JSF. Для разработки jsf примера будем использовать JDK 1.7, apache-tomcat-7.xx и среду разработки **Eclipse**. В качестве среды разработки можно использовать любую другую IDE (Integrated Development Enviroment). Последовательность действий не изменится.

**Разработка jsf примера welcome**

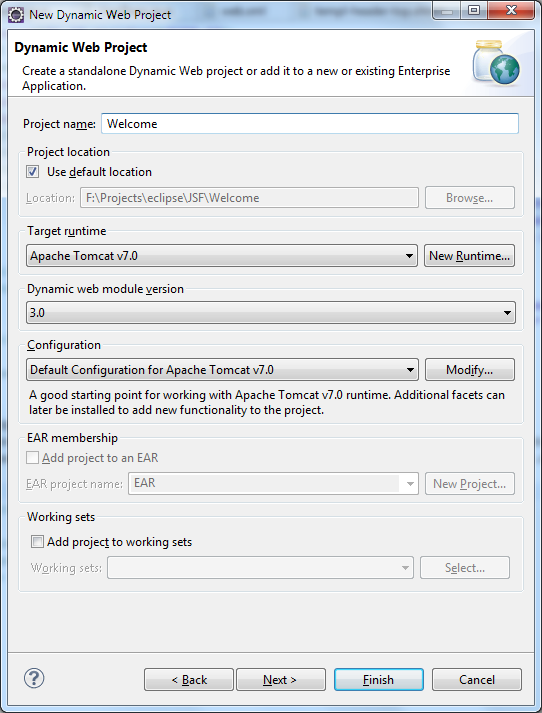
В jsf примере Welcome будет представлена авторизация. Структура jsf example будет включать 3 страницы и [JavaBean](http://java-online.ru/jsp-javabean.xhtml) компонент. На первой странице **main.xhtml** будет выполняться авторизация пользователя. Проверка логина и пароля будет выполняться в компоненте **MainBean.java**. При удачной авторизации будет выполняться переход на страницу приветствия пользователя loginsuccess.xhtml. При неудачной - на страницу ошибки loginfailed.xhtml.

**1-ый шаг**

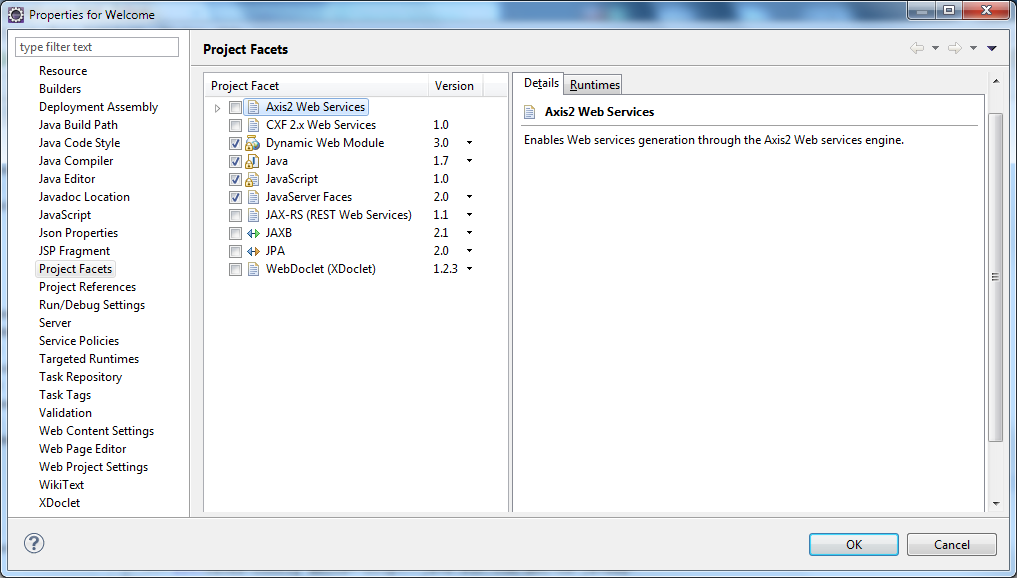
На первом шаге создаем проект **Dynamic Web Project**.



и определяем его наименование **Welcome**.



В настройках проекта подключаем фреймворк **JSF**.



**2-ой шаг**

Добавляем в каталог WEB-INF файлы **web.xml** и **faces-config.xml**. Можно было бы не добавлять файл конфигурации faces-config.xml. Ведь с выходом JSF 2.0 конфигурирование намного упростилось и параметры Managed Bean компонента (наименование и время жизни) можно определить в виде [аннотации](http://java-online.ru/java-annotation.xhtml) в коде.

В файле **web.xml** указываем управляющий сервлет и открываемую страницу по умолчанию в теге welcome-list для нашего jsf примера.

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

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemalocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" version="2.5">

<welcome-file-list>

<welcome-file>main.xhtml</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Faces Servlet</servlet-name>

<servlet-class>javax.faces.webapp.FacesServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>Faces Servlet</servlet-name>

<url-pattern>\*.xhtml</url-pattern>

</servlet-mapping>

</web-app>

При подключении фреймворка **JSF** в директории WEB-INF автоматически должен был создаться файл **faces-config.xml** следующего вида :

<!--?xml version='1.0' encoding='UTF-8'?-->

<faces-config xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemalocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-facesconfig\_2\_0.xsd" version="2.0">

</faces-config>

**3-ий шаг**

В директорию WEB-INF/lib копируем библиотеку javax.faces-2.ХХХ.jar. Данная библиотека необходима для создания Managed Bean компонентов в IDE.

**4-ый шаг**

Добавляем в проект компонент **MainBean.java**. Предварительно в проекте определяем package (jsf.common) для компонента.

package jsf.common;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.SessionScoped;

// Аннотации managed bean компонента

@ManagedBean (name="Main") // определение managed bean и его наименования

@SessionScoped // определение времени жизни - сессия

public class MainBean

{

private String login;

private String password;

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

/\* Метод простейшей авторизации.

\* Выполняется проверка имени и пароля пользователя.

\* Результат проверки - наименование страницы перехода

\*/

public String checkLogin(){

if (login.equalsIgnoreCase("alex") && password.equalsIgnoreCase("qwerty")){

return "loginsuccess?faces-redirect=true";

} else {

return "loginfailed?faces-redirect=true";

}

}

}

Метод **checkLogin** определяет навигацию по страницам. Возвращаемое значение - **jsf страница**, на которую должен быть выполнен переход. В предыдущей версии JSF правила навигации надо было описывать в **faces-config.xml**.

Для действительного перенаправления на страницу с изменением URL в коде к наименованию страницы добавляется **?faces-redirect=true**. О динамической навигации и перенаправлении подробно сказано на странице ["Навигация и перенаправление"](http://java-online.ru/jsf-navigation.xhtml).

**На 5-ом шаге создаем страницы JSF.**

JSF страница ошибки **loginfailed.xhtml**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Страница ошибки</title>

</h:head>

<h:body>

<h:outputText value="Ошибка авторизации. Проверьте логин и пароль" />

</h:body>

</html>

Компонент outputText выводит указанный текст.

Для подключения к странице JSF соответствующих тегов необходимо указать следующие строки :

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets"

JSF страница успешной авторизации **loginsuccess.xhtml**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Успешная авторизация</title>

</h:head>

<h:body>

<h:outputText value="Добро пожаловать: #{Main.login}!" />

</h:body>

</html>

main jsf страница авторизации **main.xhtml**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<title>Страница авторизации</title>

</h:head>

<h:body>

<h:form>

<h:panelGrid columns="2" border="0">

<h:outputText value="Логин"/>

<h:inputText value="#{Main.login}" />

<h:outputText value="Пароль"/>

<h:inputSecret value="#{Main.password}" />

</h:panelGrid>

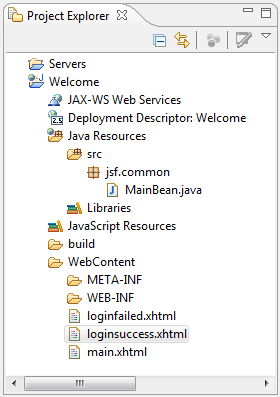
<h:commandButton value="Вход" action="#{Main.checkLogin}" />

</h:form>

</h:body>

</html>

Структура проекта jsf примера Welcome в среде Eclipse представлена на следующем скриншоте.



**Старт приложения**

Для запуска приложения из среды Eclipse необходимо выбрать пункт меню "Run As/Run on Server". Eclipse автоматически стартует Tomcat и откроет в браузере по умолчанию страницу "http://localhost:8080/Welcome".

Можно Tomcat в среде Eclipse стартовать отдельно и открыть страницу авторизации (main.xhtml) в любом браузере, набирая ту же строку URL.

Исходный код JSF примера Welcome, рассмотренного в тексте страницы, можно [скачать здесь](http://java-online.ru/sources/jsf-welcome.zip).