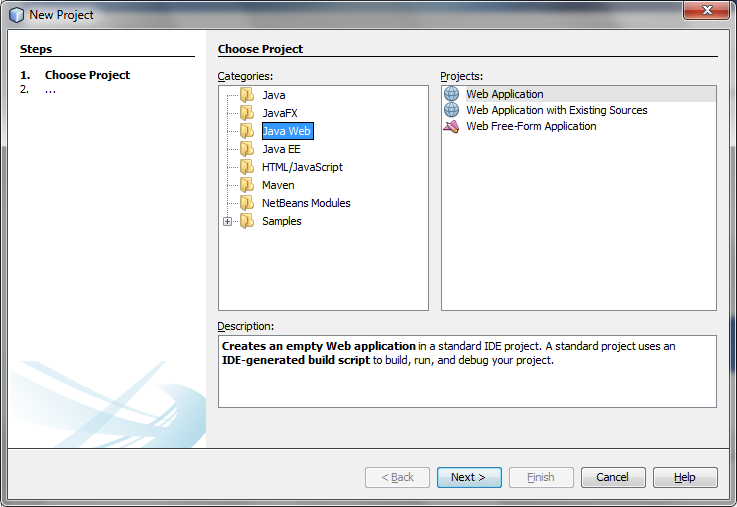
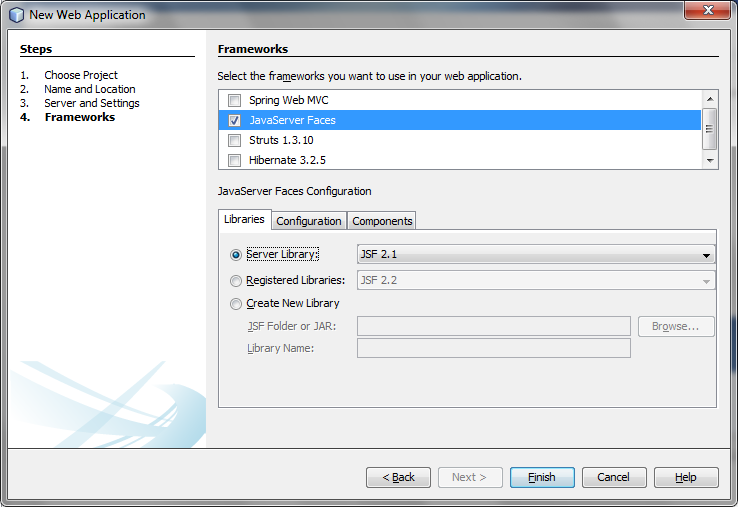
Первый проект с JSF

1. **Создаём проект:**

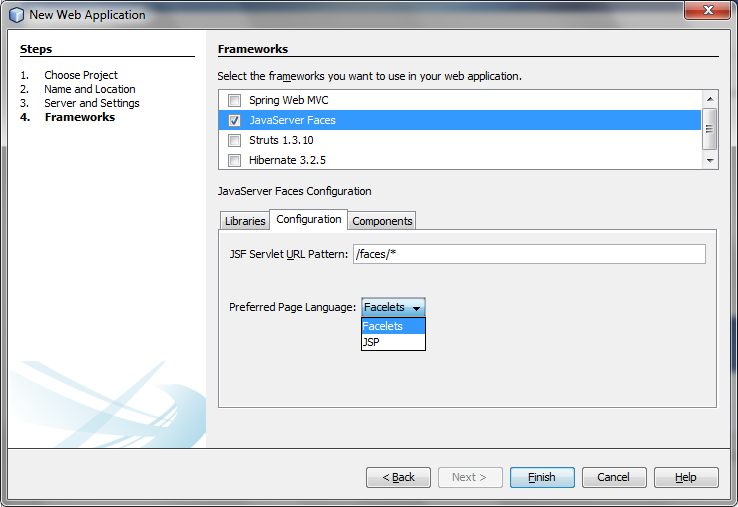


1. **Назовём проект TestJSF, выберем сервер GlassFish 3.1. Подключаем библиотеку JSF**



Выберем одну из доступных библиотек, важно чтобы библиотека была не ниже 2 версии

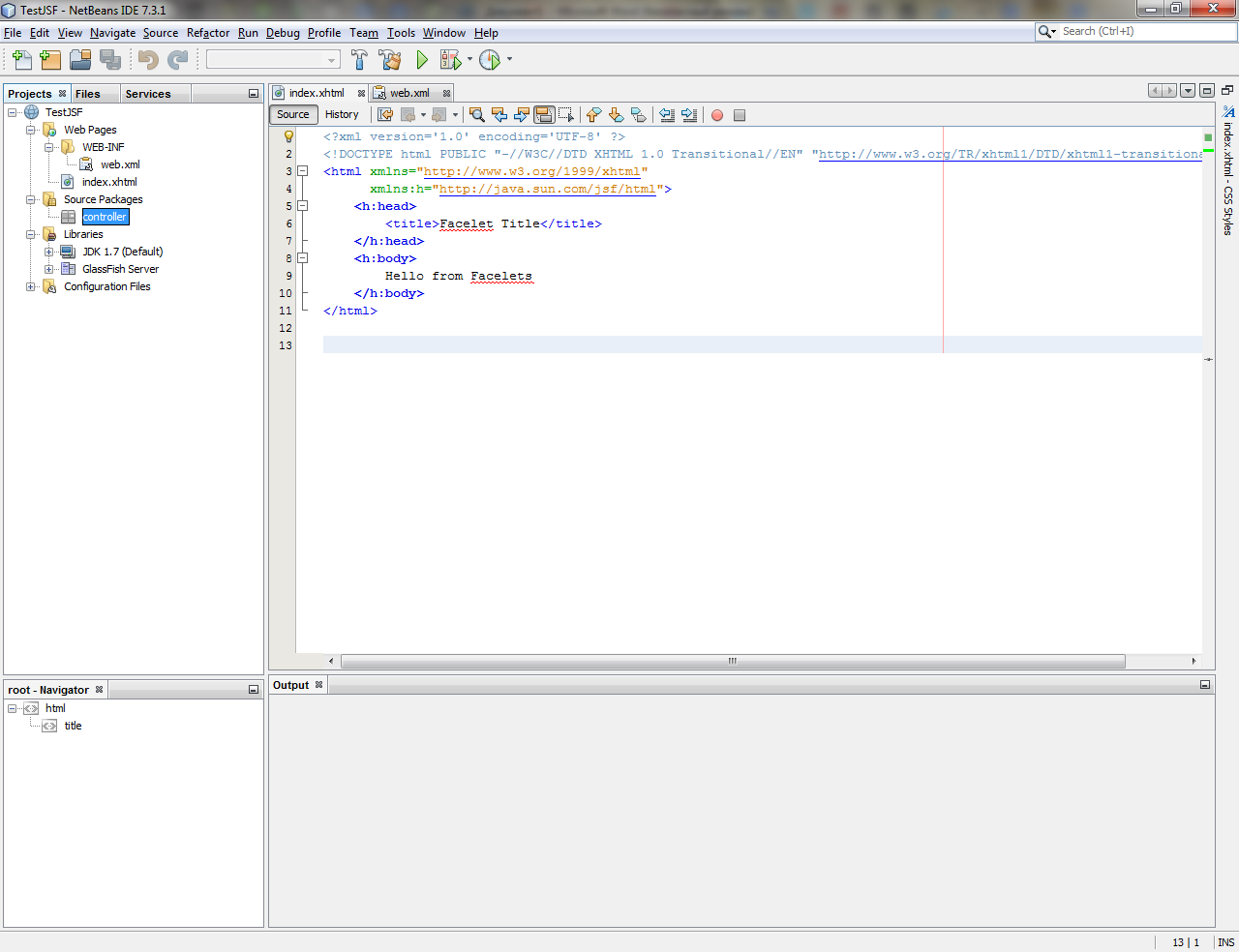
Перейдем на вкладку Configuration



Оставляем Facelets

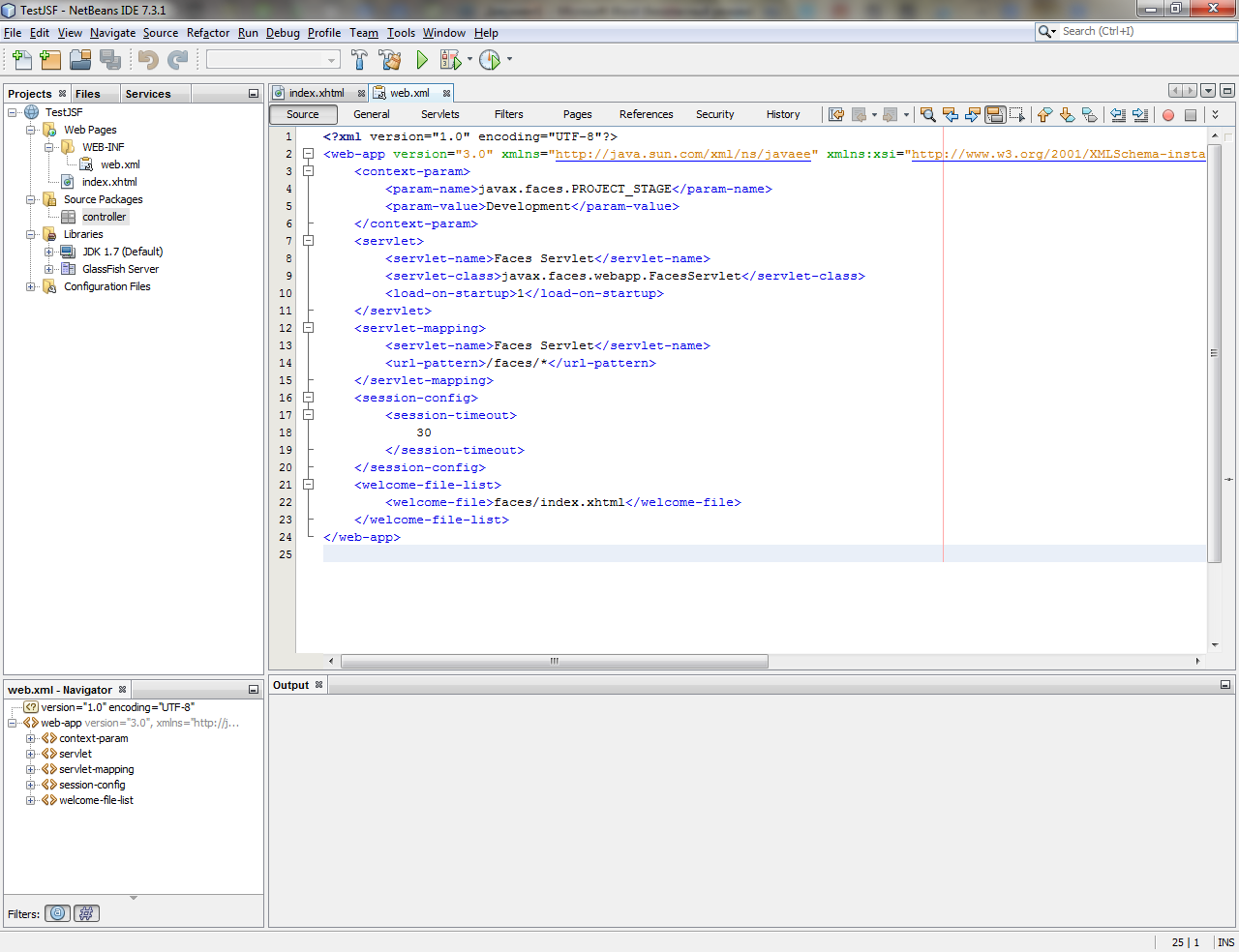
Для подключения доп. библиотек, поставляемых с JSF переходим на вкладку Components(но в данной лаб.р это не обязательно)

На следующем рисунке показано дерево проекта, заметьте, что доп.библиотек не видно, т.к они идут в поставке с GlassFish.

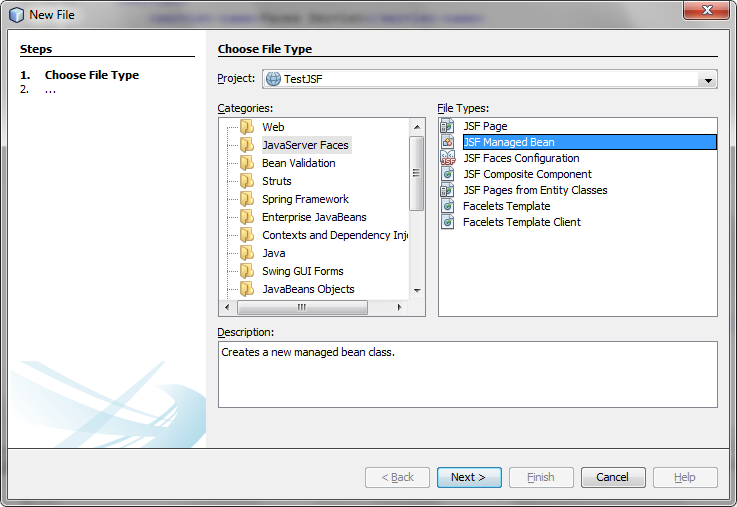


1. **Создадим новый пакет controller, в к-ом будут храниться управляющие компоненты, наши managed beans.**

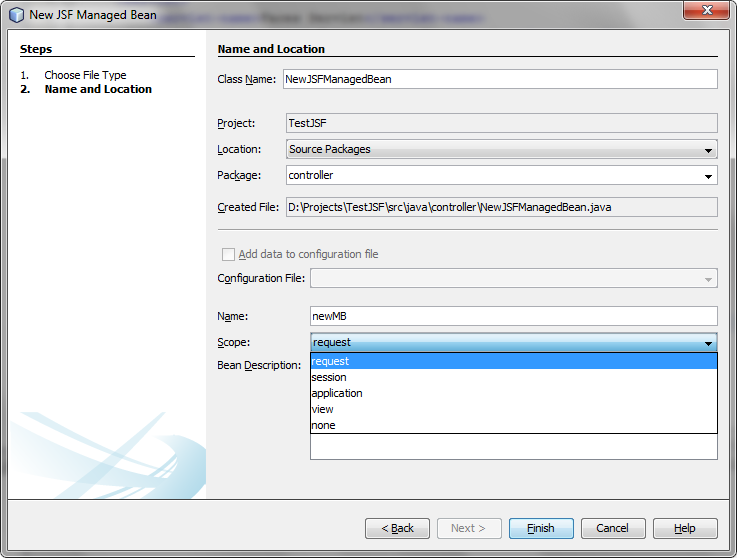
Так же обратите внимание на web.xml. Какой сервлет будет обрабатывать наши запросы и какой путь у начальной (приветствующей нас) страницы.



1. **Создадим управляющий бин**



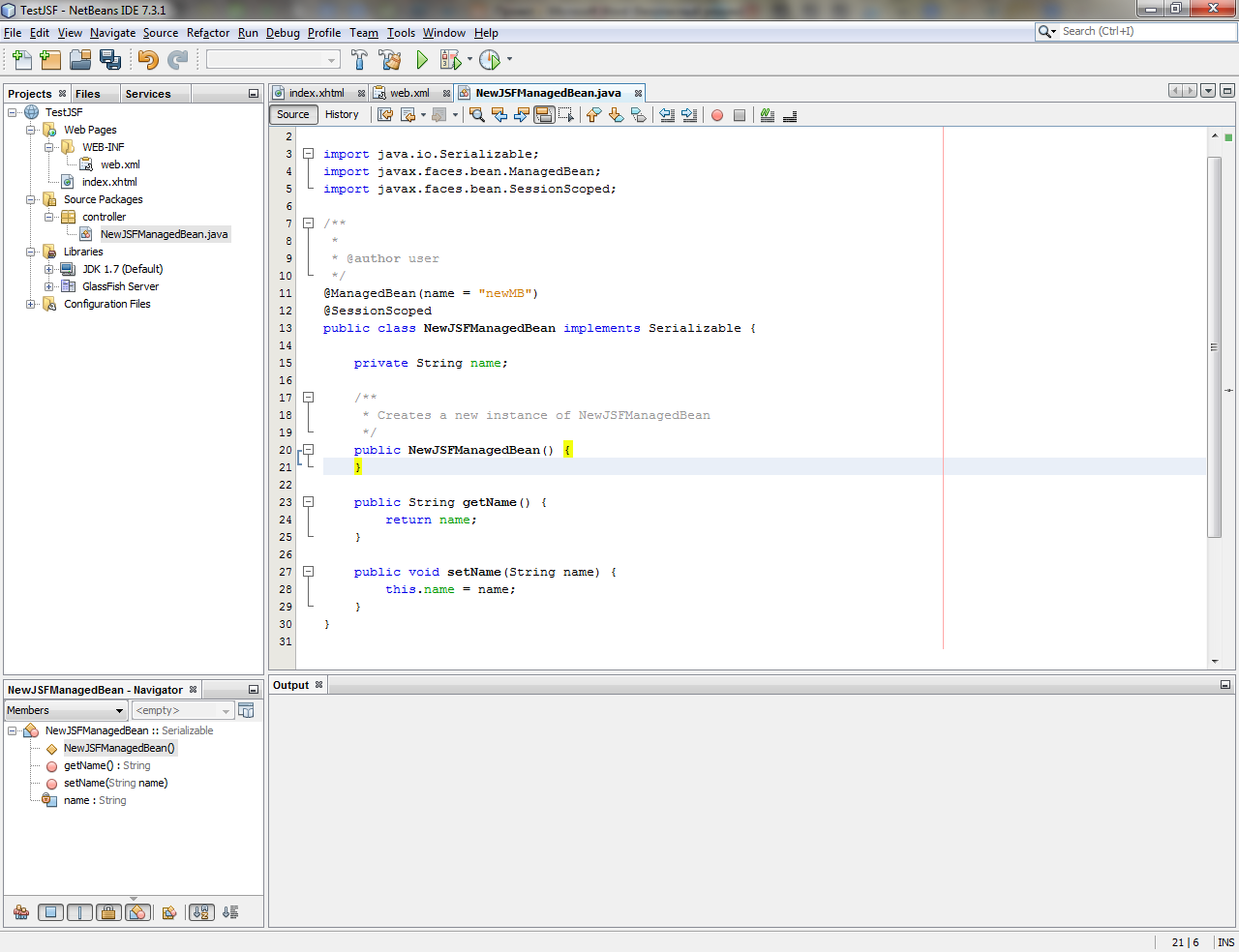
Заметьте что имя у него переименовано, именно так мы будем обращаться к нему с наших страниц. Scope (область видимости) выбираем session.



Небольшие изменения для получившегося бина:

добавляем Serializable т.к. scope выбран session;

описываем поле name + get ()set(), оно должно быть private. Как-то так:

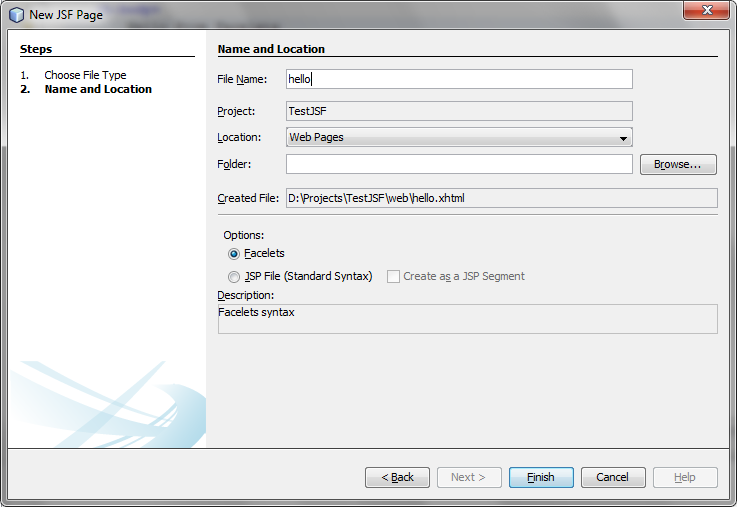


Заметьте, что В JSF 1.x, нам необходимо было объявить бины в faces-config.xml, но это больше не требуется в JSF 2.0., тк мы используем анотации.

1. **Создадим страницы, управляемые этим бином.**

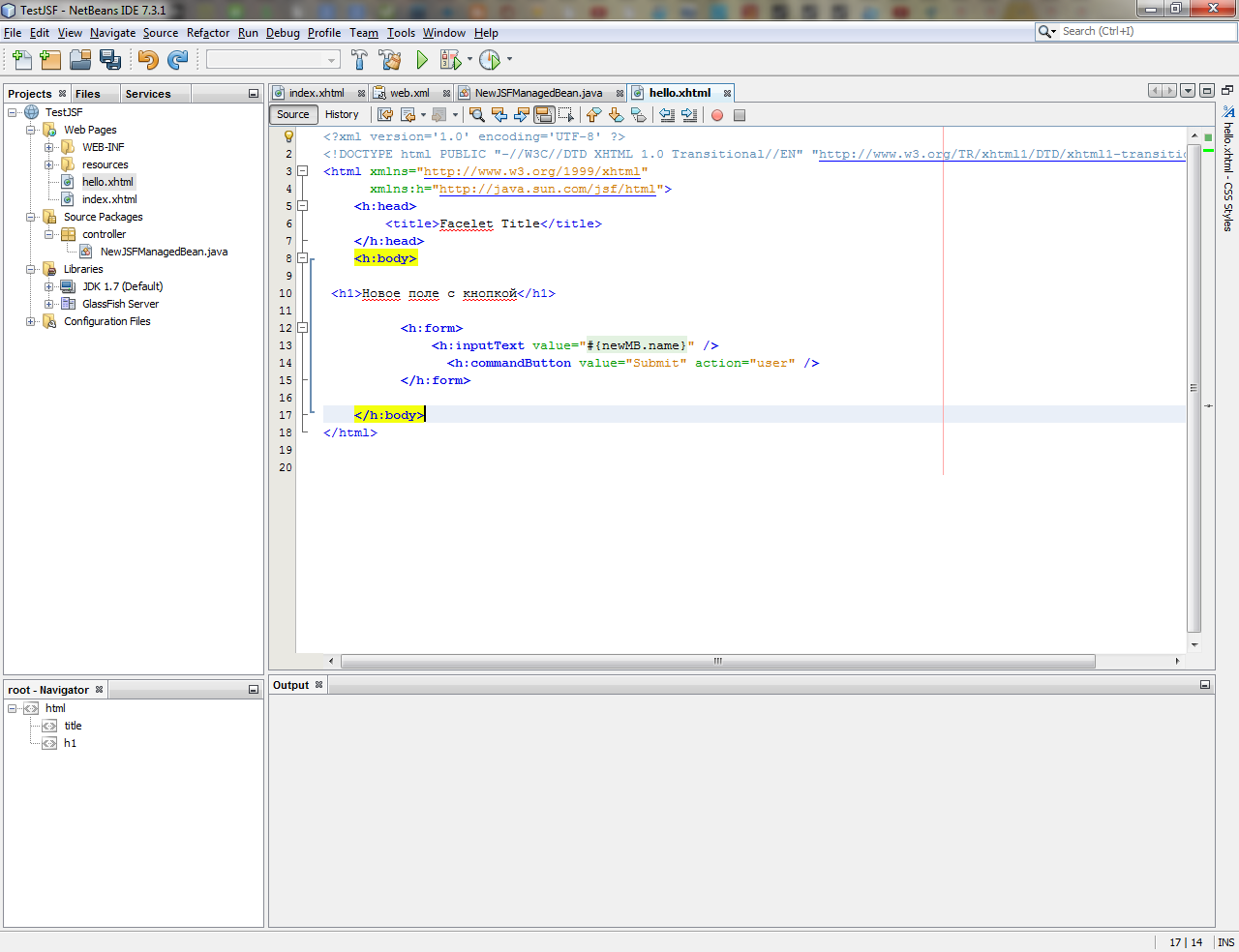
В JSF 2.0, рекомендуется создавать страницу JSF в XHTML формат файла, файл с расширением XHTML..

Кликаем правой кнопкой по папке Web pages(дерево проектов)->переходим в папку JSF->Создаём JSF page

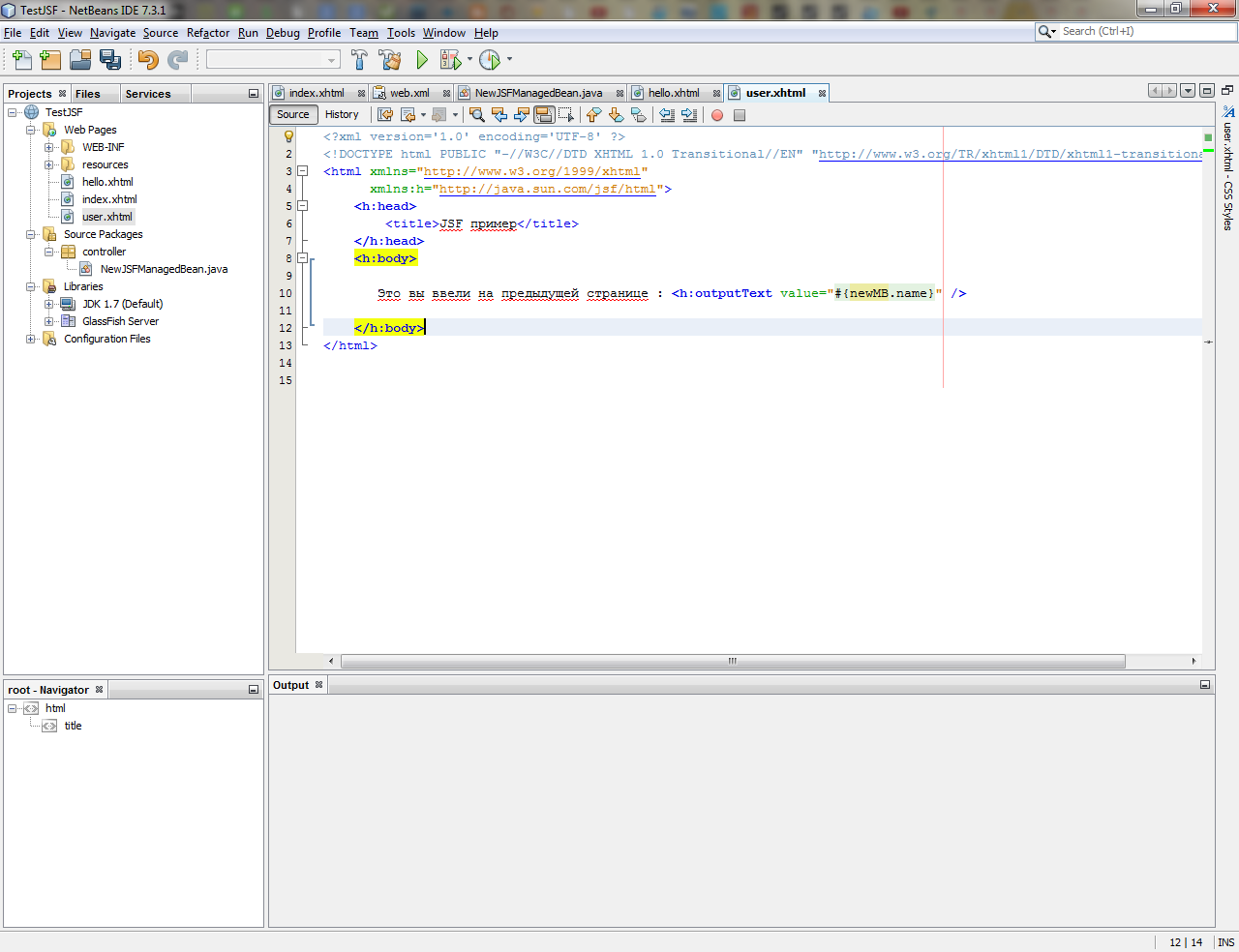


После, вставляем код, к-ый говорит нам, что мы хотим использовать форму, в к-ой есть поле для ввода значения (его значение будет обрабатываться бином с именем newMB и храниться в поле name.)

Так же есть кнопка Submit, результат к-ой будет виден на user + .xhtml, следовательно создайте по примеру ещё одну страничку user.xhtml



1. **На странице user.xhtml нам надо принять значение**, для этого используем тэг <h:outputText> и говорим, что в поле name у бина newMB уже есть(set) значение, к-ое бы хотим получить.



1. Вернёмся к странице по умолчанию index.xhtml**. Cделаем навигацию по нашему проекту**.

(ниже приведен один из возможных способов управлять страничками, то есть переходами.)

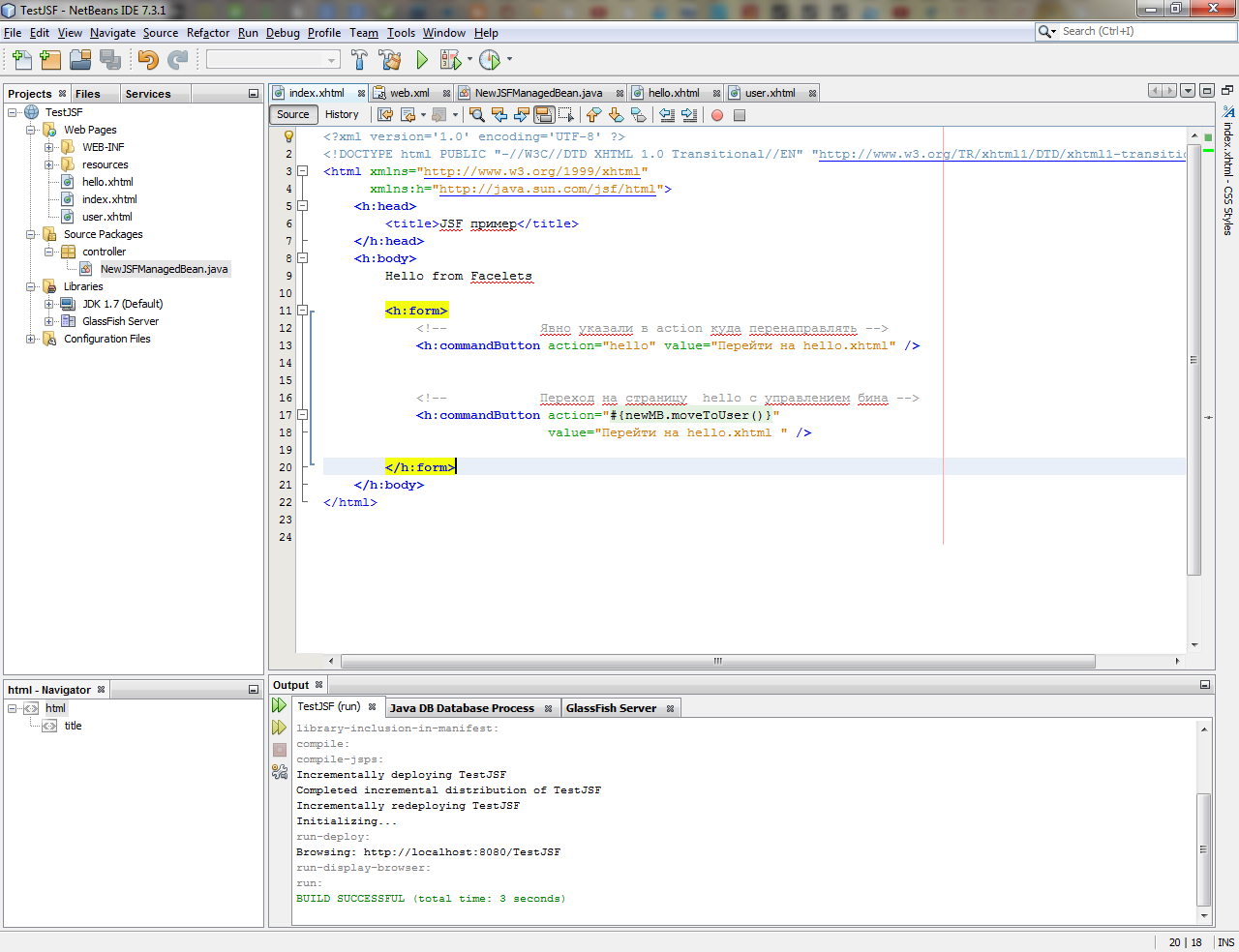
Добавим метод в наш Managed Bean

public String moveToUser(){

return "hello"; //outcome

}

А в index.xhtml

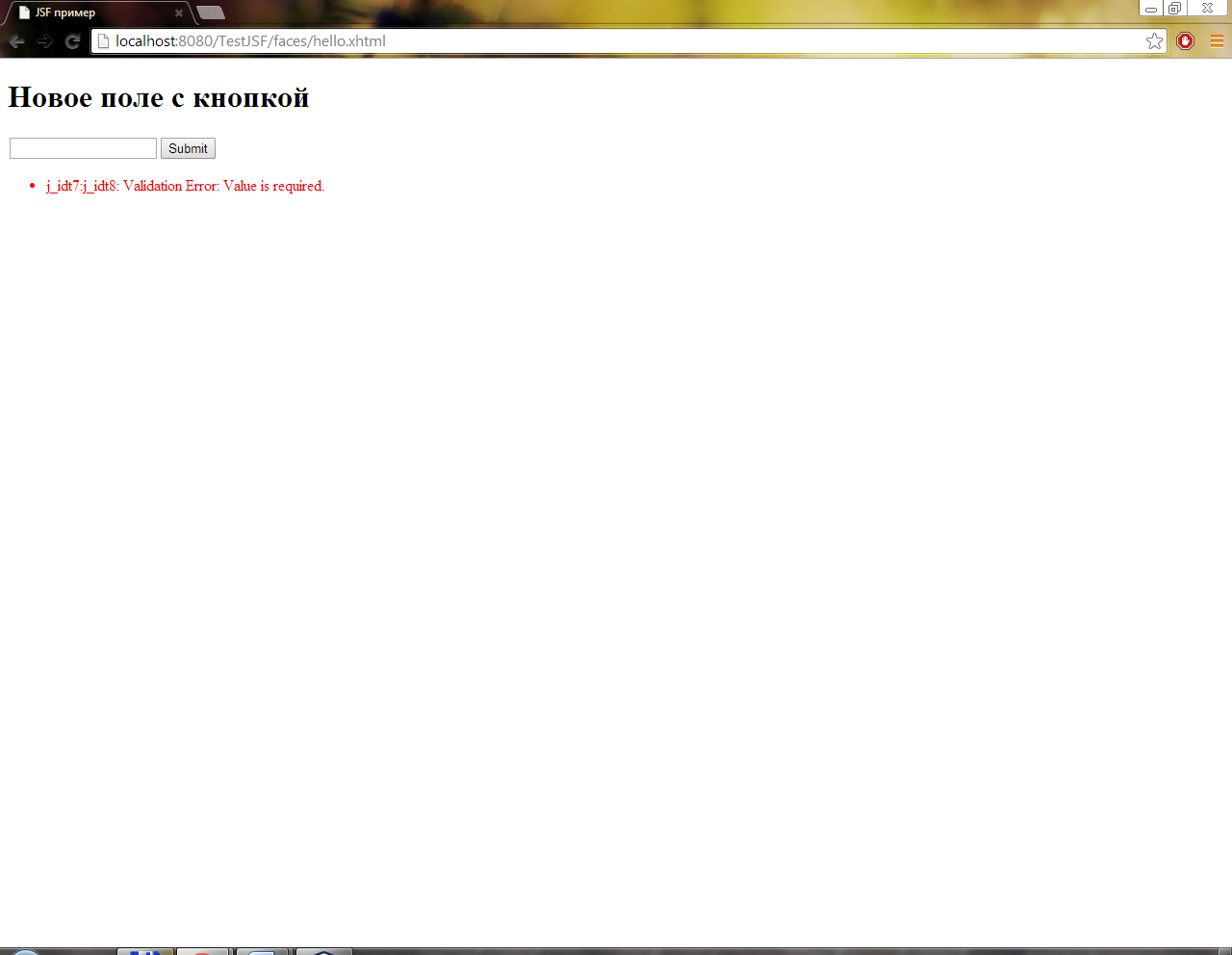


**Запустим приложение**

Если вы ничего не введёте на странице со строкой ввода, то ошибки не будет и в <h:outputText> придёт “ничего”. Чтобы этого избежать, давайте у пользователя запросим ввести данные. Изменим код, добавив опцию в <h:inputText> **required="true".**

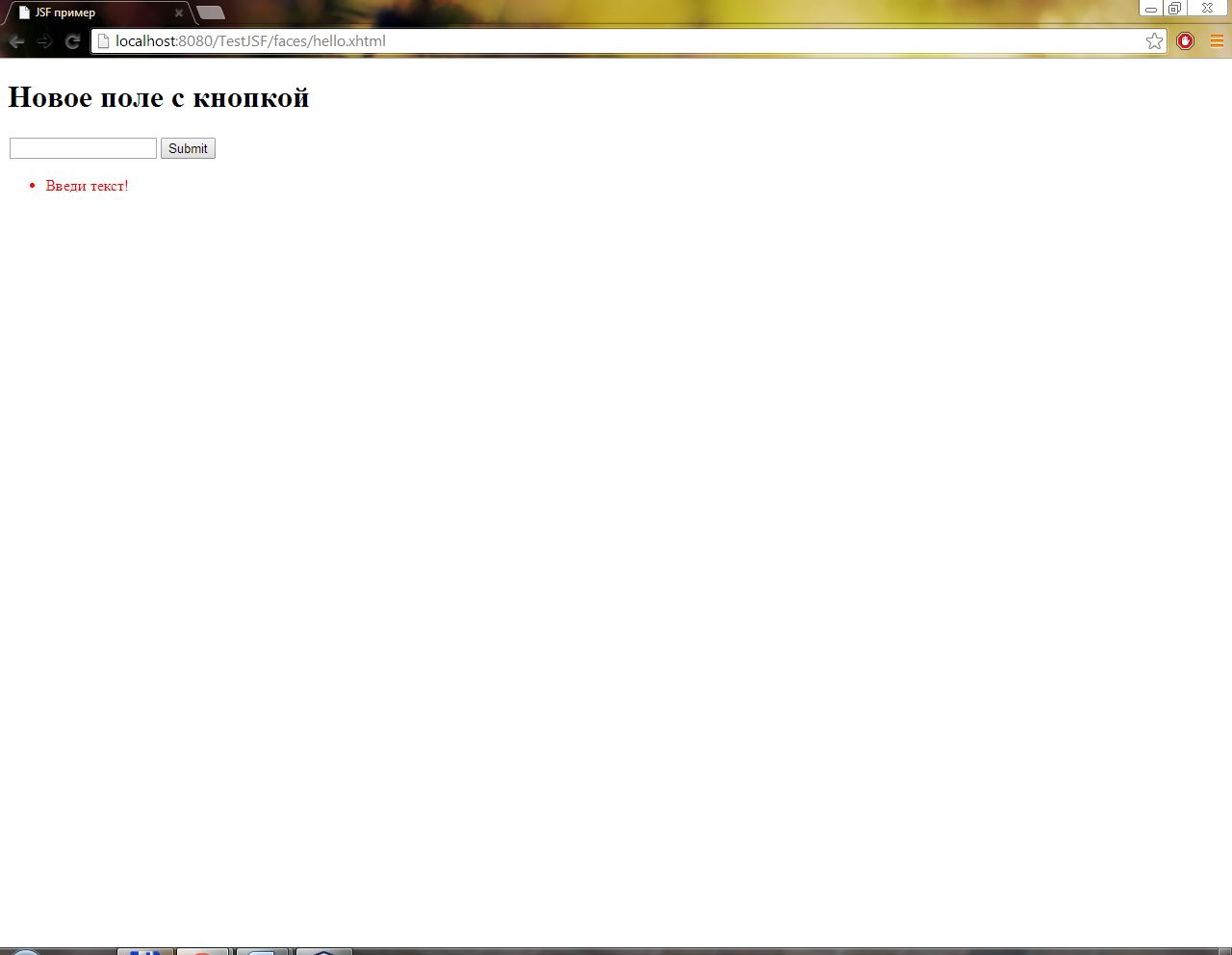
Запустим приложение

Сейчас у нас всплывёт стандартная ошибка, если мы нажали на кнопку и ничего не ввели.



Сделаем ошибку понятную, пользователю. Опять изменяем <h:inputText> добавляем requiredMessage="Введи текст!"

**Запустим приложение**



Итог:

