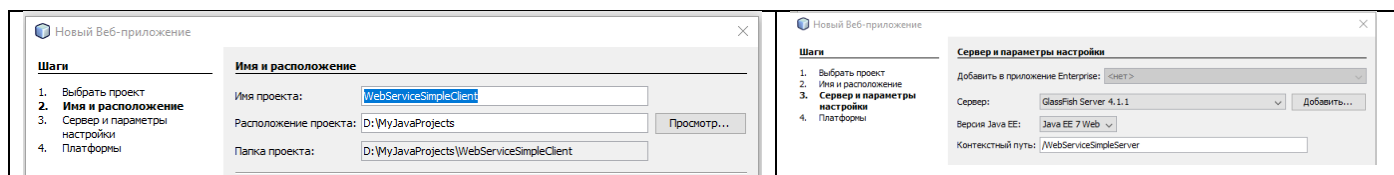
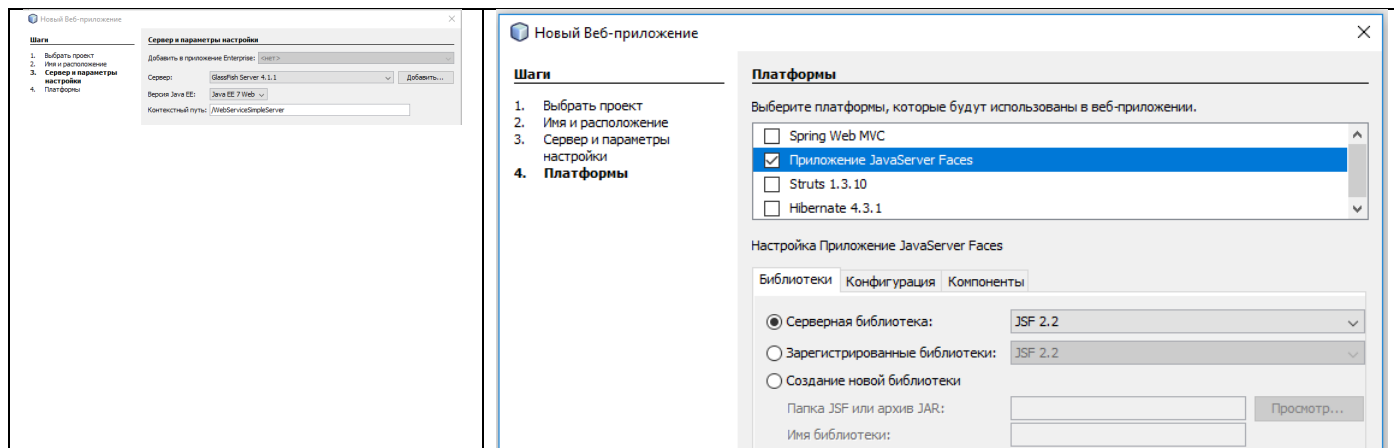


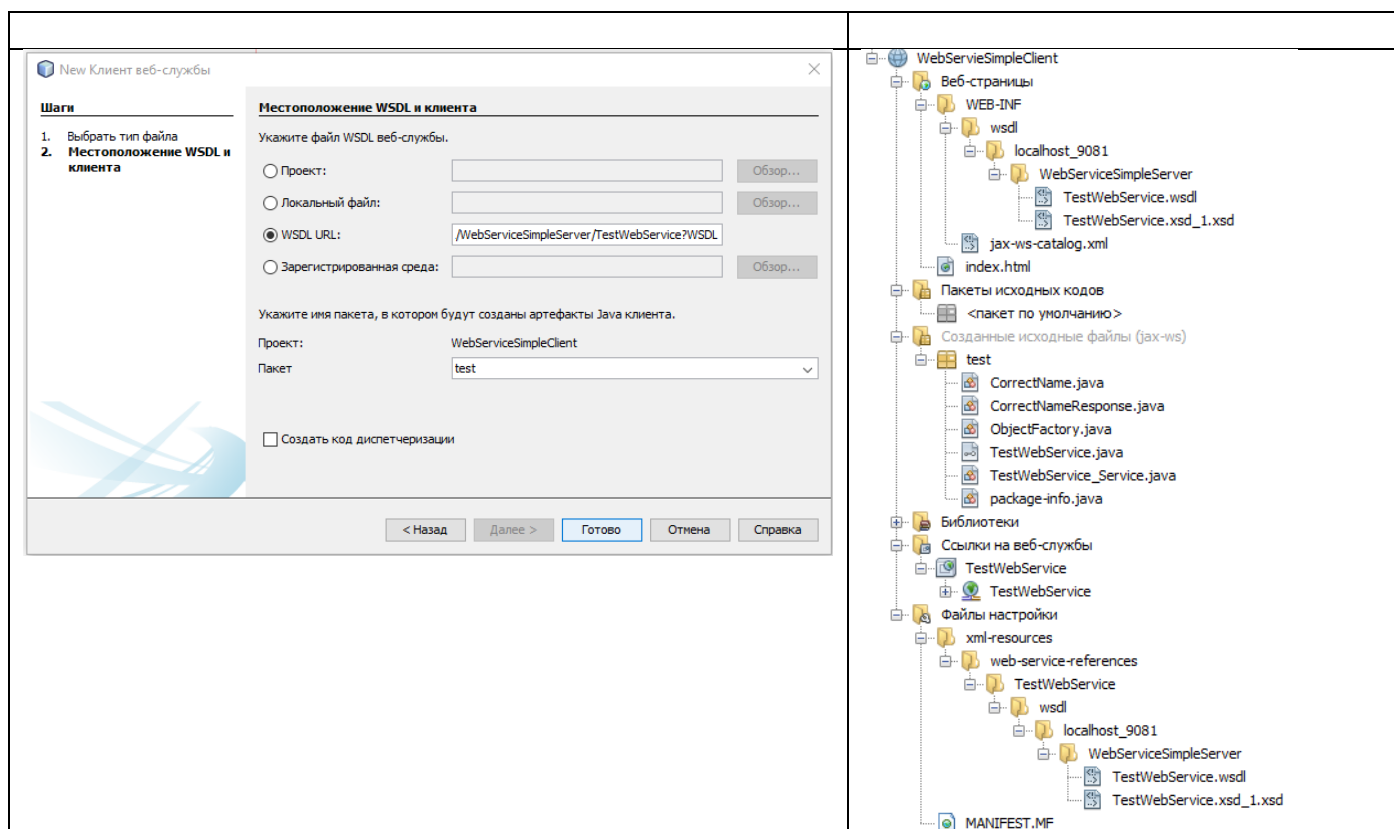
Пример 2 Создание клиента при помощи JFS форм (см. проект WebServiceSimpleClient). Создаем обычное Java Web приложение типа Java EE 7.



Указываем, что мы используем интерфейс JSF



После создаем клиента web-службы из WSDL файла (перед этим необходимо развернуть сервер службы и запустить его тест):



Далее необходимо добавить класс CallWS , который будет содержать команды обращения из клиента к серверу.

Итоговая структура проекта	Код файла CallWS.java
	<pre> 1 import javax.faces.bean.ManagedBean; 2 import javax.faces.bean.RequestScoped; 3 import test.TestWebService_Service; 4 5 6 @ManagedBean 7 @RequestScoped 8 public class CallWS { 9 10 private String name; 11 private String correctedName; 12 13 public String getName() { 14 return name; 15 } 16 17 public void setName(String name) { 18 this.name = name; 19 } 20 21 public String getCorrectedName() { 22 return correctedName; 23 } 24 25 public void setCorrectedName(String correctedName) { 26 this.correctedName = correctedName; 27 } 28 29 public void correctName() { 30 TestWebService_Service service = new TestWebService_Service(); 31 test.TestWebService port = service.getTestWebServicePort(); 32 correctedName = port.correctName(name); 33 } 34 35 } </pre>

Вносим изменения в файл разметки index.htm, в итоге он должен выглядеть так:

```

1  <?xml version='1.0' encoding='UTF-8' ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3  <html xmlns="http://www.w3.org/1999/xhtml"
4        xmlns:h="http://xmlns.jcp.org/jsf/html">
5
6      <h:head>
7          <title>Facelet Title</title>
8      </h:head>
9      <h:body>
10         Hello from Facelets
11         <h:form>
12             <h:inputText value="#{callWS.name}"/><br/>
13             <h:outputText value="#{callWS.correctedName}"/> <br/>
14             <h:commandButton action="#{callWS.correctName}" value="correct!"/>
15         </h:form>
16     </h:body>
17 </html>

```