# Création d'une Application Web / JSP

**Rédacteurs** : Alexandre Baillif, Philippe Lacomme et Raksmey Phan **Date** : juillet 2010 **Avertissement :** ce document est une reprise d'une partie d'un document écrit par Serge Tahé. En particulier la base de données utilisée.

Nous proposons deux solutions utilisant l'EJBModule1 et l'EJBModule2 et une solution standalone.

*NB* : Il faut savoir qu'il existe de nombreuses variantes et donc de nombreuses architectures possibles, mais celles présentées ici conviennent dans la grande majorité des cas.

Solution 1.



Cette solution permet de créer une application web à partir d'un EJB existant. Cette solution permet de réutiliser un EJB présent sur un serveur à travers un web service. Ici le web service et le client sont dans la même application, mais on peut très bien imaginer un client dans une application différente.

Solution 2.



Cette solution permet de créer une application web à partir d'un web service existant. Cette solution est la plus portable car elle permet de faire tourner une application web de manière distante et indépendante de l'application métier (EJB). Cette solution est à préférer dans le cas de grosses applications (une variante sans le concept d'EJB peut très bien exister).

Solution 3.



Cette solution permet de créer une application web sans avoir à utiliser d'EJB ou de web service. Elle possède une architecture plus simple (une seule application) et convient pour de petits projets qui tournent sur un seul serveur.

## Solution 1.



## 1) Conception de l'application client

New Project		×
Steps	Choose Project	
1. Choose Project 2	Categories: Android Java JavaFX Java Web Java Web Java KE Java Card Java ME Maven HP Nuby C/C++	Projects: Web Application Web Application with Existing Sources Web Free-Form Application
	Creates an empty Web application in a IDE-generated build script to build, run	a standard IDE project. A standard project uses an , and debug your project. Next > Einish Cancel Help

New Web Application	1.000	<b>X</b>
Steps	Name and Location	
1. Choose Project 2. Name and Location	Project Name: WebApplicationJSP	
<ol> <li>Server and Settings</li> <li>Frameworks</li> </ol>	Project Location: C:\Users\acomme.T3500-PC\Desk	top\essai_gwt\JEE2\verification Browse
	Project <u>F</u> older: e.T3500-PC\Desktop\essai_gwt\JE	E2\verification\WebApplicationJSP
	Use Dedicated Folder for Storing Libraries	
	Libraries Folder:	Browse
	Different users and projects can sha (see Help for details).	are the same compilation libraries
	👿 Set as Main Project	
X		
	< <u>B</u> ack Next	t > Einish Cancel Help

New Web Application	Cannot cannot its interest.	×
Steps	Server and Settings	
<ol> <li>Choose Project</li> <li>Name and Location</li> </ol>	Add to Enterprise Application: <pre> </pre>	~
<ol> <li>Server and Settings</li> <li>Frameworks</li> </ol>	Server: Personal GlassFish v3 Domain	A <u>d</u> d
	Use dedicated library folder for server JAR files	
	Java EE Version: Java EE 6 Web ↓	
	Context Path: //WebApplicationJSP	
	< Back Next > Einish Cancel	Help

On peut choisir comme framework Java Server Faces par exemple. Pour le serveur Glassfish, utiliser celui dont le port est disponible.

New Web Application	X
Steps	Frameworks
1. Choose Project	Select the frameworks you want to use in your web application.
3. Server and Settings	Google Web Toolkit
4. Frameworks	Spring Web MVC
	JavaServer Faces
	Struts 1.3.8
	JavaServer Faces Configuration
	Libraries Configuration
	• Use default library which comes with Server (JSF 1.2).
	© Create New Library
	JSF Folder: Browse
	Library Name:
/ VARA	
	< <u>Back</u> Next > <u>Finish</u> Cancel <u>H</u> elp

Faire un clic droit sur la partie « librairies » de l'application Web et choisir « Add Jar » Choisir ensuite EJBModule1.jar. La partie Libraries de WebApplicationJSP devrait se présenter alors comme suit :

🖕 🕞	Libraries
<b>.</b>	EJBModule 1. jar
÷[	JDK 1.6 (Default)
÷	Personal GlassFish v3 Domain

## 2) Défintion d'un client web

2.1. Vérifier dans la partie Services que l'EJBModule1 a bien été déployé. Vous devez obtenir un écran comme celui-ci :



2.2. Faire un clic droit et ajouter un web service client.



Aller ajouter une session Bean en cliquant sur Browser.

Choisir comme nom de package rendezvous par exemple et nommer le WSDaoJpa.

New Web Service		
Steps	Name and Location	
1. Choose File Type 2. Name and Location	Web Service Name: WSDaoJpa	
	Project: WebApplicationJSP	
	Location: Source Packages	
	Package: rendezvous	
	© Create Web Service from Scratch	
	Create Web Service from Existing Session Bean	
	Enterprise Bean: Browse	
	Implement Web Service as Stateless Session Bean	
	<ol> <li>Select an Enterprise Bean.</li> </ol>	
	< <u>Back</u> Next > Einish Cancel Help	

Browse Enterprise Bean	X
Enterpri <u>s</u> e Beans:	
Solution     Solution	
	OK Cancel

New Web Service	×
Steps	Name and Location
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Web Service Name: WSDaoJpa
	Project: WebApplicationJSP
	Location: Source Packages
	Package: rendezvous
	Create Web Service from Scratch
	Oreate Web Service from Existing Session Bean
	Enterprise Bean: Enterprise Beans#DaoJpa Browse
	Implement Web Service as Stateless Session Bear
	< <u>Back</u> Next > Finish Cancel Help

Ouvrir le fichier WSDaoJpa.java et créer un web service comme suit :

```
package rendezvous;
import javax.ejb.EJB;
import javax.jws.*;
import jpa.*;
import dao.*;
import dao.IdaoLocal;
import java.util.*;
@WebService()
public class WSDaoJpa implements Idao {
   0EJB
   private IdaoLocal dao;
   // web service numero 1
  // liste des clients
  @WebMethod
 public List<Clients> getAllClients() {
    return dao.getAllClients();
  }
 // web service numero 2
  // liste des clients
  @WebMethod
 public List<Medecins> getAllMedecins() {
   return dao.getAllMedecins();
 }
 // liste des créneaux horaires d'un médecin donné
 // medecin : le médecin
 @WebMethod
public List<Creneaux> getAllCreneaux(Medecins medecin) {
  return dao.getAllCreneaux(medecin);
 }
 // liste des Rv d'un médecin donné, un jour donné
 // medecin : le médecin, jour : le jour
 @WebMethod
 public List<Rv> getRvMedecinJour(Medecins medecin, String jour) {
   return dao.getRvMedecinJour(medecin, jour);
```

```
// ajout d'un Rv, jour : jour du Rv
 // creneau : créneau horaire du Rv, client : client pour lequel est pris le Rv
 @WebMethod
public Rv ajouterRv(String jour, Creneaux creneau, Clients client) {
   return dao.ajouterRv(jour, creneau, client);
}
// suppression d'un Rv, rv : le Rv supprimé
@WebMethod
public void supprimerRv(Rv rv) {
dao.supprimerRv(rv);
}
// récupérer un client donné
@WebMethod
public Clients getClientById(Long id) {
return dao.getClientById(id);
}
// récupérer un médecin donné
@WebMethod
public Medecins getMedecinById(Long id) {
   return dao.getMedecinById(id);
}
// récupérer un Rv donné
@WebMethod
public Rv getRvById(Long id) {
return dao.getRvById(id);
}
// récupérer un créneau donné
@WebMethod
public Creneaux getCreneauById(Long id) {
return dao.getCreneauById(id);
```

## 2.3. Compiler et déployer

Aller dans l'onglet Services et par un clic droit sur EJBModule1 faire Undeploy.

Faire un clic droit sur WebApplicationJSP puis « Clean and Build » et « Deploy ».



Il est alors possible de tester les services web en faisant un clic droit sur Web Service :



Le résultat devrait être :

🕲 WSDaoJpaService Web Service Tester - Mozilla Firefox
Eichier Édition Affichage Historique Marque-pages Qutils ?
C X 🟠 🗋 http://localhost8080/WebApplicationPh/WSDaoJpaService?Tester 🖄 🖷 🔞 - Search the web (Babylon) 🔎
🗷 Les plus visités 🏶 Débuter avec Firefox 🔜 À la une 📋 essai.html 📄 essai.html
WSDaoJpaService Web Service Tester +
WSDaoJpaService Web Service Tester
This form will allow you to test your web service implementation (WSDL File)
To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.
Methods :
public abstract java.util.List rendezvous.WSDaoJpa.getAllClients() getAllClients ()
public abstract java.util.List rendezvous.WSDaoJpa.getAllMedecins() getAllMedecins ()
public abstract java.util.List rendezvous.WSDaoJpa.getAllCreneaux(rendezvous.Medecins)          getAllCreneaux

2.4. Générer les services SOAP

Aller sur Web Services et faire un clic droit sur WSDaoJpa.

🖻 膭 Web Services	
⊕	Open
🕀 🇓 RESTful We	Refresh
🞰 👧 Configurati lavigator 👼 Tasks	Add Operation
	Test Web Service
Create	Edit Web Service Attributes
	Configure Handlers
	Generate and Copy WSDL
	Generate SOAP-over-HTTP Wrapper
· ≡ ☴ ☵ 💷 ք	Delete
ançais (Fr 💷	Properties
NotRease IDE 6.0	Vi Nans titre - Maint 🛛 🖇 (

Et faire Generate SOAP.

REST Resources Configuration		X
Specify the way REST resources will b	e registered in the application:	
Netbeans will generate a subcla all REST resources will be registed.	ss of javax.ws.rs.core.Application, ered by this class automatically(JavaEE 6).	
User is responsible for REST res e.g. by implementing a specific or by registering a specific service	ources registration, subclass of javax.ws.rs.core.Application, et adaptor in web.xml.	
Create default REST servlet ada	aptor in <u>w</u> eb.xml.	
<u>R</u> EST Resources Path:	/resources	
		OK Cancel

Finalement le projet ressemble à ce qui suit :



## Remarquons la partie Web Service References :

🖶 💫 Enterprise Beans
🖕 🕞 Web Services
🖕 ዀ 🎦 Web Service References
wSDaoJpaService
🗄 👷 WSDaoJpaService
🖶 🖟 RESTful Web Services
🗄 🗐 🚱 WSDaoJpaPort [wsdaojpaport]
🗄 🖟 💦 Configuration Files

## 2.5. Tester le Web Service

Il faut déployer l'application.

Faire un clic droit sur WSDaoJpa et choisir Test Web Service

🖨 🔂 Web Servic	ces	
	Open	
MANIF	Refresh	
	Add Operation	
	Test Web Service	
; WSDaoJpa - Naviga	Edit Web Service Attributes	₩ ₩
	Configure Handlers	
	Generate and Copy WSDL	
	Generate SOAP-over-HTTP Wrapper	
	Delete	
	Properties	

Vous devriez obtenir ceci :

🕲 WSDaoJpaService Web Service Tester - Mozilla Firefox		×
Eichier Édition Affichage Historique Marque-pages Qutils ?		
C X 🟠 📽 🗋 http://localhost:10397/WebApplicationJSP/WSDaoJpaService?Tester	☆ 🝷 🚼 - Google	٩
🖉 Les plus visités 🗋 Débuter avec Firefox 🔜 À la une		
WSDaoJpaService Web Service Tester		-
WSDaoJpaService Web Service Tester		
This form will allow you to test your web service implementation (WSDL File)		
To invoke an operation, fill the method parameter(s) input boxes and click on the b method name.	utton labeled with the	E
Methods :		
public abstract java.util.List rendezvous.WSDaoJpa.getAllClients() getAllClients ()		_
public abstract java.util.List rendezvous.WSDaoJpa.getAllMedecins() getAllMedecins ()		
public abstract java.util.List rendezvous.WSDaoJpa.getAllCreneaux(rendezvous.M getAllCreneaux ()	ledecins)	
public abstract java.util.List rendezvous.WSDaoJpa.getRvMedecinJour(rendezvous.Medecins,java.lang.String) getRvMedecinJour (		_
Terminé	4	-

Et un clic sur **getAllClients** doit donner :

Method invocation trace - Mozilla Firefox	
ichier Éditio <u>n A</u> ffichage <u>H</u> istorique <u>M</u> arque-pages <u>O</u> utils <u>?</u>	
🚱 🕞 C X 🏠 😵 📋 http://localhost:10397/WebApplicationJSP/WSDaoJpaService?Tester	्री 👻 Google
Les plus visités 📄 Débuter avec Firefox 🔜 À la une	
Method invocation trace	
getAllClients Method invocation	
Method parameter(s)	
Type Value	
Method returned	
Method returned java.util.List : "[rendezvous.Clients@5062d9, rendezvous.Clients@10=1d5	ients@189a628,
Method returned java.util.List : "[rendezvous.Clients@5062d9, rendezvous.Cl rendezvous.Clients@1fbb2d6, rendezvous.Clients@19a1d5	ients@189a628, 7]"
Method returned java.util.List : "[rendezvous.Clients@5062d9, rendezvous.Cl rendezvous.Clients@1fbb2d6, rendezvous.Clients@19a1d5 SOAP Request	ients@189a628, 7]"
Method returned java.util.List : "[rendezvous.Clients@5062d9, rendezvous.Clients@1fbb2d6, rendezvous.Clients@19a1d5 SOAP Request xml version="1.0" encoding="UTF-8"? <s:envelope <="" td="" xmlns:s="http://schemas.xmlsoap.org,&lt;br&gt;&lt;S:Header/&gt;&lt;/td&gt;&lt;td&gt;ients@189a628,&lt;br&gt;7]"></s:envelope>	
Method returned java.util.List : "[rendezvous.Clients@5062d9, rendezvous.Clients@1fbb2d6, rendezvous.Clients@19a1d5 SOAP Request xml version="1.0" encoding="UTF-8"? <s:envelope http:="" ren<br="" xmlns:s="http://schemas.xmlsoap.org,&lt;br&gt;&lt;S:Header/&gt;&lt;br&gt;&lt;S:Body&gt;&lt;br&gt;&lt;ns2:getAllClients xmlns:ns2="></s:envelope>	ients@189a628, 7]" /soap/envelope/"> ndezvous/"/>
Method returned java.util.List : "[rendezvous.Clients@5062d9, rendezvous.Clients@1fbb2d6, rendezvous.Clients@19a1d5 SOAP Request xml version="1.0" encoding="UTF-8"? <s:envelope http:="" ren<br="" xmlns:s="http://schemas.xmlsoap.org,&lt;br&gt;&lt;S:Header/&gt;&lt;br&gt;&lt;S:Body&gt;&lt;br&gt;&lt;ns2:getAllClients xmlns:ns2="></s:envelope>	ients@189a628, 7]" /soap/envelope/"> ndezvous/"/>

2.6. Vérifions que le fichier  $index.xhtml \ll fonctionne \gg$ 

Faire un clic droit sur index.xhtml et choisir view.

🖃 🌐 WebApplicationJS	🗄 🌐 WebApplicationJSP		
🖨 🔥 Web Pages			
🕀 🕠 WEB-INF			
index.,	Open		
e rendez	View		
Gan Carlos Carl	Run File Debug File	Maj+F6 Ctrl+Mai+F5	
EJBMo	Local History	•	
⊕	Cut Copy Paste	Ctrl+V	
ia∰ JUnit 4 iaia.ia. Person iaia.ia. Enterprise	Delete Rename		
🗇 🐻 Web Servic	Save As Template		
in we	Tools Properties	×	
i 🛐 web.xml			

Le résultat doit être le suivant :



## 3) Création d'un fichier JSP

## 3.1. Créer un fichier nommé index.jsp

Faire un click droit sur WebApplicationJSP et choisir JSP.

			_			In Import Javax.ejb.E0.
WebApplication	New	•	٩	Web Se	ervice	
🗄 🖟 🚺 WEB-INF	Build		8	Java Cl	ass	
index.xhtr	Clean and Ruild			Java Pa	ckag	e
🖨 🚡 Source Packag	Clean and build		Ð	JSP		
🖻 🔛 rendezvou	Clean		٢	Web Se	rvice	Client
WSDa	Generate Javadoc			JSF Pac	ie	
	Run		P	Empty	, File	<u>بر</u>
E Libraries	Deploy			Session	Real	as For Entity Classes
🗄 🔚 EJBModule	Debug			Cossion	Dear	as For Entity Classes
	Profile			Jession	Deal	is for Entry classes
🗄 📲 Personal G	Test DECTE JWeb Consistent			Java In	епас	
🖨 🍃 Test Libraries	Test RESTFULWeb Services			JPA Co	ntrol	ler Classes from Entity Classes
🕀 📄 JUnit 3.8.1	Test	Alt+F6		Entity (	Classe	es from Database
JUnit 4.5 -	Set as Main Project		8	Entity (	Class.	și
Enterprise Bea	Open Required Projects		Ð	Persiste	ence	Unit
Web Services	Close		8	Servlet.		
	ciose		<b>a</b> na			22
🖃 💫 Configuration	Rename	l		Other		-
MANIFEST	Move				: 0	tout
	Copy					Jaure DR Database Drawne & u. Pe
web.xml 📷	Delete					Java Do Database Process
						INFO: Instantiated an inst
WebApplicationJSP [M	Find			₿ %		INFO: ECHIPSELINK, Version INFO: file:/C:/Users/lacon
	Share on Kenai				×	INFO: parsing WSDL
	Versioning	+			3	
	Local History	+				INFO: generating code
	Properties					INFO: compiling code
	riopenies					

Dans les options, choisir JSP File (Standard Syntax).

New JSP File	-	×
Steps	Name and L	ocation
1. Choose File Type 2. Name and Location	File <u>N</u> ame:	index
	Project:	WebApplicationJSP
	Location:	Web Pages 🗸
	F <u>o</u> lder:	Browse
	Created File:	$\verb C: Users   a comme.T3500-PC   pesktop   essai\_gwt   JEE2   verification   WebApplication JSP   web   ndex.jsp   a commentation and the set of the set $
	Options: (a) <u>J</u> SP File	(Standard Syntax) Create ag a JSP Segment
	Description:	ument (XML syntax)
	A JSP file usi	ng JSP standard syntax.
		< Back

Un fichier nommé index.jsp apparait dans la partie Web Pages du projet.



La page contient uniquement un « Hello World ».



Faire un clic doit sur index.jsp et choisir Run File.

🗄 🛞 WebApplicationJSP				
🖨 🐻 Web Pages				
🗄 🕖 WEB-INF				
index.) index.x Source Pad	Open View Servlet Compile File	F9	E	
🖨 🚡 Test Packaç	Run File	Mai+F6		
 Libraries	Debug File Profile File	Ctrl+Maj+F5		
⊕	Local History	•		
⊡ · <mark>)</mark> Test Librari ⊕ · · <del>]</del> JUnit 3. ⊕ · · <del>]</del> JUnit 4. ⊕ · · <del>]</del> Persona	Cut Copy Paste	Ctrl+V		
Configurati	Add Delete Rename			
	Save As Template		<b>.</b>	
index.jsp - Navigator			₩ \$	
⊡…≪≫ html	Tools	+		
head	Properties			

Le résultat est alors le suivant :

🕘 JSP Page - Mozilla Firefox	
<u>Fichier</u> Édition Affichage <u>H</u> istorique <u>M</u> arque-pages <u>O</u> utils <u>?</u>	
C X 🏠 📽 🗋 http://localhost:10397/WebApplicationJSP/faces/index. 🏠 🚽 🚷 🖬 Googl	e 🔎
🛃 Les plus visités 🗋 Débuter avec Firefox 🔜 À la une	
JSP Page ÷	-
Hello World!	
Teminé	30 A
Termine	<b>#</b> 🔍

## 3.2. Ajouter du code Java dans un fichier JSP

Les balises <% et %> permettent l'insertion de code Java.

Considérons par exemple le code suivant (les ajouts sont en rouge) :

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
             La date courante est : <%= new java.util.Date() %>
             < %
             String[] langages = {"Java", "C++", "Smalltalk", "Simula 67"};
             out.println("<h3>Principaux langages orientés objets : </h3>");
             for (int i=0; i < langages.length; i++) {
    out.println("<p>" + langages[i] + "");
             }
             25
    </body>
</html>
```

Le résultat sera :



Si on visualise le code de la page, on constate que le code Java n'apparait pas ce qui montre clairement que le fichier JSP est interprété par le serveur et non pas par le client.

```
- O -X
Source de : http://localhost:10397/WebApplicationJSP/faces/index.jsp - Mozilla Firefox
Fichier Édition Affichage ?
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
            La date courante est : Sun Jul 25 12:11:44 CEST 2010
            <h3>Principaux langages orientés objets : </h3>
Java
C++
Smalltalk
Simula 67
    </body>
</html>
```

3.3. Création d'un fichier jsp pour consulter la base de données

Nous allons créer un package nommé **Bean** dans lequel nous allons définir une classe nommée **IndexClient** qui va servir d'intermédiaire vers JPA.

Faire un clic droit sur Source Package et choisir Java Package.

				-	
WebApplicationJSP			32		
🔆 🔥 Web Pages					
🗄 🕡 🚺 WEB-INF	⊕				
index.jsp			33		
index.xhtml					
🖨 🕞 Source Packages				ı.	
🖃 📻 rendez	New 🕨		Web Service	l	
💽 WS			JSF Page	l	
🕀 🊡 Test Packag	Find	۲	Web Service Client	l	
🖨 🔂 Libraries	Paste Ctrl+V	<b>B</b>	Empty File	l	
🗄 🛛 🔂 EJBMod	- contra				
⊕ <b>⊒</b> JDK 1.6	Local History	Ē	Java Package	l	
🗄 🖷 📑 GlassFis		1	Entity Class	l	
🖨 🍃 Test Librari	Tools •	8	Session Bean	l	
🕀 🚔 JUnit 3.	Descention	8	Session Beans For Entity Classes	l	
🕀 📑 JUnit 4.	Properties		lava Class	l	
GlassFish Ser	ver 3 1 (embeddable conta			l	
Enterprise Beans			Entity Classes from Database	ľ	
🖨 🍓 Web Services		<b>i</b>	Persistence Unit	l	
		8	Session Beans For Entity Classes	l	
🖅 🔥 Configuration File	S	<u>a</u> =	JDialog Form	l	
		Դել	Cascading Style Sheet	l	
			Applet	1	
			Appletin	ľ	
		2	Other		

New Java Package	<b>X</b>
Steps	Name and Location
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Package Name: bean
	Project: WebApplicationJSP
	Location: Source Packages
	Created Folder: essai_gwt\essai_gwt_dimanche\JEE2_2\verification\WebApplicationJSP\src\java\bean
	< Back

Faire ensuite un clic droit sur bean puis choisir Java Class.

webApplicationJSP						
🖶 🕞 Web Pages						
index.js	index.isp					
index.x	html					
🖨 🚡 Source Pack	ages					
Bean					_	
🖻 🖷 re	New	•	<b>=</b>	Java Package		
· 💽	Find	Ctrl+F	۷	Web Service	F	
🗈 🚹 Test P	C-4	CHL V	Ð	JSF Page		
E Librari	Cut	Ctri+X	0	Web Service Client		
	Сору	Ctrl+C		Empty File		
	Paste	Ctrl+V		Empty mean		
				Entity Class	1	
E Est L	Delete	Supprimer	1	Session Bean	-	
	Refactor	•	8	Session Beans For Entity Classes	1.	
i <b>≣</b> G	Compile Package	FQ		Java Class		
🕀 🔥 Enter	Local History		8	Entity Classes from Database	1	
🖨 🔂 Web S	Local History	,		Persistence Unit		
	Tools	•	-	Session Beans For Entity Classes		
🗄 😽 Configuration	on Files		<u>a</u> =	JDialog Form		
			Դղ	Cascading Style Sheet		
			-	Applet		
Navigator 👼 Tasks	Results		2	Other		

## Choisir comme nom IndexClient.

New Java Class		×
Steps	Name and I	ocation
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Class <u>N</u> ame:	IndexClient
	Project:	WebApplicationJSP
	Location:	Source Packages 🔹
	Package:	bean 🗸
	Created File:	$[i\_gwt\_dimanche\] JEE2\_2\] verification\] WebApplication\] SP\] src\] ava\] bean\] IndexClient.java\] ava \] ava$
		< Back Next > Finish Cancel Help

On peut alors définir la classe IndexClient comme suit :

```
package bean ;
import java.util.ArrayList;
import java.util.List;
import rendezvous.WSDaoJpaPort;
import rendezvous_client.WSDaoJpaService;
import rendezvous_client.Clients;
public class IndexClient {
    private List<Clients> clients;
   public void setClients(List<Clients> clients) {
        this.clients = clients;
    }
    public List<Clients> getClients()
       WSDaoJpaService ws = new WSDaoJpaService ();
       System.out.println("--- affichage liste des clients ---");
       List<Clients> myArr = new ArrayList<Clients>();
       myArr = ws.getWSDaoJpaPort().getAllClients();
       return myArr;
    }
```

#### 3.4. Modification de la page index.jsp

```
<%@page import="bean.IndexClient,rendezvous client.Clients, java.util.*, RendezVous.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
           <script language="javascript">
             function getClients()
                   <%
                              IndexClient ind = new IndexClient();
                              StringBuffer val = new StringBuffer();
                              for (Clients c : ind.getClients()) {
                                  val.append(c.getNom()+" <br/> ");
                              }
                   %>
                   var chaine = "<%= new String(val)%>";
                   <%--var elt = document.getElementById("res");</pre>
                   elt.value = chaine;--%>
                   document.getElementById("text").innerHTML= chaine;
              }
           </script>
    </head>
    <body>
        <h1>Hello World!</h1>
            La date courante est : <%= new java.util.Date() %>
            <୫
            String[] langages = {"Java", "C++", "Smalltalk", "Simula 67"};
            out.println("<h3>Principaux langages orientés objets : </h3>");
            for (int i=0; i < langages.length; i++) {</pre>
                out.println("" + langages[i] + "");
            ୫>
        <input type="submit" value="getClients" onclick="getClients()"/>
```

```
</body>
</html>
```

Ce qui donne comme résultat d'exécution :





## 1) Déployer l'EJBModule2

Vérifier que l'EJBModule2 compile.





Faire un clic droit sur DaoJpa et choisir Test Web Service.

÷	EJBModule2	ckages	s
	🚡 Test Packa	iges	
	🚡 Libraries		
	🚡 Test Librar	ies	
	🐻 Enterprise	Beans	s
¢.	🚡 Web Servi	ces	
	⊡ DaoJp		
÷	🔈 Configurat		Open
ė.	🙀 Server Res		Refresh
	🔄 🖑 sun-re		
<b>⊕</b> …≜	Test_JSP		Add Operation
🖻 ·· 🌐	WebApplicatio		Tert Web Service
÷)	WebApplicatio		Test web service
			Edit Web Service Attributes
			Configure Handlers
			Generate and Copy WSDL
			Generate SOAP-over-HTTP Wrapper
			Delete
			Properties

## Vous devez obtenir ceci :

DaoJpaService Web Service Tester - Mozilla Firefox	
Eichier Édition Affichage Historique Marque-pages Qutils ?	
C X 🔬 📽 🗋 http://localhost:10397/DaoJpaService/DaoJpa?Tester	🖓 👻 Google 🔎
🖉 Les plus visités 🗋 Débuter avec Firefox 🔜 À la une	
DaoJpaService Web Service Tester	-
DaoJpaService Web Service Tester	_
This form will allow you to test your web service implementation (WSDL File)	
To invoke an operation, fill the method parameter(s) input boxes and click on th name.	e button labeled with the method
Methods :	
public abstract java.util.List dao.DaoJpa.getAllClients() getAllClients ()	
public abstract java.util.List dao.DaoJpa.getAllMedecins() getAllMedecins ()	
public abstract java.util.List dao.DaoJpa.getAllCreneaux(dao.Medecins) getAllCreneaux ()	
public abstract java.util.List dao.DaoJpa.getRvMedecinJour(dao.Medecins.java. getRvMedecinJour ( , )	lang.String)
public abstract dao.Rv dao.DaoJpa.ajouterRv(java.lang.String,dao.Creneaux,da	o.Clients)
Terminé	* &

method invocation trace - Mozilia		
hier Éditio <u>n A</u> ffichage <u>H</u> istori	que <u>M</u> arque-pages <u>O</u> utils <u>?</u>	
🌗 🖓 🗙 🔂 🖣	http://localhost:10397/DaoJpaService/DaoJpa?Tester	☆ - Google
Les plus visités 📄 Débuter avec	Firefox 🔊 À la une	
Method invocation trace	*	
getAllClient	s Method invocation	
Method paramete Type Value	r(s)	
Aethod returned ava.util.List : "[dao OAP Request	o.Clients@1dd8d46, dao.Clients@662911, dao.Clien	ts@1121f65, dao.Clients@cb56cd]"
Method returned ava.util.List : "[dad GOAP Request <s:envelope <s:head <s:body </s:body <td>o.Clients@1dd8d46, dao.Clients@662911, dao.Clien on="1.0" encoding="UTF-8"?&gt; e xmlns:S="http://schemas.xmlsoap.org/soap/enve ler/&gt; '&gt; 2:getAllClients xmlns:ns2="http://dao/"/&gt; ly&gt; pe&gt;</td><td>is@1121f65, dao.Clients@cb56cd]"</td></s:head </s:envelope 	o.Clients@1dd8d46, dao.Clients@662911, dao.Clien on="1.0" encoding="UTF-8"?> e xmlns:S="http://schemas.xmlsoap.org/soap/enve ler/> '> 2:getAllClients xmlns:ns2="http://dao/"/> ly> pe>	is@1121f65, dao.Clients@cb56cd]"

# 2) Créer une web application

New Project	
Steps	Choose Project
<ol> <li>Choose Project</li> <li></li> </ol>	Categories:       Projects:         Image: Solution of the second sec
	Description:  Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.
	< Back

New Web Application	NE same for the impage of the impage of	X
Steps	Name and Location	
1. Choose Project	Project Name: WebApplicationJSP_2	
Server and Settings     Frameworks	$\label{eq:project_location: 2:UsersPhilippelDesktoplessai_gwtlessai_gwt_dimanchelJEE2_2lverification} Project \\ \_ Desktoplessai_gwtlessai_gwt_dimanchelJEE2_2lverification \\ \_ Desktoplessai_gwt_lessai_gwt_dimanchelJEE2_2lverification \\ \_ Desktoplessai_gwt_lessai_gwt_lessai_gwt_dimanchelJEE2_2lverification \\ \_ Desktoplessai_gwt_$	Browse
	$\label{eq:project_folder: sktop} \end{tabular} {\tt Project\_folder: sktop} \end{tabular} satisfy the the the the the the the the the the$	]
	I se Dedicated Folder for Storing Libraries	
	Libraries Folder:	Browse
	Different users and projects can share the same compilation libraries (see Help	
	I Gecalis). I Set as Main Project	
	< <u>B</u> ack Next > Einish Cancel	<u>H</u> elp

New Web Application		tell suggest for other languages and	Address for the file	x
Steps	Server and Set	tings		
<ol> <li>Choose Project</li> <li>Name and Location</li> </ol>	Add to Enterprise	Application: <pre></pre>		-
<ol> <li>Server and Settings</li> <li>Frameworks</li> </ol>	Server:	GlassFish Server 3.1	•	A <u>d</u> d
	Java EE Version	Use dedicated library folder for server JAR files		
	Java EE Version:	Enable Contexts and Dependency Injection		
	Context Path:	/WebApplicationJSP_2		
C C C C C C C C C C C C C C C C C C C				
		< Back Next > Fin	ish Cancel	Help

New Web Application	Mitroper Service and	age of the	X
Steps	Frameworks		
1. Choose Project	Select the frameworks you want to use in your v	eb application.	
3. Server and Settings	Google Web Toolkit		
4. Frameworks	Spring Web MVC		E
	JavaServer Faces		
	Struts 1.3.8		<b>T</b>
	JavaServer Faces Configuration		
	Libraries Configuration		
	Use <u>d</u> efault library which comes with Serve	r (JSF 2.0).	
	<u>R</u> egistered Libraries: JSF 2.0		▼
	Create New Library		
	<u>l</u> SF Folder:		Browse
	Library Name:		
	< <u>8</u>	ack Next > Finish	Cancel <u>H</u> elp

Pour terminer, supprimer EJBModule2 des applications déployées sur le serveur GlassFish.



3) Création d'un service web client





Ceci doit donner la visualisation suivante :



3.2. Ensuite faire un clic droit sur Libraries et choisir Add Jar.



Choisir ensuite le .jar de l'EJBModule2.

3.3. Création d'un web service client



New Web Service Client	×
Steps	WSDL and Client Location
1. Choose File Type 2. WSDL and Client Location	Specify the WSDL file of the Web Service.
	Project:     Browse
	O Local File:
	O WSDL URL: Set Proxy
	Specify a package name where the client java artifacts will be generated:
	Project: WebApplicationJSP_2
	P <u>a</u> ckage:
	Client <u>S</u> tyle: JAX-WS Style
	Cenerate Dispatch code
	(i) Enter the URL of the service you wish to use.
	< Back Next > Einish Cancel Help

Browse Web	Services	X
Web Services:		
	dule2 getAllClients: List getAllCreneaux: List getAllCreneaux: List getRvMedecinJour: List ajouterRv: Rv supprimerRv: void getClientById: Clients getMedecinById: Medecins getRvById: Rv getCreneauById: Creneaux	
		OK Cancel

Attention, pensez à redéployer le EJBModule2 avant.

Steps	WSDL and Client Location			
<ol> <li>Choose File Type</li> <li>WSDL and Client Location</li> </ol>	Specify the WSDL file of the Web Service.			
	Project:	ittp://localhost:8080/DaoJpaService/DaoJpa?wsdl	Browse	
	<u>Local File:</u>		Browse	
	O WSDL URL:		Set Proxy	
	geen	WebApplicationJSP_2		
	P <u>a</u> ckage:	vebApplicationJSP_2 rendezvous	•	
	P <u>a</u> ckage:	rendezvous		
	P <u>a</u> ckage: Client <u>S</u> tyle:	VebApplicationJSP_2 rendezvous JAX-WS Style	•	
	P <u>a</u> dkage: Client <u>S</u> tyle:	WebApplicationJSP_2 rendezvous JAX-WS Style spatch code		

Le projet doit ressembler à ce qui suit :



3.4. Création d'un package bean et d'une classe IndexClient

Nous allons créer un package nommé **Bean** dans lequel nous allons définir une classe nommée **IndexClient** qui va servir d'intermédiaire vers JPA.

Faire un clic droit sur Source Package et choisir Java Package.



🕞 New Java Package	X
Steps	Name and Location
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Package <u>N</u> ame: bean
	Project: WebApplicationJSP_2
	Location: Source Packages
	<u>C</u> reated Folder: iai_gwt\essai_gwt_dimanche\JEE2_2\verification\WebApplicationJSP_2\src\java\bean
2-11/	< Back

Faire un clic droit sur Source Package sur bean et choisir Java Class

E Test P	New	•	<b>=</b>	Java Package
	Find	Ctrl+F	<u> </u>	Web Service Client
🗈 🚺 Genera	Cut	Ctrl+X		Java Class
🖨 🍃 Librarie	Conv	Ctrl+C	Ð	JSP
	Copy	Chille V	۷	Web Service
	Paste	Ctrl+v	Ð	JSF Page
🕀 🍃 Test Li	Delete	Supprimer		Empty File
UC 🚍 🗉	Refactor	+	8	Entity Class
UL 🚍 🕀	Compile Package	FQ	8	Session Bean
HIT Gla	Local History		8	Session Beans For Entity Classes
	Locarrinstory	,	8	Entity Classes from Database
🕀 🛞 Cri	Tools	+	ð	Persistence Unit
🗄 🛞 DaoJpa			8	Session Beans For Entity Classes
MedecinsFacade			<u>a</u> =	JDialog Form
H KvFacade			$\mathcal{T}_{\mathcal{T}}}}}}}}}}$	Cascading Style Sheet
🖶 💦 Configuratio	n Files			Utner

🕽 New Java Class	X
Steps	Name and Location
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Class Name: IndexClient
	Project: WebApplicationJSP_2
	Location: Source Packages
	Package: bean 🗸
	Created File: ]wt_dimanche\JEE2_2\verification\WebApplicationJSP_2\src\java\bean\IndexClient.java
	< Back Next > Finish Cancel Help

Ouvrir le fichier IndexClient.java et insérer le code suivant :

```
package bean;
import java.util.ArrayList;
import java.util.List;
import rendezvous.DaoJpaService;
import rendezvous.Clients;
public class IndexClient {
    private List<Clients> clients;
    public void setClients (List<Clients> clients) {
        this.clients = clients;
    }
```

```
public List<Clients> getClients()
{
    DaoJpaService ws = new DaoJpaService ();
    System.out.println("--- affichage liste des clients ---");
    List<Clients> myArr = new ArrayList<Clients>();
    myArr = ws.getDaoJpaPort().getAllClients();
    return myArr;
  }
}
```

3.5. Création d'une page JSP

Faire un clic doit sur Web Page et choisir JSP.



New JSP File	out having	X	
Steps	Name and L	ocation	
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	File <u>N</u> ame:	index	
	Project:	WebApplicationJSP_2	
	Location:	Web Pages 🔹	
	F <u>o</u> lder:	Browse	
	Created File:	$\label{eq:c:UsersPhilippelpesktoplessai_gwtlessai_gwt_dimanchelyEE2_2lverification} \\ WebApplication \\ JSP_2lveb \\ \\ index. \\ jsp \\ label{eq:spin} \\ label{eq$	
	Options: (a) JSP File (Standard Syntax)  (c) Create as a JSP Segment (c) Standard Syntax)		
	© JSP Jocument (ZML Syntax) Description:		
	A JSP file usir	ng JSP standard syntax.	
		< Back	

Il faut ensuite, faire « undeploy » de l'EJBModule2 et arrêter le serveur GlassFish.

Faire un clic droit sur index.jsp et choisir Run File.



Le résultat est alors le suivant :



# 3.6. Modification de la page JSP Modifiez la page comme suit :

```
<%@page import="bean.IndexClient,rendezvous.Clients, java.util.*, rendezvous.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
           <script language="javascript">
             function getClients()
             ł
                   <%
                             IndexClient ind = new IndexClient();
                             StringBuffer val = new StringBuffer();
                             for (Clients c : ind.getClients()) {
                                  val.append(c.getNom()+" <br/> ");
                              }
                   8>
                   var chaine = "<%= new String(val)%>";
                   <%--var elt = document.getElementById("res");</pre>
                   elt.value = chaine;--%>
                   document.getElementById("text").innerHTML= chaine;
              }
           </script>
    </head>
    <body>
        <h1>Hello World!</h1>
            La date courante est : <%= new java.util.Date() %>
            <%
            String[] langages = {"Java", "C++", "Smalltalk", "Simula 67"};
            out.println("<h3>Principaux langages orientés objets : </h3>");
            for (int i=0; i < langages.length; i++) {</pre>
                out.println("" + langages[i] + "");
            ୫>
        <input type="submit" value="getClients" onclick="getClients()"/>
        </body>
</html>
```

Ce qui donne comme résultat d'exécution :

🕹 JSP Page - Mozilla Firefox	0.00		
Eichier Édition Affichage Historique	Marque-pages Qutils ?		
	http://localhost:10397/WebApplicationJSP/faces/index.jsp	Google 🔎	4
Les plus visités Débuter avec Firef	A la une		
JSP Page	*		-
Hello Wor	ld!		
La date courante est :	Mon Jul 26 15:12:07 CEST 2010		
Principaux langa	ges orientés objets :		
Java			
C++			
Smalltalk			
Simula 67			
getClients			
MARTIN			
GERMAN			
JACQUARD			
BISTROU			
Terminé		* &	



La particularité de cette solution est que l'on n'utilise pas le concept d'EJB. En effet, les beans et pages web vont directement venir se greffer sur la couche DAO. Cette solution peut paraître plus longue car on recrée tout depuis le début (à partir de JPA) mais en fait, elle est rapide à mettre en place.

1) Créer une web application

New Project	NE saper for street	
Steps	Choose Project	
<ol> <li>Choose Project</li> <li></li> </ol>	Categories: Android Android Java JavaFX Java FX Java Web Java EE Java Card Java ME Java ME Maven HPP Ruby	Projects: Web Application Web Application with Existing Sources Web Free-Form Application
	C/C++ Description: Creates an empty Web application in IDE-generated build script to build, run	a standard IDE project. A standard project uses an n, and debug your project.

New Web Application	A Neuros Researces	x
Steps	Name and Location	
Choose Project     Name and Location     Server and Settings	Project Name: WebApplicationJSP_3 Project Location: C:\Users\Jacomme.T3500-PC\Deskton\essai.owt\JEE2\verification Browse	
4. Frameworks	Project Eolder: 2.T3500-PC\Desktop\essai_gwt\JEE2\verification\WebApplicationJSP_3	
	Use Dedicated Folder for Storing Libraries	
	Libraries Folder: Browse	
	Different users and projects can share the same compilation libraries (see Help for details).	
	☑ Set as Main Project	
	< <u>B</u> ack Next > Einish Cancel Help	

New Web Application	A Record Tools Reactions	x
Steps	Server and Settings	
<ol> <li>Choose Project</li> <li>Name and Location</li> </ol>	Add to Enterprise Application:	
<ol> <li>Server and Settings</li> <li>Frameworks</li> </ol>	Server: Personal GlassFish v3 Domain	A <u>d</u> d
	Use dedicated library folder for server JAR files	
	Zava EE Version: Java EE 6 web 🗸	
	< Back Next > Einish Cancel	

New Web Application	×
Steps	Frameworks
1. Choose Project	Select the frameworks you want to use in your web application.
3. Server and Settings	Google Web Toolkit
4. ITalleworks	Spring Web MVC 2.5 ☑ JavaServer Faces
	Struts 1.3.8
1	Hibernate 3.2.5
	JavaServer Faces Configuration
	Libraries Configuration
	Use default library which comes with Server (JSF 2.0).
	© <u>C</u> reate New Library
	1SF Folder: Browse
	Library Name:
	< Back Nevt > Finish Cancel Help

## 2) Créer la couche dao et jpa

Cette partie reprend certains point du cours 1.

Dans un **premier temps** nous allons créer une unité de persistance (faire Clic Droit sur WebApplicationJSP\_3 et choisir New->Others).

New File		×
Steps 1. Choose File Type 2	Choose File Type Project: WebApplicationJSP_3	Fån Turnen:
		<ul> <li>Entity Class</li> <li>Entity Classes from Database</li> <li>JPA Controller Classes from Entity Classes</li> <li>JSF Pages from Entity Classes</li> <li>Persistence Unit</li> <li>Database Schema</li> <li>Session Beans For Entity Classes</li> <li>RESTful Web Services from Entity Classes</li> <li>RESTful Web Services from Database</li> </ul>
	Creates an JSR220 persistence unit. If per	sistence.xml doesn't exist, it is created.
	< <u>B</u> ack	Next > Einish Cancel Help

Choisir ensuite comme Data Source : base\_medecin\_JNDI Et comme fournisseur de service de persistance EclipseLink.

New Persistence Unit		
Steps	Provider and Databa	ise
<ol> <li>Choose File Type</li> <li>Provider and Database</li> </ol>	Persistence <u>U</u> nit Name: Specify the persistence	WebApplicationJSP_3PU provider and database for entity dasses.
	Data Source:	EclipseLink(JPA 2.0)(default)
	🔽 Use Java Transactio	n APIs
	Table Generation Strate	egy: O Create O Drop and Create O None
		< <u>B</u> ack Next > Finish Cancel Help

Attention à ne pas choisir une stratégie englobant la génération des tables. Elles existent déjà grâce au script SQL que nous avons utilisé au départ.

Dans la partie « Configuration Files », le fichier persistence.xml apparaît.



Dans un **deuxième temps**, nous allons créer des entités JPA. Comme précédemment faire Clic Droit / New / Other.



New Entity Classes from Database	•		States in Suprembry States	×
Steps	Database Tables			
1. Choose File Type 2. Database Tables	<u>Data Source:</u>	jdbc/base_medecin_JNDI		-
A. Mapping Options	O Database Schema	<no database="" in="" project="" schemas="" the=""></no>		
	Available <u>T</u> ables:		Selected Tables:	
		Add >	creneaux medecins	
		< <u>R</u> emove	rv	
		<pre></pre>		
		<< Remove All		
< <u>B</u> ack Next > Einish Cancel Help				<u>H</u> elp

Etant donné qu'il s'agit de la génération d'entités JPA, on peut choisir JPA comme nom de package.

New Entity Classes from Database	•		×
Steps	Entity Class	25	
1. Choose File Type 2. Database Tables	Specify the na	mes and the location of the entity classes.	
3. Entity Classes 4. Mapping Options	<u>Class Names:</u>	Database Table	Class Name
		clients creneaux	Creneaux
		medecins	Medecins
		rv	Rv
	Project:	WebApplicationJSP_3	
	Location:	Source Packages	•
	Pac <u>k</u> age:	jpa	•
	🔽 Generate I	Named Query Appotations for Persistent Fields	
	<u>o</u> cherater	tanica query principations for reconstent rielas	
		< <u>B</u> ack Next	> <u>Finish</u> Cancel <u>H</u> elp

New Entity Classes from Database	
Steps	Mapping Options
Choose File Type     Database Tables     Entity Classes     Mapping Options	Specify the default mapping options. Association Fetch: default       Qollection Type:     iava.util.List       Fully Qualified Database Table Names
	Attributes for <u>Regenerating</u> Tables
	< <u>Back</u> Next > Einish Cancel Help

Veuillez à bien choisir « java.util.List » pour Collection Type (pour avoir plus de lisibilité).

Dans la classe Rv nous ajoutons un nouveau constructeur permettant de créer un rendez vous en donnant une date, un client et un jour.

```
public Rv(Date jour, Clients client, Creneaux creneau)
{
    this.jour = jour ;
    this.idClient = client ;
    this.idCreneau = creneau ;
```

Enfin nous allons créer un package nommée **dao**, qui représente l'interface de l'EJB pour l'accès aux données.

New Java Package		×
Steps	Name and Loc	ation
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Package <u>N</u> ame:	dao
	Project:	WebApplicationJSP_3
	Location:	Source Packages
	Created Folder:	re.T3500-PC\Desktop\essai_gwt\JEE2\verification\WebApplicationJSP_3\src\java\dao
		< <u>Back</u> Next > <u>Finish</u> Cancel <u>H</u> elp

Le projet se présente alors comme suit :



Ajoutons une classe DaoJpa qui contiendra l'intelligence métier.

Il faut rajouter le code suivant dans le fichier DaoJpa.java.

```
* To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.text.SimpleDateFormat;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;
import jpa.Clients;
import jpa.Creneaux;
import jpa.Medecins;
import jpa.Rv;
/**
 * @author lacomme
 */
public class DaoJpa {
private EntityManager em;
    private EntityManagerFactory emf;
   private EntityTransaction tx ;
    public void init()
    Ł
        emf = Persistence.createEntityManagerFactory("WebApplicationJSP 3PU");
        em = emf.createEntityManager();
       tx = em.getTransaction();
        tx.begin();
    ł
    public void close()
       em.close();
        emf.close();
   }
    // liste des clients
    public List<Clients> getAllClients() {
        try {
           return em.createQuery("select c from Clients c").getResultList();
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        }
   }
    // liste des médecins
    public List<Medecins> getAllMedecins() {
       try {
           return em.createQuery("select m from Medecins m").getResultList();
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        }
   }
    // liste des créneaux horaires d'un médecin donné
    // medecin : le médecin
    public List<Creneaux> getAllCreneaux(Medecins medecin) {
        try {
           return em.createQuery("select c from Creneaux c join c.medecin m where
m.id=:idMedecin").setParameter("idMedecin", medecin.getId()).getResultList();
       } catch (Exception e) {
         e.printStackTrace();
         return null;
        }
    }
```

```
// liste des Rv d'un médecin donné, un jour donné
    // medecin : le médecin
    // jour : le jour
    public List<Rv> getRvMedecinJour(Medecins medecin, String jour) {
        try {
            return em.createQuery("select rv from Rv rv join rv.creneau c join c.medecin m where
m.id=:idMedecin and rv.jour=:jour").setParameter("idMedecin", medecin.getId()).setParameter("jour",
new SimpleDateFormat("yyyy:MM:dd").parse(jour)).getResultList();
        }catch (Exception e) {
         e.printStackTrace();
        return null;
        ł
    }
    // ajout d'un Rv
    public Rv ajouterRv(String jour, Creneaux creneau, Clients client) {
        try {
           Rv rv = new Rv(new SimpleDateFormat("yyyy:MM:dd").parse(jour), client, creneau);
            em.persist(rv);
            return rv;
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        ł
    }
    // suppression d'un Rv
    // rv : le Rv supprimé
   public void supprimerRv(Rv rv) {
       try {
            em.remove(em.merge(rv));
        }catch (Exception e) {
         e.printStackTrace();
        }
   }
    // recuperer un client donné
   public Clients getClientById(Long id) {
       try {
           return (Clients) em.find(Clients.class, id);
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        ł
    ł
    // recuperer un médecin donné
    public Medecins getMedecinById(Long id) {
        try {
            return (Medecins) em.find(Medecins.class, id);
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        }
    }
    // recuperer un Rv donné
    public Rv getRvById(Long id) {
        try {
           return (Rv) em.find(Rv.class, id);
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        }
   }
    // recuperer un créneau donné
    public Creneaux getCreneauById(Long id) {
        try {
           return (Creneaux) em.find(Creneaux.class, id);
        } catch (Exception e) {
         e.printStackTrace();
        return null;
        ł
    ł
```

## 3) Ajout de page JSP



3.1. Remarquons le fichier index.xtml qu'on peut visualiser simplement par un clic droit.

Ceci doit donner la visualisation suivante :



3.2. Ajout d'une page JSP

WebApplication	JSP_3			
i → D WEB-	New 🕨	Ð	JSF Page	
index.	Find		JSP	
E Test Pack	Darte (trl+V	8	Java Class	
💼 <defa< th=""><td>Paste Ctil+V</td><td><b>•••</b></td><td>Java Package Web Service Client</td><td></td></defa<>	Paste Ctil+V	<b>•••</b>	Java Package Web Service Client	
EJBM	Local History •		Web Service	
DK 1	Properties		Empty File	
Test Libraries	assi ish və boman	<u>8</u>	Session Beans For Entity Classes	
🖶 🔥 Enterprise Bea	ns Files	<u>8</u>	Session Beans For Entity Classes	
Web Pages - Navigator	100	<u></u>	JPA Controller Classes from Entity Classes	
		8	Entity Classes from Database	
		8	Entity Class	
			Persistence Unit	
		8	Servlet	
		2	Other	

New JSP File		X		
Steps	Name and Location			
1. Choose File Type 2. Name and Location	File <u>N</u> ame:	index		
	Project:	WebApplicationJSP_3		
	Location:	Web Pages 👻		
	F <u>o</u> lder:	Browse		
	Created File:	le: C:\Users\Jacomme.T3500-PC\Desktop\essal_gwt\JEE2\verification\WebApplicationJSP_3\web\jindex.jsp		
	Options:			
	③ <u>J</u> SP File	ISP File (Standard Syntax) Create as a JSP Segment		
	O JSP Doc Description:	JSP Document (XML Syntax)		
	A JSP file using JSP standard syntax.			
		<pre><back next=""> Finish Cancel Help</back></pre>		

Le projet doit ressembler à ce qui suit :



Il faut tester la page JSP en faisant un clic droit et Run File.



Fichier Édition Affichage Historique Marque-pages Qutils ?     C      I tes plus visités © bébuter avec Firefox À la une   JSP Page      Hello World?	JSP Page - Mozilla Firefox		
C A     Les plus visités     JSP Page     Hello World!	<u>Fichier</u> Édition <u>Affichage</u> <u>H</u> istorique	<u>M</u> arque-pages <u>O</u> utils <u>?</u>	
Les plus visits Debuter avec Firefox   JSP Page    Hello World! Image: Image	🔇 🕑 - C 🗙 🏠 😵	http://localhost:10397/WebApplicationJSP_3/faces/index.jsp	☆ - Kar Google 🔎
i JSP Page i	🔊 Les plus visités 📄 Débuter avec Firefo	🗴 🔜 À la une	
<section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	🗋 JSP Page	*	
Terminé 😵 🌾	Hello Wor	ld!	
	Terminé		¥ 6.

3.3. Création d'un package bean et d'une classe IndexClient

Nous allons créer un package nommé **Bean** dans lequel nous allons définir une classe nommée **IndexClient** qui va servir d'intermédiaire vers JPA.

\_

🖨 🌄 Web Pages			19 /hodys
			TO C/DODA
			19 //html>
index.jsp			20
index.xhtml			
🖶 🚡 Source Parles			
🖨 🚹 Test Packa	New 🕨	œ۲	JSP
💼 <defa< th=""><th>Land Library A</th><th>Ð</th><th>JSF Page</th></defa<>	Land Library A	Ð	JSF Page
🖨 🔓 Libraries		8	Java Class
EJBMo	Find		Java Package
UK I.		۲	Web Service Client
E Test Librar	Paste Ctrl+V		Web Service
🕀 🔥 Enterprise	Tools •		Empty File
🗄 🖟 Configurat	Properties	8	Session Beans For Entity Classes
Source Packages - wave	Jator	8	Session Beans For Entity Classes
		8	Java Interface
		8	JPA Controller Classes from Entity Classes
		8	Entity Classes from Database
		8	Entity Class
		٩	Persistence Unit
		8	Servlet
<no available="" view=""></no>		2	Other

Faire un clic droit sur **Source Package** et choisir **Java Package**.

Steps	Name and Loc	ation
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Package <u>N</u> ame:	bean
	Project:	WebApplicationJSP_3
	Location:	Source Packages
	Created Folder:	.T3500-PC\Desktop\essai_gwt\JEE2\verification\WebApplicationJSP_3\src\java\bea
× · · ·		

## Faire un clic droit sur Source Package sur bean et choisir Java Class

				~~
i 🔡 be	New	Þ	<b>#</b>	Java Package
	Find	Ctrl+F	<u> </u>	Web Service Client
🗈 🚹 Genera	Cut	Ctrl+X		Java Class
🖨 🍃 Librarie	Conv	Ctrl+C	Ð	JSP
	Copy	Challe V	۷	Web Service
	Paste	Ctri+v	D	JSF Page
🕀 🍃 Test Li	Delete	Supprimer		Empty File
ut 🚭 🗄	Refactor	Þ	8	Entity Class
Ut 🚰 🗉	Compile Package	FQ	8	Session Bean
H. Entern	Local History		8	Session Beans For Entity Classes
⊡ - A Cli	Local History	,	8	Entity Classes from Database
🗈 🛞 Cri	Tools	Þ	ð	Persistence Unit
🕀 👧 DaoJpa			8	Session Beans For Entity Classes
HereinsFacade			<u>a</u> =	JDialog Form
Web Service References		$\mathcal{T}_{\mathcal{T}_{\mathcal{T}}}$	Cascading Style Sheet	
			-	i.
🗇 🐻 Configuration Files				Uther

New Java Class	×
Steps	Name and Location
<ol> <li>Choose File Type</li> <li>Name and Location</li> </ol>	Class Name: IndexClient
	Project: WebApplicationJSP_3
	Location: Source Packages
	Pad <u>k</u> age: bean 🗸
	Created File: sktop\essai_gwt\JEE2\verification\WebApplicationJSP_3\src\java\bean\IndexClient.java
	<back next=""> Einish Cancel Help</back>

Ouvrir le fichier IndexClient.java et insérer le code suivant :

```
package bean;
import java.util.ArrayList;
import java.util.List;
import dao.DaoJpa;
import jpa.Clients;
public class IndexClient {
    private DaoJpa dao;
    private List<Clients> clients;
   public void setClients(List<Clients> clients) {
        this.clients = clients;
    }
    public List<Clients> getClients()
      dao = new DaoJpa ();
      dao.init(); //On initialise la couche de persistence
       System.out.println("--- affichage liste des clients ---");
       List<Clients> myArr = new ArrayList<Clients>();
       myArr = dao.getAllClients();
       dao.close();
       return myArr;
    3
}
```

#### 3.4. Modification de la page JSP

Modifiez la page comme suit :

```
<%@page import="bean.IndexClient,jpa.Clients, java.util.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
           <script language="javascript">
            function getClients()
            ł
                  <୫
                            IndexClient ind = new IndexClient();
                            StringBuffer val = new StringBuffer();
                            for (Clients c : ind.getClients()) {
                                val.append(c.getNom()+" <br/> ");
                            }
                  %>
                  var chaine = "<%= new String(val)%>";
                  <%--var elt = document.getElementById("res");</pre>
                  elt.value = chaine;--%>
                  document.getElementById("text").innerHTML= chaine;
             }
```

Ce qui donne comme résultat d'exécution :



----- fin -----