

# Création et utilisation d'applications client/serveur avec JEE

« Le royaume des web services »

Jeudi 5 Septembre 2013 – JDev2013

Romain Guidoux, Libo Ren,  
Jonathan Fontanel, Philippe Lacomme



# Création et utilisation d'applications client/serveur avec JEE

« Le royaume des web services »

Document sous Licence Creative Commons

Copyright (c) 2013. **Fontanel-Guidoux-Lacomme-Ren**



« Cette licence permet aux autres de remixer, arranger, et adapter votre œuvre à des fins non commerciales tant qu'on vous crédite en citant votre nom et que les nouvelles œuvres sont diffusées selon les mêmes conditions. »



# Qui sommes-nous ?

- Stand INRA au salon de l'agriculture
- Application smartphone couplée à un web service
- Expert des systèmes mobiles Android



Romain Guidoux (INRA) 3

<http://www.isima.fr/~lacomme/pagewebsevice/websevice>



Jonathan Fontanel (Qualiac)  
Philippe Lacomme (UBP, LIMOS)  
Libo Ren (UdA, LIMOS)



# Plan de l'atelier

---

**Partie I : Quelques rappels et Pré requis**

**Partie II : Création d'un premier web service**

**Partie III : Test d'un WS de géolocalisation d'une adresse IP**

**Partie IV : Test d'un WS de conversion de monnaie**

**Partie V : Réalisation d'un client de conversion de monnaie**

**Partie VI : Réalisation d'un client de gestion d'adresse IP**

# **Partie I : Quelques rappels et Pré requis**

# Avant de commencer ...

## Outils de développement & pilotes

### Outils de développement

Java EE SDK (92 Mo) : <http://www.oracle.com/technetwork/java/javaee/downloads> ou [Téléchargement local](#)

NetBeans JEE + GlassFish (204 Mo) : <http://download.netbeans.org> ou [Téléchargement local](#)

### Consignes

#### Téléchargement de Java EE SDK :

Aller sur la page <http://www.oracle.com/technetwork/java/javaee/downloads/> et choisir **Java EE 7 SDK** :

Overview Downloads Documentation Community Technologies Training

### Java EE 7 SDK Downloads

 Java EE with JDK DOWNLOAD	 Java EE DOWNLOAD	 Java EE Web Profile DOWNLOAD	 GlassFish DOWNLOAD
Java EE 7 SDK with JDK 7 U21	Java EE 7 SDK	Java EE 7 SDK Web Profile	Oracle GlassFish Server 3.1.2.2 (Java EE 6 Compatible)

Previous Releases

#### Téléchargement de NetBeans :

Aller sur la page <https://netbeans.org/downloads/> et choisir le bouton Download qui se trouve en bas de la colonne **Java EE** :

NetBeans IDE 7.3.1 Download

7.3 | 7.3.1 | Development | Archive



Installer **JAVA 7 JDK**  
Puis Installer **NetBeans 7.3 version Java EE**  
**Test d'ouverture de NetBeans**

# Java...NetBeans... GlassFish

## Fonctionne en Java



Environnement de développement



Fonction principale : Créer des applications Java



Serveur d'application

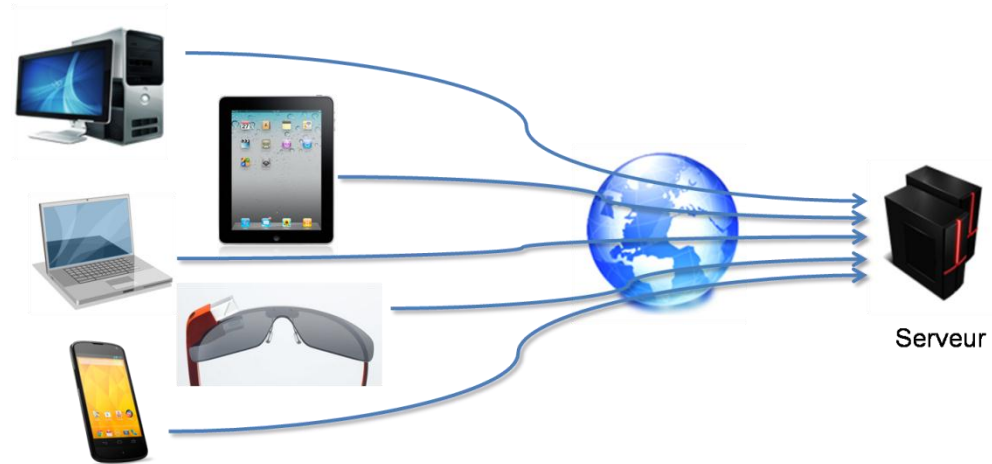


Fonction principale : héberger des applications web

# Architecture clients-serveur

- **Dans l'univers grand public :**

- Application météo
- Programme TV
- GPS Waze



- **Dans l'univers de la recherche :**

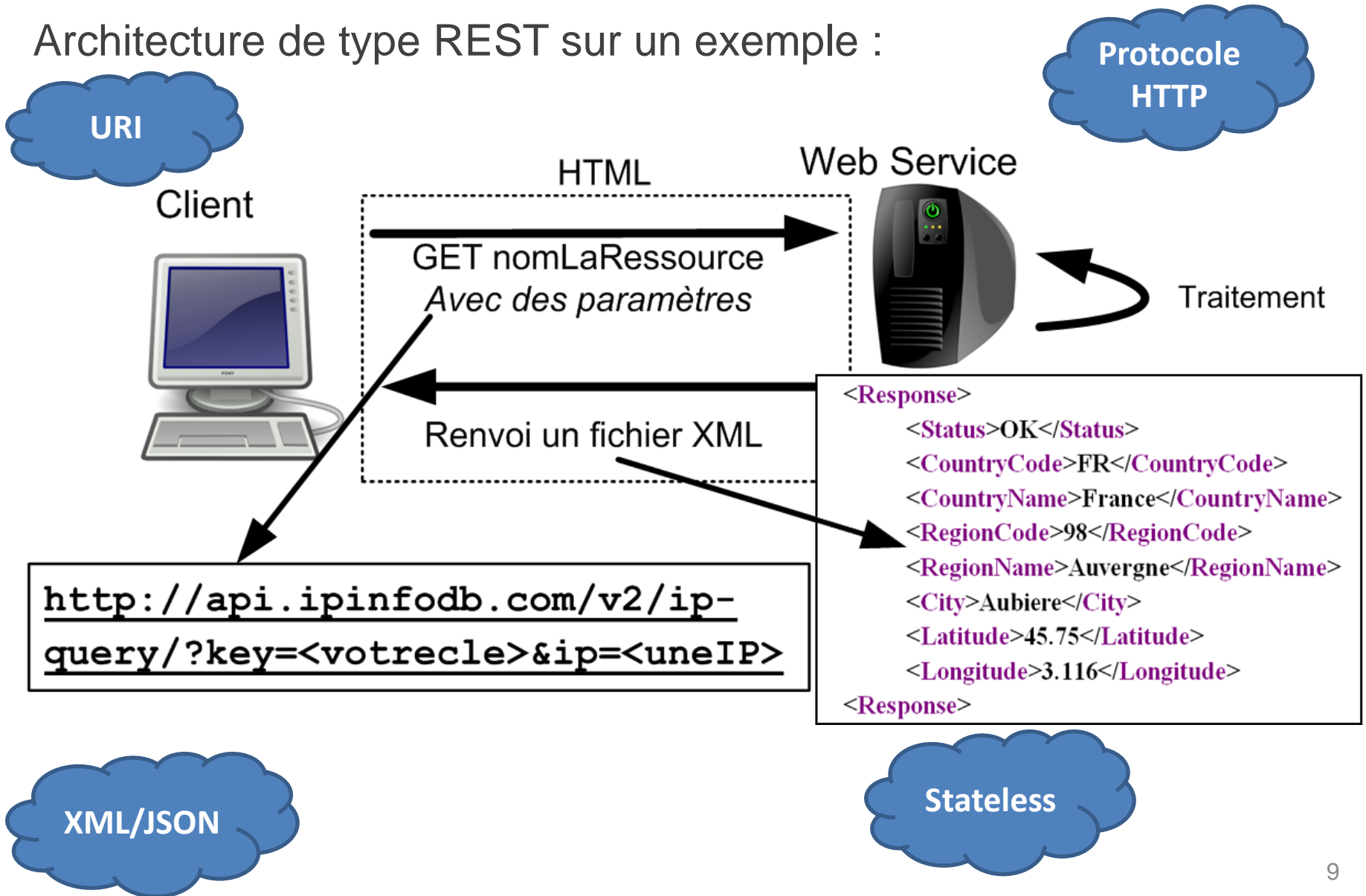
- Archive HAL
- Développement d'une application d'estimation de la dépense énergétique à l'INRA
- Mise à disposition de métaheuristiques par le LIMOS
- ...



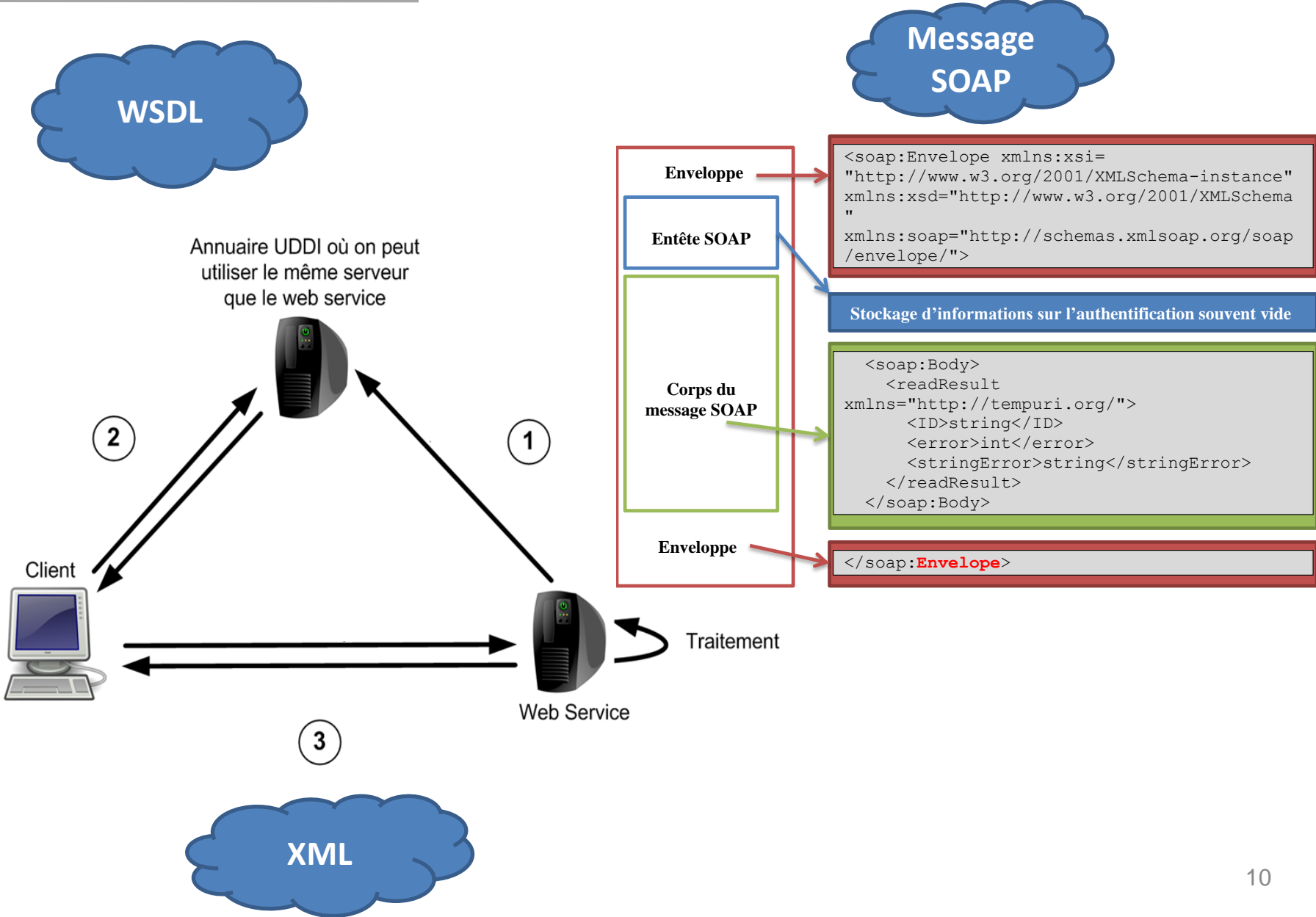


# L'architecture REST

Architecture de type REST sur un exemple :



# Le protocole SOAP





# Où trouver des WS ?

**Web Services Directory** Subscribe to get the latest APIs

Sort by: **Name** | Date | Popularity | Category

Hide Filters ▲

Keywords:

Category:

Company:

Protocols / Styles:

Data Format:

Date:

Managed By:

Viewing 1 to 3000 of 5977 APIs

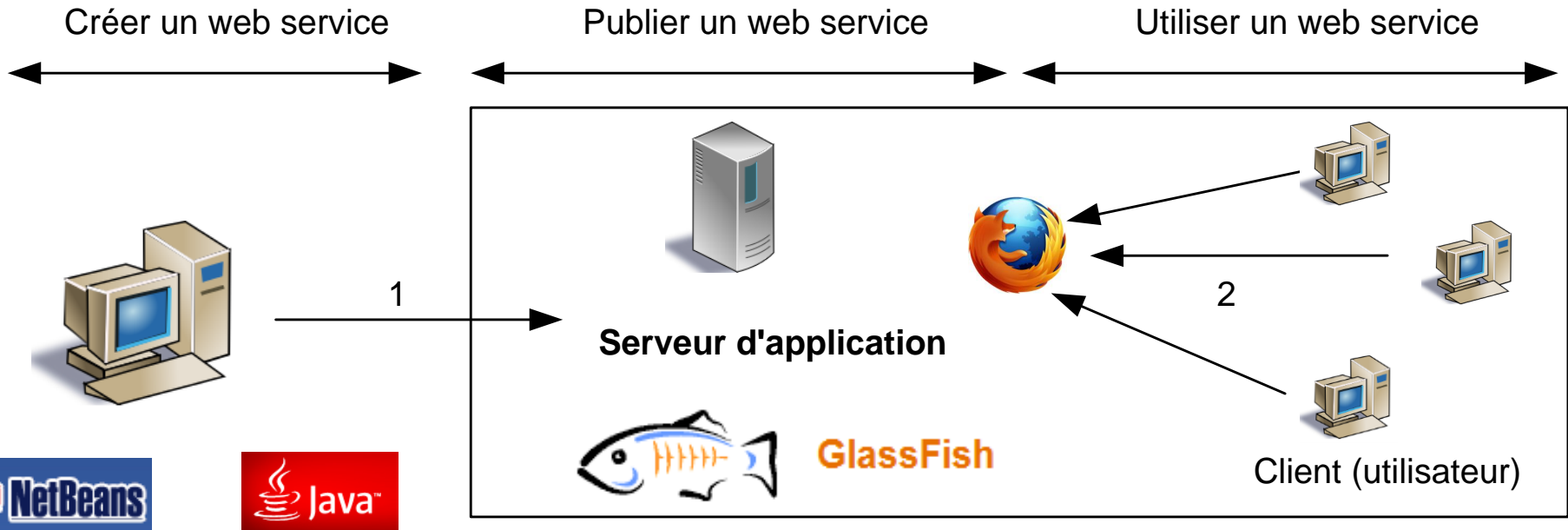
API	Description	Category	Protocols / Styles
Hoptoad	Application error notification service	Tools	5
OneLogin	Single sign-on solution	Security	3
Ordr.in	Restaurant E-Commerce System	Food	2
#blue	Text messaging storage service	Messaging	3
.NET Daily Fact	.NET daily fact widget	Internet	9
.tel	Access to .tel DNS	Internet	0
100 Facts About Me	Fact sharing service	Social	9

Protocols / Styles dropdown menu items: AIM (OSCAR), Atom, Blogger, DTC-XML, GData, GET, hCalendar, iCal, JavaScript, JSON-RPC, POST, REST, RSS, SMS, SOAP, XML-RPC, XMPP

<http://www.programmableweb.com/apis/directory>

# **Partie II : Création d'un premier web service**

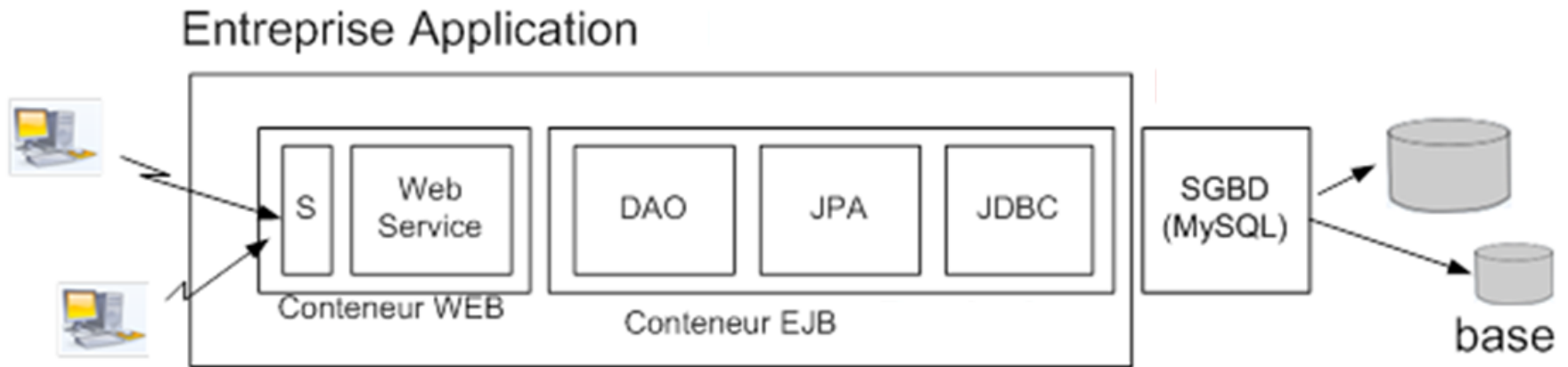
# 1..2..3 On crée un Web Service



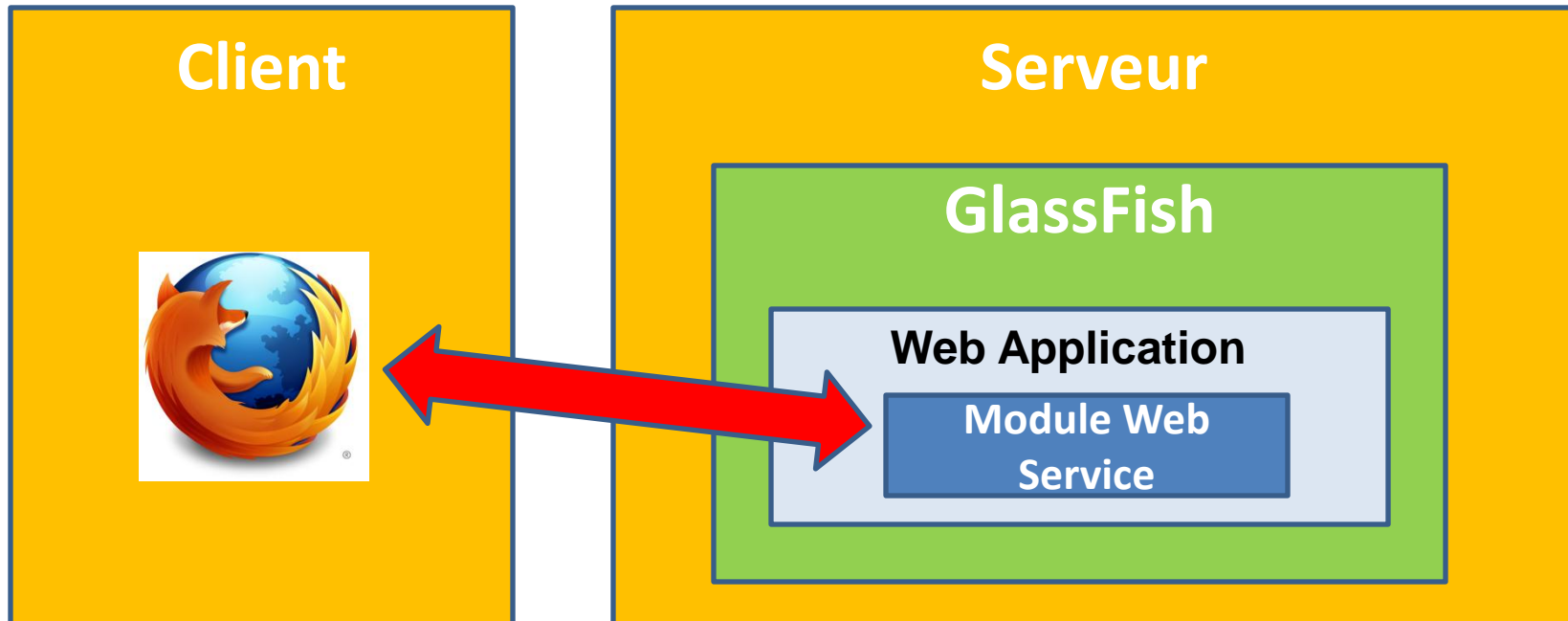
1

2

3



On va faire un peu plus simple pour commencer :



# **Partie II : Création d'un premier web service**

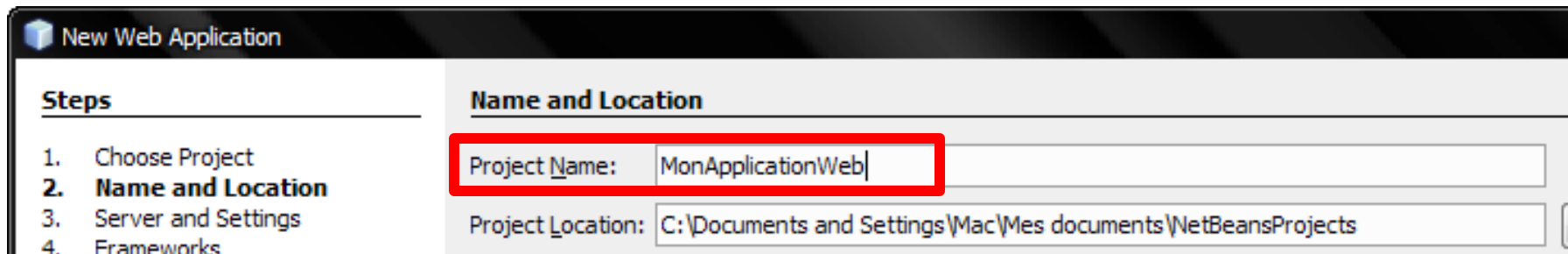
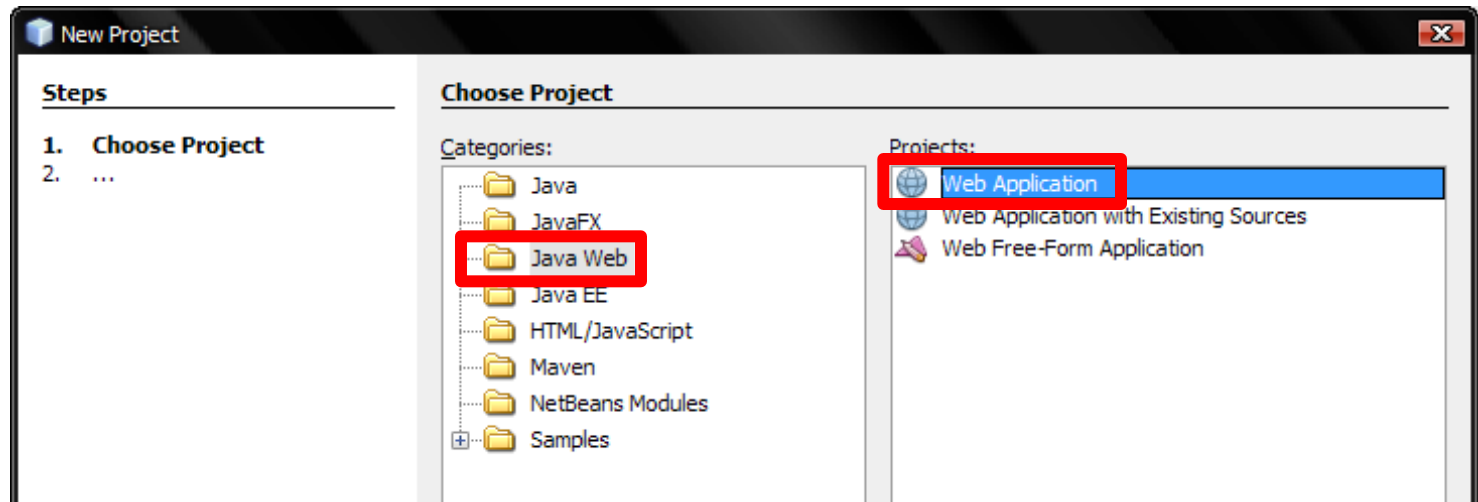
**Partie 1. Au niveau serveur**

**Partie 2. Tester et Comprendre son projet**

**Partie 3. Au niveau client**

# Définir un projet sur le serveur

- Démarrer Netbeans
- Créer un projet sur le serveur





New Web Application

**Steps**

1. Choose Project
2. Name and Location
- 3. Server and Settings**
4. Frameworks

**Server and Settings**

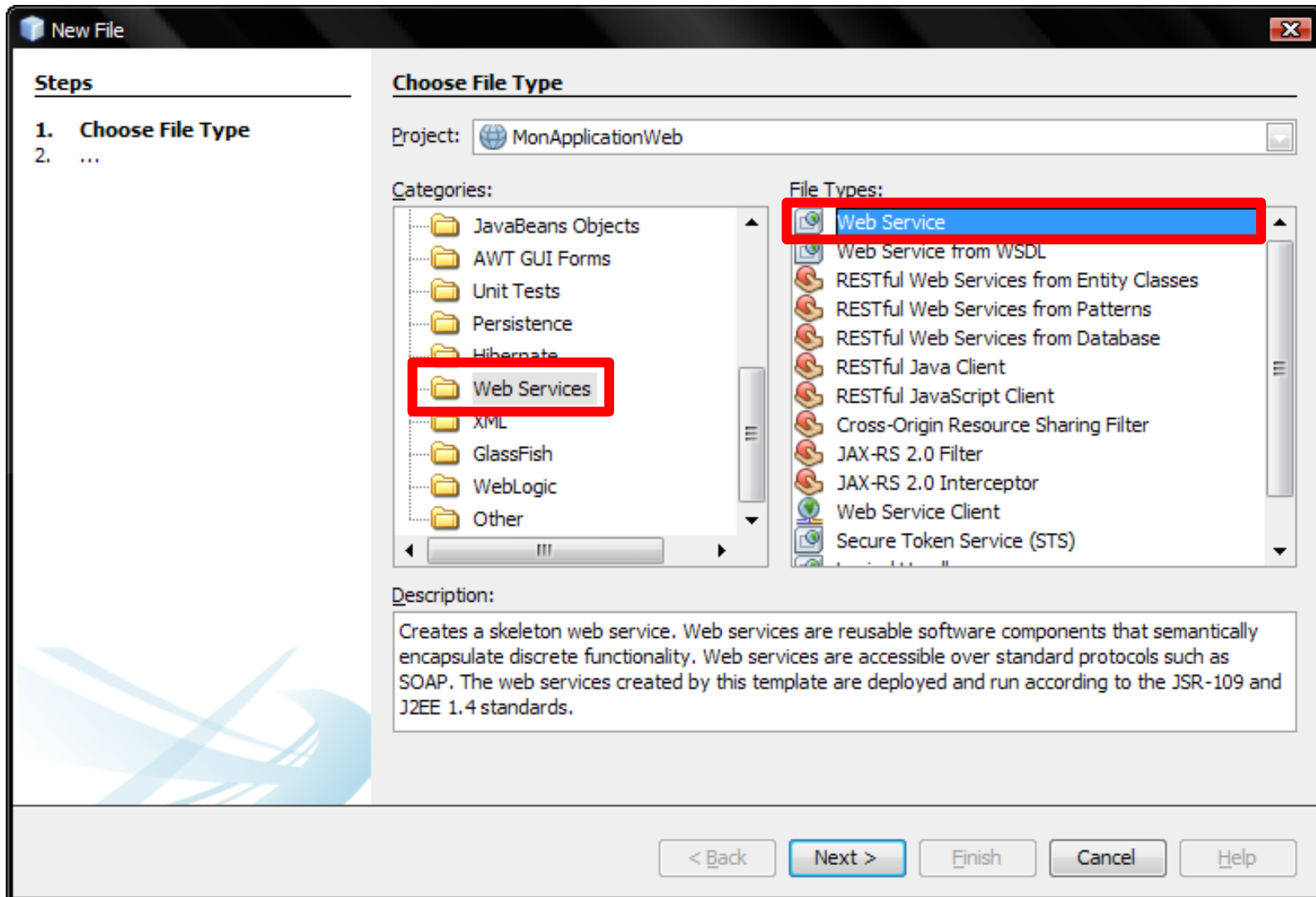
Add to Enterprise Application: <None>

Server: GlassFish Server 4.0

Java EE Version: Java EE 7 Web

Context Path: /MonApplicationWeb

< Back Next > Finish Cancel Help



**New File**

**Steps**

1. Choose File Type
2. ...

**Choose File Type**

Project: MonApplicationWeb

**Categories:**

- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Persistence
- Hibernate
- Web Services**
- XML
- GlassFish
- WebLogic
- Other

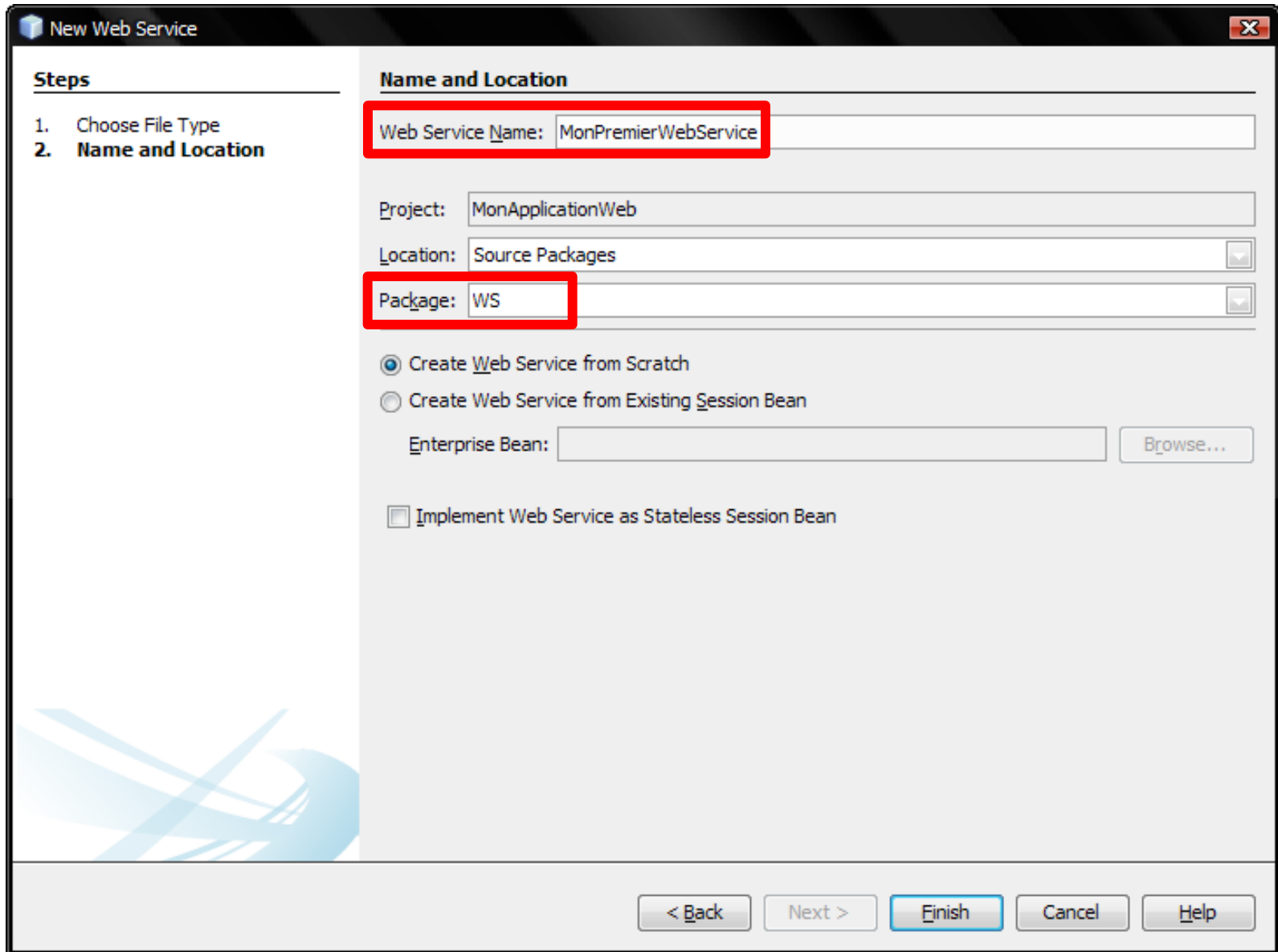
**File Types:**

- Web Service**
- Web Service from WSDL
- RESTful Web Services from Entity Classes
- RESTful Web Services from Patterns
- RESTful Web Services from Database
- RESTful Java Client
- RESTful JavaScript Client
- Cross-Origin Resource Sharing Filter
- JAX-RS 2.0 Filter
- JAX-RS 2.0 Interceptor
- Web Service Client
- Secure Token Service (STS)

**Description:**

Creates a skeleton web service. Web services are reusable software components that semantically encapsulate discrete functionality. Web services are accessible over standard protocols such as SOAP. The web services created by this template are deployed and run according to the JSR-109 and J2EE 1.4 standards.

< Back   Next >   Finish   Cancel   Help



**New Web Service**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Web Service Name: MonPremierWebService

Project: MonApplicationWeb

Location: Source Packages

Package: WS

Create Web Service from Scratch

Create Web Service from Existing Session Bean

Enterprise Bean:

Implement Web Service as Stateless Session Bean

< Back   Next >   **Finish**   Cancel   Help



# Un « nouveau » projet Java

MonApplicationWeb - NetBeans IDE 7.3.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

MonApplicationWeb

- Web Pages
  - Source Packages
    - WS
      - MonPremierWebService.java
- Libraries
  - Web Services
    - MonPremierWebService
      - hello: String
- Configuration Files

```
4  /*
5  package WS;
6
7  import javax.jws.WebService;
8  import javax.jws.WebMethod;
9  import javax.jws.WebParam;
10
11 /**
12  *
13  * @author Mac
14
15  @WebService(serviceName = "MonPremierWebService")
16  public class MonPremierWebService {
17
18  /**
19  * This is a sample web service operation
20
21  @WebMethod(operationName = "hello")
22  public String hello(@WebParam(name = "name") String txt) {
23      return "Hello " + txt + " !";
24  }
25
26 }
```

```
@WebMethod(operationName = "hello")
public String hello(@WebParam(name = "name") String txt) {
    return "Hello " + txt + " !";
}
```

```
@WebMethod(operationName = "additionner")
public int additionner(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {

    int s = 0;

    for(int i=1;i<5000;i++)
    {
        for(int j=1;j<5000;j++)
        {
            for(int k=1;k<5000;k++)
            {
                s = s + 1;
            }
        }
    }

    return (s+a+b-s);
}
}
```

```
MonApplicationWeb (run) × Java DB Database Process × GlassFish Server 4.0 ×
Created dir: C:\Documents and Settings\Mac\Mes documents\NetBeansProjects\MonApplicationWeb\build\empty
Created dir: C:\Documents and Settings\Mac\Mes documents\NetBeansProjects\MonApplicationWeb\build\generated-sources\ap
compile:
compile-jsp:
Starting GlassFish Server 4.0
GlassFish Server 4.0 is running.
In-place deployment at C:\Documents and Settings\Mac\Mes documents\NetBeansProjects\MonApplicationWeb\build\web
Initializing...
run-deploy:
Browsing: http://localhost:8080/MonApplicationWeb
run-display-browser
run:
BUILD SUCCESSFUL (total time: 40 seconds)
```

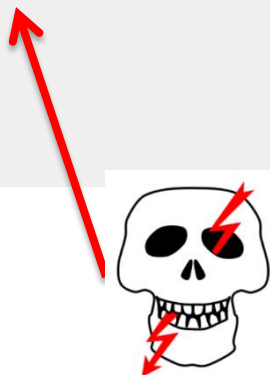


Adresse de la web application :

```
Browsing: http://localhost:8080/MonApplicationWeb
```

<http://localhost:8080/MonApplicationWeb>

```
MonApplicationWeb (run) x  Java DB Database Process x  GlassFish Server 4.0 x
INFO:  Génération du code...
INFO:  Compilation du code...
INFO:  wsimport successful
INFO:  Invoking wsimport with http://localhost:8080/MonApplicationWeb/MonPremierWebService?WSDL
INFO:  analyse du WSDL...
INFO:  Génération du code...
INFO:  Compilation du code...
INFO:  wsimport successful
INFO:  Invoking wsimport with http://localhost:8080/MonApplicationWeb/MonPremierWebService?WSDL
INFO:  analyse du WSDL...
INFO:  Génération du code...
INFO:  Compilation du code...
INFO:  wsimport successful
```



Adresse du fichier WSDL de notre web service :

<http://localhost:8080/MonApplicationWeb/MonPremierWebService?WSDL>

# **Partie II : Création d'un premier web service**

**Partie 1. Au niveau serveur**

**Partie 2. Tester et Comprendre son projet**

**Partie 3. Au niveau client**





MonPremierWebService Testeur de service Web - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

Connexion...

localhost:8080/MonApplicationWeb/MonPremierWebService?Tester

## MonPremierWebService Testeur de service Web

Ce formulaire vous permettra de tester votre implémentation de service Web [Fichier WSDL](#)

Pour appeler une opération, remplissez les zones de saisie des paramètres de la méthode et cliquez sur le bouton libellé avec

**Méthodes :**

public abstract java.lang.String ws.MonPremierWebService.hello(java.lang.String)

hello (  )

---

public abstract int ws.MonPremierWebService.additionner(int,int)

additionner (  ,  )



Trace d'appel de méthode - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ? ↓ [Print] [Star]

Trace d'appel de méthode +

localhost:8080/MonApplicationWeb/MonPremierWebService?Tester

## ajouter Appel de méthode

---

**Method parameter(s)**

Type	Value
int	2
int	3

---

**Méthode renvoyée**

int : "5"



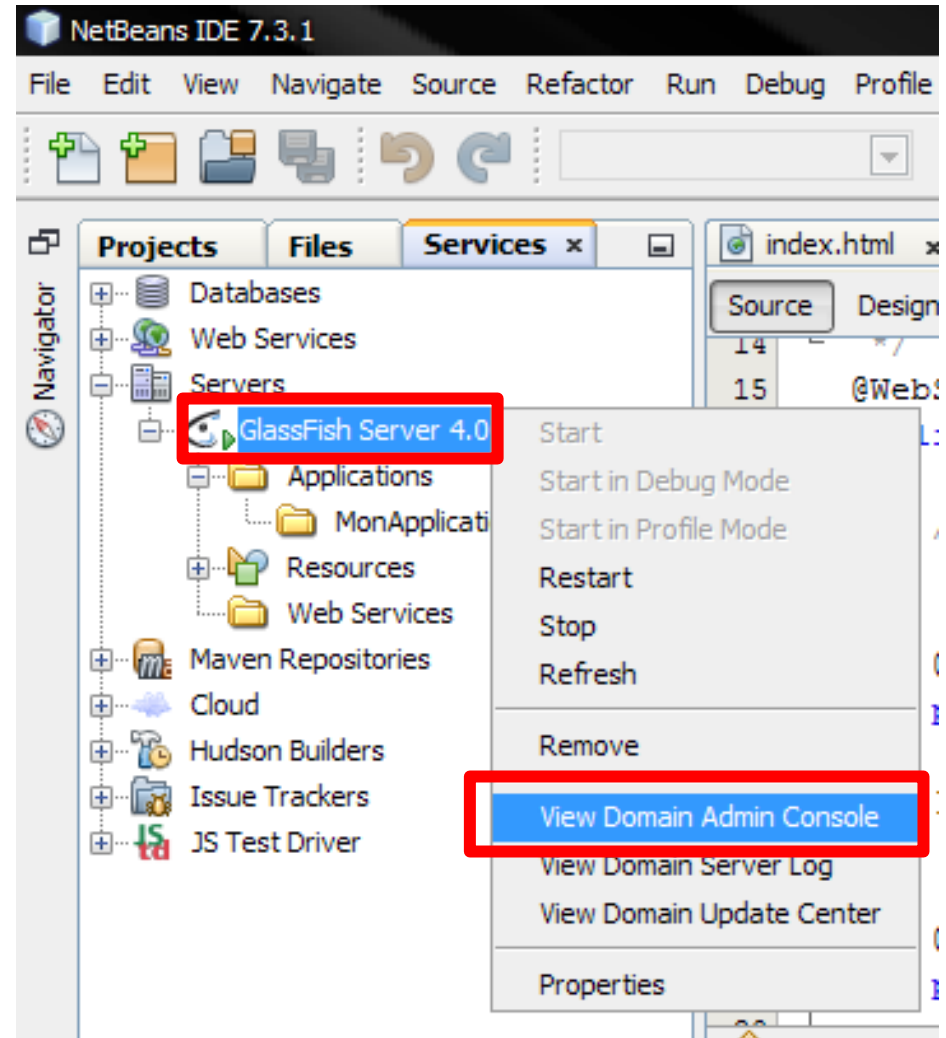
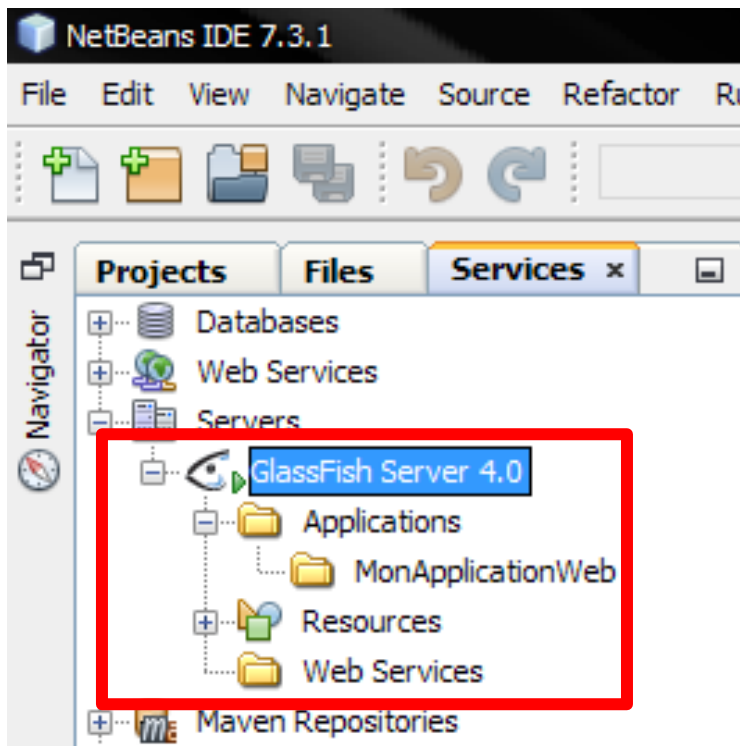
GlassFish

# Comprendre son projet

Serveur d'application



Fonction principale :  
héberger des applications  
web





Console GlassFish - Tâches courantes - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

Trace d'appel de méthode x Console GlassFish - Tâches courantes x +

localhost:4848/common/index.jsf

additionner

Accueil A propos de... Aide

Utilisateur : admin | Domaine : domain1 | Serveur : localhost

## GlassFish™ Server Open Source Edition

### Arborescence

- Tâches courantes
  - Domaine
    - serveur (serveur d'administration)
  - Clusters
  - Instances autonomes
  - ▶ Noeuds
  - ▶ Applications
  - ▶ Modules de cycle de vie
  - ▶ Données de surveillance
  - ▶ Ressources
    - ▶ Ressources simultanées
    - ▶ Connecteurs
    - ▶ JDBC
    - ▶ Ressources JMS
    - ▶ JNDI
      - ▶ Sessions JavaMail
    - ▶ Configurations d'adaptateurs de r
  - ▶ Configurations
    - ▶ default-config
    - ▶ server-config
  - ▶ Outil de mise à jour

### Console GlassFish - Tâches courantes

#### Actualités de GlassFish

- Prise en charge
- Inscription
- Actualités de GlassFish

#### Déploiement

- Lister les applications déployées
- Déployer une application

#### Administration

- Modifier le mot de passe administrateur
- Lister les alias de mot de passe

#### Surveillance

- Données de surveillance

#### Documentation

- Documentation Open Source Edition
- Guide de démarrage rapide
- Guide d'administration
- Guide du développement d'application
- Guide du déploiement d'application

#### Centre de mises à jour

- Composants installés
- Mises à jour disponibles
- Extensions disponibles

#### Ressources

- Créer une ressource JDBC
- Créer un pool de connexions JDBC



Applications

Les applications peuvent être de type Enterprise ou Web, ou différentes sortes de modules. Pour redémarrer une application ou un module, cliquez sur le lien de rechargement ; cette action sera uniquement appliquée aux cibles sur lesquelles l'application ou le module est activé.

**Applications déployées (1)**

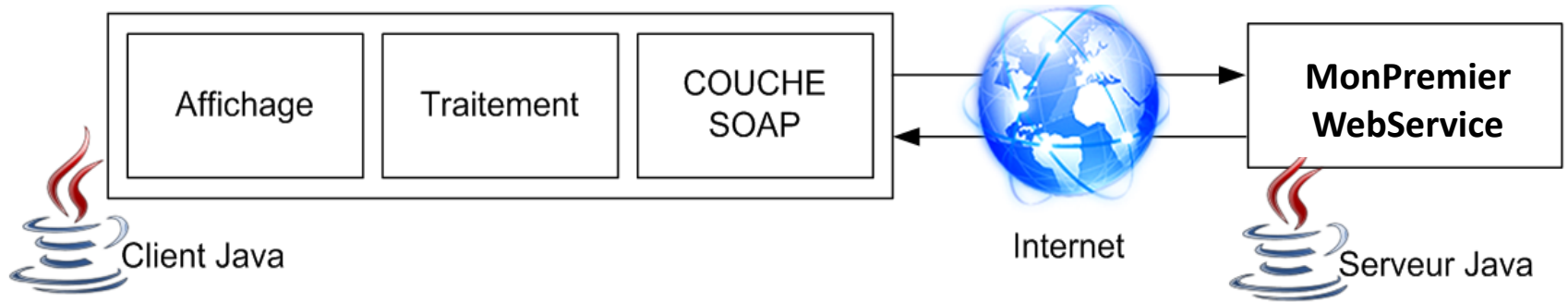
Select	Nom	Ordre de déploiement	Activé	Moteurs	Action
<input type="checkbox"/>	MonApplicationWeb	100	<input checked="" type="checkbox"/>	webservices, web	Lancer   Redéploiement   Recharger

# **Partie II : Création d'un premier web service**

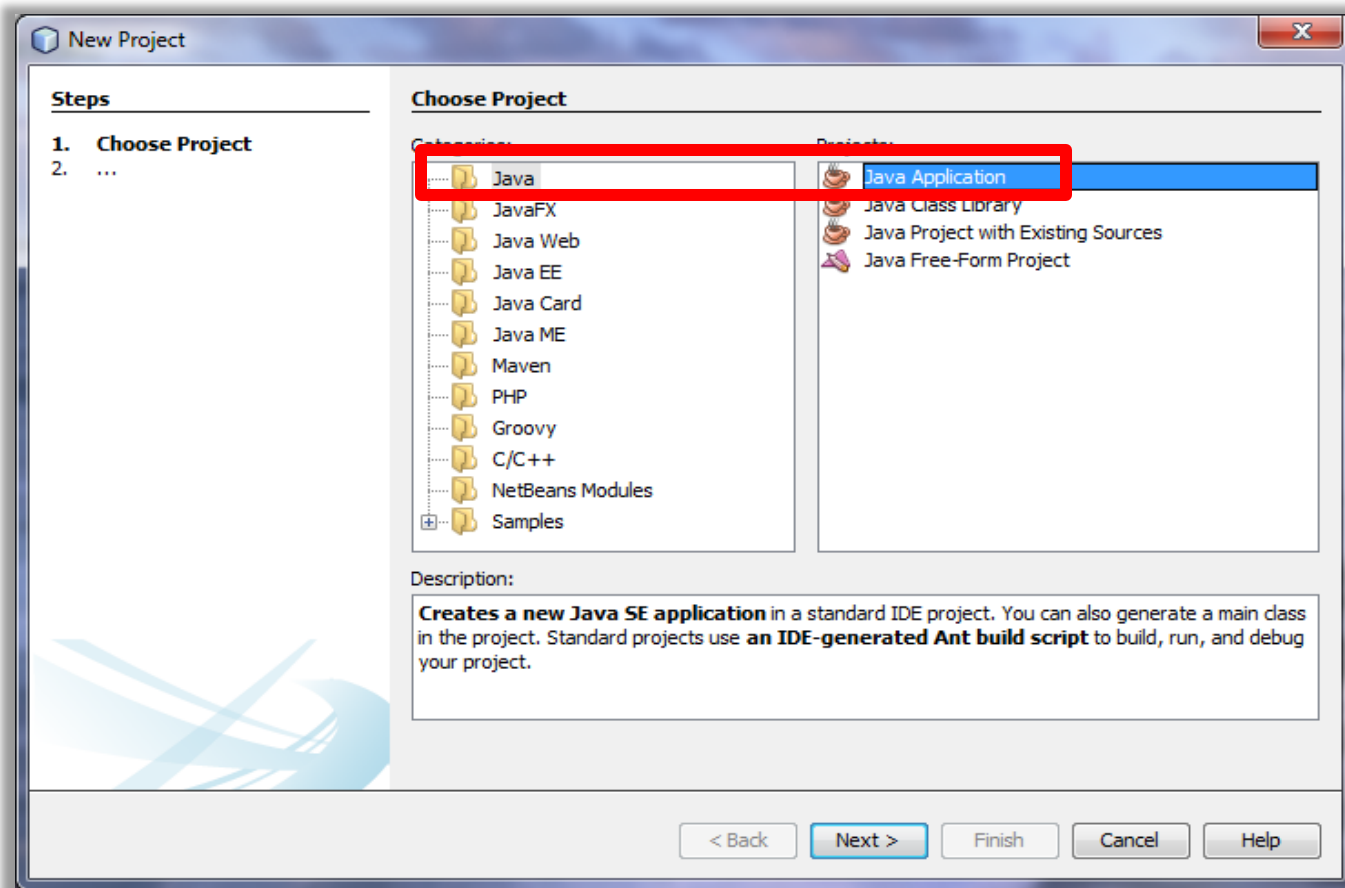
**Partie 1. Au niveau serveur**

**Partie 2. Tester et Comprendre son projet**

**Partie 3. Au niveau client**



- Démarrer Netbeans
- Créer un projet





- Démarrer Netbeans
- Créer un projet

**Name and Location**

---

Project Name:

Project Location:

Project Folder:

---

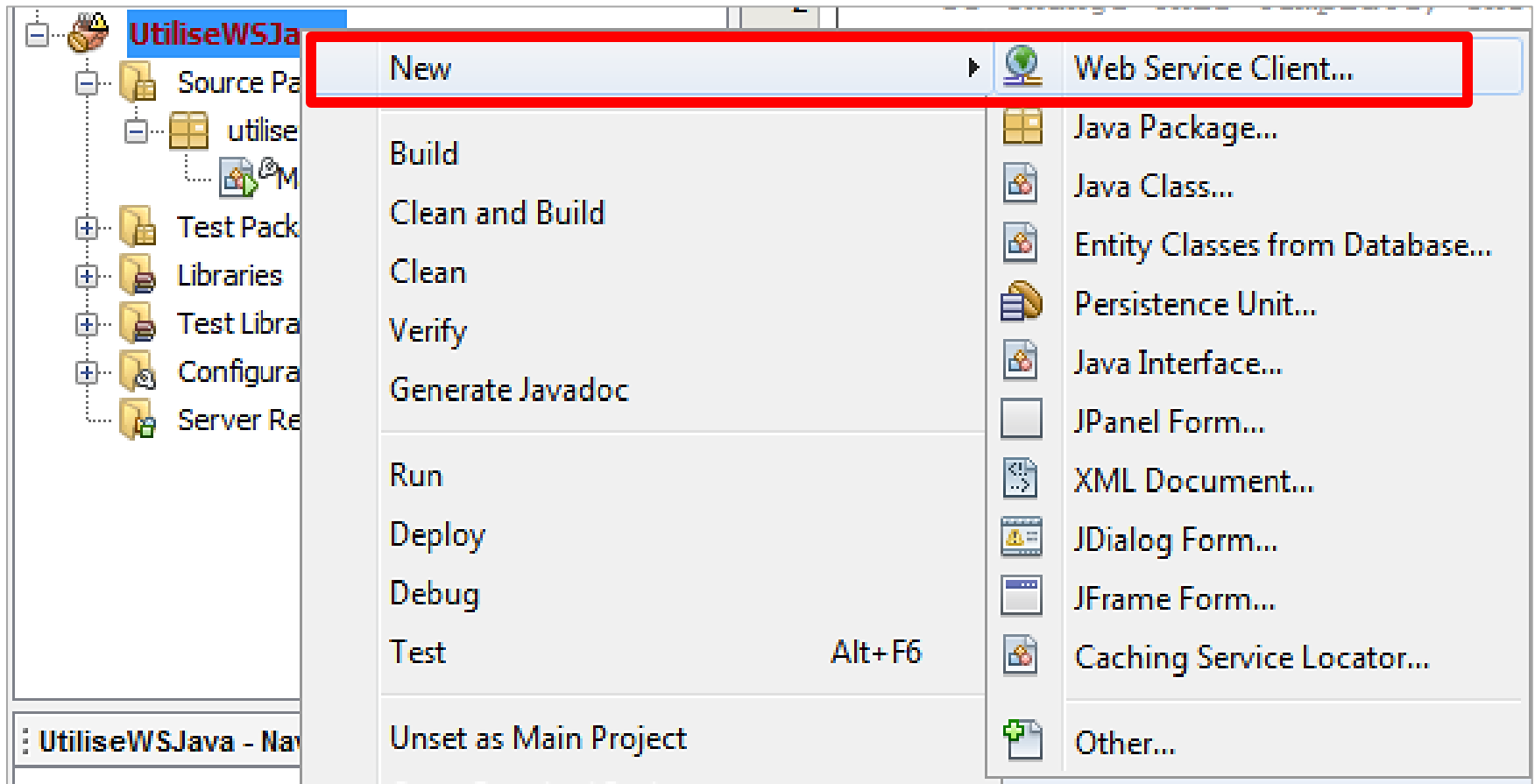
Use Dedicated Folder for Storing Libraries

Libraries Folder:

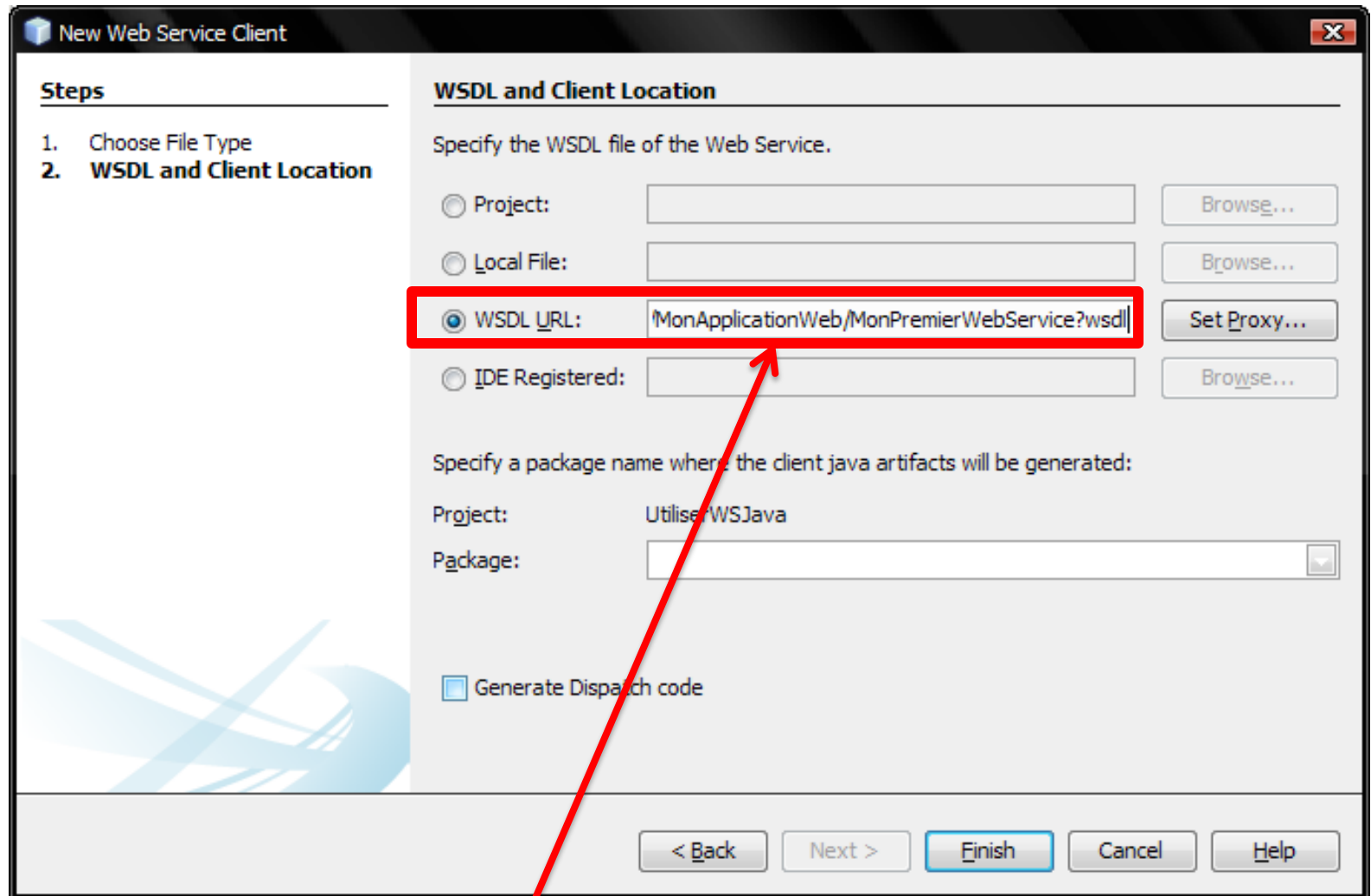
Different users and projects can share the same compilation libraries  
(see Help for details).

Set as Main Project

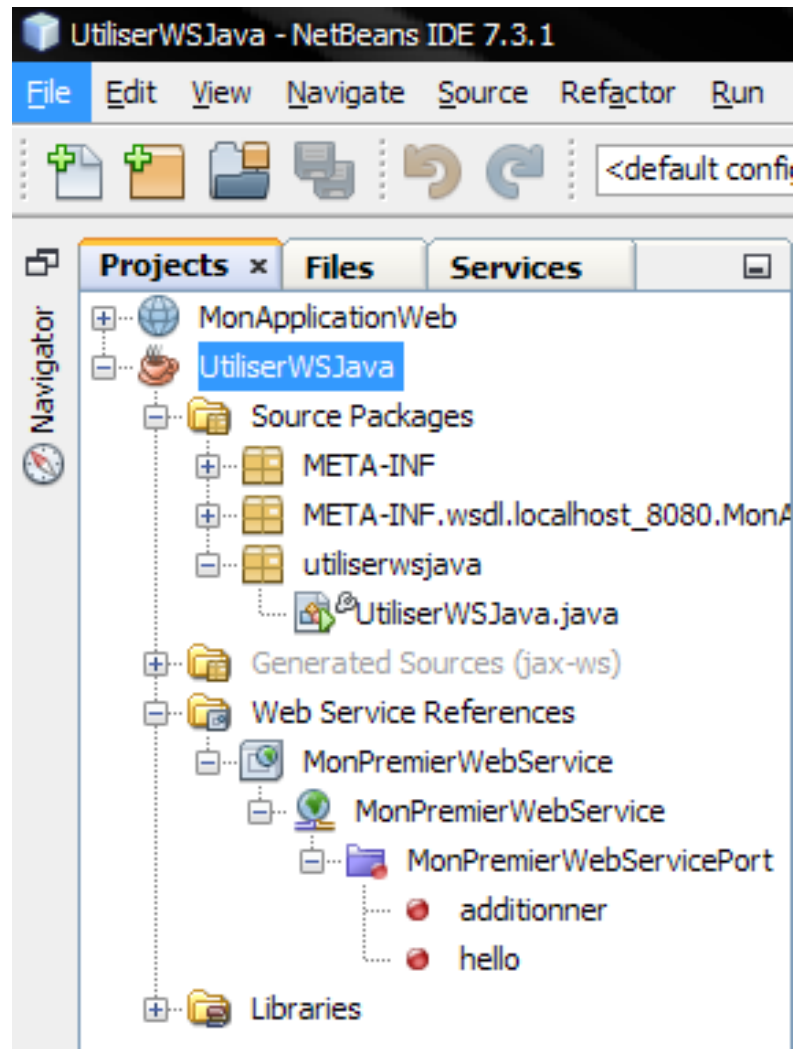
# Ajout d'une Ref. de service

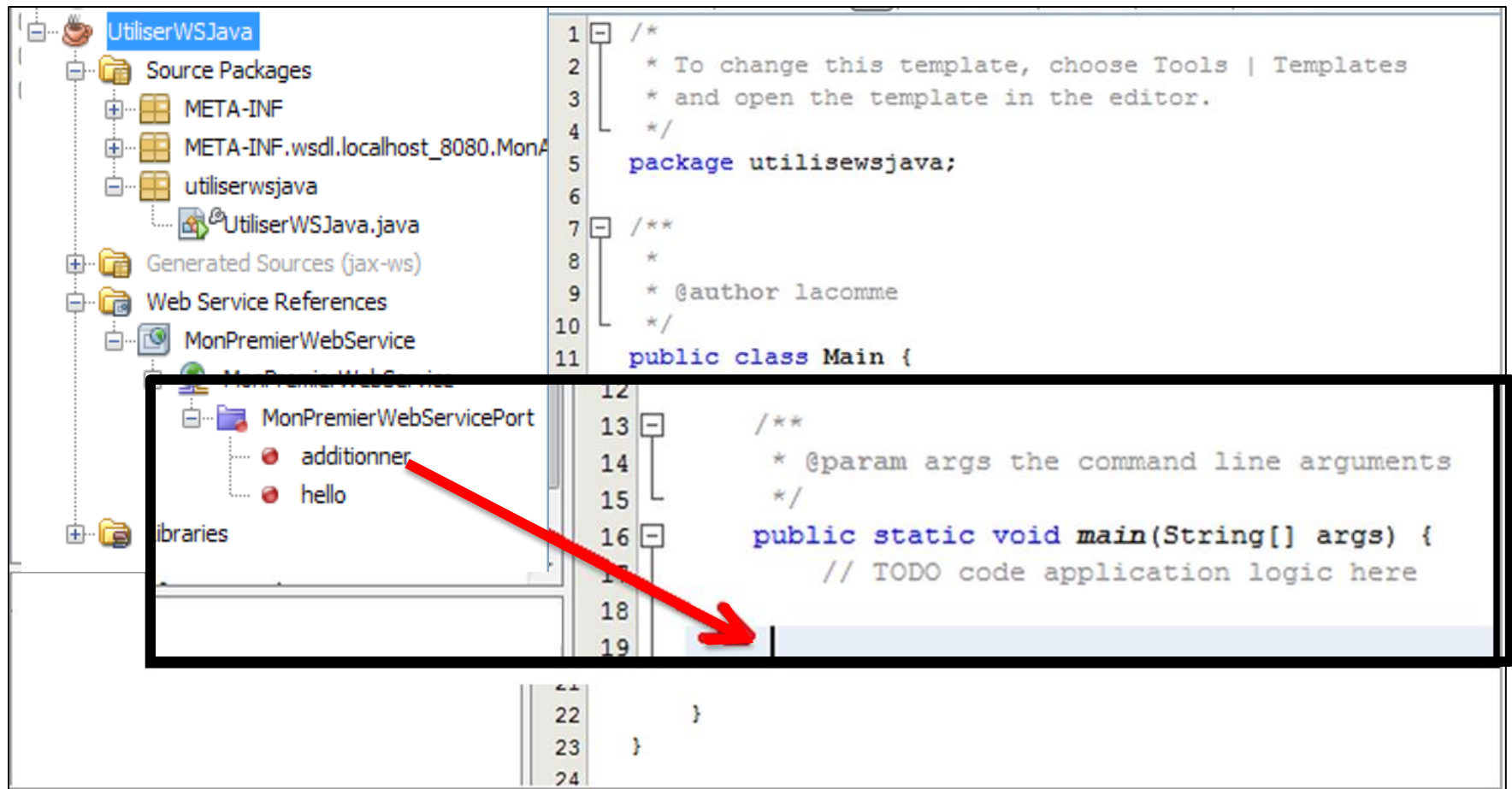


# Ajout d'une Ref. de service



<http://localhost:8080/MonApplicationWeb/MonPremierWebService?wsdl>

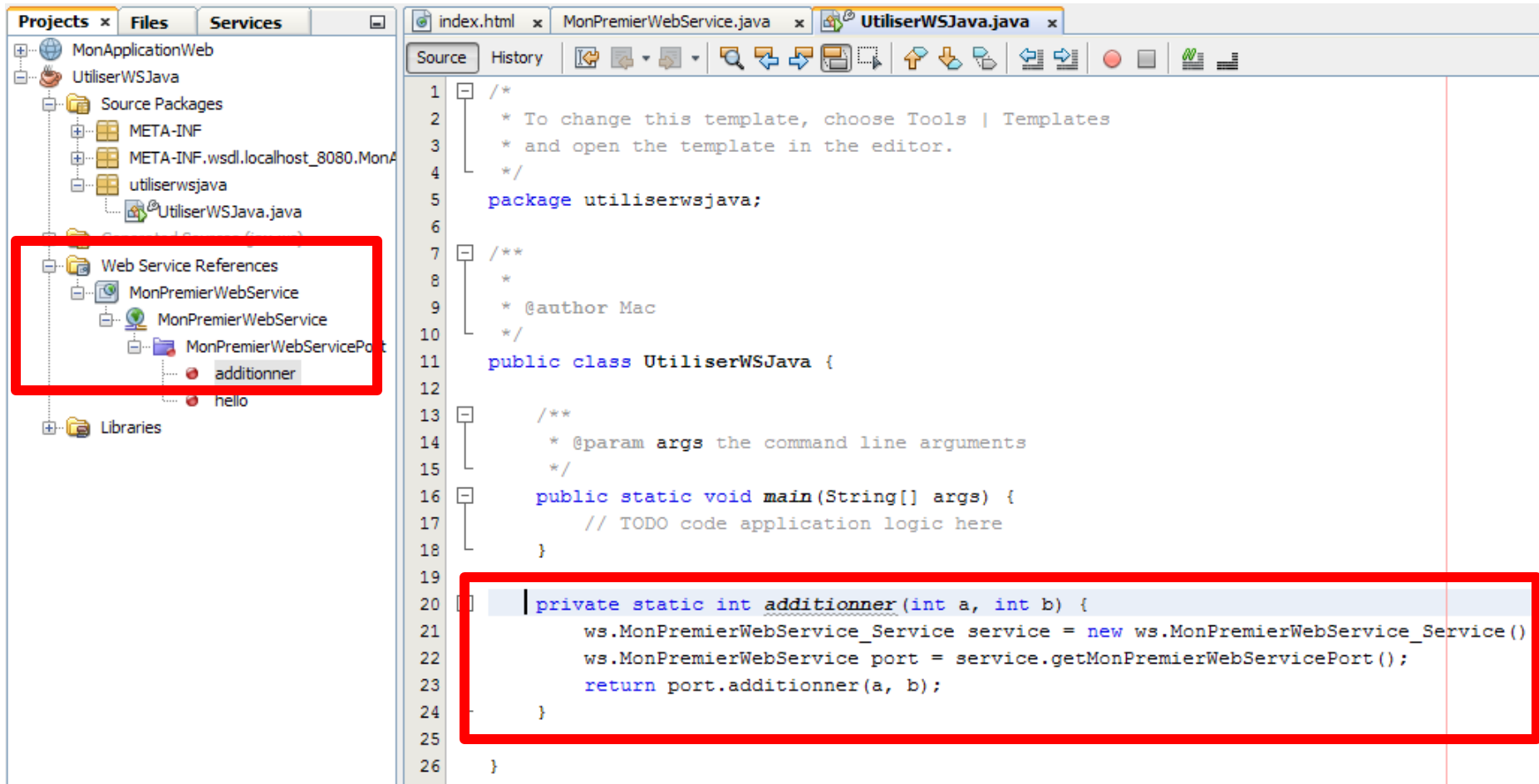




The screenshot shows an IDE with a project named 'UtiliserWSJava'. The project structure includes 'Source Packages' (META-INF, META-INF.wsdl.localhost\_8080.MonA, utiliserswsjava), 'Generated Sources (jax-ws)', 'Web Service References' (MonPremierWebService), and 'Libraries'. The code editor displays the following Java code:

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package utiliserswsjava;
6
7  /**
8  *
9  * @author lacomme
10 */
11 public class Main {
12
13     /**
14     * @param args the command line arguments
15     */
16     public static void main(String[] args) {
17         // TODO code application logic here
18
19
20
21     }
22
23 }
24
```

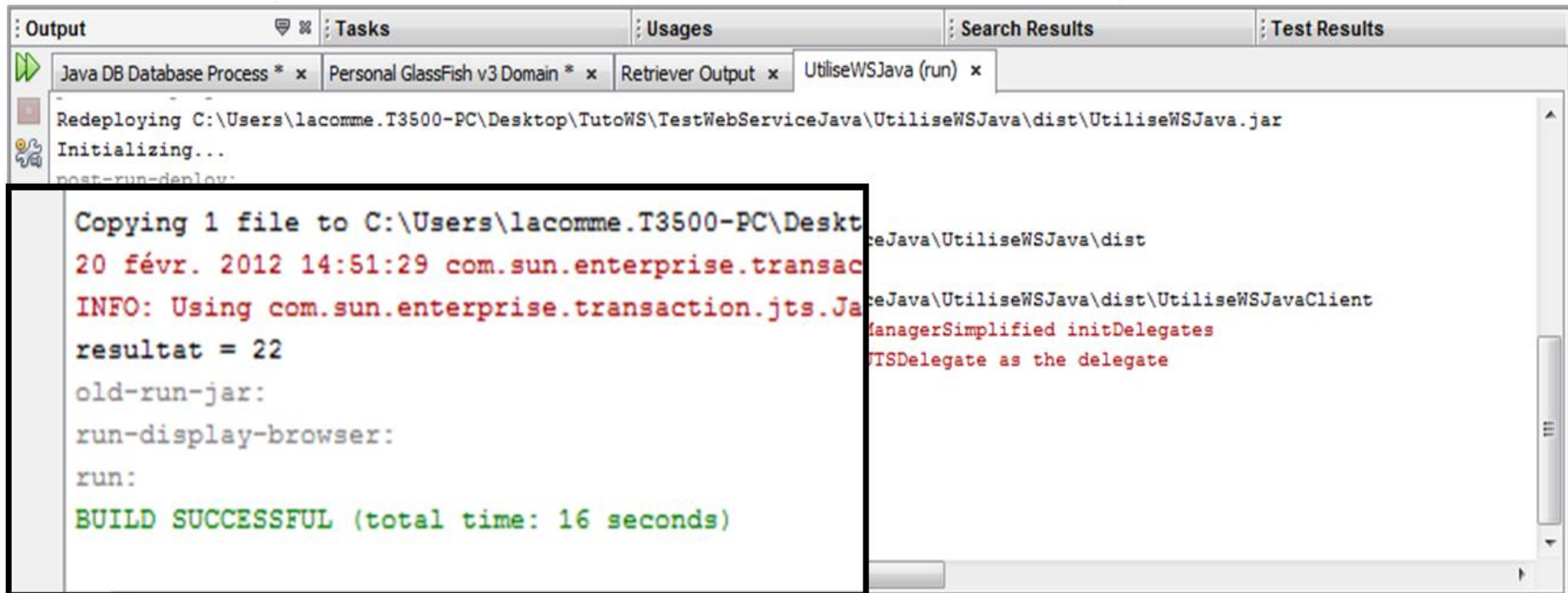
A red arrow points to the end of the main method on line 19.



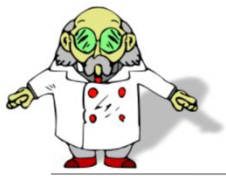
The screenshot displays an IDE interface with the following components:

- Projects Panel (Left):** Shows a project named 'MonApplicationWeb' containing a sub-project 'UtiliserWSJava'. Under 'Source Packages', there are 'META-INF' files and a package 'utilisrwsjava'. A 'Web Service References' folder is highlighted in red, containing 'MonPremierWebService' and 'MonPremierWebServicePort'.
- Source Editor (Right):** Shows the code for 'UtiliserWSJava.java'. The code includes package declarations, a class definition, and a main method. A red box highlights the 'additionner' method call within the main method.

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package utilisrwsjava;
6
7  /**
8   *
9   * @author Mac
10  */
11  public class UtiliserWSJava {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18      }
19
20      private static int additionner(int a, int b) {
21          ws.MonPremierWebService_Service service = new ws.MonPremierWebService_Service()
22          ws.MonPremierWebService port = service.getMonPremierWebServicePort();
23          return port.additionner(a, b);
24      }
25
26  }
```



```
Output Tasks Usages Search Results Test Results
Java DB Database Process * x Personal GlassFish v3 Domain * x Retriever Output x UtiliseWSJava (run) x
Redeploying C:\Users\lacomme.T3500-PC\Desktop\TutoWS\TestWebServiceJava\UtiliseWSJava\dist\UtiliseWSJava.jar
Initializing...
post-run-deploy:
Copying 1 file to C:\Users\lacomme.T3500-PC\Desktop\TutoWS\TestWebServiceJava\UtiliseWSJava\dist
20 févr. 2012 14:51:29 com.sun.enterprise.transaction.JTSDelegate
INFO: Using com.sun.enterprise.transaction.jts.JTSDelegate as the delegate
resultat = 22
old-run-jar:
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 16 seconds)
```



# Éléments à retenir

- Un serveur → une application web
- Déployer → Glassfish
- Rôle de Glassfish
  - héberger
  - administrer
  - simplifier
- Un client → programme Java presque « standard »



# **Partie III : Test d'un web service de géolocalisation d'une adresse ip**



The screenshot shows the IPInfoDB website interface. At the top left is the logo with the text "Free IP Address Geolocation Tools". A dark navigation bar contains links: "IP Location", "IP Location API" (highlighted with a blue arrow), "Block IP by Country", "IP Database", "Fraud Detection", and "Account". Below this is a light blue sub-navigation bar with links: "XML API", "JSON API", "Flash IP info", "Listeners Map", and "Linux tools". The main content area has the heading "IP Address Geolocation XML API". The text below explains that the API returns location data in XML format and provides code samples for PHP, Javascript, Ruby, Python, and ASP. A "Usage" section follows, explaining that for city precision, a query with an IP parameter is used. A text box contains the example URL: `http://api.ipinfodb.com/v3/ip-city/?key=<your_api_key>&ip=74.125.45.100`. The text concludes by stating that for country precision, a different API endpoint is used.

[http://ipinfodb.com/ip\\_location\\_api.php](http://ipinfodb.com/ip_location_api.php)

- Précision au niveau de la ville :

[http://api.ipinfodb.com/v3/ip-city/?key=<api\\_key>&ip=<ip>](http://api.ipinfodb.com/v3/ip-city/?key=<api_key>&ip=<ip>)

- Précision au niveau du pays :

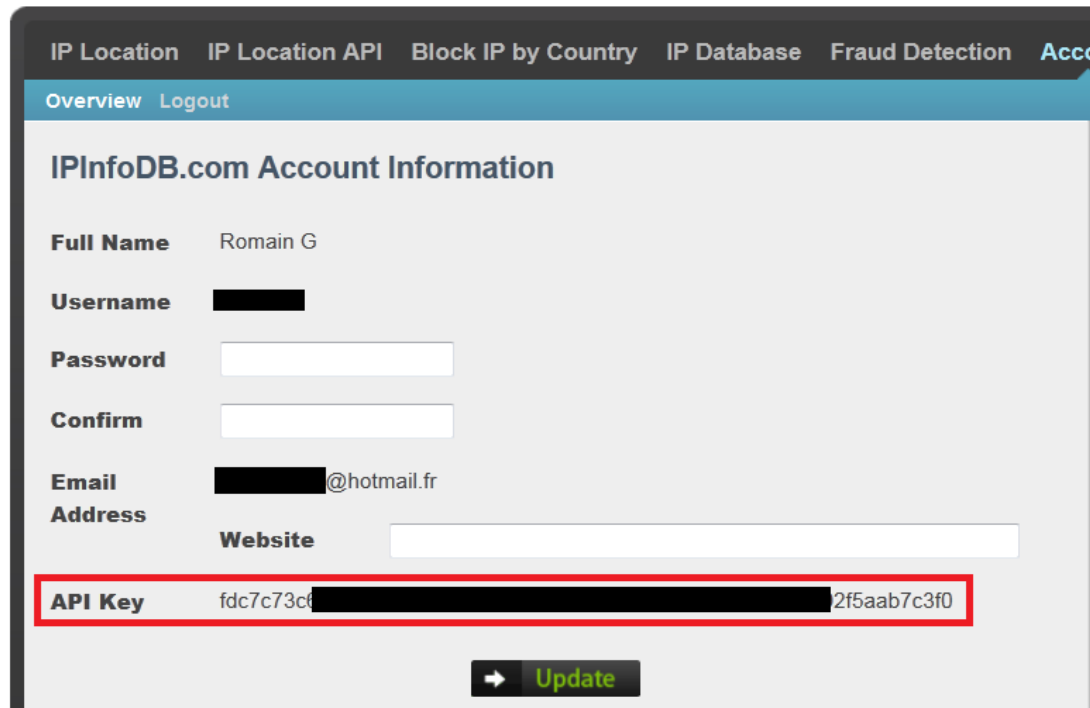
[http://api.ipinfodb.com/v3/ip-country/?key=<api\\_key>&ip=<ip>](http://api.ipinfodb.com/v3/ip-country/?key=<api_key>&ip=<ip>)

## API parameters

Parameter	Required	Default	Value
key	Yes	<empty>	API key provided with your free account.
ip	No	Client IP	IP address
format	No	raw	raw, xml, json
callback	No	<empty>	Required when using json callback.

- But de la clé :
  - Identifier les utilisateurs du WS
  - Contrôler le nombre de requêtes

1. Création d'un compte : <http://ipinfodb.com/register.php>
2. La clé est disponible dans votre compte



The screenshot shows the 'IPInfoDB.com Account Information' page. The page has a dark navigation bar with links: 'IP Location', 'IP Location API', 'Block IP by Country', 'IP Database', 'Fraud Detection', and 'Acco'. Below the navigation bar is a blue header with 'Overview' and 'Logout'. The main content area is titled 'IPInfoDB.com Account Information' and contains several fields: 'Full Name' (Romain G), 'Username' (redacted), 'Password' (input field), 'Confirm' (input field), 'Email Address' (redacted@hotmail.fr), and 'Website' (input field). At the bottom, the 'API Key' is displayed as 'fdc7c73ct[redacted]2f5aab7c3f0' and is highlighted with a red border. Below the API key is an 'Update' button with a right-pointing arrow.

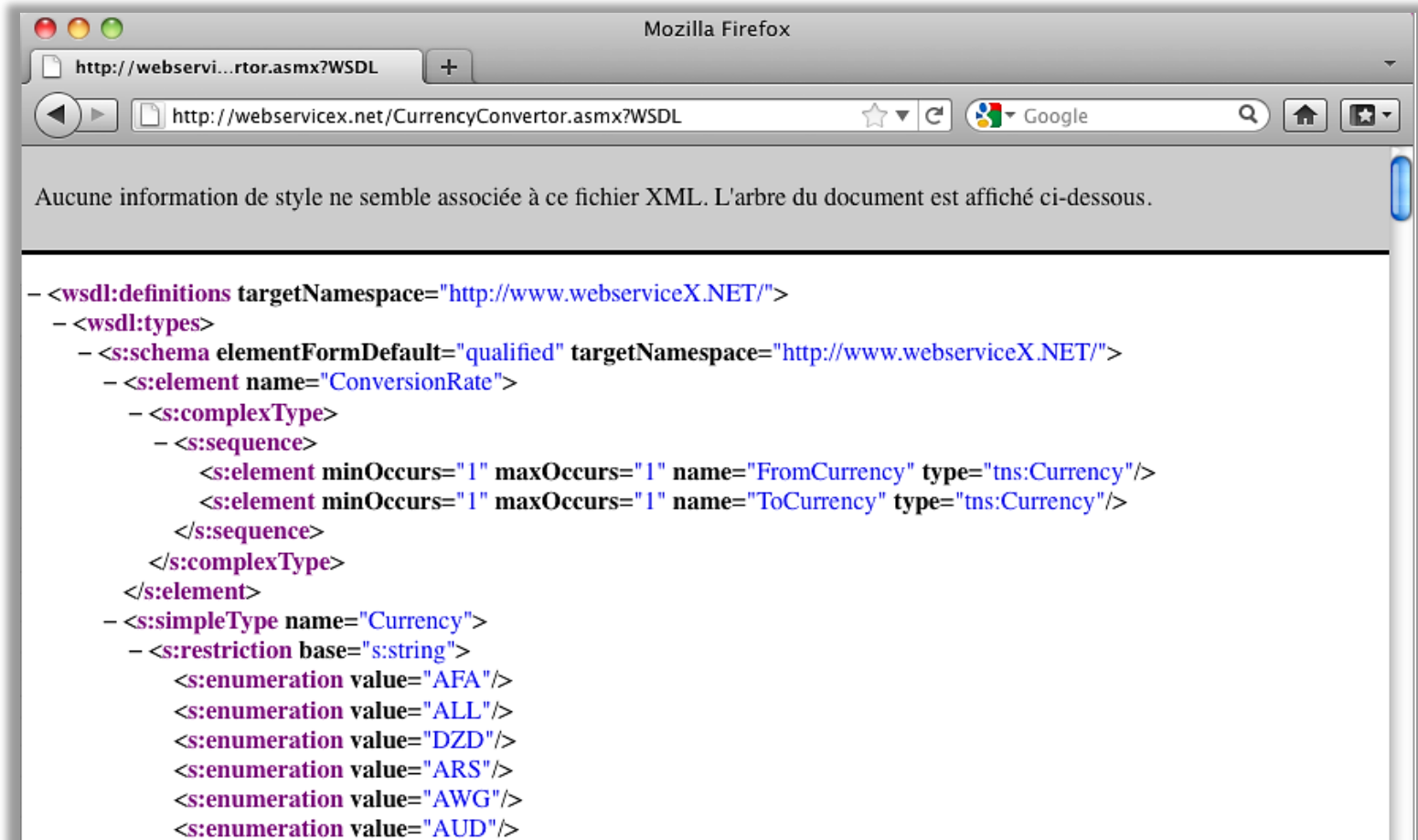
<http://api.ipinfodb.com/v3/ip-city/?key=<cle>&ip=193.55.95.1&format=xml>

Aucune information de style ne semble associée à ce fichier XML.

```
- <Response>
  <statusCode>OK</statusCode>
  <statusMessage/>
  <ipAddress>193.55.95.1</ipAddress>
  <countryCode>FR</countryCode>
  <countryName>FRANCE</countryName>
  <regionName>AUVERGNE</regionName>
  <cityName>CLERMONT-FERRAND</cityName>
  <zipCode>-</zipCode>
  <latitude>45.783</latitude>
  <longitude>3.083</longitude>
  <timeZone>+01:00</timeZone>
</Response>
```

# **Partie IV : Test d'un web service de conversion de monnaie**

<http://www.webservices.net/CurrencyConvertor.asmx?WSDL>



Aucune information de style ne semble associée à ce fichier XML. L'arbre du document est affiché ci-dessous.

```
- <wsdl:definitions targetNamespace="http://www.webserviceX.NET/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
- <s:element name="ConversionRate">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="FromCurrency" type="tns:Currency"/>
  <s:element minOccurs="1" maxOccurs="1" name="ToCurrency" type="tns:Currency"/>
</s:sequence>
</s:complexType>
</s:element>
- <s:simpleType name="Currency">
- <s:restriction base="s:string">
  <s:enumeration value="AFA"/>
  <s:enumeration value="ALL"/>
  <s:enumeration value="DZD"/>
  <s:enumeration value="ARS"/>
  <s:enumeration value="AWG"/>
  <s:enumeration value="AUD"/>
```

<http://www.websvcicex.net/CurrencyConvertor.aspx?op=ConversionRate>

AED-UAE Dirham  
UGX-Ugandan Shilling  
UAH-Ukraine Hryvnia  
UYU-Uruguayan New Peso  
VUV-Vanuatu Vatu  
VEB-Venezuelan Bolivar  
VND-Vietnam Dong  
YER-Yemen Riyal  
YUM-Yugoslav Dinar  
ZMK-Zambian Kwacha  
ZWD-Zimbabwe Dollar

## Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

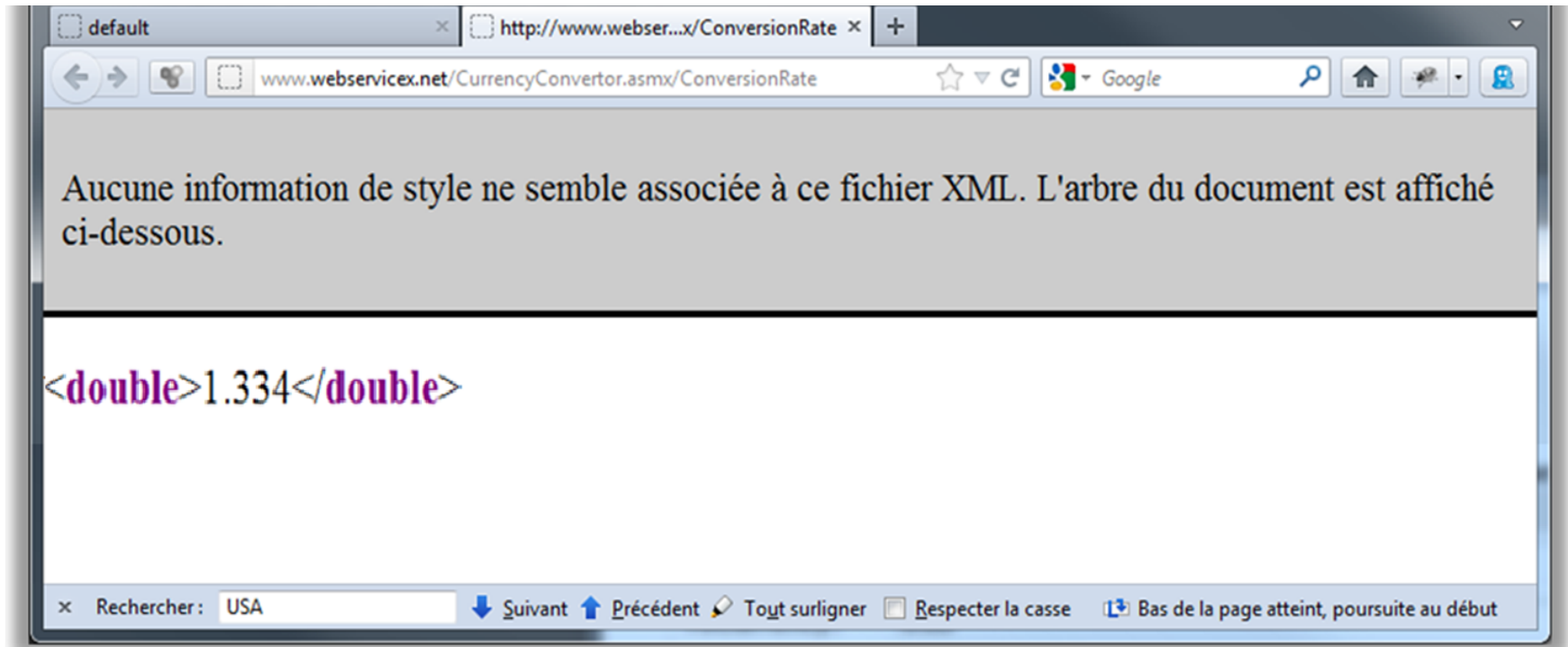
Parameter	Value
FromCurrency:	<input type="text" value="EUR"/>
ToCurrency:	<input type="text" value="USD"/>

## SOAP 1.1

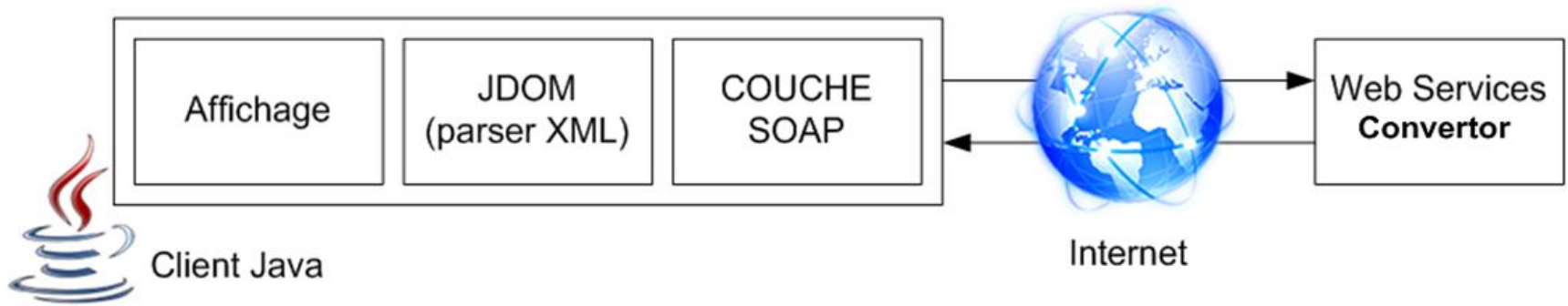
The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

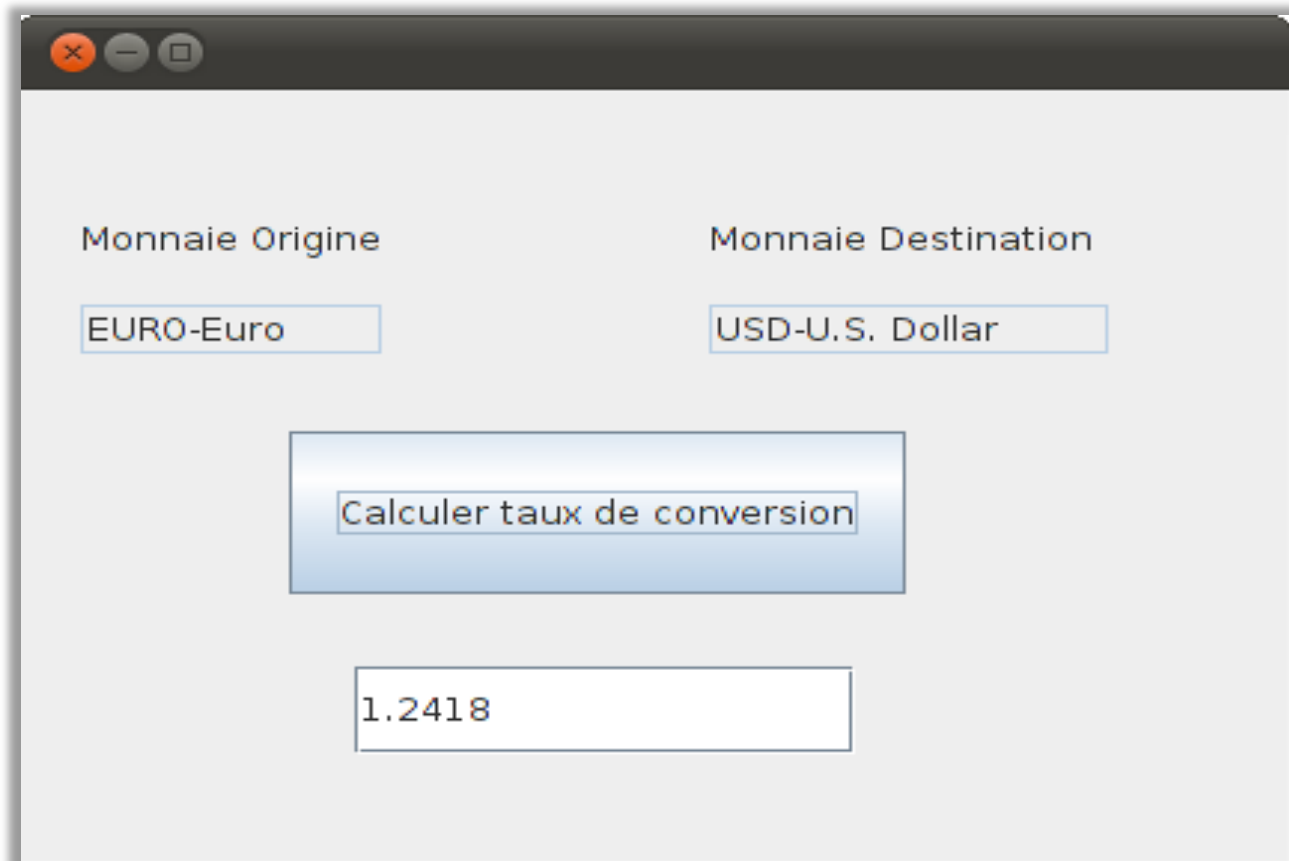
```
POST /CurrencyConvertor.aspx HTTP/1.1
Host: www.websvcicex.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.webserviceX.NET/ConversionRate"
```





# **Partie V : Réalisation d'un client pour la conversion de monnaie**



A screenshot of a Java application window with a dark title bar containing standard window controls (close, minimize, maximize). The window has a light gray background and contains the following elements:

- Two labels: "Monnaie Origine" and "Monnaie Destination".
- Two text input fields: the first contains "EURO-Euro" and the second contains "USD-U.S. Dollar".
- A large, light blue button with a gradient and a drop shadow, labeled "Calculer taux de conversion".
- A text input field below the button containing the value "1.2418".

- Démarrer Netbeans
- Créer un projet

**Name and Location**

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same libraries (see Help for details).

Create Main Class

Set as Main Project

**Name and Location**

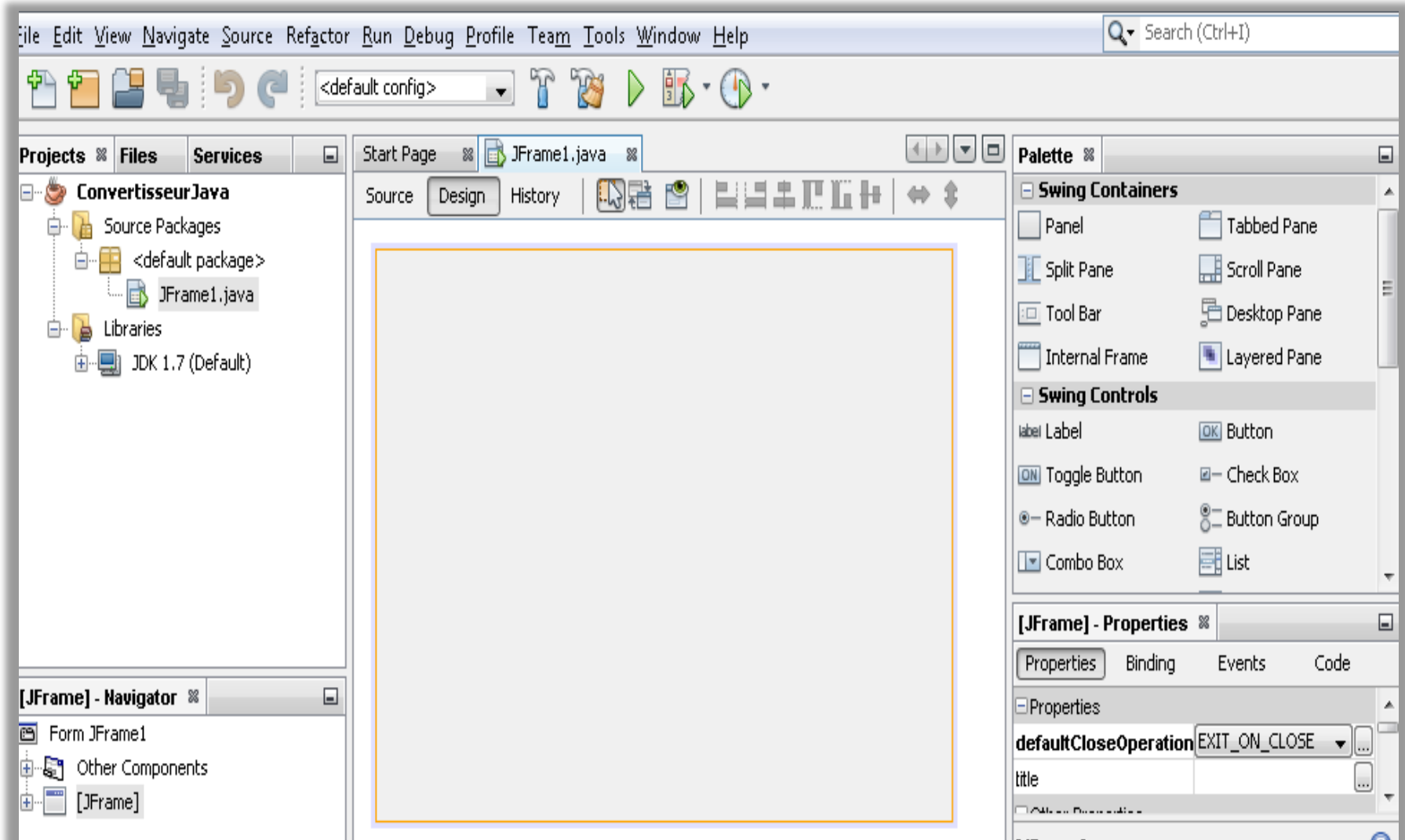
Class Name:

Project:

Location:

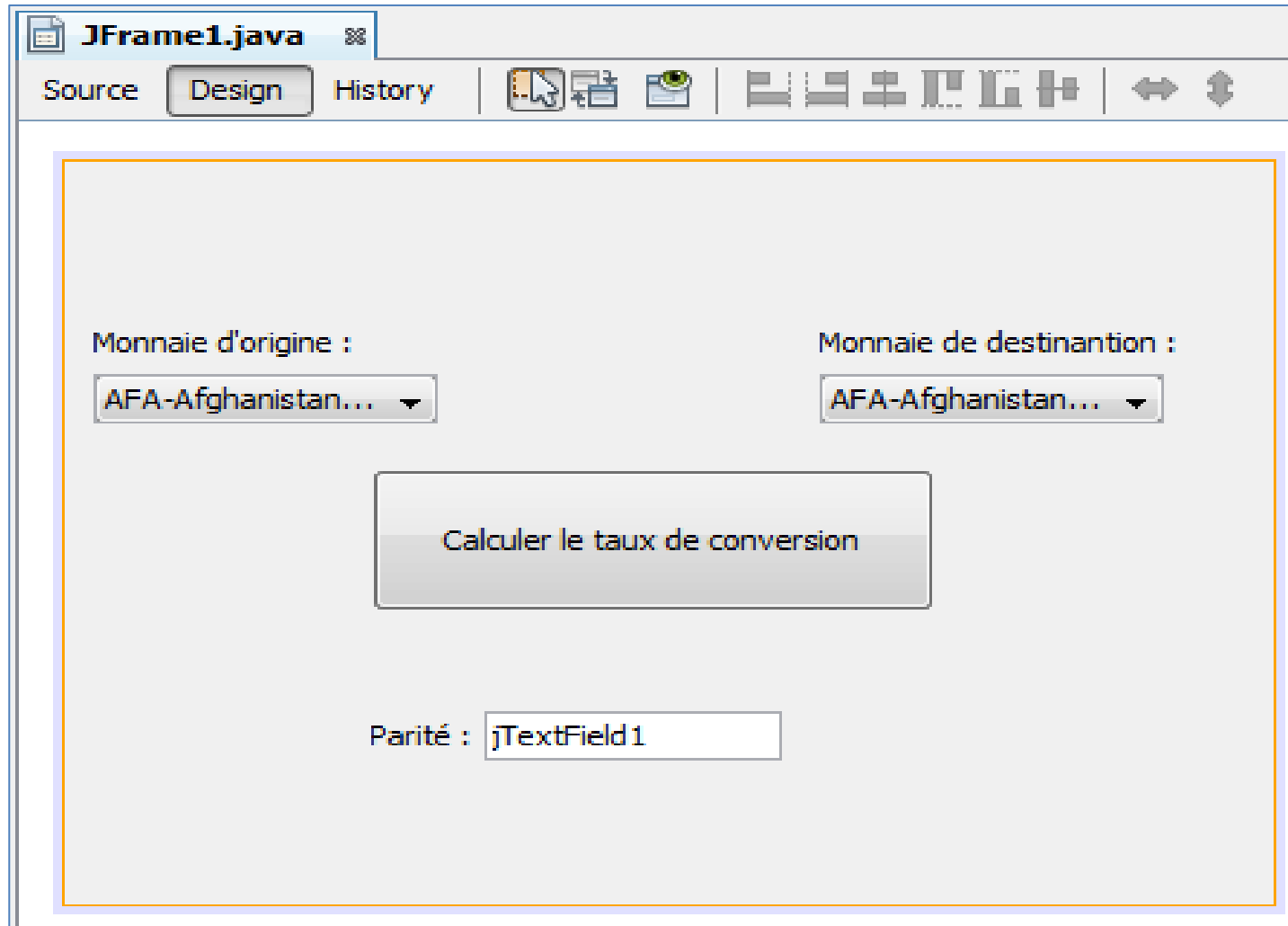
Package:

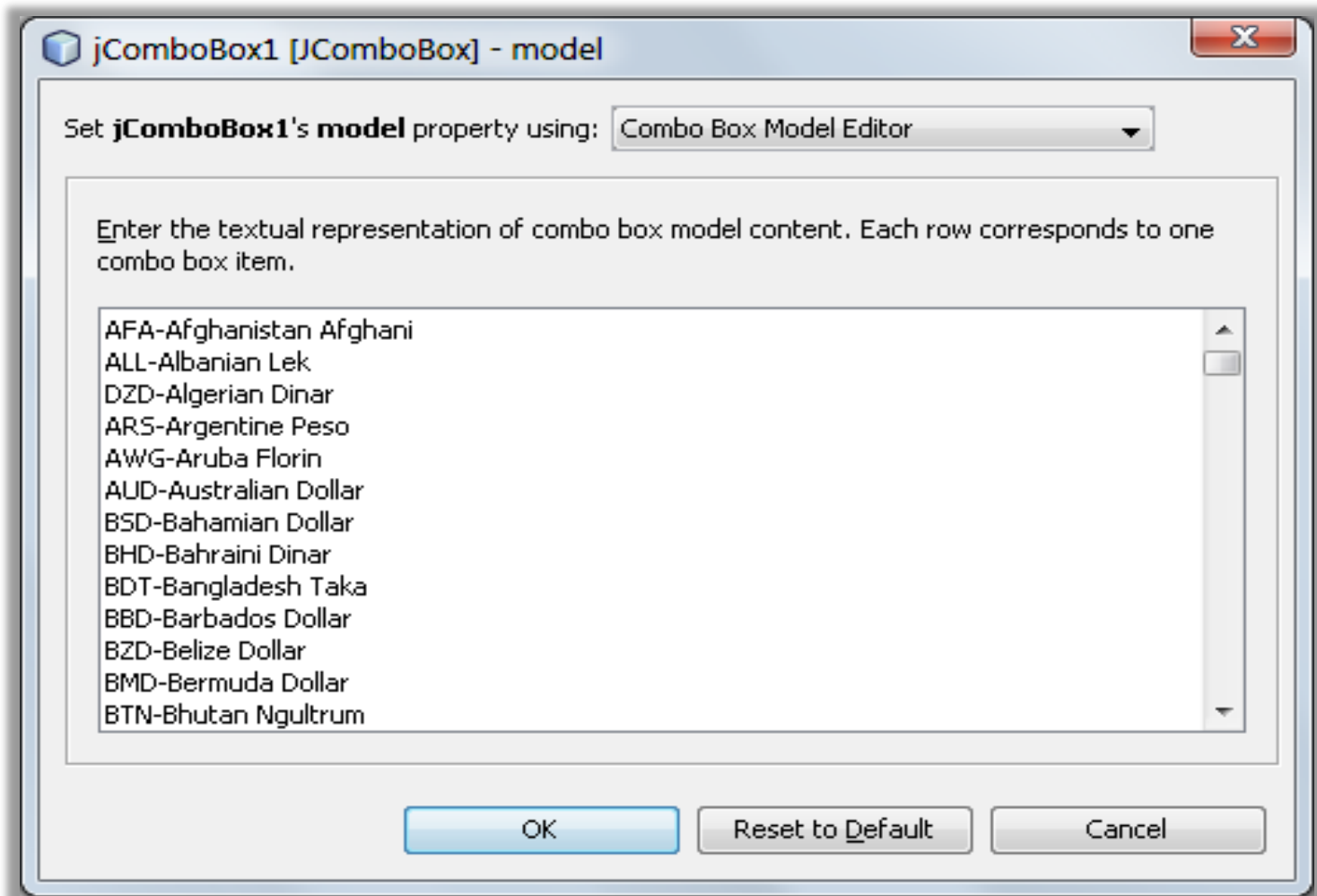
Created File:



The screenshot displays the NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. A search bar is located on the right side of the menu bar. Below the menu bar is a toolbar with various icons for file operations, configuration, and execution. The main workspace is divided into several panes:

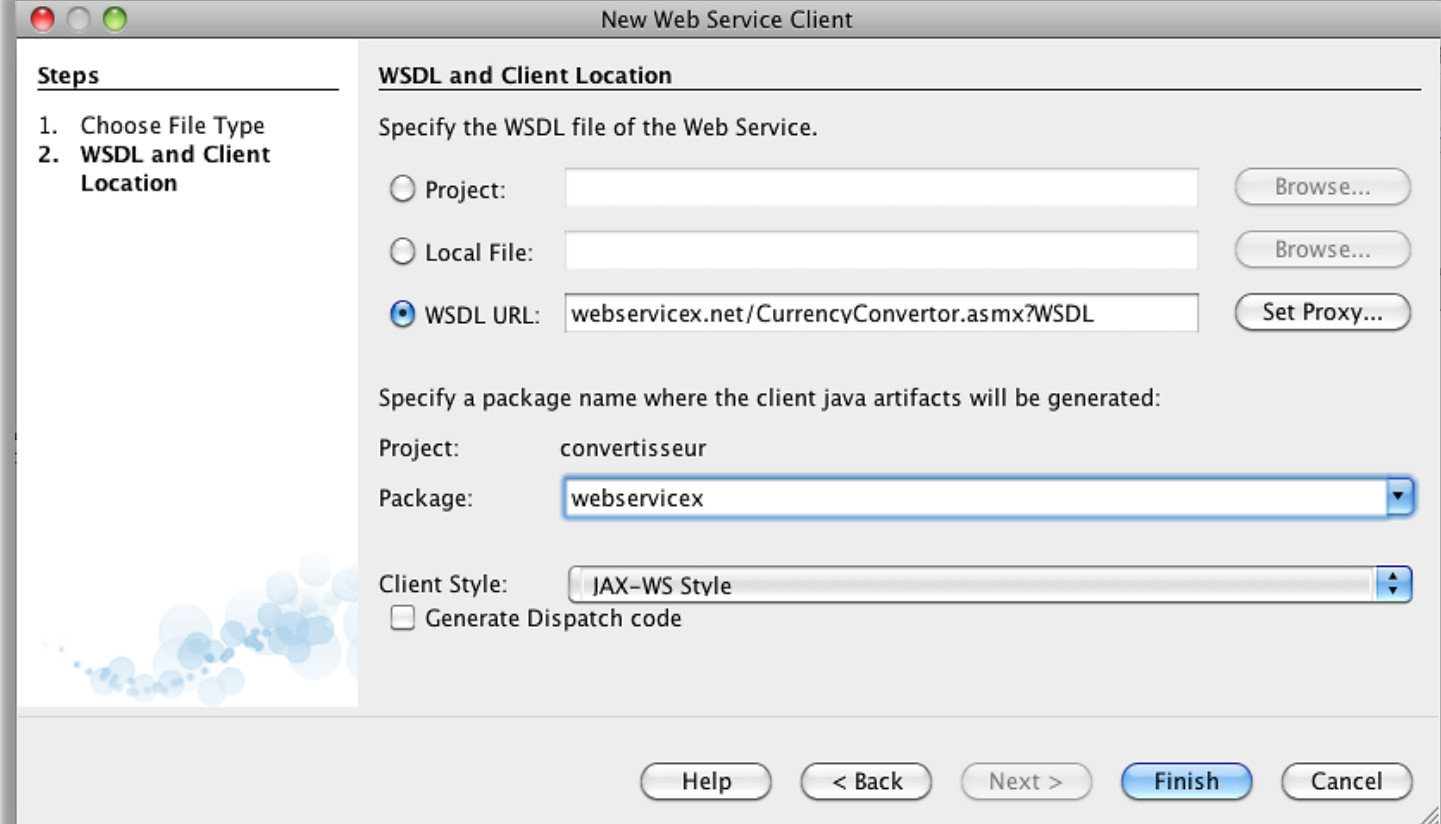
- Projects:** Shows a project named "ConvertisseurJava" with a source package structure including "<default package>" and "JFrame1.java".
- Files:** Shows the current file "JFrame1.java" in the "Design" mode.
- Services:** Shows the "JDK 1.7 (Default)" library.
- Palette:** Contains a list of Swing components under "Swing Containers" (Panel, Split Pane, Tool Bar, Internal Frame, Tabbed Pane, Scroll Pane, Desktop Pane, Layered Pane) and "Swing Controls" (Label, Toggle Button, Radio Button, Combo Box, OK Button, Check Box, Button Group, List).
- [JFrame] - Properties:** Shows the properties for the selected JFrame component, including "defaultCloseOperation" set to "EXIT\_ON\_CLOSE" and "title".





Lien vers le code : <http://www.isima.fr/~lacomme/ateliers/t8a1/?idx=2>





**New Web Service Client**

**Steps**

1. Choose File Type
2. **WSDL and Client Location**

**WSDL and Client Location**

Specify the WSDL file of the Web Service.

Project:

Local File:

WSDL URL:

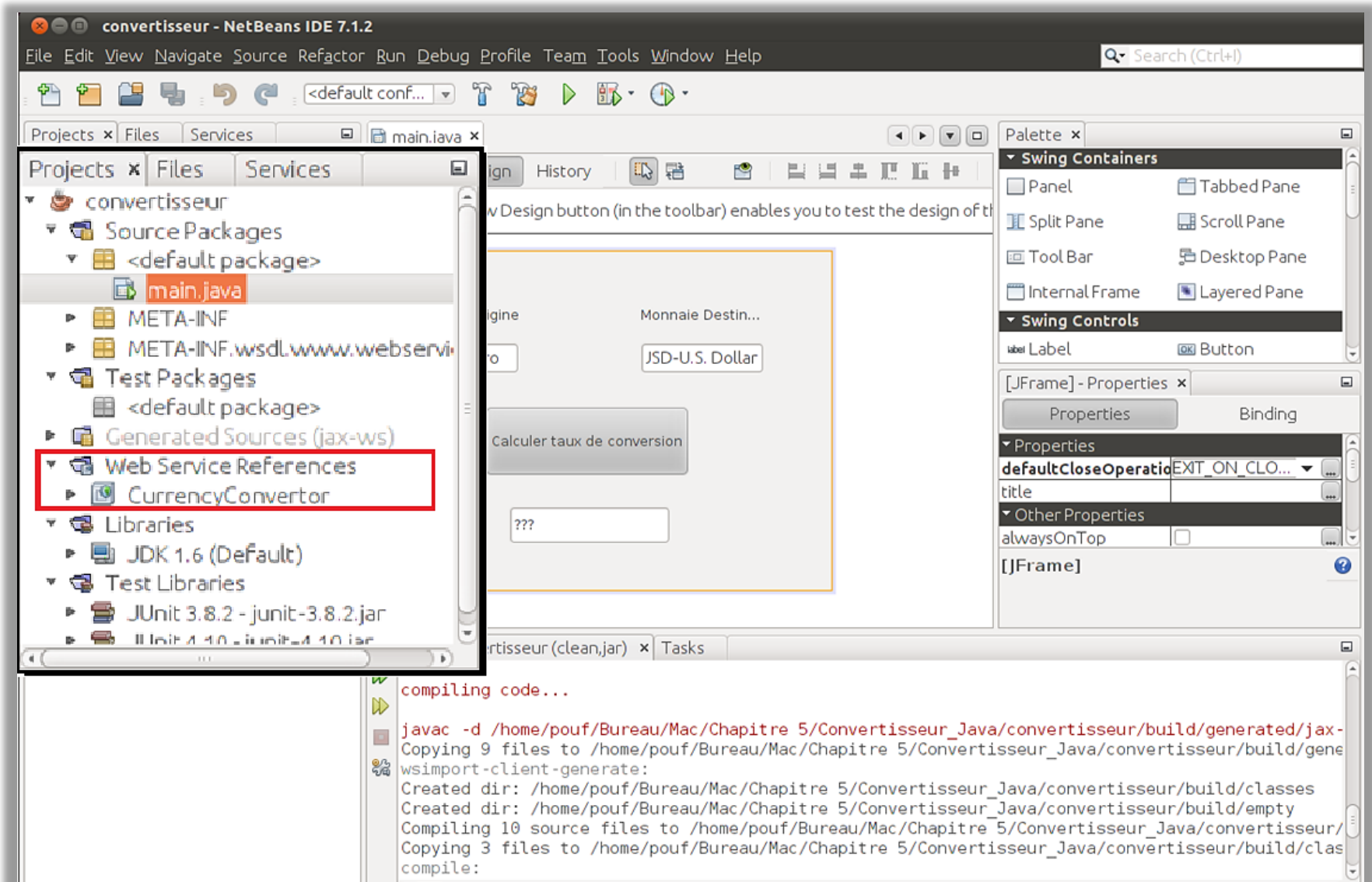
Specify a package name where the client java artifacts will be generated:

Project:

Package:

Client Style:

Generate Dispatch code



convertisseur - NetBeans IDE 7.1.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects x Files Services main.java x

Projects x Files Services

- convertisseur
  - Source Packages
    - <default package>
      - main.java
    - META-INF
    - META-INF.wsdl.www.webservi...
    - Test Packages
      - <default package>
    - Generated Sources (jax-ws)
    - Web Service References**
      - CurrencyConvertor**
    - Libraries
      - JDK 1.6 (Default)
      - Test Libraries
        - JUnit 3.8.2 - junit-3.8.2.jar
        - JUnit 4.10 - junit-4.10.jar

Design button (in the toolbar) enables you to test the design of th...

Monnaie Destin...

JSD-U.S. Dollar

Calculer taux de conversion

???

Swing Containers

- Panel
- Tabbed Pane
- Split Pane
- Scroll Pane
- Tool Bar
- Desktop Pane
- Internal Frame
- Layered Pane

Swing Controls

- Label
- Button

[JFrame] - Properties x

Properties Binding

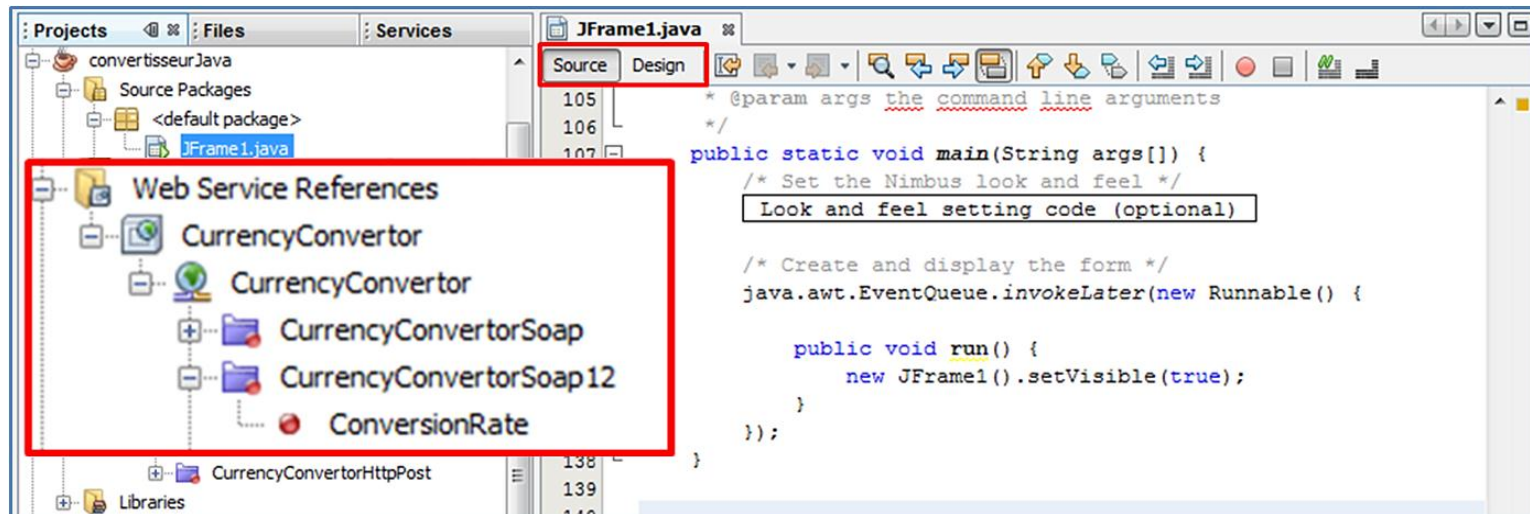
Properties

- defaultCloseOperation: EXT\_ON\_CLO...
- title
- Other Properties
  - alwaysOnTop:

[JFrame]

compiling code...

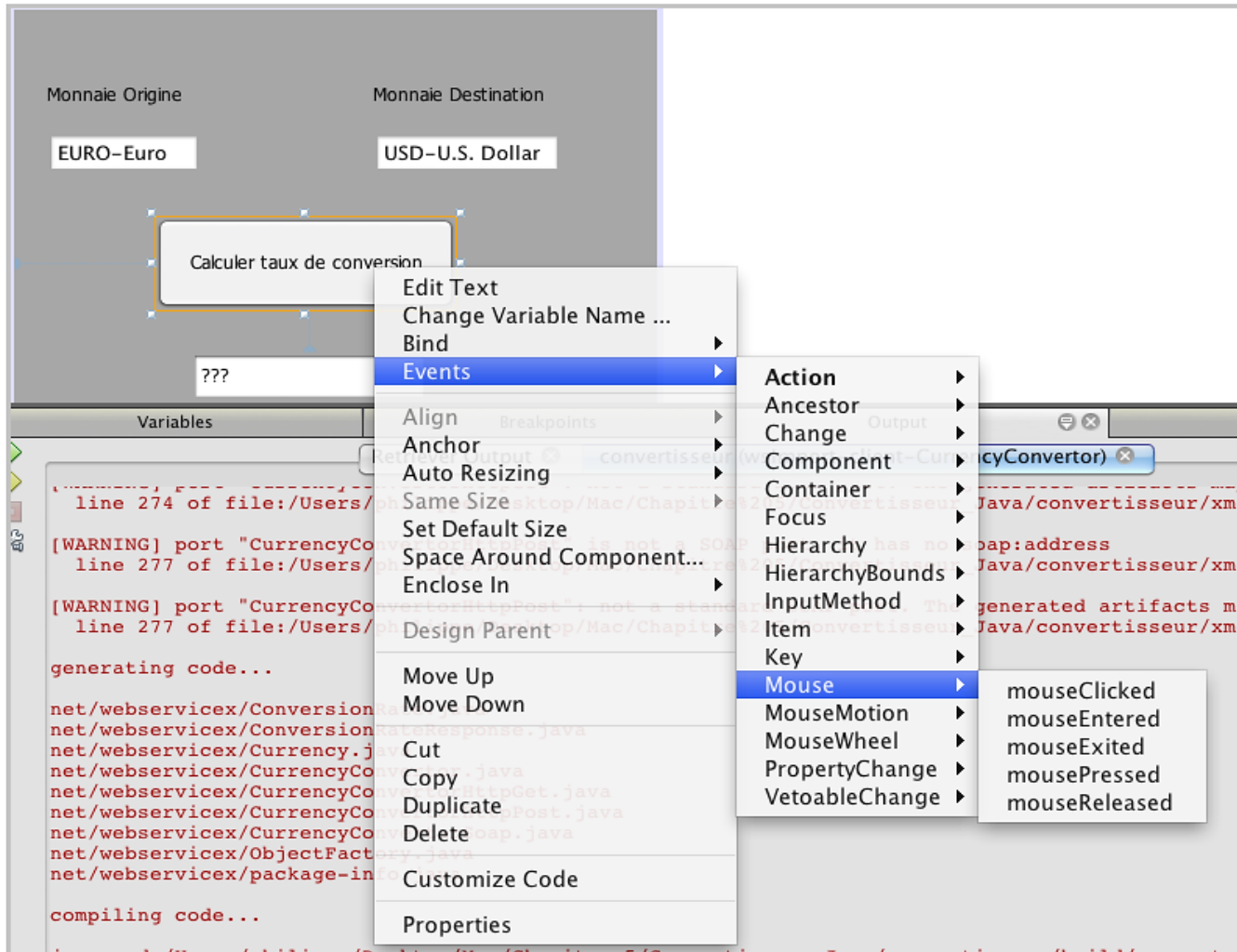
```
javac -d /home/pouf/Bureau/Mac/Chapitre 5/Convertisseur_Java/convertisseur/build/generated/jax-
Copying 9 files to /home/pouf/Bureau/Mac/Chapitre 5/Convertisseur_Java/convertisseur/build/gene
wsimport-client-generate:
Created dir: /home/pouf/Bureau/Mac/Chapitre 5/Convertisseur_Java/convertisseur/build/classes
Created dir: /home/pouf/Bureau/Mac/Chapitre 5/Convertisseur_Java/convertisseur/build/empty
Compiling 10 source files to /home/pouf/Bureau/Mac/Chapitre 5/Convertisseur_Java/convertisseur/
Copying 3 files to /home/pouf/Bureau/Mac/Chapitre 5/Convertisseur_Java/convertisseur/build/clas
compile:
```



```

private static double conversionRate(webservicex.Currency fromCurrency, webservicex.Currency toCurrency)
webservicex.CurrencyConvertor service = new webservicex.CurrencyConvertor();
webservicex.CurrencyConvertorSoap port = service.getCurrencyConvertorSoap12();
return port.conversionRate(fromCurrency, toCurrency);
}
}

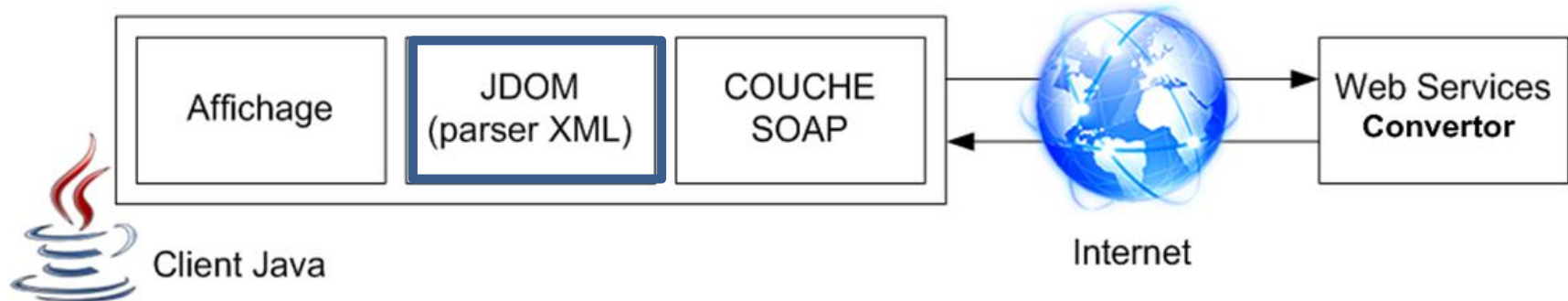
```



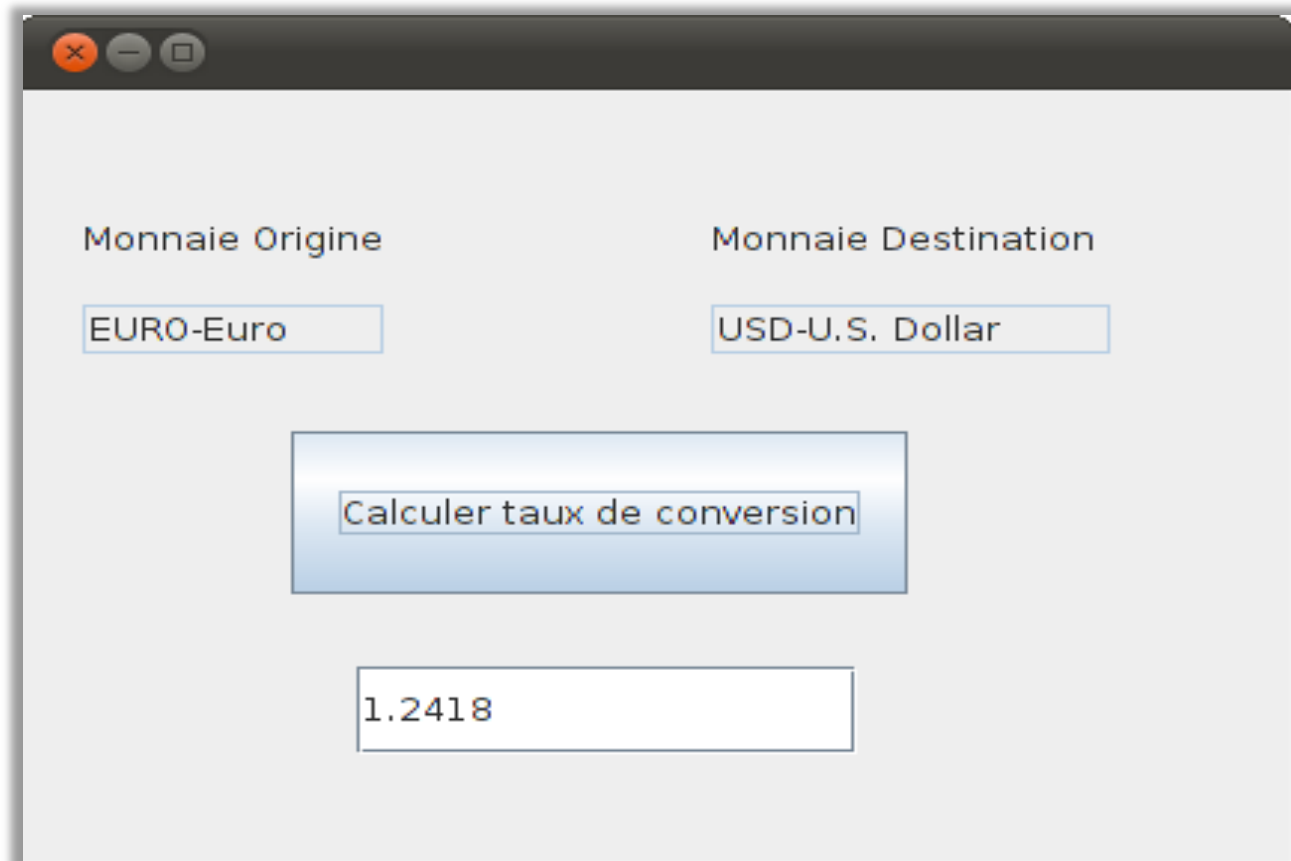
The screenshot shows an IDE window with a GUI design. The GUI has two labels: 'Monnaie Origine' with a dropdown menu containing 'EURO-Euro', and 'Monnaie Destination' with a dropdown menu containing 'USD-U.S. Dollar'. Below these is a button labeled 'Calculer taux de conversion'. A context menu is open over the button, with the following items:

- Edit Text
- Change Variable Name ...
- Bind
- Events** (highlighted)
  - Action
  - Ancestor
  - Change
  - Component
  - Container
  - Focus
  - Hierarchy
  - HierarchyBounds
  - InputMethod
  - Item
  - Key
  - Mouse** (highlighted)
    - MouseClicked
    - MouseEntered
    - MouseExited
    - mousePressed
    - mouseReleased
- Align
- Anchor
- Auto Resizing
- Same Size
- Set Default Size
- Space Around Component...
- Enclose In
- Design Parent
- Move Up
- Move Down
- Cut
- Copy
- Duplicate
- Delete
- Customize Code
- Properties

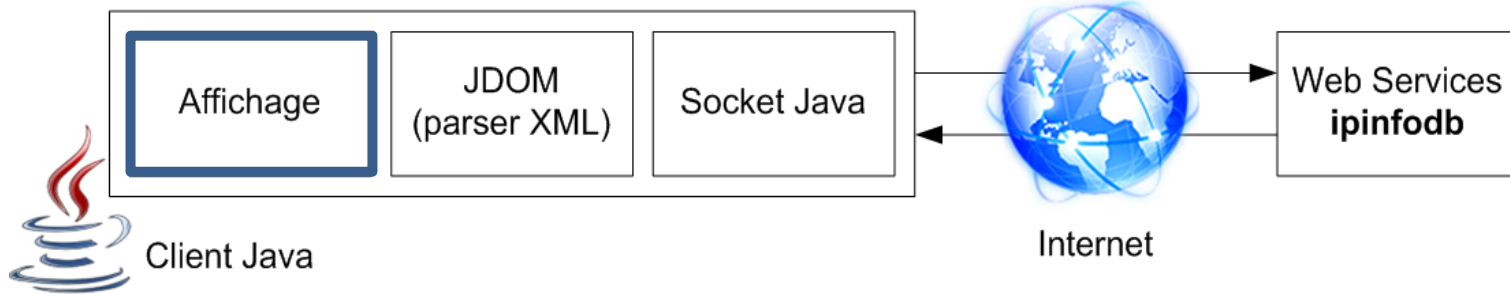
The IDE background shows a code editor with Java code and a console window with warnings and compilation messages.



```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    String src = (String)jComboBox1.getSelectedItemAt();  
    String dest = (String)jComboBox2.getSelectedItemAt();  
  
    double rate = ConversionRate(  
        webservicex.Currency.valueOf(src.substring(0,3)),  
        webservicex.Currency.valueOf(dest.substring(0,3))  
    );  
    jTextField1.setText(" = " + rate);  
}
```

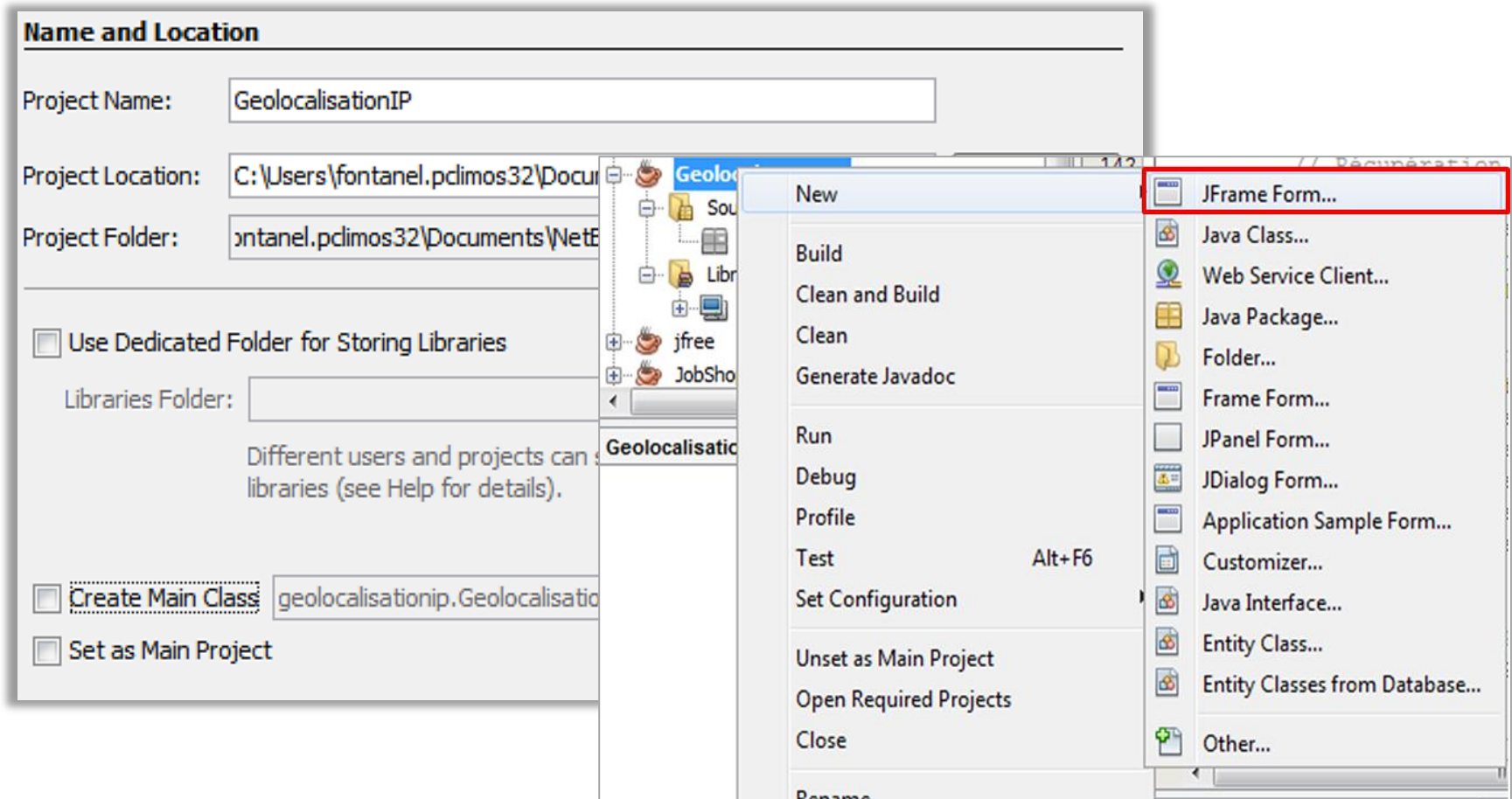


# **Partie VI : Réalisation d'un client pour la gestion d'adresse IP**





- Démarrer Netbeans
- Créer un projet



Adresse ip :

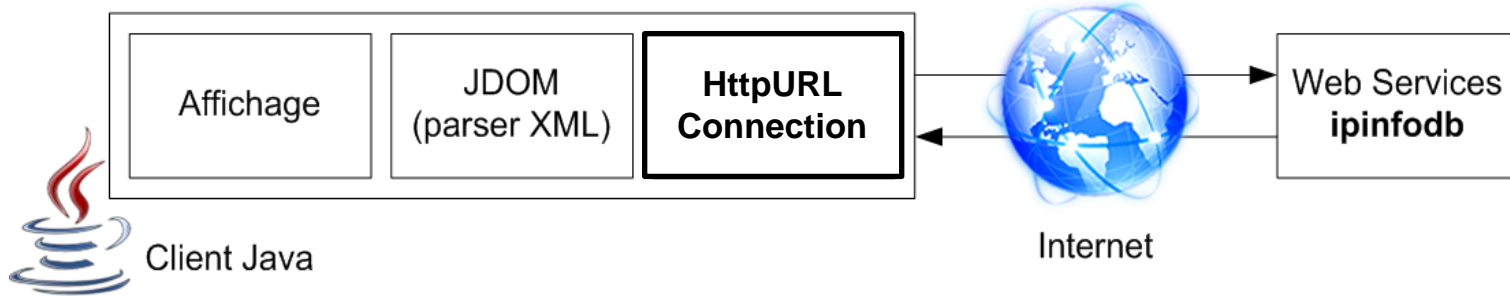
Pays :

Région :

Ville :

Latitude :

Longitude :





# La procédure complète

```
// Methode permettant d'envoyer et de receptionner des donnees provenant
// des web services grace à la bibliotheque "URLConnection "
private void envoiEtReceptionDonneesAvecHttpURL(String adresse)
{
    try {
        // Configuration de l'URL
        URL url = new URL("http://api.ipinfodb.com/v3/ip-city/?key=5xxxxxxxxxx6" + "&ip=" + adresse +
"&format=xml");

        // Configuration de la connexion au web service
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        // Configuration de la methode HTTP utilisee pour joindre le web service
        conn.setRequestMethod("GET");
        // Connexion au web service
        conn.connect();
        // Declaration du flux et du buffer de reception
        InputStream in = conn.getInputStream();
        BufferedReader BufferReception = new BufferedReader(new InputStreamReader(in));
        // Lecture de la réponse
        String ligneReponse = "";
        try {
            do
            {
                ligneReponse = BufferReception.readLine();

                reponseXMLHttpURL.append(ligneReponse + "\n");
                // Récupération de la taille du fichier XML
            }
            while (ligneReponse.length() > 0);
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
        // Fermeture de la connexion
        conn.disconnect();
    } catch (IOException ex) {
        Logger.getLogger(InterfaceGraphique.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

## *Binaries*

We provide several means of accessing the JDOM™ code for our users and developers. A binary download which includes a JDOM *jar* is sufficient for using JDOM. Binary releases come with source, but you're not required to build the code yourself.

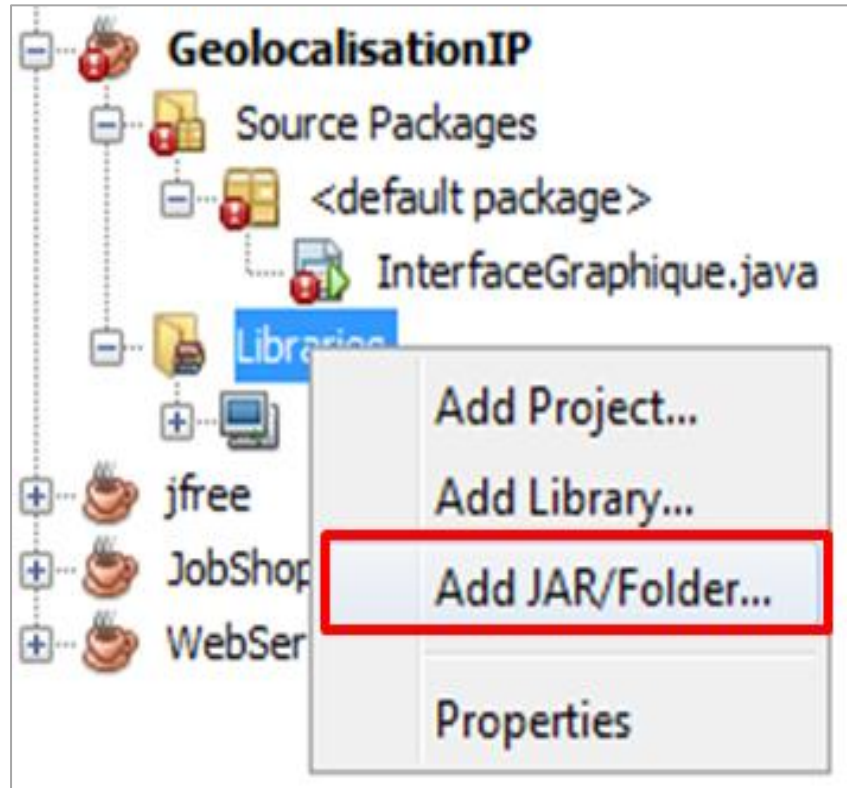
### Release Builds

These are the stable, publicly released JDOM builds. Rigorous testing goes into a release build, ensuring you can run it in your production applications without fear of major bugs or trouble.

- [JDOM 2.0.2](#) is the latest stable build.
- [JDOM 1.1.3](#) is the latest stable and supported version of JDOM from the 1.x branch. Use JDOM 1.1.3 if you require Java 1.2 through 1.4
- The [JDOM Archive](#) contains all releases (old and new).

Lien fichier : <http://www.isima.fr/~lacomme/ateliers/t8a1/jdom-2.0.2.jar>

# Ajouter le parser au projet



```
import org.jdom2.Document;
import org.jdom2.Element;
import org.jdom2.input.SAXBuilder;
```

```
// Methode permettant le traitement du fichier XML
// passe en paramètre
private void TraitementFichierXML(String fichierXML)
{
    |
}
```

```
private void TraitementFichierXML(String fichierXML)
{
    // Declaration des variables pour traiter le fichier XML
    SAXBuilder sxb = new SAXBuilder();
    Document document = new Document();
    try
    {
        // Traitement du fichier XML
        StringReader strReader = new StringReader(fichierXML);
        document = sxb.build(strReader);
    } catch (Exception e)
        {e.printStackTrace(); }

    // Declaration de la racine du document XML
    Element racine = document.getRootElement();

    // Recupération des resultats dans le fichier XML
    Element CountryName = racine.getChild("countryName");
    jTextField2.setText(CountryName.getText());

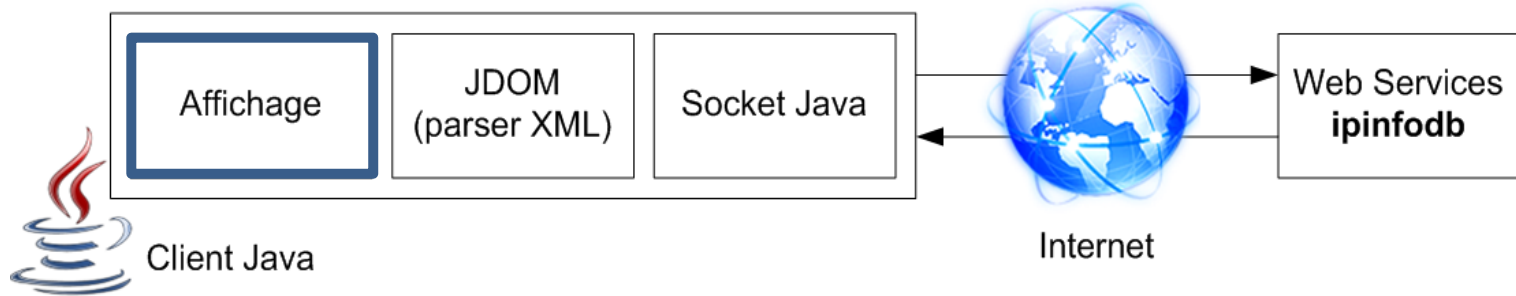
    Element RegionName = racine.getChild("regionName");
    jTextField3.setText(RegionName.getText());

    Element City = racine.getChild("cityName");
    jTextField4.setText(City.getText());

    Element Latitude = racine.getChild("latitude");
    jTextField5.setText(Latitude.getText());

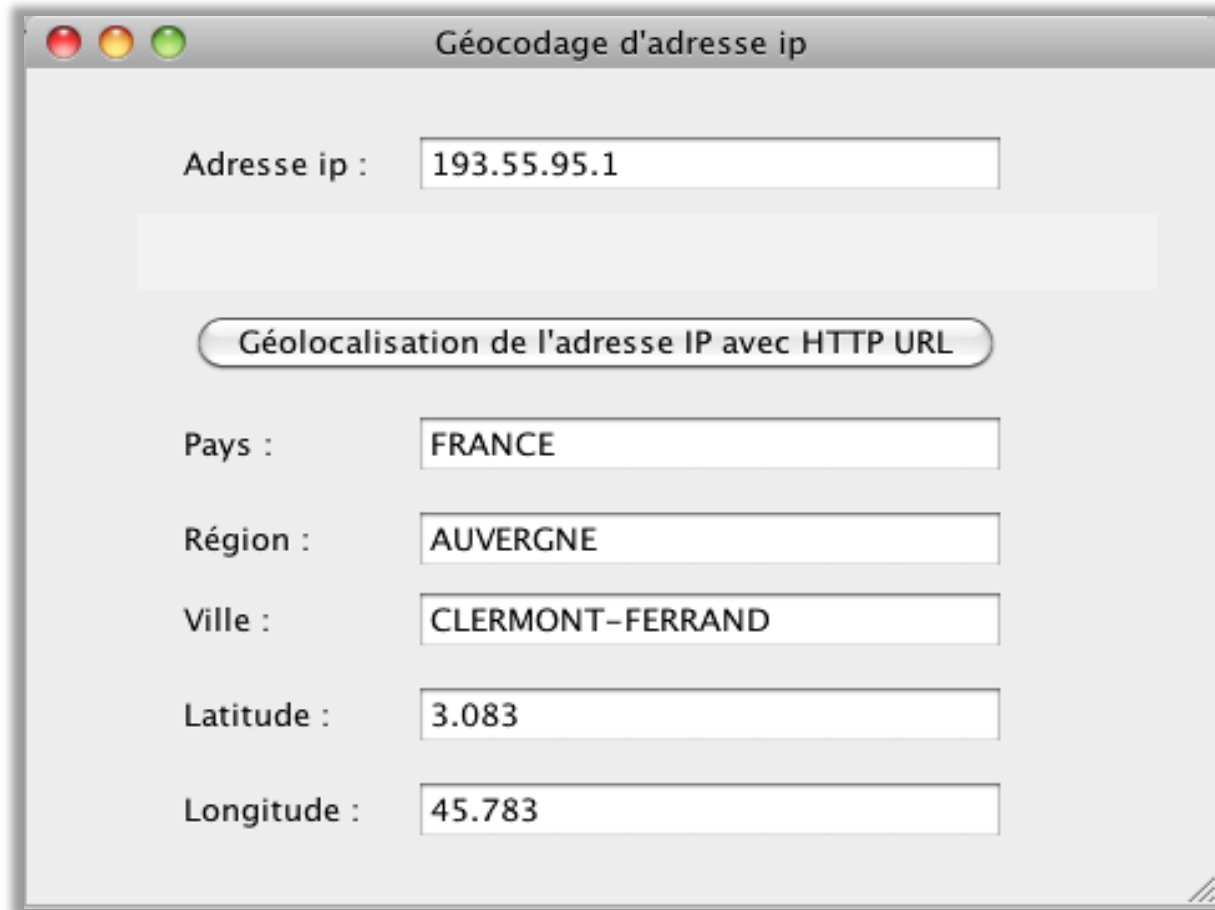
    Element Longitude = racine.getChild("longitude");
    jTextField6.setText(Longitude.getText());
}
```





```
reponseXMLHttpURL = new StringBuilder();  
  
// Envoi et reception des donnees avec les sockets  
envoieEtReceptionDonneesAvecHttpURL(jTextField1.getText());  
  
// Creation du string contenant le fichier XML  
String reponseXMLHttp = reponseXMLHttpURL.toString();  
reponseXMLHttp = reponseXMLHttp.replace("null", "");  
System.out.println(reponseXMLHttp);  
  
// Chercher les informations dans le fichier XML avec JDOM  
TraitementFichierXML(reponseXMLHttp);
```

**Lien vers le code : <http://www.isima.fr/~lacomme/ateliers/t8a1/?idx=7>**

A screenshot of a Java application window titled "Géocodage d'adresse ip". The window has a standard macOS-style title bar with red, yellow, and green window control buttons. The main content area is light gray and contains several text input fields and a button. At the top, the label "Adresse ip :" is followed by a text box containing "193.55.95.1". Below this is a large, empty rectangular area. A rounded button with the text "Géolocalisation de l'adresse IP avec HTTP URL" is centered below the empty area. Below the button are five more text input fields, each with a label to its left: "Pays :" with "FRANCE", "Région :" with "AUVERGNE", "Ville :" with "CLERMONT-FERRAND", "Latitude :" with "3.083", and "Longitude :" with "45.783". A small diagonal icon is visible in the bottom right corner of the window's content area.

Géocodage d'adresse ip

Adresse ip : 193.55.95.1

Géolocalisation de l'adresse IP avec HTTP URL

Pays : FRANCE

Région : AUVERGNE

Ville : CLERMONT-FERRAND

Latitude : 3.083

Longitude : 45.783

# Conclusion



# Des technologies récentes

- **JEE a envahi les entreprises**

Offres d'emplois APEC

Mots clés	2011	2013
Système d'Information	4579	4880
Bases de données	2436	5684
Java	1737	1963
Oracle	1498	1564
SAP	1225	1503

- **Domaine de recherche à part entière**

- Thèses, Revues, Conférences

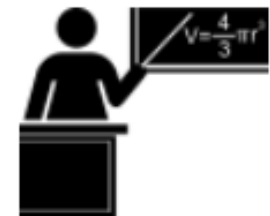
- **Besoins en formation, consulting...**

- Contactez nous !

- [placomme@isima.fr](mailto:placomme@isima.fr), [ren@isima.fr](mailto:ren@isima.fr),

- [Romain.guidoux@clermont.inra.fr](mailto:Romain.guidoux@clermont.inra.fr),

- [J.fontanel@qualiac.com](mailto:J.fontanel@qualiac.com)



- **Ellipses**
  - 15 exemplaires gratuits
- **Nos organismes/employeurs**
  - Ressources
  - Disponibilité
  - ...

