

YAPS Pet Store sous NetBeans

par Saïd Eloudhiri

Date de publication : 08-Dec-2008

Dernière mise à jour : 23-May-2009

Cet article va présenter le portage de l'application "Yaps Pet Store", présentée par Antonio Goncalves dans son livre "Java EE 5", sous NetBeans.

| | |
|--|----|
| I - Introduction..... | 5 |
| II - Conventions..... | 7 |
| II-A - Langue..... | 7 |
| II-B - Mots clés..... | 7 |
| II-C - Répertoires..... | 7 |
| II-D - Commandes..... | 7 |
| II-E - Messages..... | 7 |
| III - Installation..... | 8 |
| III-A - Prérequis..... | 8 |
| III-B - NetBeans..... | 8 |
| III-B-1 - Récupérer NetBeans..... | 8 |
| III-B-2 - Installer NetBeans..... | 10 |
| III-B-2-a - Page d'accueil..... | 10 |
| III-B-2-b - License Agreement..... | 13 |
| III-B-2-c - Répertoires d'installation..... | 13 |
| III-B-2-d - Propriétés de GlassFish v2..... | 14 |
| III-B-2-e - GlassFish v3..... | 15 |
| III-B-2-f - Apache Tomcat..... | 16 |
| III-B-2-g - Résumé..... | 17 |
| III-B-2-h - Finish..... | 18 |
| III-B-3 - Démarrer NetBeans..... | 19 |
| III-B-4 - Localisation..... | 20 |
| LLIII-C - Variables d'environnement..... | 21 |
| LLIII-D - Récupérer l'application Yaps..... | 22 |
| LLIII-E - Récupérer l'application YapsON..... | 23 |
| IV - Configurer YapsON..... | 24 |
| LLIV-A - Variables d'environnement..... | 24 |
| LLIV-B - Scripts Ant..... | 24 |
| V - Configurer GlassFish..... | 25 |
| LLV-A - Positionner les variables d'environnement..... | 25 |
| LLV-B - Créer le domaine « petstore »..... | 26 |
| LLV-C - Démarrer le domaine « petstore »..... | 28 |
| LLV-D - Localisation de GlassFish..... | 30 |
| LLV-E - Créer les ressources Database..... | 30 |
| LLV-E-1 - Pool de connexions JDBC..... | 31 |
| LLV-E-2 - Démarrer le serveur Derby..... | 33 |
| LLV-E-3 - Créer la base de données..... | 33 |
| LLV-E-4 - Créer la source de données..... | 34 |
| LLV-F - Créer les ressources JMS..... | 35 |
| LLV-F-1 - Créer la fabrique de connexion..... | 35 |
| LLV-F-3 - Vérification..... | 36 |
| LLV-G - Créer les loggers..... | 37 |
| LLV-H - Fin de la configuration..... | 38 |
| LLV-G-1 - Stopper le serveur Derby..... | 38 |
| LLV-G-2 - Stopper le domaine « petstore »..... | 38 |
| VI - Configurer NetBeans..... | 39 |
| LLVI-A - Serveur « Petstore »..... | 39 |
| LLVI-A-1 - Ajout du serveur « Petstore »..... | 39 |
| LLVI-A-1-a - Type de serveur..... | 39 |
| LLVI-A-1-b - Sélection du domaine..... | 40 |
| LLVI-A-1-c - Profil de l'administrateur..... | 41 |
| LLVI-A-2 - Liste des serveurs..... | 42 |
| LLVI-A-3 - Affichage des traces..... | 43 |
| LLVI-A-4 - Démarrer le serveur Petstore..... | 44 |
| LLVI-A-5 - Explorer les ressources..... | 47 |
| LLVI-A-6 - Accès à la console d'administration..... | 49 |
| LLVI-B - Changer de browser..... | 49 |
| LLVI-C - Base de données..... | 51 |

| | |
|--|-----|
| LLVI-C-1 - Arrêt et démarrage du serveur de base de données..... | 52 |
| LLVI-C-2 - Vérifier les propriétés..... | 53 |
| LLVI-C-3 - Connexion vers la base de données..... | 56 |
| LLVI-D - Conclusion..... | 56 |
| VII - Importer le projet BarkBank..... | 57 |
| LLVII-A - Créer un nouveau projet..... | 57 |
| LLVII-A-1 - Fermeture des projets..... | 57 |
| LLVII-A-2 - Type de projet..... | 58 |
| LLVII-A-3 - Nom et répertoires..... | 58 |
| LLVII-A-4 - Configuration du serveur..... | 59 |
| LLVII-A-5 - Les frameworks..... | 60 |
| LLVII-A-6 - Génération du projet..... | 61 |
| LLVII-B - Transférer le projet sous NetBeans..... | 62 |
| LLVII-B-1 - Positionner les variables d'environnement..... | 62 |
| LLVII-B-2 - Transférer le projet..... | 62 |
| LLVII-B-3 - Créer le groupe de projet..... | 64 |
| LLVII-C - Compiler le projet..... | 66 |
| LLVII-4 - Déployer le projet..... | 68 |
| LLVII-E - Tester le projet..... | 70 |
| LLVII-E-1 - Méthode 1 : Web Service Tester de GlassFish..... | 71 |
| LLVII-E-2 - Méthode 2 : Web Service Tester de NetBeans..... | 71 |
| LLVII-F - Conclusion..... | 79 |
| VIII - Importer le projet PetEx..... | 80 |
| LLVIII-A - Créer un nouveau projet..... | 80 |
| LLVIII-A-1 - Fermeture des projets..... | 80 |
| LLVIII-A-2 - Type de projet..... | 81 |
| LLVIII-A-3 - Nom et répertoires..... | 81 |
| LLVIII-A-4 - Configuration du serveur..... | 82 |
| LLVIII-A-5 - Les frameworks..... | 83 |
| LLVIII-A-6 - Génération du projet..... | 84 |
| LLVIII-B - Transférer le projet sous NetBeans..... | 85 |
| LLVIII-B-1 - Positionner les variables d'environnement..... | 85 |
| LLVIII-B-2 - Transférer le projet..... | 85 |
| LLVIII-B-3 - Créer le groupe de projet..... | 87 |
| LLVIII-C - Compiler le projet..... | 89 |
| LLVIII-D - Déployer le projet..... | 91 |
| LLVIII-E - Tester le projet..... | 93 |
| LLVIII-E-1 - Méthode 1: Web Service Tester de GlassFish..... | 94 |
| LLVIII-E-2 - Méthode 2: Web Service Tester de NetBeans..... | 94 |
| LLVIII-F - Conclusion..... | 101 |
| IX - Importer le projet Yaps - côté serveur..... | 102 |
| LLIX-A - Créer un nouveau projet..... | 102 |
| LLIX-A-1 - Fermeture des projets..... | 102 |
| LLIX-A-2 - Type de projet..... | 103 |
| LLIX-A-3 - Nom et répertoires..... | 103 |
| LLIX-A-4 - Configuration du serveur..... | 104 |
| LLIX-A-5 - Génération du projet..... | 105 |
| LLIX-B - Transférer le projet sous NetBeans..... | 106 |
| LLIX-B-1 - Positionner les variables d'environnement..... | 106 |
| LLIX-B-2 - Transférer le projet..... | 107 |
| LLIX-B-3 - Référencer les Web Services..... | 110 |
| LLIX-B-3-a - Référencer DeliveryService..... | 110 |
| LLIX-B-3-b - Référencer ValidationService..... | 113 |
| LLIX-B-4 - Créer le groupe de projet..... | 116 |
| LLIX-C - Changer le Context Root..... | 118 |
| LLIX-D - Créer le fichier de déploiement standard..... | 119 |
| LLIX-E - Compiler le projet..... | 122 |
| LLIX-F - Déployer le projet..... | 123 |

| | |
|--|-----|
| LLIX-G - Base de données..... | 125 |
| LLIX-G-1 - Créer une connexion..... | 125 |
| LLIX-G-2 - Connexion à la DB..... | 129 |
| LLIX-G-3 - Explorer la DB..... | 131 |
| LLIX-G-4 - Données de test..... | 134 |
| LLIX-G-4-a - Méthode 1 : Chargement manuel des données de test..... | 135 |
| LLIX-G-4-b - Méthode 2 : Chargement automatique des données de test..... | 138 |
| LLIX-H - Tester le projet..... | 146 |
| LLIX-I - Conclusion..... | 150 |
| X - Importer le projet Yaps - côté client..... | 151 |
| LLX-A - Créer un nouveau projet..... | 151 |
| LLX-A-1-a - Fermer les projets..... | 151 |
| LLX-A-1-b - Type de projet..... | 151 |
| LLX-A-1-c - Nom et répertoires..... | 152 |
| LLX-A-1-d - Génération du projet..... | 154 |
| LLX-B - Transférer le projet sous NetBeans..... | 154 |
| LLX-B-1 - Positionner les variables d'environnement..... | 154 |
| LLX-B-2 - Transférer le projet..... | 154 |
| LLX-B-3 - Référencer les bibliothèques..... | 156 |
| LLX-B-3-a - Référencer les classes de Yaps-Server..... | 157 |
| LLX-B-3-b - Référencer les bibliothèques utilitaires..... | 161 |
| LLX-B-3-c - Référencer les bibliothèques JMS..... | 163 |
| LLX-B-3-d - Référencer les bibliothèques TopLink..... | 171 |
| LLX-B-4 - Identifier la classe principale..... | 174 |
| LLX-B-5 - Créer le groupe de projet..... | 177 |
| LLX-C - Compiler le projet..... | 179 |
| LLX-D - Tester le projet..... | 182 |
| LLX-D-1 - Démarrer le serveur PetStore..... | 182 |
| LLX-D-2 - Exécuter l'application Yaps-Client..... | 182 |
| LLX-E - Conclusion..... | 185 |
| XI - Conclusion..... | 186 |
| XII - Références..... | 187 |
| XIII - Glossaire..... | 188 |

I - Introduction

Dans le livre d'Antonio Goncalves consacré à Java **EE5**, l'auteur illustre les principaux concepts de JEE 5 au travers d'une application fictive baptisée Yaps Pet Store (ou Yaps pour faire court). Cette application est elle-même une émanation du Java Petstore de Sun issue du programme Java BluePrints.

Le livre de référence couplé à l'application Yaps est un excellent support pédagogique pour qui souhaite aborder et expérimenter le vaste sujet qu'est **JEE 5**.

Les sujets du livre de référence sont traités de manière à construire pas à pas les différents composants de l'application en abordant les principales spécifications de **JEE5** : **EJB**, **JPA**, **JMS**, **JAX-WS**, JavaMail, etc.

L'auteur a pris le parti d'être indépendant d'un quelconque IDE. Des tâches Ant sont utilisées pour configurer l'environnement de travail, ainsi que pour construire et déployer les projets de l'application.

Ce document a pour objectif de décrire les différentes étapes nécessaires pour porter l'application Yaps sous l'IDE de Sun : NetBeans 6.5

Pourquoi NetBeans ?

Loin de nous l'idée de débattre entre tel ou tel **IDE**. Chaque **IDE** a ses forces et faiblesses. Le plus important est que l'**IDE** choisi (Eclipse, NetBeans, IntelliJ, etc.) soit un moteur et non un frein dans le processus d'industrialisation de logiciels.

Par ailleurs, connaître plus d'un **IDE** est une valeur ajoutée. En tant que consultants informatiques, nous sommes souvent amenés à adapter nos habitudes par rapport à l'environnement de l'entreprise dans laquelle nous intervenons. Le choix des outils logiciels est souvent dicté par des contraintes tant financières, techniques ou culturelles.

Dans le cas de notre entreprise, NetBeans est très apprécié notamment pour sa couche **SOA**. Les outils **SOA** intégrés à NetBeans ainsi que l'**ESB** (Enterprise Service Bus) **OpenESB** sont excellents pour concevoir des prototypes applicatifs. Mais c'est un autre sujet que nous aurons probablement l'occasion d'aborder dans le cadre d'un autre document.

NetBeans couplé à GlassFish est le partenaire idéal pour aborder les concepts **JEE5**.

Et c'est justement l'objectif de ce document.

À partir du livre de référence et de l'application Yaps, nous allons importer l'ensemble des projets sous NetBeans 6.5 pour obtenir un environnement complet et gratuit pour expérimenter les différents sujets du livre et aller plus loin dans la découverte des concepts entourant les spécifications **JEE 5**.

Ce document est subdivisé comme suit :

| | |
|---|--|
| Installation | Installation de NetBeans et des fichiers nécessaires au processus d'importation de l'application Yaps Pet Store. |
| Configurer YapsON | Décris la configuration à apporter aux fichiers livrés avec l'application initiale YapsON. |
| Configurer GlassFish | La configuration de GlassFish nécessaire avant de pouvoir déployer les différents projets. |
| Configurer NetBeans | La configuration de NetBeans avant d'accueillir les projets à déployer. |
| Importer le projet BarkBank | L'importation et le déploiement du projet BarkBank. |
| Importer le projet PetEx | L'importation et le déploiement du projet PetEx. |
| Importer le projet Yaps - côté serveur | L'importation et le déploiement de la partie serveur du projet Yaps. |
| Importer le projet Yaps - côté client | L'importation et le déploiement de la partie cliente du projet Yaps. |
| Conclusion | La conclusion de ce document. |

II - Conventions

II-A - Langue

Dans ce document, les copies d'écrans des logiciels utilisés (Microsoft Windows, NetBeans, etc.) sont en anglais. Ils peuvent donc différer de votre environnement.

II-B - Mots clés

Le tableau suivant fournit l'ensemble des mots clés utilisés dans le document :

| | |
|--------------------|---|
| Livre de référence | Par "Livre de référence", nous entendons le livre sur Java EE5 d'Antonio Goncalves (voir [R1]). |
| Yaps | Identifie l'application originale Yaps Pet Store fournie par l'auteur du livre de référence. |
| YapsON | Identifie l'application Yaps importée sous NetBeans (Yaps Pet Store on NetBeans). |

II-C - Répertoires

| | |
|------------------|---|
| {netbeans_home} | Répertoire d'installation de NetBeans. |
| {glassfish_home} | Répertoire d'installation de GlassFish v2. |
| {yaps_home} | Répertoire hébergeant l'application originale Yaps Pet Store. |
| {yaps_on_home} | Répertoire de l'application Yaps créé sous NetBeans. |

II-D - Commandes

Les commandes sont représentées de la manière suivante :

```
C:\> echo "This is a command"
Cette commande affiche à l'écran la phrase suivante: "This is a command".
```

II-E - Messages

Dans le document, des messages peuvent apparaître pour alerter ou informer le lecteur.



Un message d'alerte.



Un message d'information.

III - Installation

III-A - Prérequis

Ce document a été conçu avec la configuration logicielle suivante:

- Microsoft Windows Vista Service Pack 1 ou supérieur
- Sun Java Development Kit (JDK) 1.6 Update 7 ou supérieur
- Ant 1.7.0 ou supérieur
- NetBeans 6.5 installé avec le Bundle « All »
- Les fichiers de Yaps
- Les fichiers de YapsON

Le chapitre 3 du livre de référence décrit comment installer le JDK et Ant.

III-B - NetBeans

NetBeans 6.5 est l'IDE (Integrated Development Environment) Open Source de Sun conçu pour concevoir, déployer et tester des applications développées sous différents langages (Java, PHP, Ruby, C/C++, etc.).

Tout comme Eclipse, NetBeans utilise la notion de plug-ins pour enrichir son environnement de développement.

En ce qui concerne JEE5, NetBeans fournit un environnement riche et complet pour concevoir des applications (visuelles ou non) et les déployer au sein de serveurs tels que GlassFish ou Tomcat qui font partie intégrante du kit d'installation.

NetBeans est disponible sous différents bundles. Chacun couvrant un ensemble de fonctionnalités.

Ce document a été conçu sur base du bundle « All » de NetBeans. Mais vous pouvez vous contenter du bundle « Java » et récupérer plus tard les plug-ins qui vous intéressent (par exemple le stack SOA).

III-B-1 - Récupérer NetBeans

NetBeans 6.5 peut être téléchargé gratuitement à partir du site de Sun à l'adresse suivante:

<http://www.netbeans.org>

La page affiche la dernière version en date de NetBeans.



Figure 3-1: Version de NetBeans

Cliquez sur le bouton "Download NetBeans IDE" pour continuer.

La page de téléchargement permet de définir les propriétés du kit NetBeans :

- Langue de travail
- Plateforme

Language: Platform:

Note: The UML Pack is available on the Update Center.

Figure 3-2: Langue de travail et plateforme

Ensuite, il vous reste à sélectionner le type de bundle :

NetBeans IDE Download Bundles

| Supported technologies * | Java SE | Java | Ruby | C/C++ | PHP | All |
|--------------------------|--|---|--|--|--|---|
| Java SE | • | • | | | | • |
| Java Web and EE | | • | | | | • |
| Java ME | | • | | | | • |
| Ruby | | | • | | | • |
| C/C++ | | | | • | | • |
| PHP | | | | | • | • |
| MOA | | | | | | • |
| Cloud servers | | | | | | • |
| GlassFish V2 UR2 | | • | | | | • |
| GlassFish v3 Prelude | | • | • | | | • |
| Apache Tomcat 6.0.18 | | • | | | | • |
| | <input type="button" value="Download"/> Free, 37 MB | <input type="button" value="Download"/> Free, 212 MB | <input type="button" value="Download"/> Free, 58 MB | <input type="button" value="Download"/> Free, 23 MB | <input type="button" value="Download"/> Free, 24 MB | <input type="button" value="Download"/> Free, 250 MB |

Figure 3-3: Type de bundle

Ce document est conçu à partir du bundle "All".
Cliquez sur le bouton "Download" pour télécharger NetBeans.

Le site de NetBeans regorge d'information utile qui facilitera votre prise en main de cet IDE.
Vous trouverez notamment bon nombre de tutoriaux estampillés avec la version de l'IDE qui est supportée.

III-B-2 - Installer NetBeans

Vous allez passer en revue les différentes étapes requises pour installer NetBeans.
Commencez par exécuter le fichier "**netbeans-6.5-m1-windows.exe**".

III-B-2-a - Page d'accueil

La page d'accueil permet d'emblée de définir la liste des composants à installer :

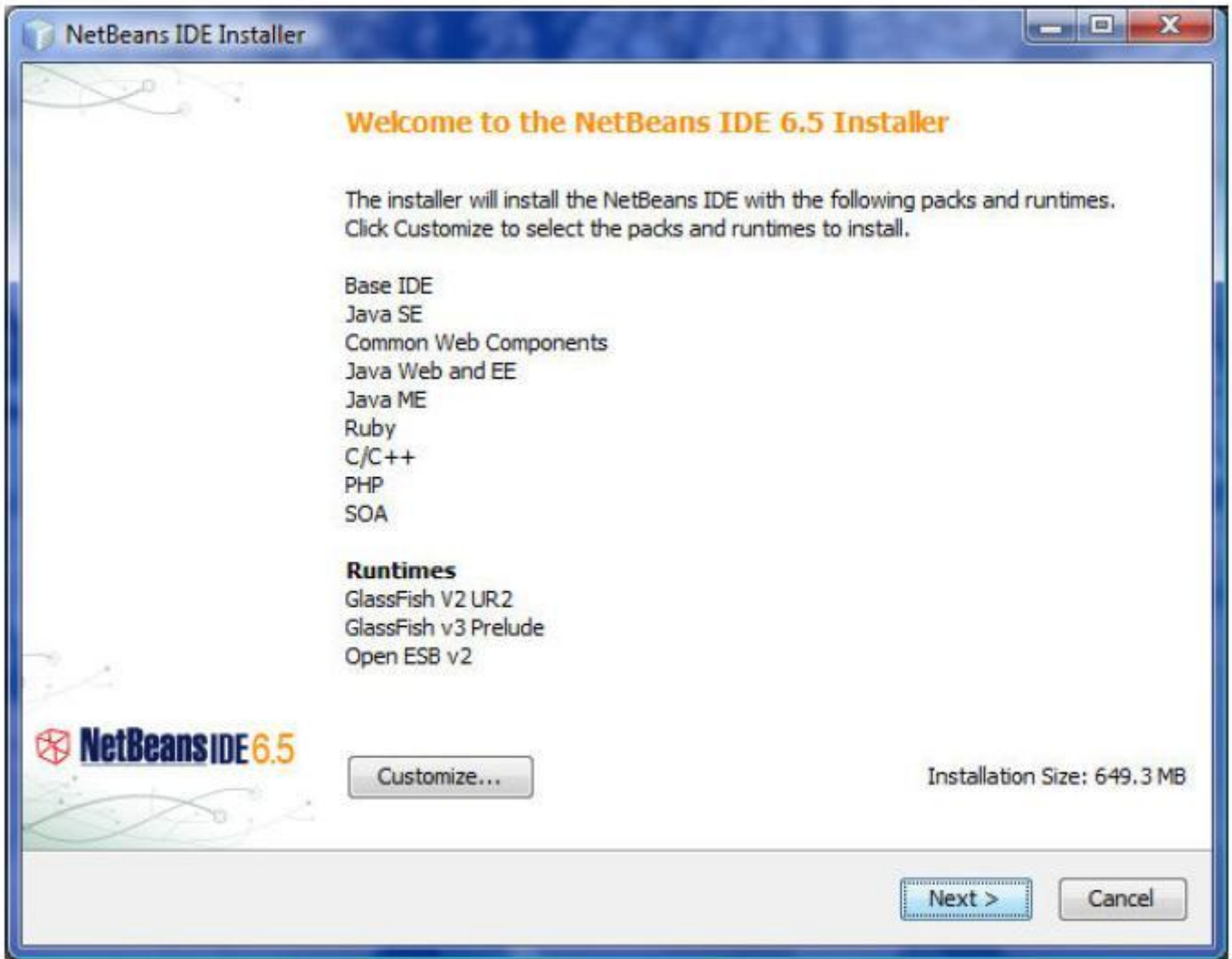


Figure 3-4: Page d'accueil

Le seul composant optionnel est le serveur d'applications « Tomcat version 6 ».

Si vous souhaitez modifier la liste des composants à installer, cliquez sur le bouton « Customize » :

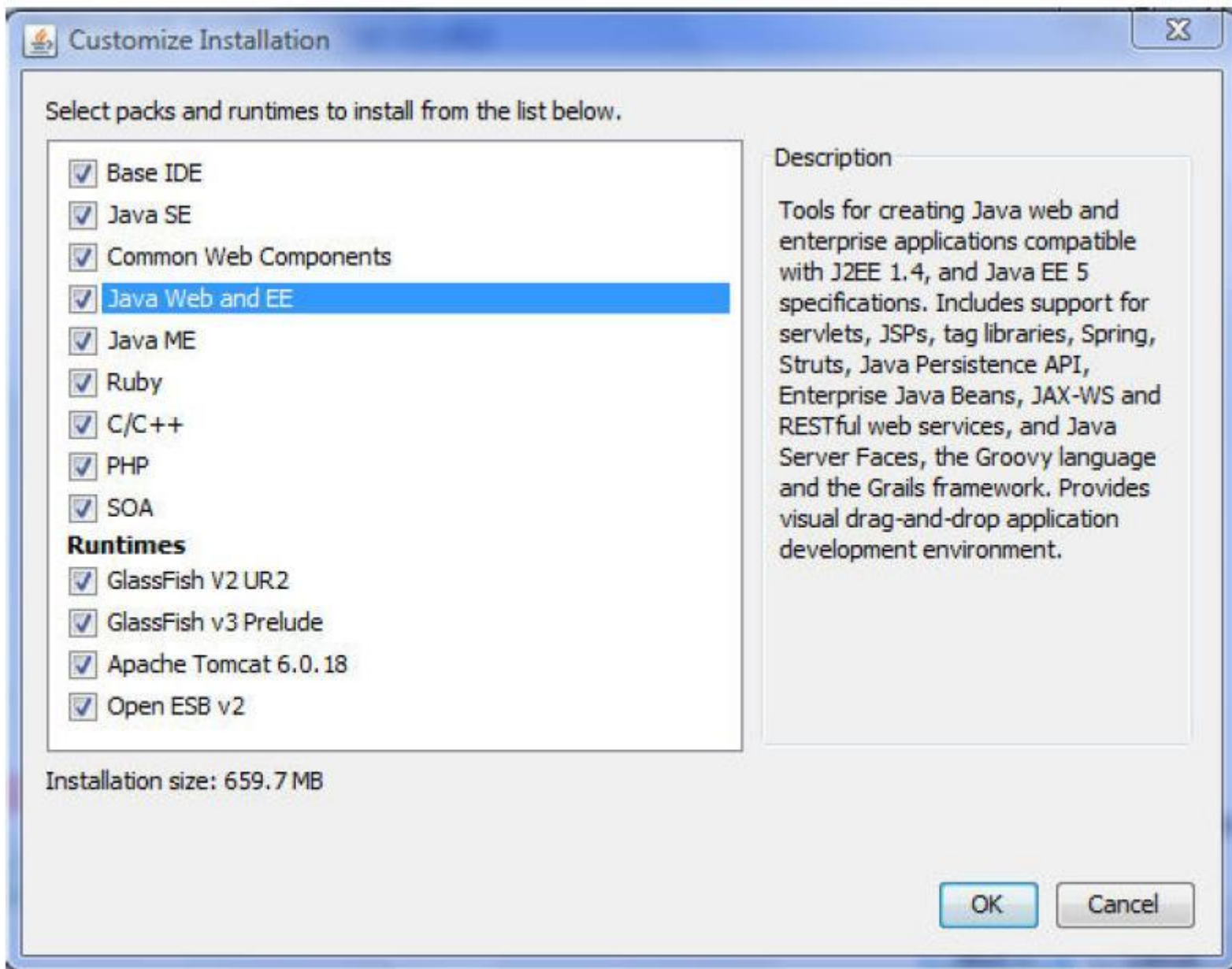


Figure 3-5: Paramétrer (Customize) l'installation

Dans l'écran ci-dessus, on remarque que l'ensemble des composants a été sélectionné.

Dans le contexte de l'application YapsON, veuillez à sélectionner les composants suivants :

- Base IDE
- Java SE
- Common Web Components
- Java Web and EE
- GlassFish V2 R2

Cliquez sur « OK » pour valider votre configuration.
Ensuite, cliquez sur « Next » pour continuer.

III-B-2-b - License Agreement

Cette étape présente la licence Open Source utilisée par NetBeans :

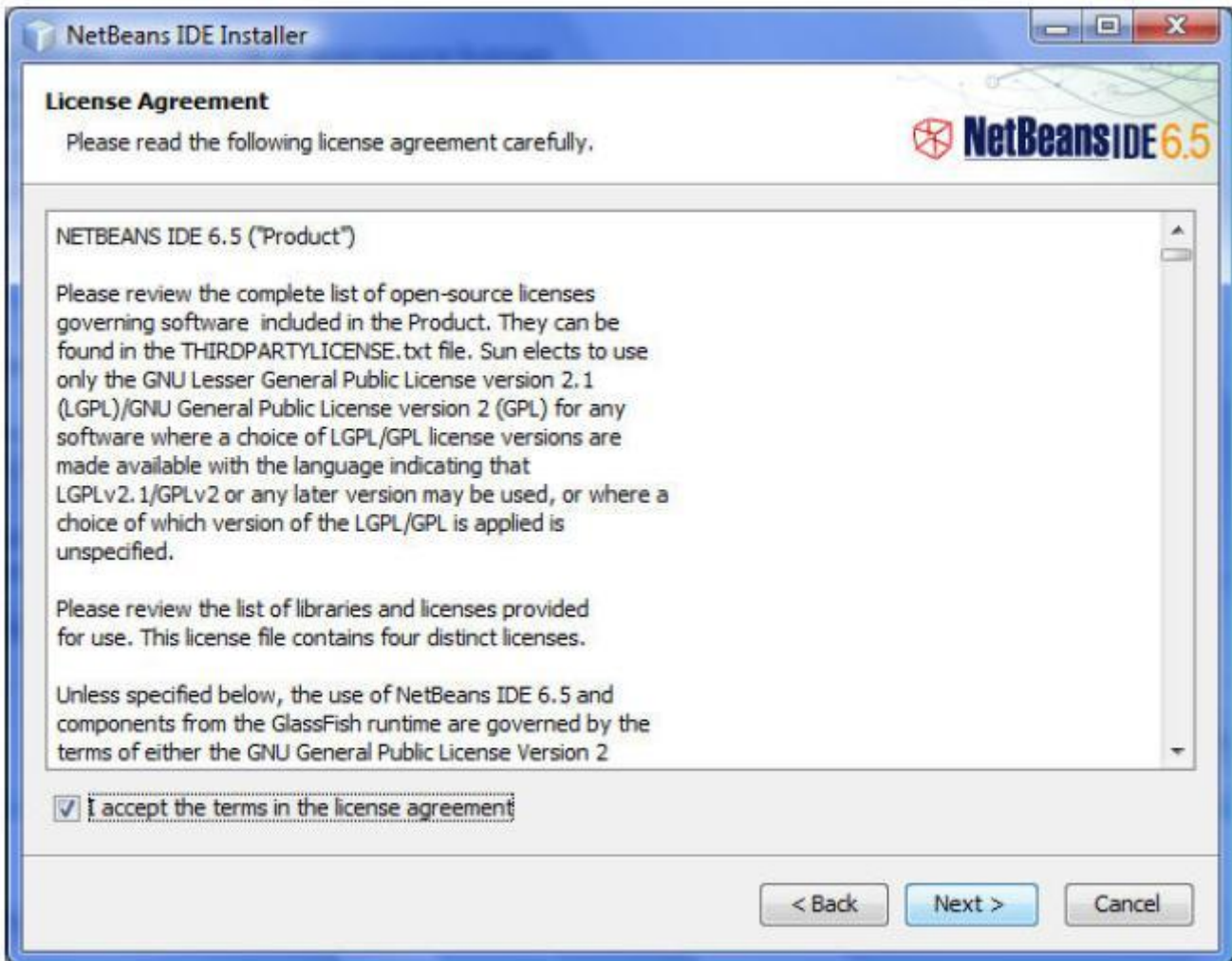


Figure 3-6: Licence de NetBeans

Acceptez les termes de la licence et cliquez sur "Next" pour continuer.

III-B-2-c - Répertoires d'installation

Cette étape permet de définir les répertoires d'installation de NetBeans et de localiser le JDK préinstallé dans votre environnement :

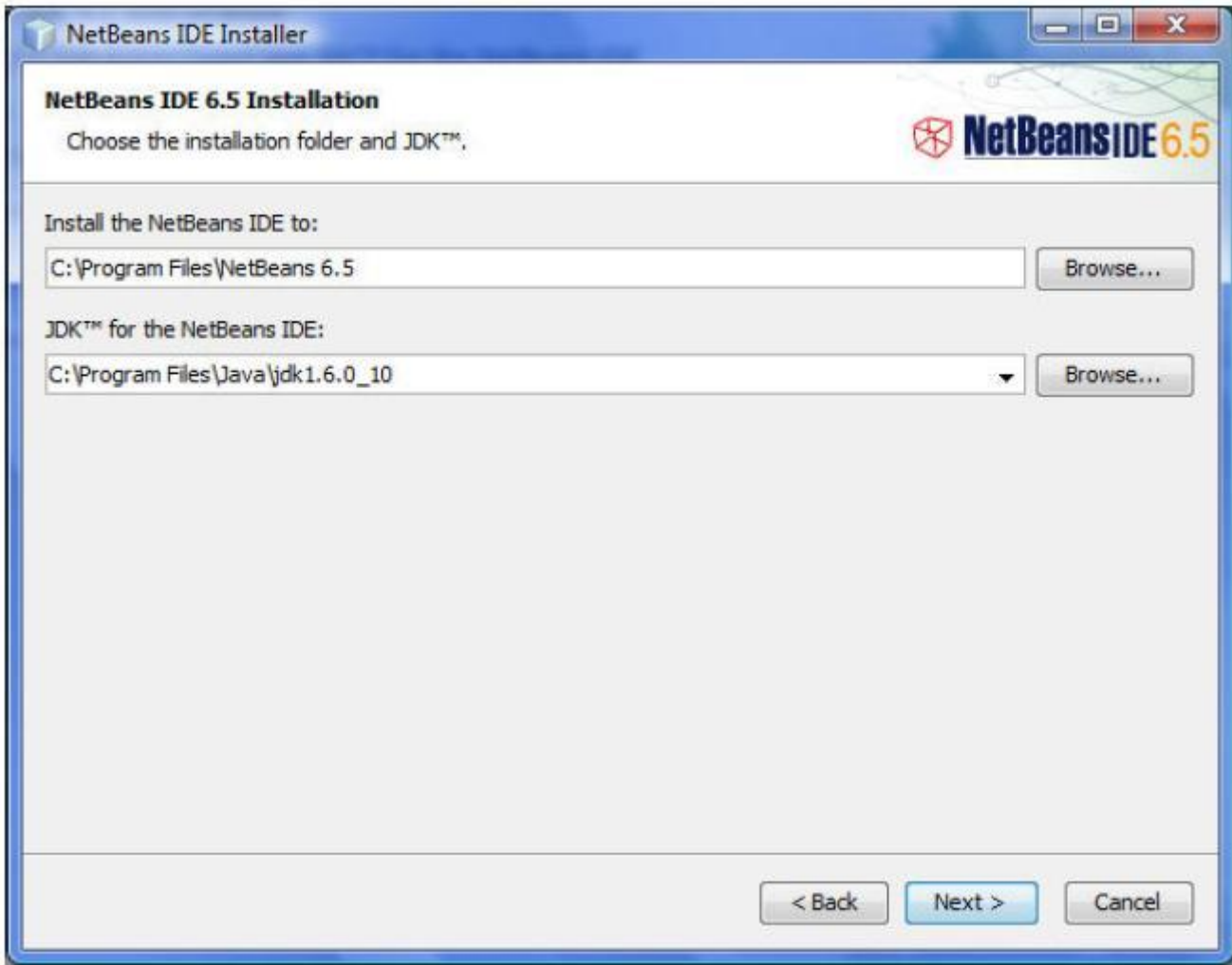


Figure 3-7: Répertoires d'installation

Cliquez sur « Next » pour continuer.

III-B-2-d - Propriétés de GlassFish v2

Propriétés de GlassFish v2

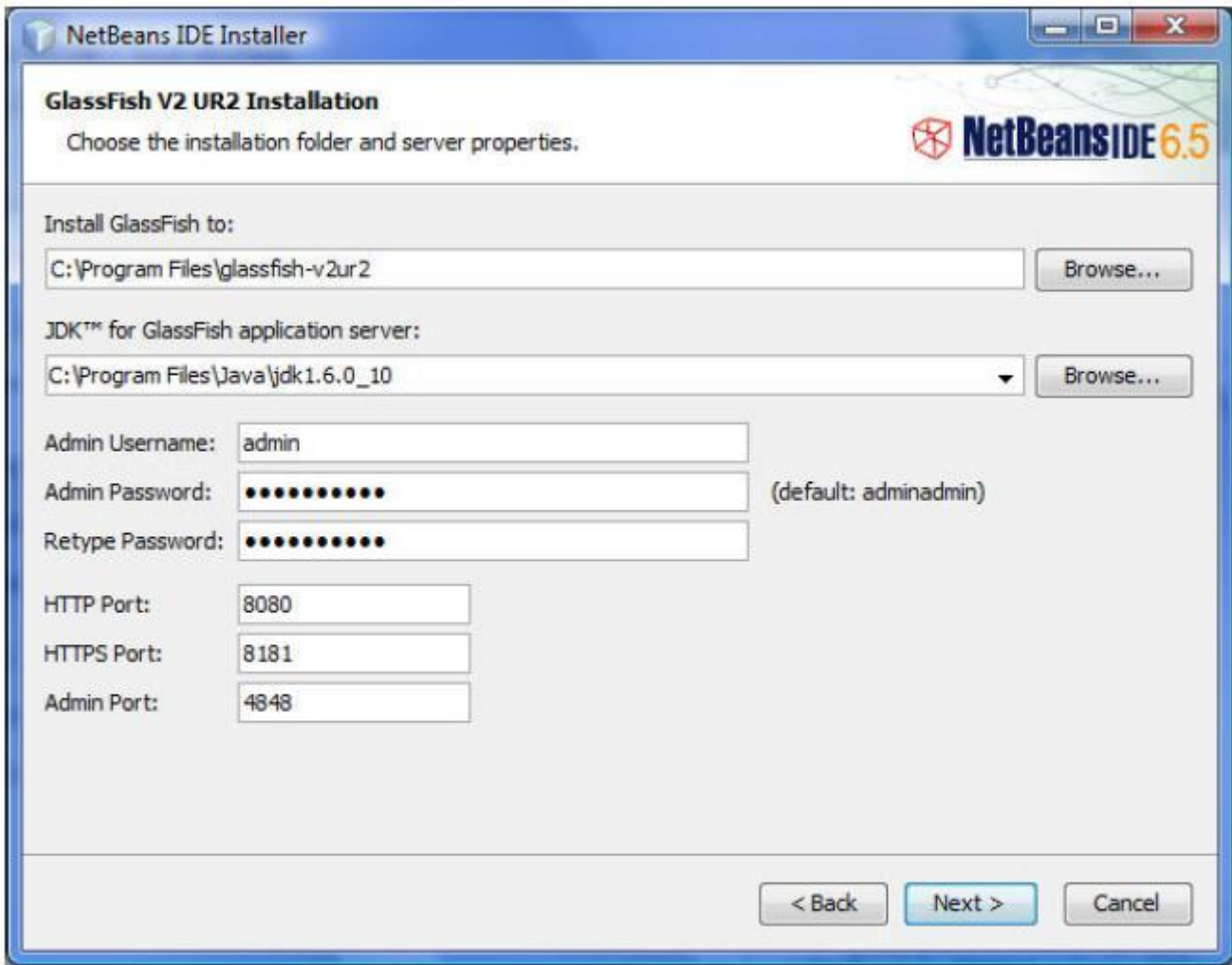


Figure 3-8: Paramètres de GlassFish V2

Nous avons gardé les paramètres par défaut à savoir :

| | |
|-----------------------|---|
| GlassFish | Le dossier où sera installé GlassFish v2. |
| JDK | Le dossier où a été installé le JDK . |
| Admin Username | Le nom qui sera utilisé pour administrer GlassFish. Le défaut est: admin |
| Admin Password | Le mot de passé de l'administrateur de GlassFish. Le défaut est: adminadmin |
| Ports | Les numéros de port. |

Cliquez sur « Next » pour continuer.

III-B-2-e - GlassFish v3

La version 3 de GlassFish est livrée à titre d'évaluation (Prelude version).

L'application YapsON n'utilise pas GlassFish v3 :

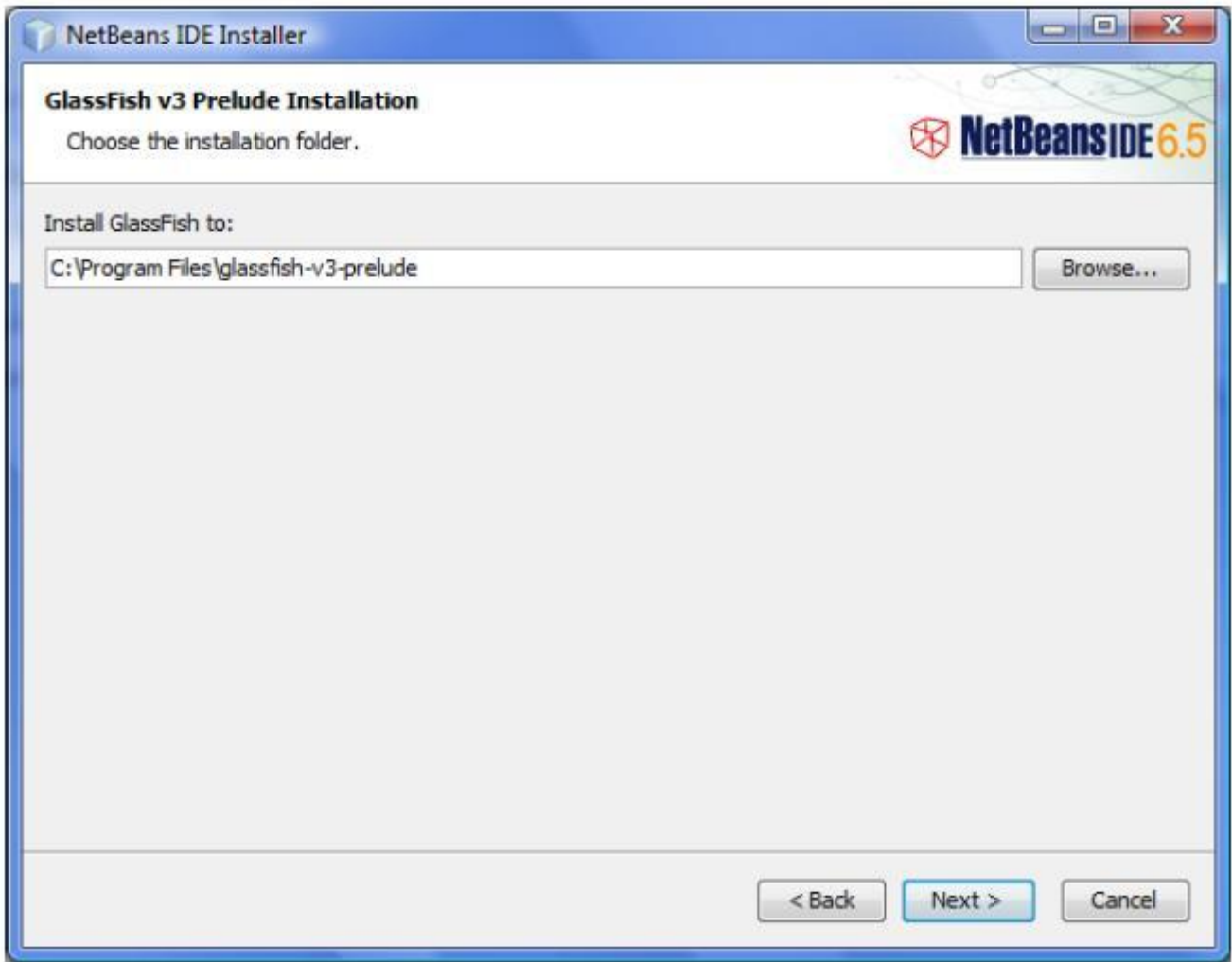


Figure 3-9: Répertoire de GlassFish v3

Vous pouvez garder le défaut et cliquer sur « Next » pour continuer.

III-B-2-f - Apache Tomcat

Cette étape apparaît uniquement si vous avez sélectionné le package optionnel de Tomcat.

Dans ce cas, l'écran suivant vous demande de choisir le répertoire d'installation de Tomcat v6 ou d'accepter le défaut :

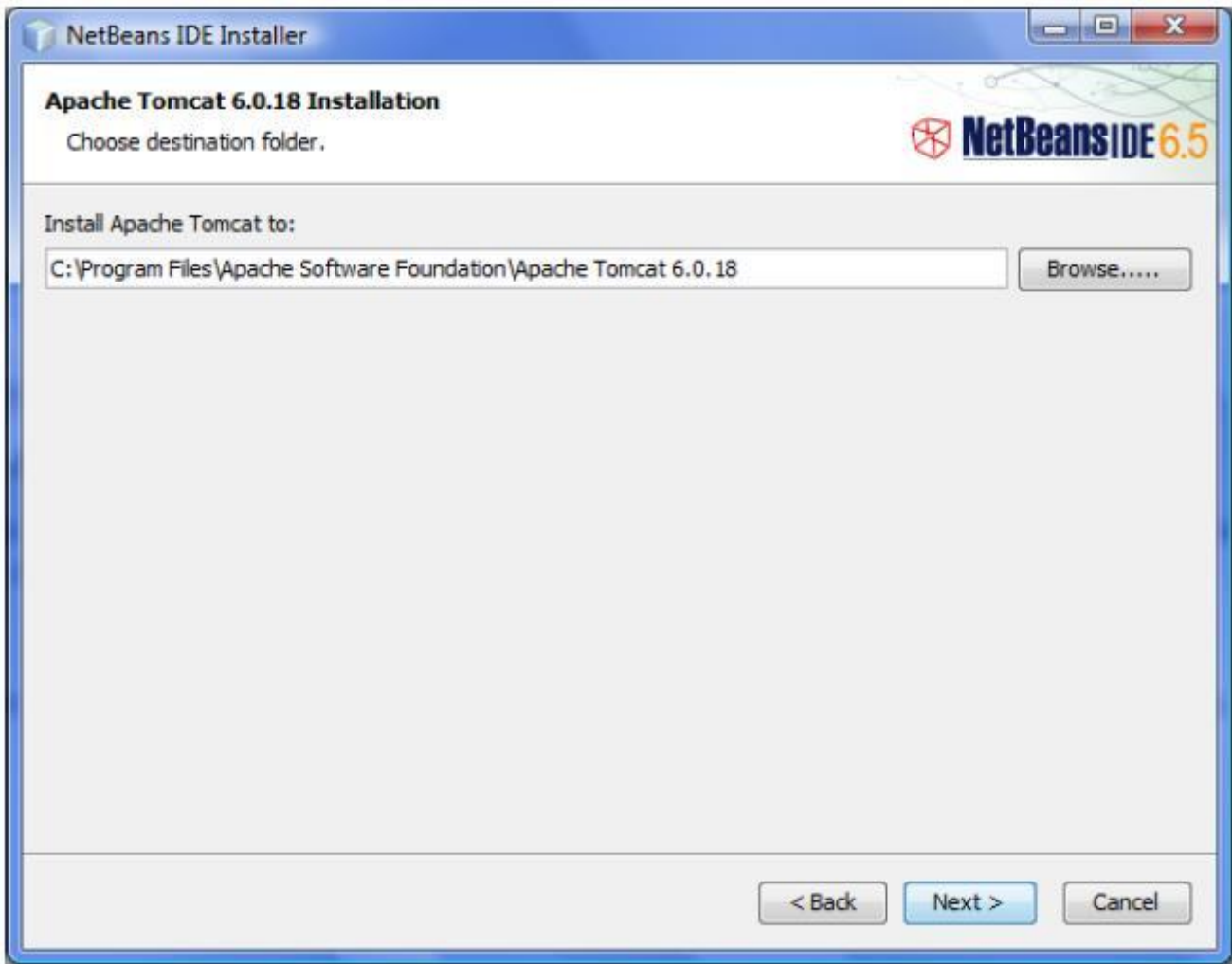


Figure 3-10: Répertoire de Tomcat v6

Cliquez sur « Next » pour continuer.

III-B-2-g - Résumé

La page suivante présente un résumé du paramétrage :

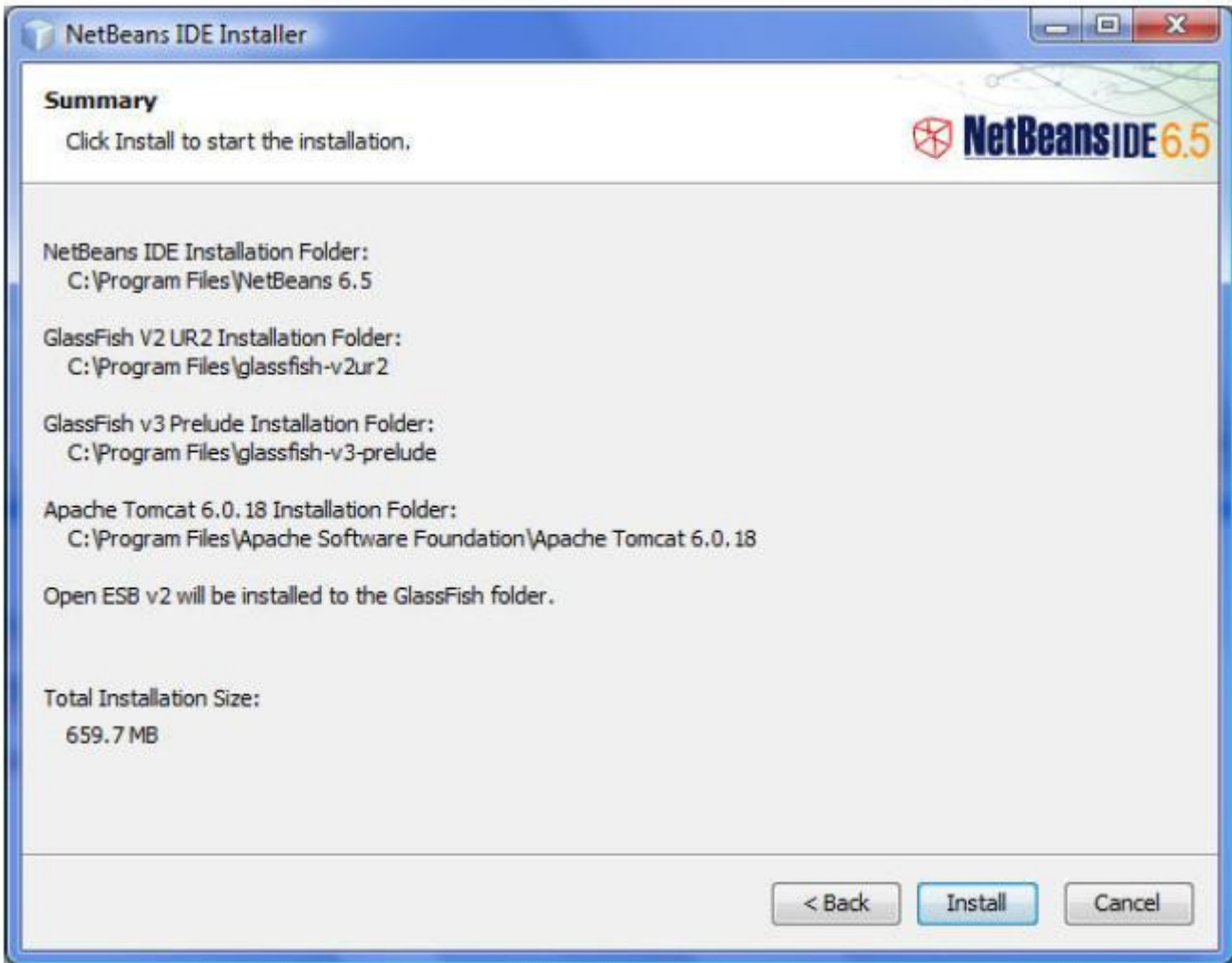


Figure 3-11: Résumé d'installation

Cliquez sur « Install » pour démarrer le processus d'installation.

III-B-2-h - Finish

À la fin de l'installation, la dernière étape vous demande d'enregistrer NetBeans et GlassFish :

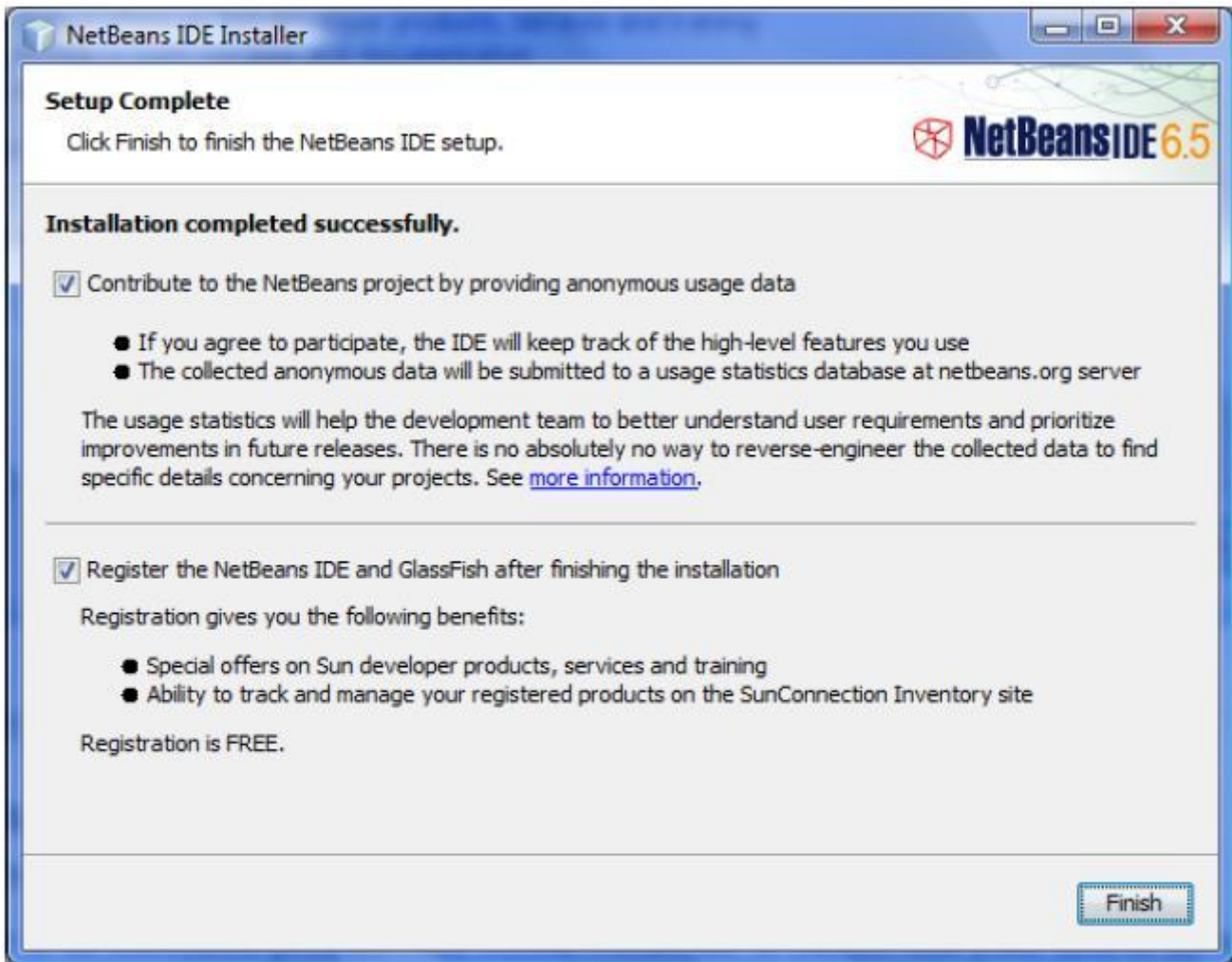


Figure 3-12: Enregistrer NetBeans et GlassFish

Cliquez sur « Finish » pour terminer le processus d'installation.

III-B-3 - Démarrer NetBeans

Démarrez NetBeans sous le lien:

Start -> All Programs -> NetBeans -> NetBeans IDE 6.5

NetBeans offre un processus automatique qui lui permet de vérifier la présence de mises à jour. Ce processus est enclenché à chaque démarrage de NetBeans :



Figure 3-13: Vérification des mises à jour

Un message dans la barre de statut vous informe de la présence des mises à jour :



Figure 3-14: Mises à jour détectées

Pour visualiser la liste des mises à jour et les appliquer, cliquez sur l'icône de la barre de statut.

Il est bien entendu recommandé d'appliquer ces mises à jour.

III-B-4 - Localisation

Par défaut, NetBeans utilise la localisation courante pour afficher ses boîtes de dialogues.

Vous pouvez forcer une localisation spécifique en modifiant le fichier « netbeans.conf » de NetBeans qui se trouve à l'emplacement suivant :

```
{netbeans_home}\etc\netbeans.conf
```

Dans ce fichier, modifiez l'entrée "**netbeans_default_options**" pour y ajouter l'option :


```
--locale <code_langue>
```

Par exemple, pour forcer le code langue en anglais :

```
--locale EN
```

Ci-dessous, un extrait du fichier de configuration montre le code langue fixé à **EN** :

```
...
# Options used by NetBeans launcher by default, can be overridden by
explicit
# command line switches:
netbeans_default_options="-JDorg.
netbeans.modules.tomcat.autoregister.token=1227909276055 -J
Dorg.netbeans.modules.tomcat.autoregister.catalinaHome=\"C:\Program
Files\Apache Software Foundation\Apache Tomcat 6.0.18\" -JDorg.
glassfish.v3.installRoot=\"C:\Program Files\glassfish-v3-prelude\" -JDcom.
sun.aas.installRoot=\"C:\Program Files\glassfish-v2ur2\" -J-client -JXverify:
none -J-Xss2m -J-Xms32m -J-XX:PermSize=32m -J-XX:MaxPermSize=200m -
J-Dapple.laf.useScreenMenuBar=true -J-Dsun.java2d.noddraw=true --locale EN"
...
```



Positionnez le code langue

Une autre méthode consiste à modifier les propriétés de démarrage de NetBeans pour y ajouter le code langue :




Figure 3-15: Code langue

LLIII-C - Variables d'environnement

Il sera quelquefois nécessaire d'exécuter des instructions via la ligne de commande.

Pour ce faire, les variables d'environnement suivantes doivent être définies dans votre système:

- GLASSFISH_HOME
- ANT_HOME
- JAVA_HOME

La variable d'environnement PATH doit également être adaptée pour localiser les logiciels requis.

Ci-dessous, nous donnons un exemple de définition de variables d'environnement sous MS-Windows:

```

C:\> set ANT_HOME=C:\Tools\Ant\apache-ant-1.7.1
C:\> set PATH=%ANT_HOME%\bin;%PATH%
C:\> set GLASSFISH_HOME=C:\Program Files\glassfish-v2ur2
C:\> set PATH=%GLASSFISH_HOME%\bin;%PATH%
C:\> set JAVA_HOME=C:\Program Files\Java\jdk1.6.0_10
C:\> set PATH=%JAVA_HOME%\bin;%PATH%
    
```

Ces variables d'environnement peuvent également être définies dans les propriétés système de MS-Windows.

LLIII-D - Récupérer l'application Yaps

Récupérez l'application Yaps que sera importée sous NetBeans.
 L'application Yaps se trouve à l'adresse suivante :
<http://www.antoniogoncalves.org/xwiki/bin/view/Book/JavaEE5Fr>

Dans cette page, reportez-vous à la section relative au téléchargement du code source :

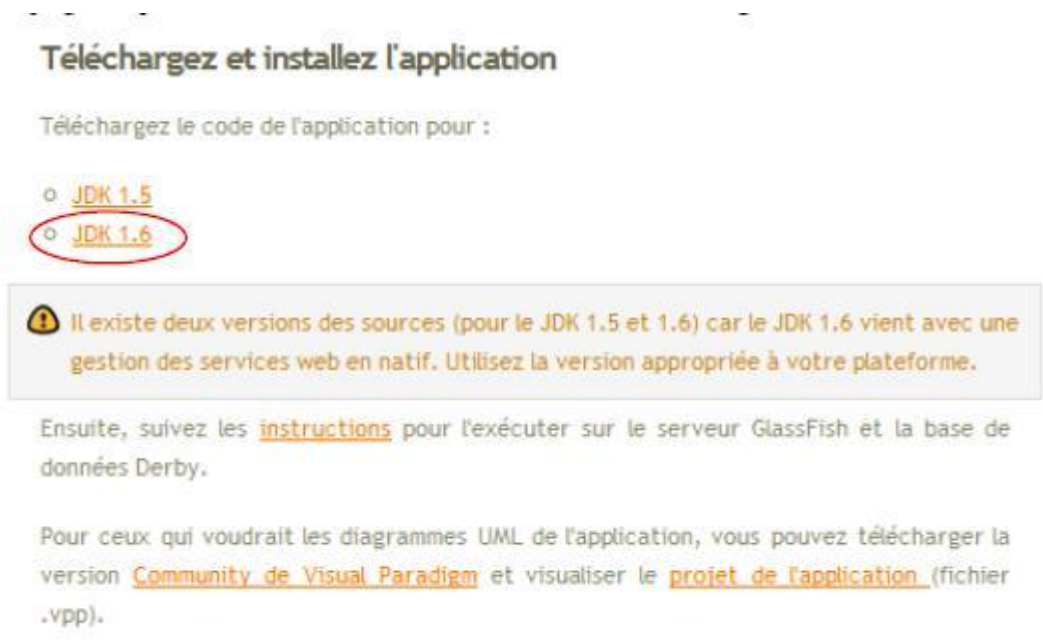


Figure 3-16: Télécharger Yaps

Ce document est basé sur une version 1.6 du JDK. Veillez à télécharger le code source de l'application Yaps correspondant au JDK 1.6.

Ensuite, vous devez extraire le fichier d'archives dans un répertoire qui sera référencé dans la suite du document sous le nom **{yaps_home}**.

Les répertoires et les documents suivants sont extraits du fichier d'archives :

| Name | Date modified | Type | Size |
|--------------|------------------|---------------|-------|
| BarkBank | 09/10/2008 10:50 | File Folder | |
| PetEx | 09/10/2008 10:50 | File Folder | |
| Yaps | 09/10/2008 10:50 | File Folder | |
| admin.xml | 01/10/2007 14:57 | XML Document | 21 KB |
| build.xml | 01/10/2007 14:57 | XML Document | 31 KB |
| derby.log | 09/10/2008 11:32 | Text Document | 1 KB |
| passwordfile | 01/10/2007 14:57 | File | 1 KB |
| readme.txt | 01/10/2007 14:57 | Text Document | 18 KB |

Figure 3-17: Structure de l'archive Yaps

Le livre de référence détaille l'ensemble des répertoires.

LLIII-E - Récupérer l'application YapsON

L'application YapsON est disponible sous deux archives différentes:

| | |
|----------------------------|--|
| YapsON-initial.zip | Ce fichier contient la version minimale de l'application YapsON utilisée comme point de départ au processus de portage décrit dans ce document. |
| YapsON-complete.zip | Ce fichier contient la version complète de l'application YapsON telle qu'obtenue après la phase de portage décrite dans ce document. |

Ce document décrit les étapes de portage à appliquer sur la **version minimale** de l'application YapsON.

La version minimale est composée des fichiers suivants :

| | |
|--------------------------|---|
| admin.xml | Ce fichier Ant provient du projet Yaps Pet Store et contient un ensemble de tâches pour configurer et gérer GlassFish et Derby. |
| setenv-yaps.cmd | Fichier de commande définissant les variables d'environnement JAVA_HOME, ANT_HOME, GLASSFISH_HOME. |
| yapson.xml | Fichier de script Ant permettant le transfert des projets Yaps vers l'arborescence de NetBeans. |
| yapson.properties | Fichier de propriétés utilisé par le script Ant « yapson.xml ». |
| yaps-data.xml | Fichier de script Ant utilisé pour charger la base de données avec les données de test. |

Les fichiers de l'application YapsON sont disponibles à l'adresse suivante :

<http://www.noratek.net/2008/12/08/yaps-sous-netbeans-65/>

Vous devez récupérer et extraire le fichier d'archives **YapsON-initial.zip** dans un répertoire qui sera référencé dans la suite du document sous le nom **{yaps_on_home}** :






| Name | Date modified | Type | Size |
|---|------------------|------------------------|-------|
|  admin.xml | 29/11/2008 10:53 | XML Document | 21 KB |
|  setenv-yaps.cmd | 29/11/2008 10:28 | Windows Command Script | 1 KB |
|  yaps-data.xml | 30/11/2008 15:40 | XML Document | 2 KB |
|  yapson.properties | 18/11/2008 16:46 | PROPERTIES File | 1 KB |
|  yapson.xml | 30/11/2008 13:54 | XML Document | 7 KB |

Figure 3-18: Structure de l'archive YapsON

IV - Configurer YapsON

Avant d'entamer le processus de configuration, vous allez configurer les fichiers de commande livrés avec le fichier d'archives de l'application YapsON.

LLIV-A - Variables d'environnement

Éditez le fichier « **setenv-yaps.cmd** » et modifiez les variables d'environnements suivantes pour y placer les répertoires d'installation adéquats :

| | |
|----------------|--|
| YAPS_HOME | Répertoire d'installation de l'application Yaps (voir ici). |
| YAPSON_HOME | Répertoire d'installation de l'application YapsON (voir ici). |
| GLASSFISH_HOME | Répertoire d'installation de GlassFish v2 (voir ici). |
| ANT_HOME | Répertoire d'installation d'Ant (voir ici). |
| JAVA_HOME | Répertoire d'installation du JDK (voir ici). |

LLIV-B - Scripts Ant

Éditez le fichier « **yapson.properties** » et modifiez les propriétés suivantes :

| | |
|---------------|--|
| yaps.folder | Répertoire d'installation de l'application Yaps (voir ici). |
| yapson.folder | Répertoire d'installation de l'application YapsON (voir ici). |

Il y a également lieu d'éditer le fichier « **yaps-data.xml** » afin de modifier la propriété suivante :

| | |
|----------------|--|
| glassfish.home | Répertoire d'installation de GlassFish v2 (voir ici). |
|----------------|--|

V - Configurer GlassFish

La configuration du serveur d'applications GlassFish est similaire à celle exposée dans le livre de référence et résumée dans le fichier « readme.txt » livré avec le fichier d'archives de Yaps.

Afin de rendre ce document autonome, nous allons rappeler les commandes nécessaires pour configurer GlassFish.

Pour cela, vous allez utiliser les tâches Ant à partir de la ligne de commande.

Problèmes avec Ant et le copié/collé

Vous pouvez rencontrer des problèmes si vous copiez/collez d'un document (MS-Word, etc.) des commandes Ant vers la ligne de commande.

Maintenant, nous illustrons un problème qui peut apparaître :

```

C:\Projects\Yaps>ant -f admin.xml start-domain
Buildfile: build.xml

BUILD FAILED
Target "uf" does not exist in the project "Petstore".

Total time: 0 seconds
    
```

Il se pourrait que dans certains cas, des caractères spéciaux accompagnent la commande qui est collée. Ant détecte ces caractères spéciaux et tente de trouver la cible correspondante. Qui n'existe bien évidemment pas.

Pour éviter ce problème, taper les commandes ou configurer votre environnement pour un codepage adéquat.

LLV-A - Positionner les variables d'environnement

Avant tout, vous devez positionner les variables d'environnement nécessaires pour l'exécution des tâches Ant de configuration.

Rendez-vous dans le répertoire de l'application YapsON et exécutez le fichier de commande « setenv-yaps.cmd » :

```
C:> cd {yapson_home}
{yapson_home}> setenv-yaps.cmd
```

LLV-B - Créer le domaine « petstore »

Sous GlassFish, vous allez créer le domaine « petstore » dans lequel seront déployés les projets de l'application YapsON.

Les propriétés du domaine « petstore » sont les suivantes :

| | |
|--------------------------|-----------|
| Name | petstore |
| http port number | 8080 |
| Admin port number | 8282 |
| Admin username | admin |
| Admin password | adminpwd |
| Master password | masterpwd |

Exécutez la commande « asadmin » suivante afin de créer le domaine « petstore » :

```
{yapson_home}> asadmin create-domain --adminport 8282 --user admin --savemasterpassword=true -
-instanceport 8080 petstore
```

La procédure de création du domaine « petstore » vous demande de définir les mots de passe « admin » et « master » :

```
...
Please enter the admin password>
Please enter the admin password again>
Please enter the master password [Enter to accept the default]:>
Please enter the master password again [Enter to accept the default]:>
...
```

Pour rester conforme au livre de référence, vous allez introduire les mots de passe suivants:

- **adminpwd** pour le « admin » password
- **masterpwd** pour le « master » password

Pour terminer, la procédure vous informe des numéros de ports assignés au domaine « petstore » :

```
...
Using port 8282 for Admin.
Using port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Domain being created with profile:developer, as specified by variable AS_ADMIN_PROFILE in
configuration file.
```

```
Security Store uses: JKS  
Domain petstore created.
```

À ce stade, le domaine "petstore" est physiquement créé sous le répertoire « domains » de GlassFish :

```
{yapson_home}> dir {glassfish_home}\domains\petstore  
...  
06/11/2008 23:59 <DIR> .  
06/11/2008 23:59 <DIR> ..  
06/11/2008 23:59 <DIR> addons  
06/11/2008 23:59 <DIR> applications  
06/11/2008 23:59 <DIR> autodeploy  
06/11/2008 23:59 <DIR> bin  
06/11/2008 23:59 <DIR> config  
06/11/2008 23:59 <DIR> docroot  
06/11/2008 23:59 <DIR> generated  
06/11/2008 23:59 <DIR> imq  
06/11/2008 23:59 <DIR> java-web-start  
06/11/2008 23:59 <DIR> jbi  
06/11/2008 23:59 <DIR> lib  
06/11/2008 23:59 <DIR> logs  
06/11/2008 23:59 500 master-password  
06/11/2008 23:59 <DIR> session-store  
1 File(s) 500 bytes  
15 Dir(s) 149.829.881.856 bytes free
```

Suppression d'un domaine sous GlassFish

Lors des phases de tests, on peut être amené à devoir supprimer un domaine.

Le script Ant « admin.xml » propose le target « delete-domain » pour supprimer un domaine de GlassFish.

On peut également directement utiliser l'interface d'administration de GlassFish comme illustré ci-dessous.

Tout d'abord, veuillez démarrer l'interface d'administration via la commande « asadmin » :

```
{yapson_home}> asadmin  
...  
Use "exit" to exit and "help" for online help.  
asadmin>
```

À l'invite de l'interface d'administration, vous pouvez supprimer le domaine en utilisant la commande « **delete-domain** » :

```
delete-domain --domaindir {glassfish_home}\domains petstore  
Domain petstore deleted.
```

Cette commande nécessite d'identifier :

- le répertoire « **domains** » de GlassFish

- le nom du domaine à supprimer (dans notre exemple, il s'agit du domaine « petstore »)

Pour sortir de l'interface « asadmin », il suffit d'introduire la commande « **exit** ».

LLV-C - Démarrer le domaine « petstore »

Le domaine « petstore » que vous venez de créer doit être démarré afin de poursuivre la configuration du serveur GlassFish.

L'application YapsON fournit le fichier « admin.xml » contenant des tâches qui permettent de démarrer et stopper le domaine.

Placez-vous dans le répertoire de l'application YapsON:

```
C:\> cd {yapson_home}
{yapson_home}>
```

Exécuter la tâche « **start-domain** » pour démarrer le domaine « petstore » :

```
{yapson_home}> ant -f admin.xml start-domain
```

Les traces doivent indiquer que le domaine « petstore » est correctement démarré :

```
...
[exec] Domain [petstore] is running [Sun Java System Application Server 9.1_02 (build b04-
fcs)] with its configuration and logs at: [C:\Program Files\glassfish-v2ur2\domains].
...
```

À ce stade, il est possible d'accéder à la console d'administration du domaine « petstore » au travers du lien suivant :

<http://localhost:8282>

L'écran de login suivant apparaît :



Figure 5-1: Écran de login

Sur base de notre configuration (voir [ici](#)), les informations suivantes doivent être introduites :

- **User Name** : admin
- **Password** : adminpwd

La console d'administration apparaît à l'écran :

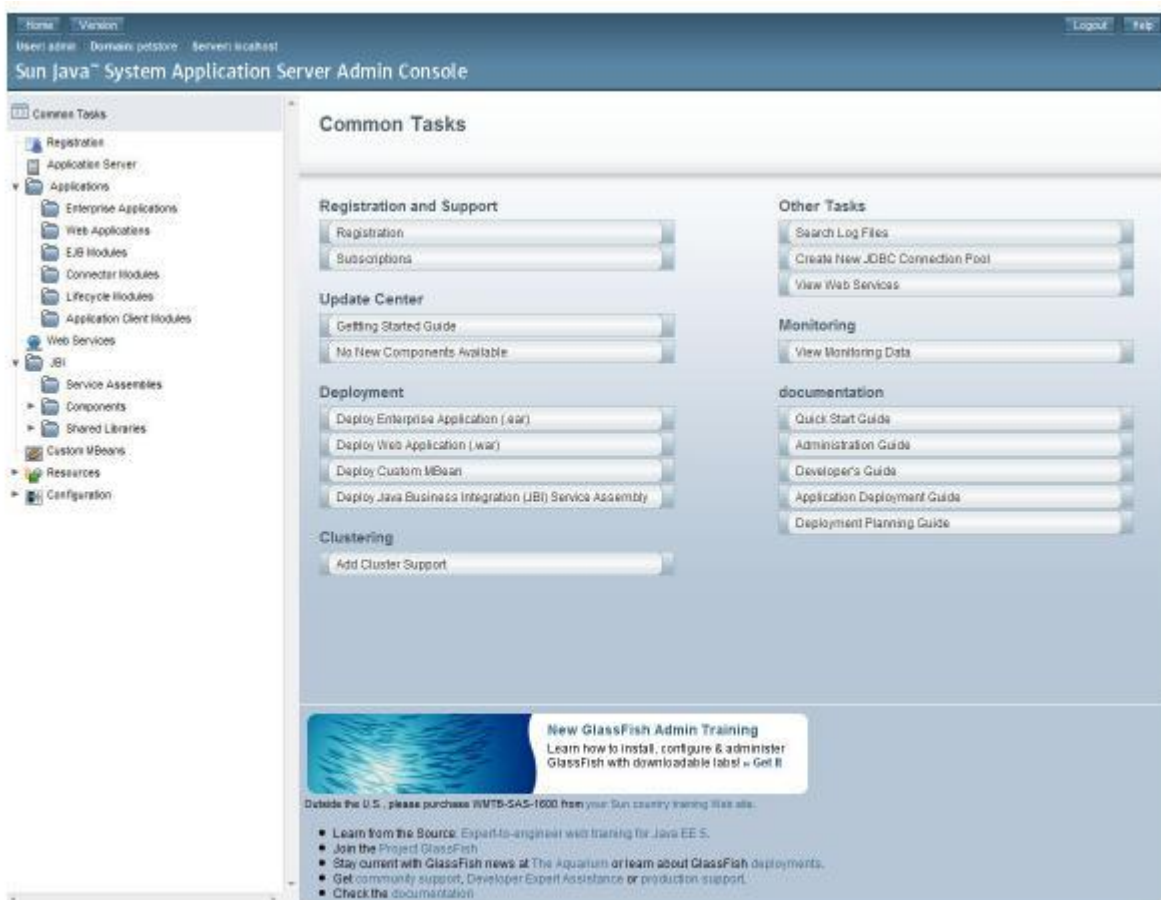


Figure 5-2: Console d'administration

LLV-D - Localisation de GlassFish

Par défaut, la console d'administration de GlassFish utilise la localisation définie sur le système.

Nous allons décrire comment changer la localisation sous GlassFish:

- Ouvrez la console d'administration accessible à l'adresse <http://localhost:8282>.
- Connectez-vous avec votre compte administrateur.
- Sélectionnez le noeud **Application Server**.
- Sélectionnez l'onglet **Advanced** et ensuite **Domain Attributes**.
- Changez la valeur de l'entrée **Locale** pour y définir le code langue que vous souhaitez utiliser.
- Sauvez vos modifications.

La modification du code langue nécessite un redémarrage du serveur GlassFish.

La figure suivante illustre l'application de l'anglais (**en-UK**) comme langue de travail :

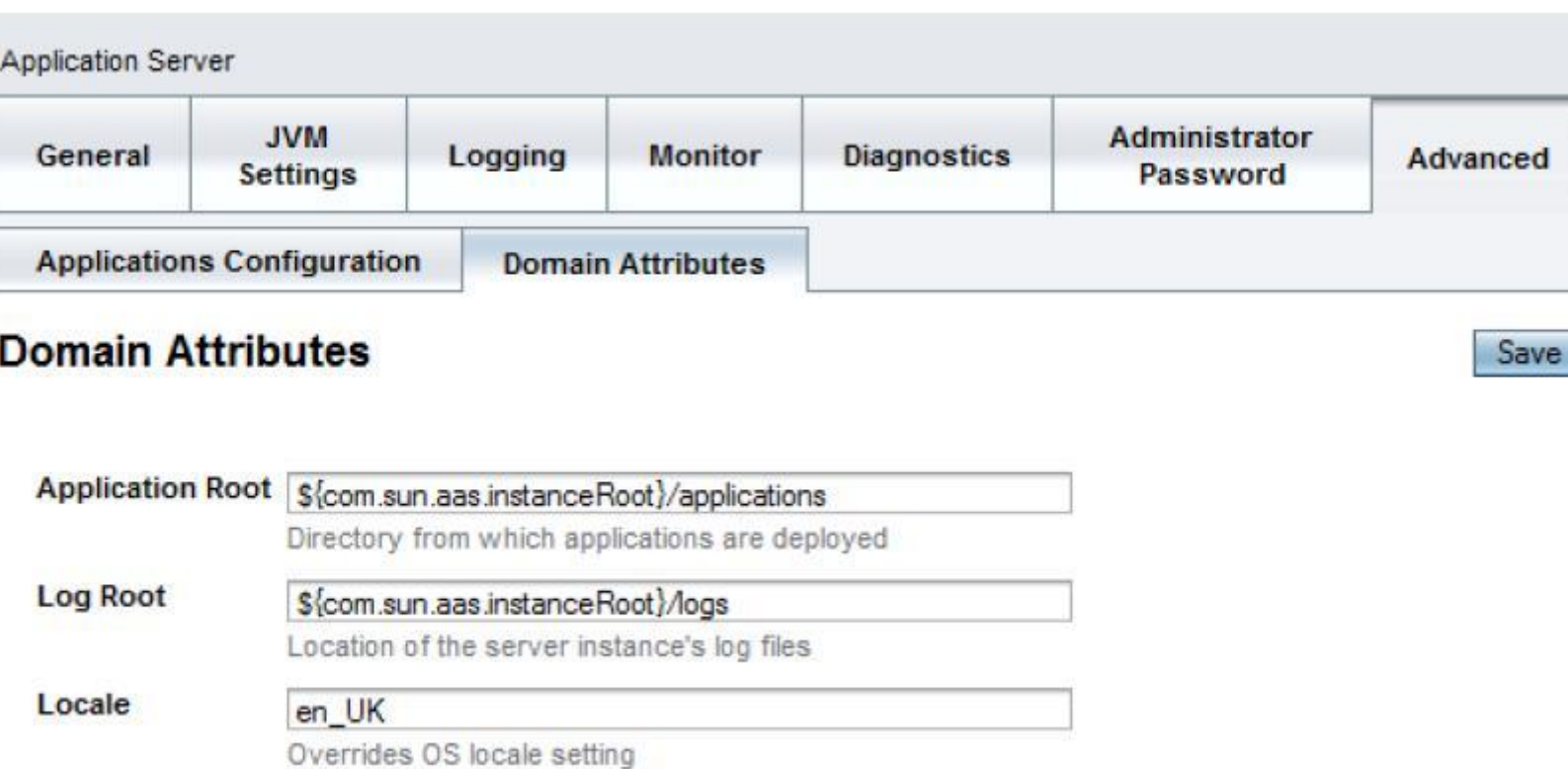


Figure 5-3: Localisation sous GlassFish

LLV-E - Créer les ressources Database

Les données de l'application Yaps sont maintenues dans une base de données de type Sun Java DB.

Sun Java DB et Apache Derby

Il n'existe pas de différence entre Sun Java DB et Apache Derby. Apache Derby est packagé par Sun sous le nom Java DB.

Cette section décrit le processus de création de l'ensemble des ressources requises pour accéder et manipuler la base de données « petstoreDB ».

Base de données sous MySQL

La base de données de l'application Yaps peut être définie sous MySQL.

L'auteur du livre de référence décrit comment modifier la configuration pour supporter MySQL :

<http://www.antoniogoncalves.org/xwiki/bin/view/Book/JavaEE5InstallMySQLFr>

Commencez par vous placer dans le répertoire de l'application YAPS:

```
C:\> cd {yapson_home}
{yapson_home}>
```

LLV-E-1 - Pool de connexions JDBC

Exécutez la tâche « **create-connection-pool** » pour créer le pool de connexions « petstorePool » :

```
{yapson_home}> ant -f admin.xml create-connection-pool
...
[exec] Command create-jdbc-connection-pool executed successfully.
...
```

En recherchant le statut « **BUILD SUCCESSFUL** », on s'assure que l'exécution de la tâche s'est correctement déroulée.

La tâche « **list-connexion-pool** » permet de vérifier que le pool de connexion JDBC a été correctement créé :

```
{yapson_home}> ant -f admin.xml list-connection-pool
...
[exec] __CallFlowPool
[exec] __TimerPool
[exec] DerbyPool
[exec] petstorePool
[exec] Command list-jdbc-connection-pools executed successfully.
[exec]
[exec] C:\Projects\YapsON>endlocal
BUILD SUCCESSFUL
```

Une autre manière consiste à se rendre sur la console d'administration de GlassFish.

Le pool de connexions JDBC se trouve à l'emplacement suivant :

Resources -> JDBC -> Connection Pools -> petstorePool

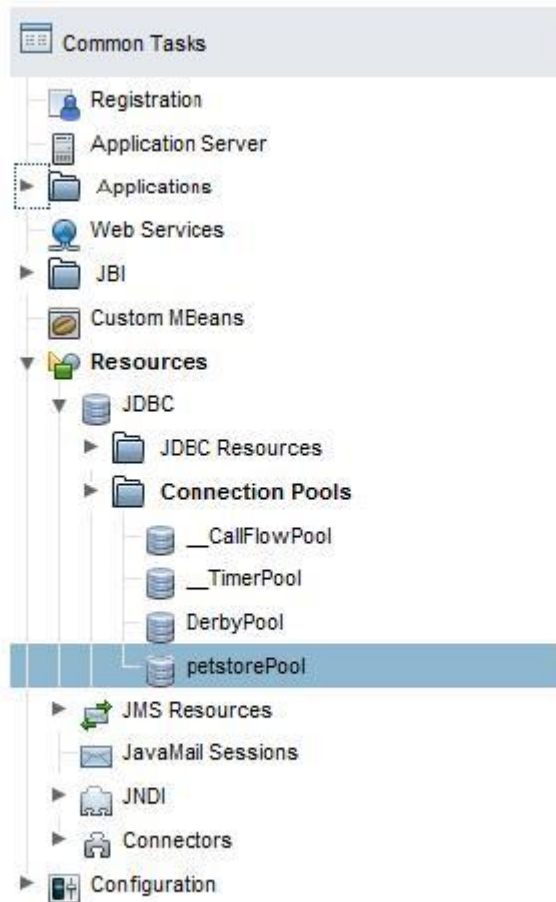


Figure 5-4: Pool de connexions JDBC

N'hésitez pas à inspecter les propriétés du pool de connexion « **petstorePool** ».

Les propriétés définies dans l'onglet « **Additional Properties** » seront utilisées plus tard lors de la création de la base de données :

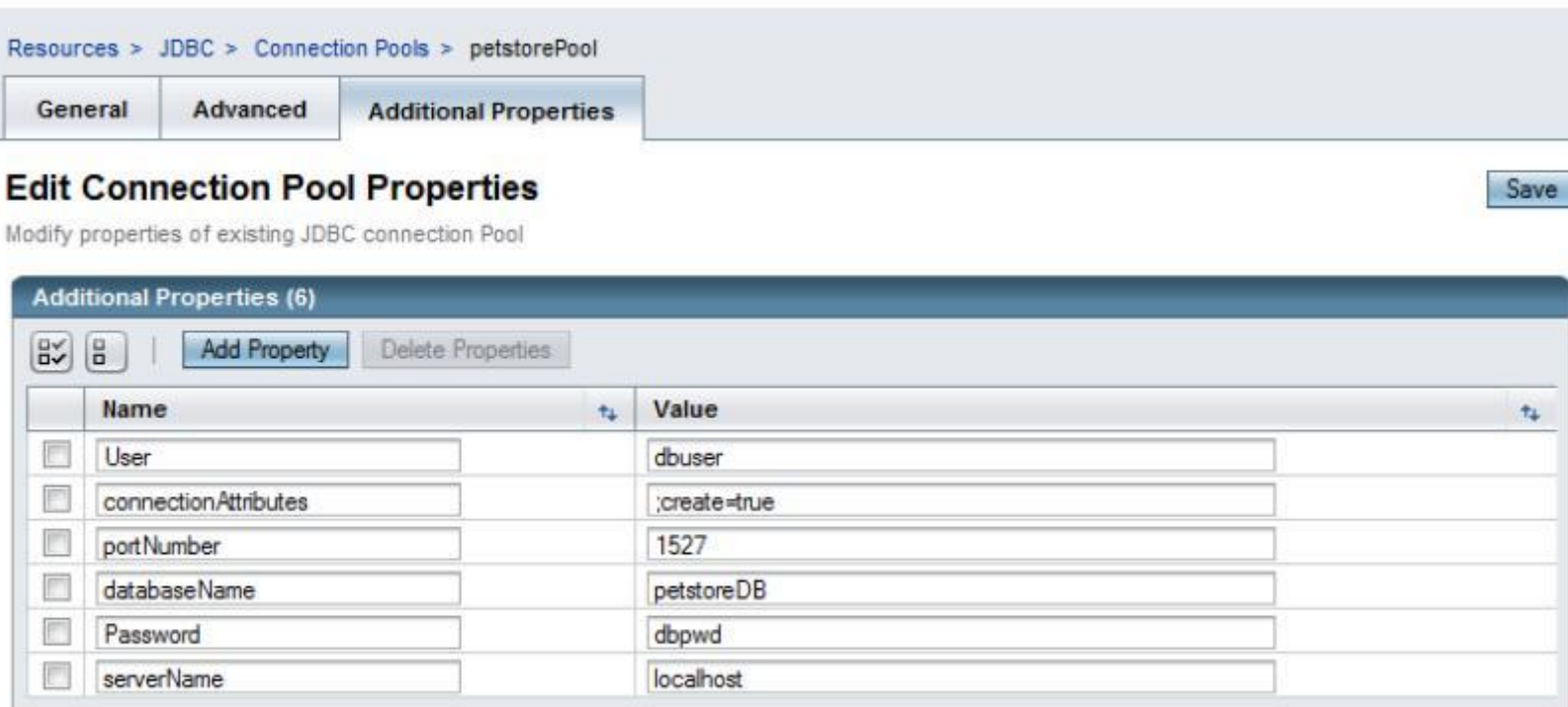


Figure 5-5: Propriétés du pool de connexion

LLV-E-2 - Démarrer le serveur Derby

Avant de poursuivre le processus, il est nécessaire de démarrer le serveur de base de données Derby.

Exécutez la tâche « **start-db** » pour démarrer Derby :

```
{yapson_home}> ant -f admin.xml start-db
...
[exec] Starting database in the background.
[exec] Log redirected to c:\Program Files\glassfish-v2ur2/javadb\derby.log.
[exec] Command start-database executed successfully.
...
```

LLV-E-3 - Créer la base de données

La base de données est créée en exécutant un « ping » vers le pool de connexion.

Exécutez la tâche « **ping-connection-pool** » pour créer la base de données « petstoreDB » :

```
{yapson_home}> ant -f admin.xml ping-connection-pool
...
[exec] Command ping-connection-pool executed successfully.
...
```

La base de données « **petstoreDB** » est maintenant créée sous le répertoire « **javadb** » de GlassFish :

```
{glassfish_home}\javadb\petstoreDB
```

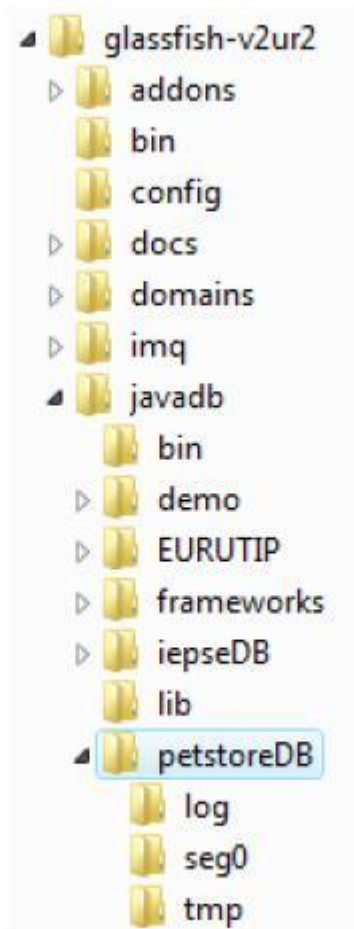


Figure 5-6: Base de données petstoreDB

LLV-E-4 - Créer la source de données

La source de données (**data source**) permet à l'application d'accéder à la base de données.

Exécutez la tâche « **create-datasource** » pour créer la source de données « petstoreDS » :

```
{yapson_home}> ant -f admin.xml create-datasource
...
[exec] Command create-jdbc-resource executed successfully.
...
```

Utilisez la tâche « **list-datasource** » pour vérifier que la data source a bien été créée :

```
{yapson_home}> ant -f admin.xml list-datasource
...
[exec] jdbc/___TimerPool
[exec] jdbc/___CallFlowPool
```

```
[exec] jdbc/_default
[exec] jdbc/petstoreDS
[exec] Command list-jdbc-resources executed successfully.
```

Une autre manière consiste à se rendre sur la console d'administration de GlassFish.

Le pool de connexions JDBC se trouve à l'emplacement suivant :

Resources-> JDBC-> JDBC Resources-> jdbc/petstoreDS

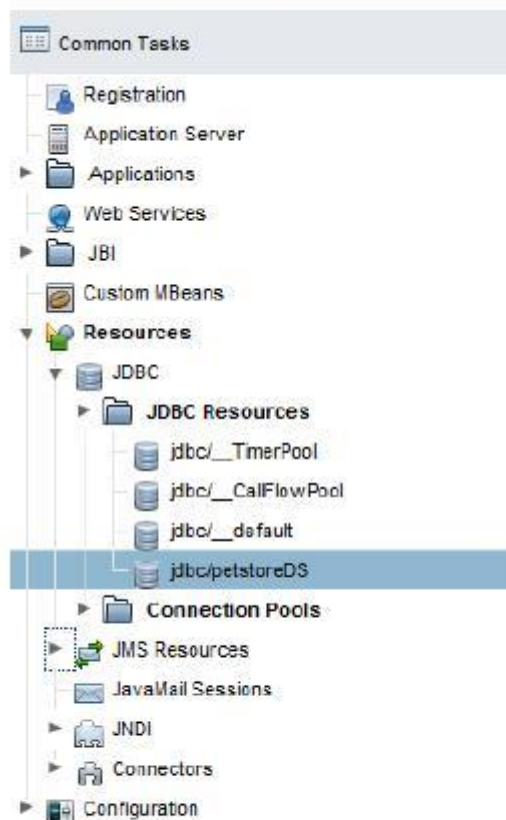


Figure 5-7: JDBC Resources

LLV-F - Créer les ressources JMS

Vous allez maintenant créer les ressources **JMS** que sont la file d'attente (**JMS topic**) et la fabrique de connexion (Connection Factory).

LLV-F-1 - Créer la fabrique de connexion

Exécutez la tâche « **create-jms-connection-factory** » pour créer la fabrique de connexion JMS (Connection Factory) « **jdbc/petstoreConnectionFactory** » :

```
{yapson_home}> ant -f admin.xml create-jms-connection-factory
...
```

```
[exec] Command create-jms-resource executed successfully.]
]></code>
  <paragraph>
    &#160;
  </paragraph>
</section>
<section id="V-F-2">
  <title>Créer la file d'attente</title>
  <paragraph>
    Exécutez la tâche « <b>create-jms-topic</b> » pour créer la file d'attente (<b>JMS</b> topic) «
jms/topic/order
  »:
  </paragraph>
<code language="other"><![CDATA[
{yapson_home}> ant -f admin.xml create-jms-topic
...
[exec] Command create-jms-resource executed successfully.
```

LLV-F-3 - Vérification

Utilisez la tâche « **list-jms-resources** » pour énumérer les ressources JMS définies sous GlassFish :

```
{yapson_home}> ant -f admin.xml list-jms-resources
...
[exec] jms/topic/order
[exec] jms/petstoreConnectionFactory
[exec] Command list-jms-resources executed successfully.
BUILD SUCCESSFUL
```

Vous devez vous assurer que la file d'attente « jms/topic/order » et la fabrique de connexion « jms/petstoreConnectionFactory » ont bien été créées.

Une autre manière consiste à se rendre sur la console d'administration de GlassFish.

Les ressources JMS se trouvent aux emplacements suivants :

- Resources -> JMS Resources -> Connection Factories
- Resources -> JMS Resources -> Destination Resources

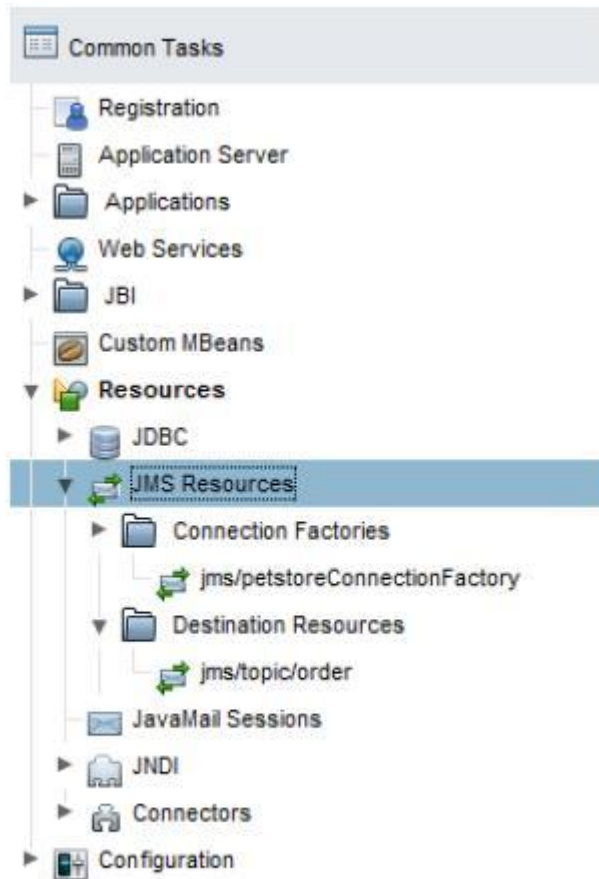


Figure 5-8: JMS Resources

LLV-G - Créer les loggers

Les loggers sont utilisés par l'application pour générer des traces (messages d'erreur, informations de debug, audit, etc.).

Exécutez la tâche « **set-loggers** » pour créer les loggers de chaque composant (petsore, barkbank, petex) :

```
{yapson_home}> ant -f admin.xml set-loggers
...
[exec] server.log-service.module-log-levels.property.com.yaps.petstore = FINEST
...
[exec] server.log-service.module-log-levels.property.com.barkbank = FINEST
...
[exec] server.log-service.module-log-levels.property.com.petex = FINEST
...
```

La console d'administration de GlassFish vous permet de vérifier la présence des loggers.

Rendez-vous à l'emplacement suivant :

Application Server -> Logging -> Log Level

La page « Log Levels » présente la section « Additional Properties » dans laquelle sont définis les loggers :

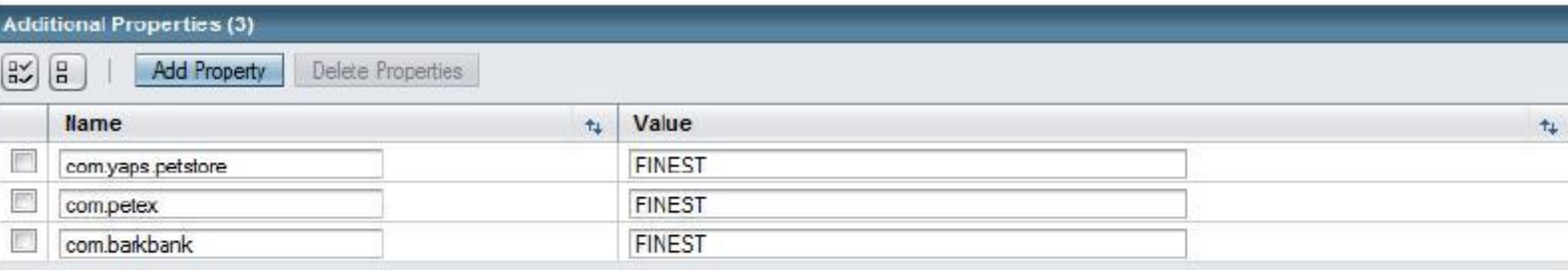


Figure 5-9: Loggers de Yaps

LLV-H - Fin de la configuration

À ce stade, nos environnements GlassFish et Derby sont configurés pour accueillir l'application Yaps.

Le reste du processus de configuration va se poursuivre sous NetBeans.

Vous allez maintenant stopper le serveur de base de données Derby et le domaine « petstore ». Ceux-ci seront démarrés en temps voulu sous NetBeans.

LLV-G-1 - Stopper le serveur Derby

Exécutez la tâche Ant « **stop-db** » pour stopper le serveur de base de données Derby :

```
{yapson_home}> ant -f admin.xml stop-db
...
[exec] Command stop-database executed successfully.
```

LLV-G-2 - Stopper le domaine « petstore »

Exécutez la tâche Ant « **stop-domain** » pour stopper le domaine applicatif « petstore » :

```
{yapson_home}> ant -f admin.xml stop-domain
...
[exec] Domain petstore stopped.
```

VI - Configurer NetBeans

Avant de procéder au portage des projets Yaps sous NetBeans, il est nécessaire de configurer l'IDE.

Veillez démarrer NetBeans comme décrit dans la section 3.2.3.

LLVI-A - Serveur « Petstore »

LLVI-A-1 - Ajout du serveur « Petstore »

Vous allez procéder à l'ajout d'un serveur de type GlassFish v2 qui sera lié au domaine « petstore » que vous avez précédemment créé.

Par défaut, le serveur GlassFish v2 défini sous NetBeans est configuré avec le domaine « domain1 ».

Sélectionnez l'onglet « Services » et ensuite un clic droit sur le noeud « Servers » :

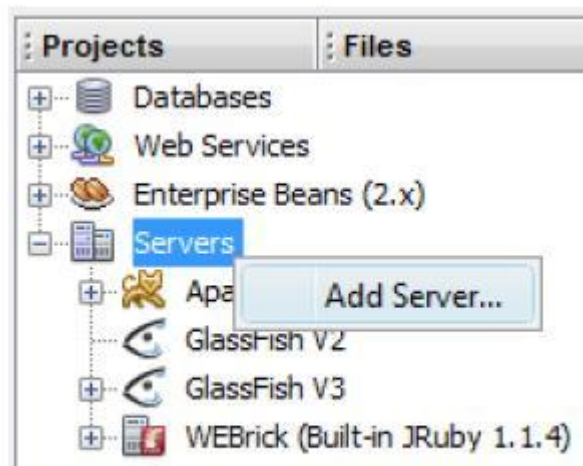


Figure 6-1: Liste des Services

Le menu contextuel apparaît et vous permet d'ajouter un nouveau serveur. Cliquez sur l'option « Add Server ».

LLVI-A-1-a - Type de serveur

La première étape du wizard consiste à définir le type de serveur d'applications.

Dans notre cas, il s'agit de « GlassFish v2 » :

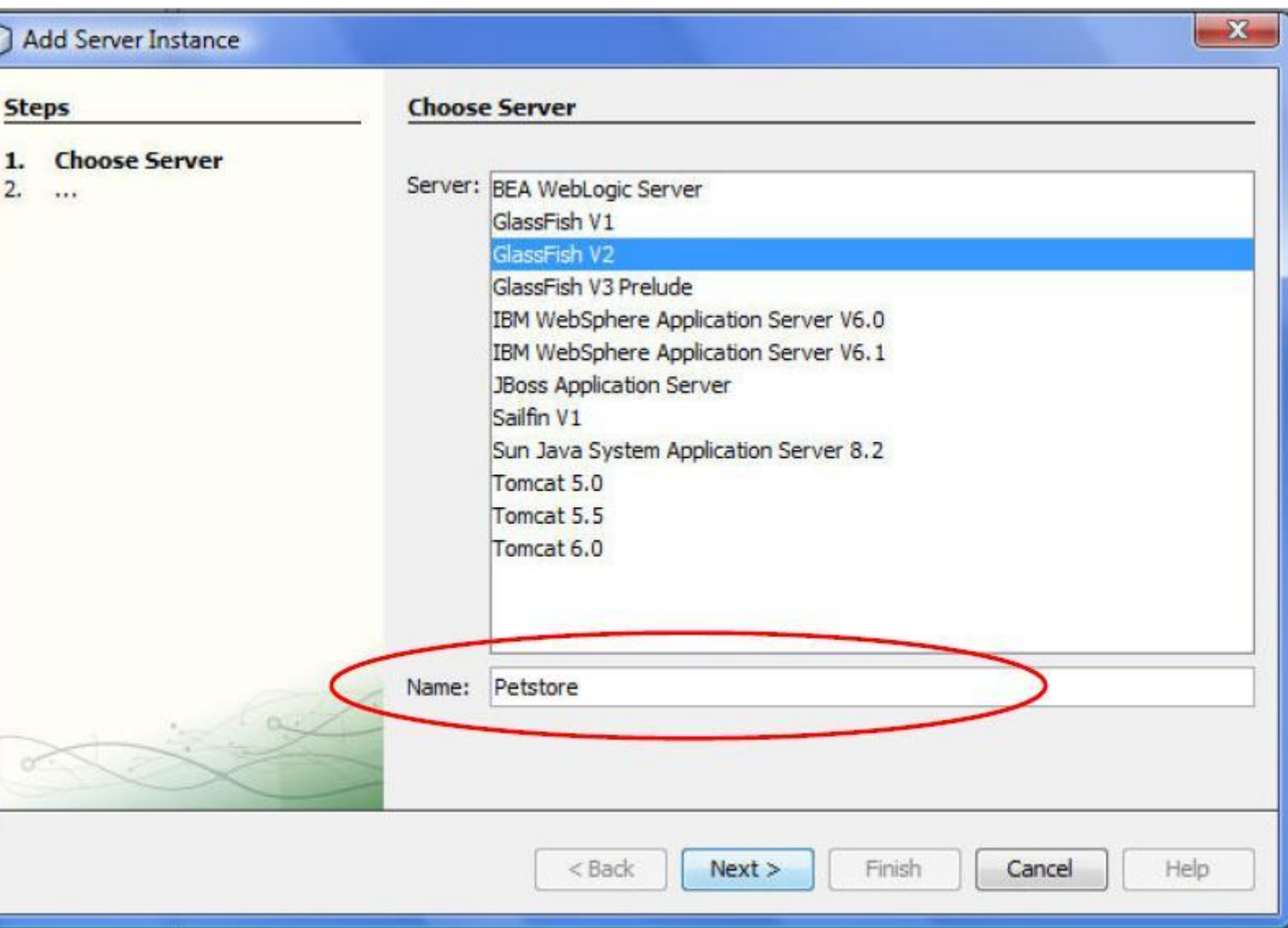


Figure 6-2: Type de serveur

Dans le champ « Name », veuillez nommer le serveur « **Petstore** ».

Ce nom est arbitraire et ne sert qu'à identifier le serveur parmi la liste définie sous NetBeans.

Cliquez sur « Next » pour continuer.

LLVI-A-1-b - Sélection du domaine

Dans cette étape, vous allez lier notre domaine « petstore » au serveur « Petstore ». Sélectionnez l'option « Register Local Default Domain » et ensuite le domaine « petstore » :

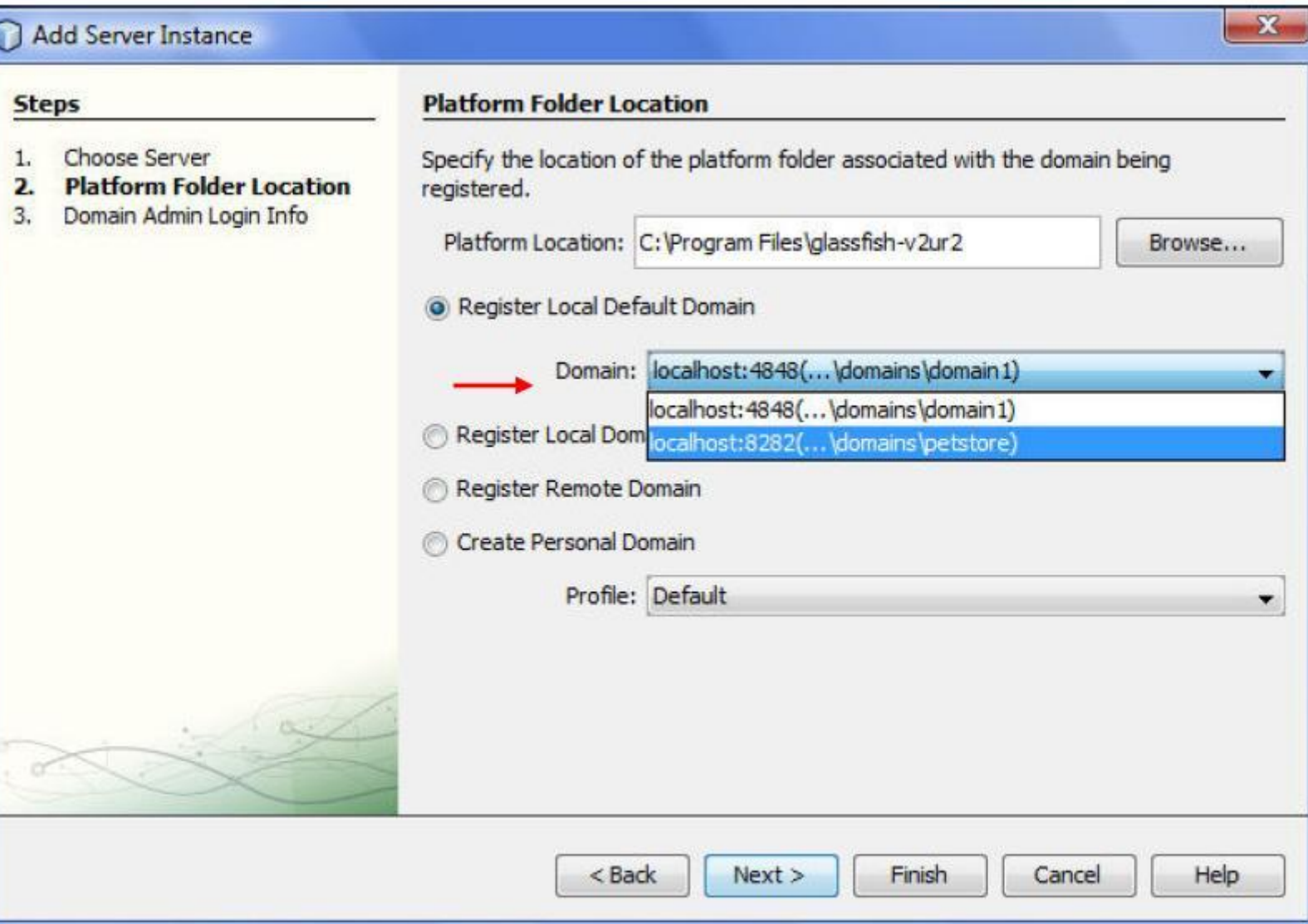


Figure 6-3: Choix du domaine

Cliquez sur « Next » pour continuer.

LLVI-A-1-c - Profil de l'administrateur

Cette étape consiste à identifier le profil de l'administrateur :

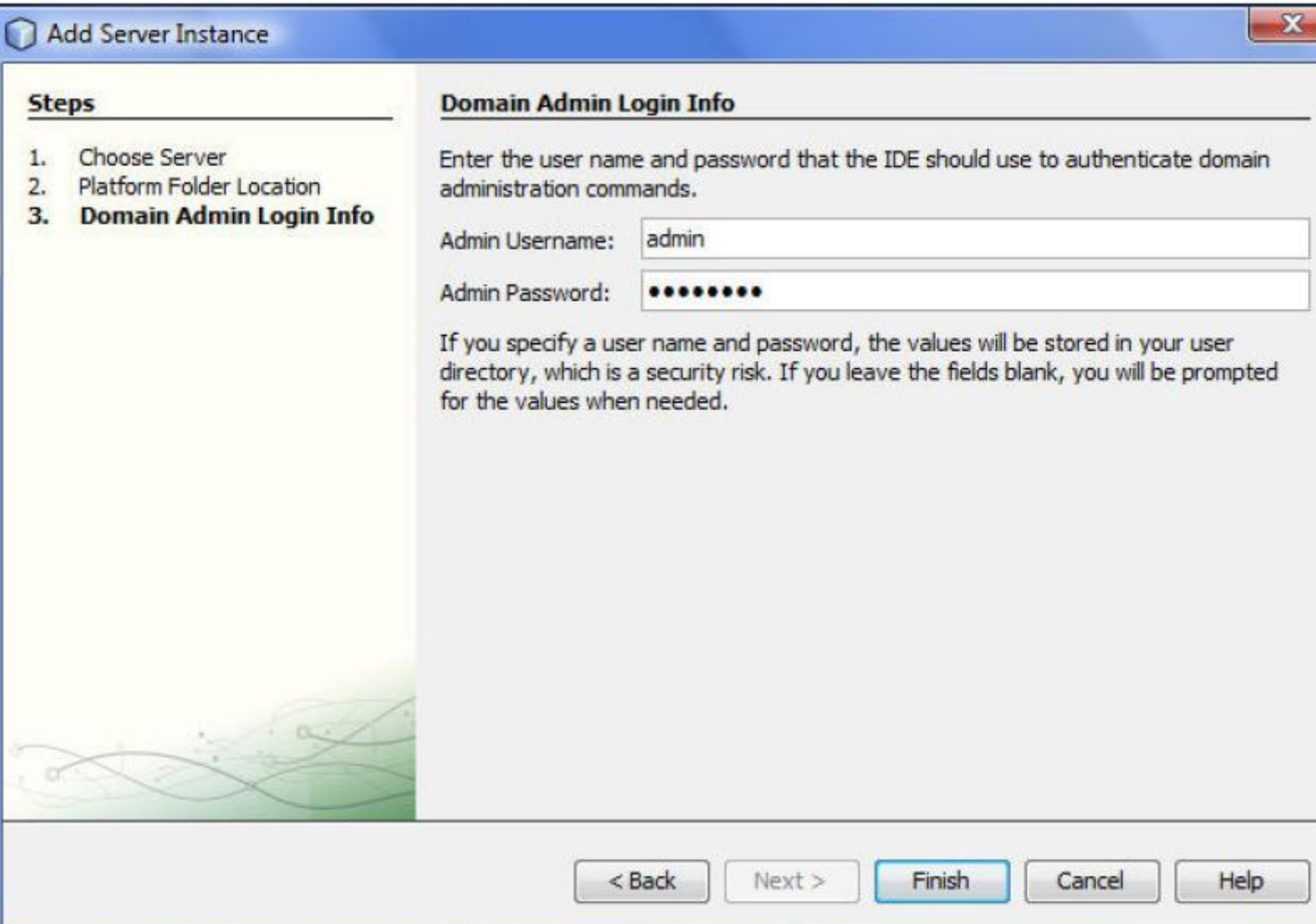


Figure 6-4: Paramètres de l'administrateur

Veillez à introduire le mot de passe que vous avez défini plus haut lors de la création du domaine (voir [ici](#)).

Cliquez sur le bouton « Finish » pour terminer le processus de création du serveur applicatif.

LLVI-A-2 - Liste des serveurs

La liste des serveurs fait maintenant apparaître notre serveur « Petstore » :

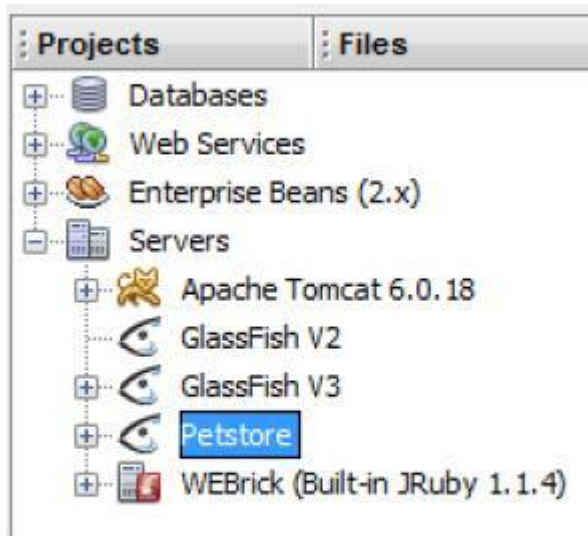


Figure 6-5: Liste des serveurs

LLVI-A-3 - Affichage des traces

Avant de démarrer le serveur « Petstore », vous devez vous assurer que les traces seront visibles. Un click droit sur le serveur « Petstore » permet d'activer l'affichage des traces. Cliquez sur l'option « View Server Log » du menu contextuel :

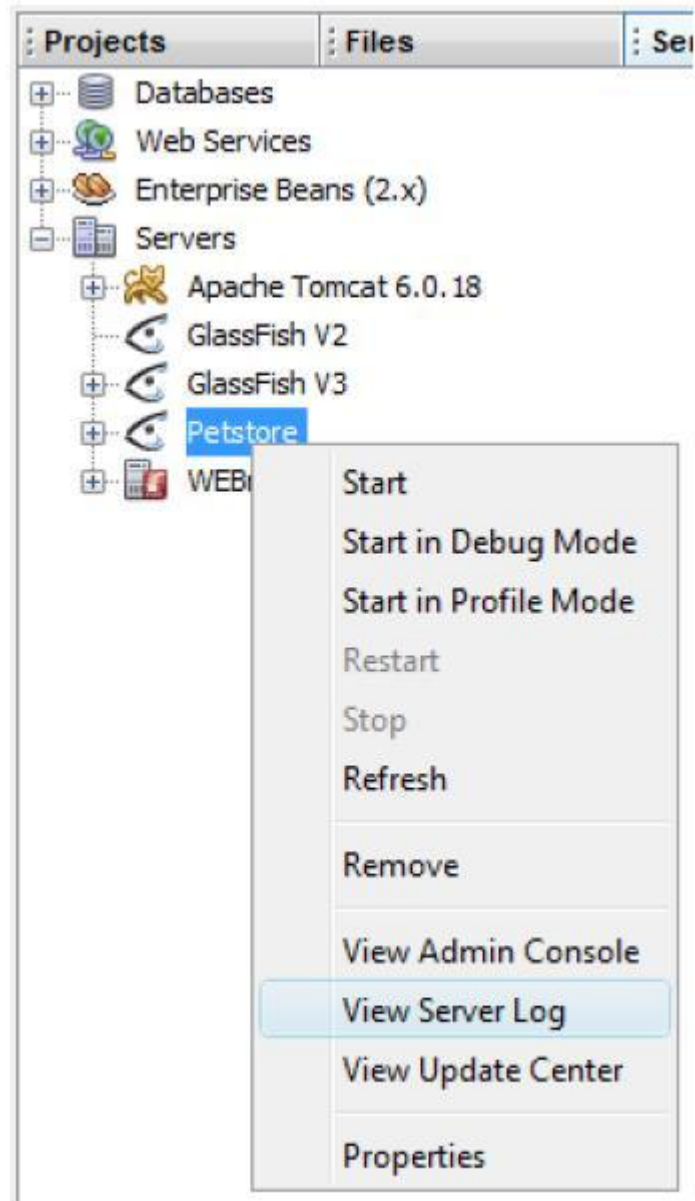


Figure 6-6: Afficher les traces

LLVI-A-4 - Démarrer le serveur Petstore

Vous pouvez maintenant démarrer le serveur « Petstore » en sélectionnant l'option « Start » du menu contextuel :

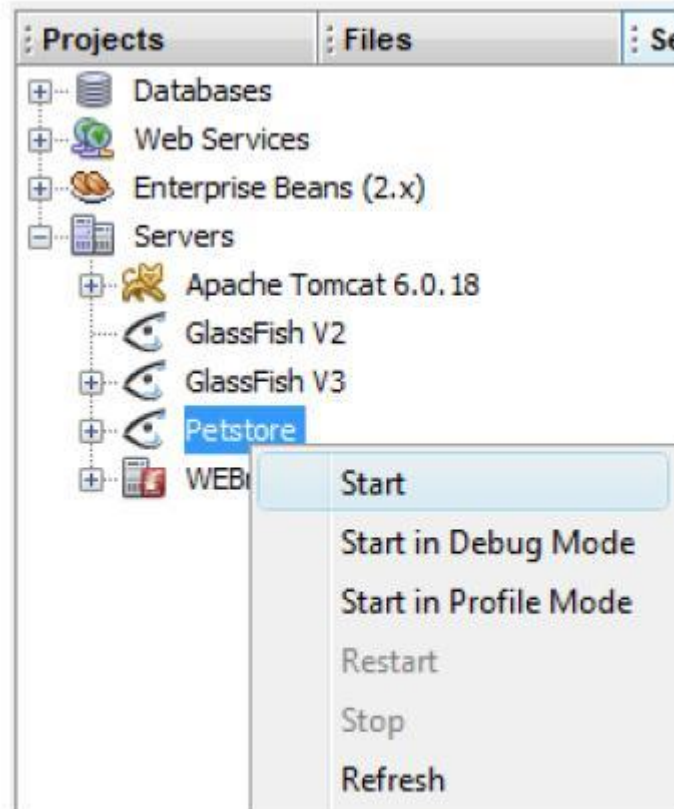


Figure 6-7: Démarrer le serveur

La fenêtre « Output » de NetBeans permet de suivre le processus de démarrage du domaine « petstore » :

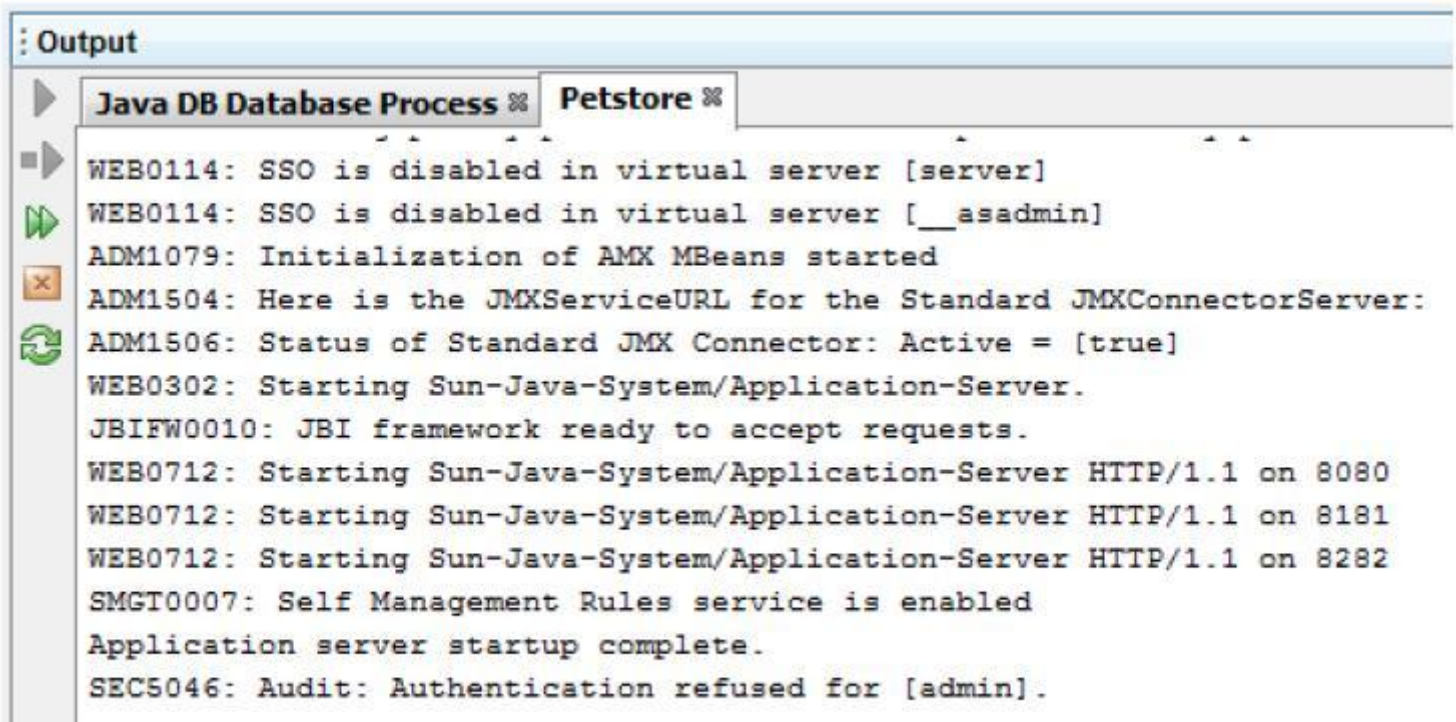


Figure 6-8: Fenêtre "Output"

Fenêtre « Output » : Affichage

Si la fenêtre « Output » n'est pas visible, activez là au travers du menu :
Window -> Output -> Output

Vous constaterez que la fenêtre « Output » contient deux sous-fenêtres :

- **Petstore:** affiche les traces du domaine « petstore »
- **Java DB Database Process:** affiche les traces de la base de données Derby

Si l'on explore la sous-fenêtre « Java DB Database Process », on constate que le démarrage du domaine « petstore » a automatiquement démarré le serveur de base de données Derby :

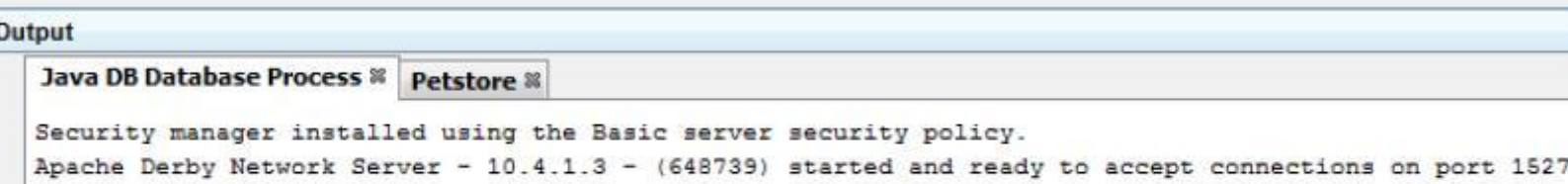


Figure 6-9: Démarrage automatique de Derby

Fenêtre « Output » : Menu contextuel

La fenêtre de trace offre un menu contextuel accessible via un click droit :

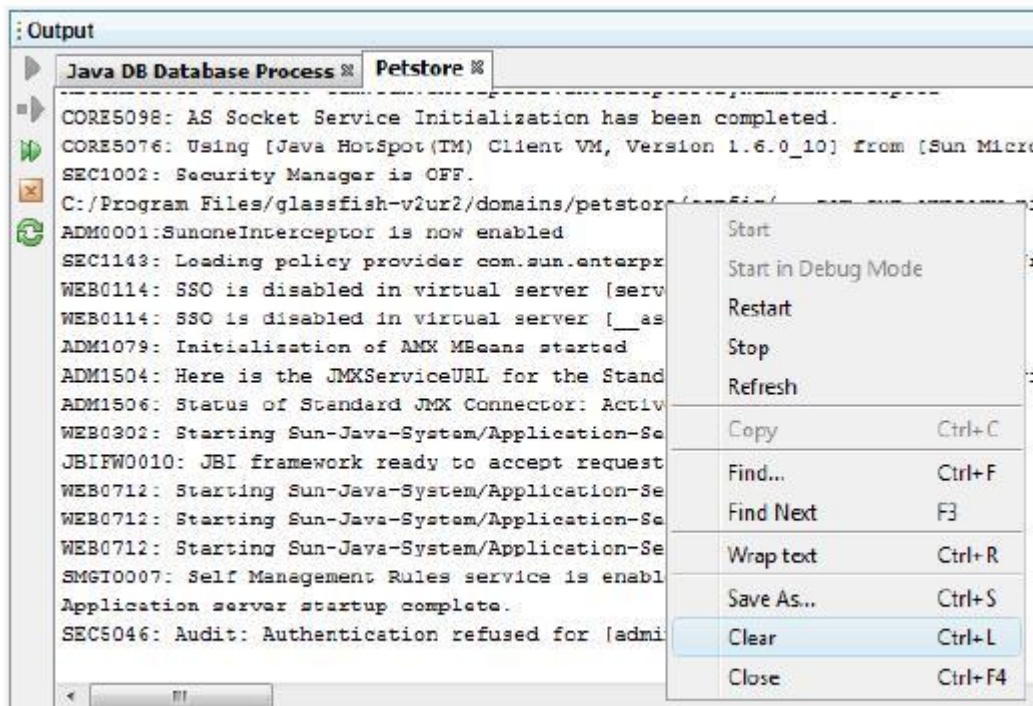


Figure 6-10: Menu contextuel de la fenêtre Output

En plus de proposer les options d'arrêt (Stop) ou de démarrage du serveur (Start), il est également possible de rechercher une chaîne de caractères (Find) ou d'effacer le contenu de la fenêtre (Clear).

LLVI-A-5 - Explorer les ressources

Une exploration du serveur « Petstore » permet de découvrir les éléments déployés dans le domaine tels que les applications, les ressources JMS ou les ressources JDBC :

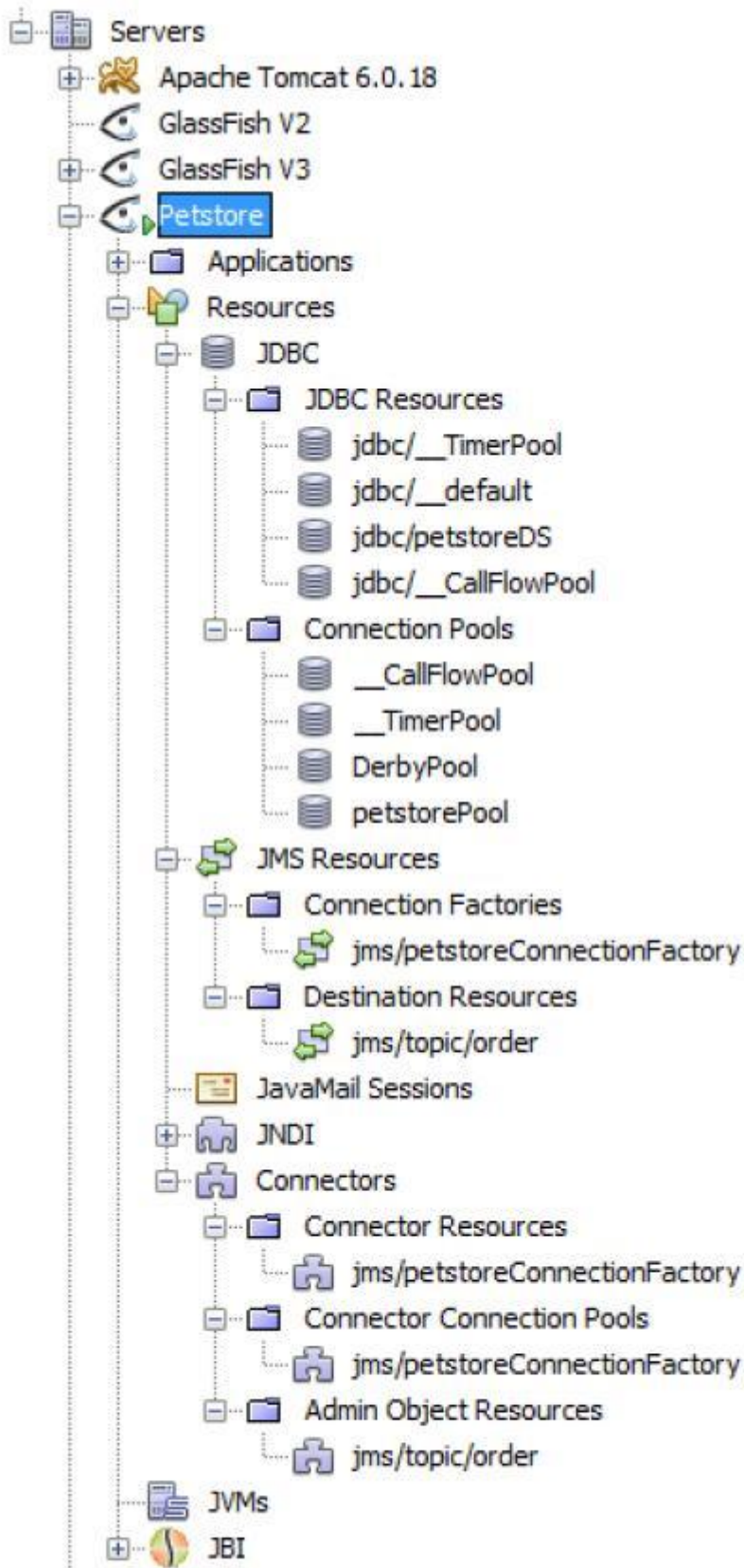


Figure 6-11: Exploration du serveur Petstore

LLVI-A-6 - Accès à la console d'administration

L'option « View Admin Console » du menu contextuel du serveur permet d'accéder à la console d'administration :

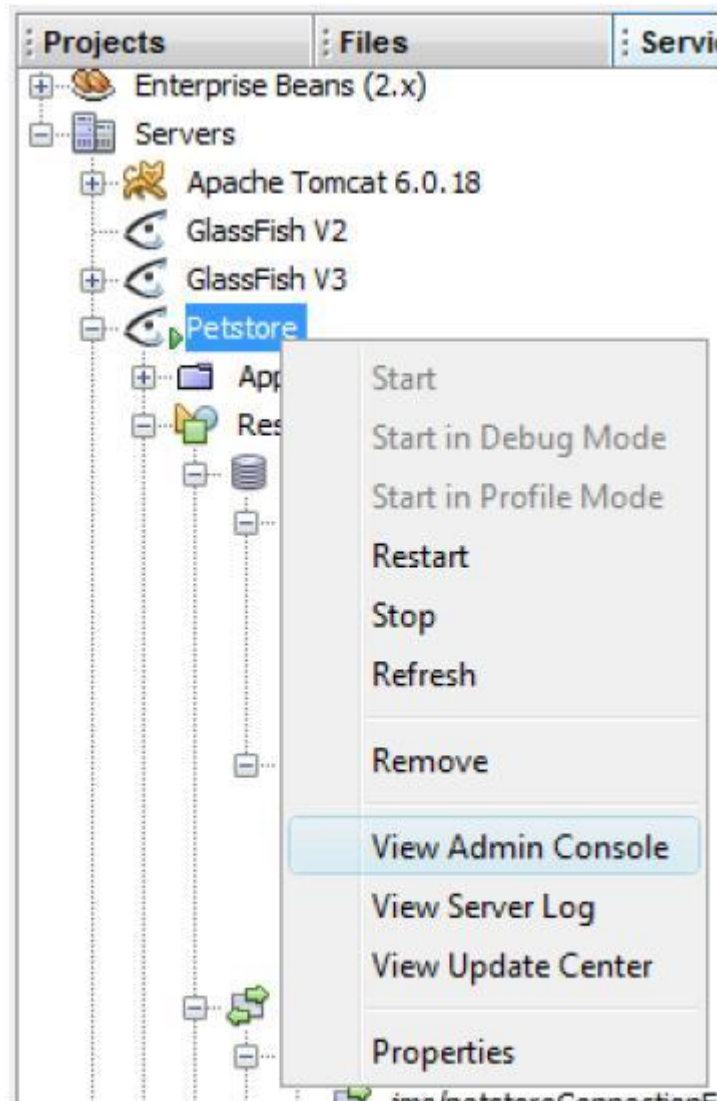


Figure 6-12: Accès à la console d'administration

NetBeans utilise le browser pour ouvrir la console d'administration : <http://localhost:8282>

LLVI-B - Changer de browser

En modifiant les options de NetBeans, il est possible de changer le browser par défaut. Sélectionnez le menu « Tools -> Options » :

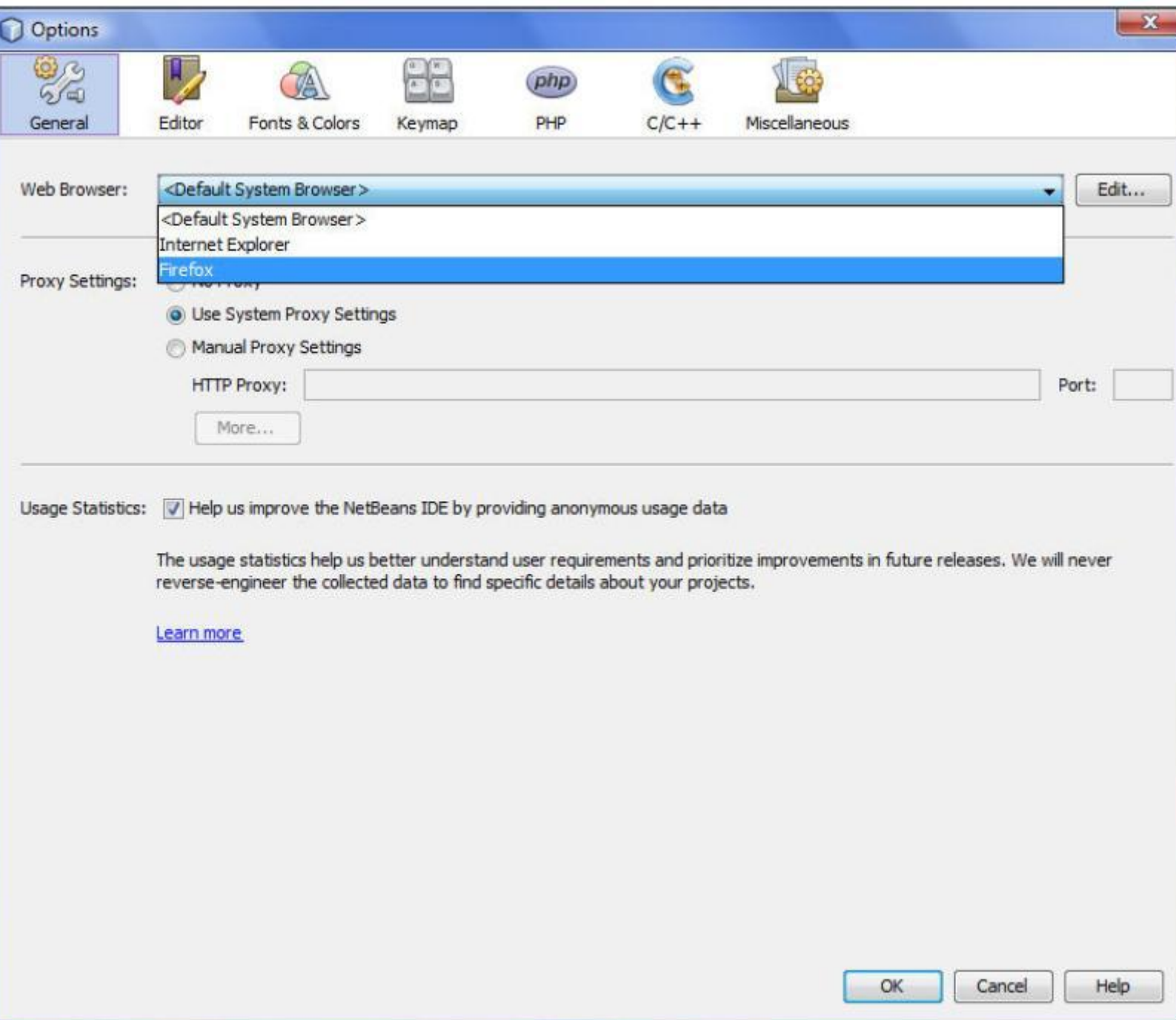


Figure 6-13: Options de NetBeans

Dans les options générales (General), vous êtes libre d'opter pour un browser et même d'en définir en cliquant sur le bouton « Edit » :

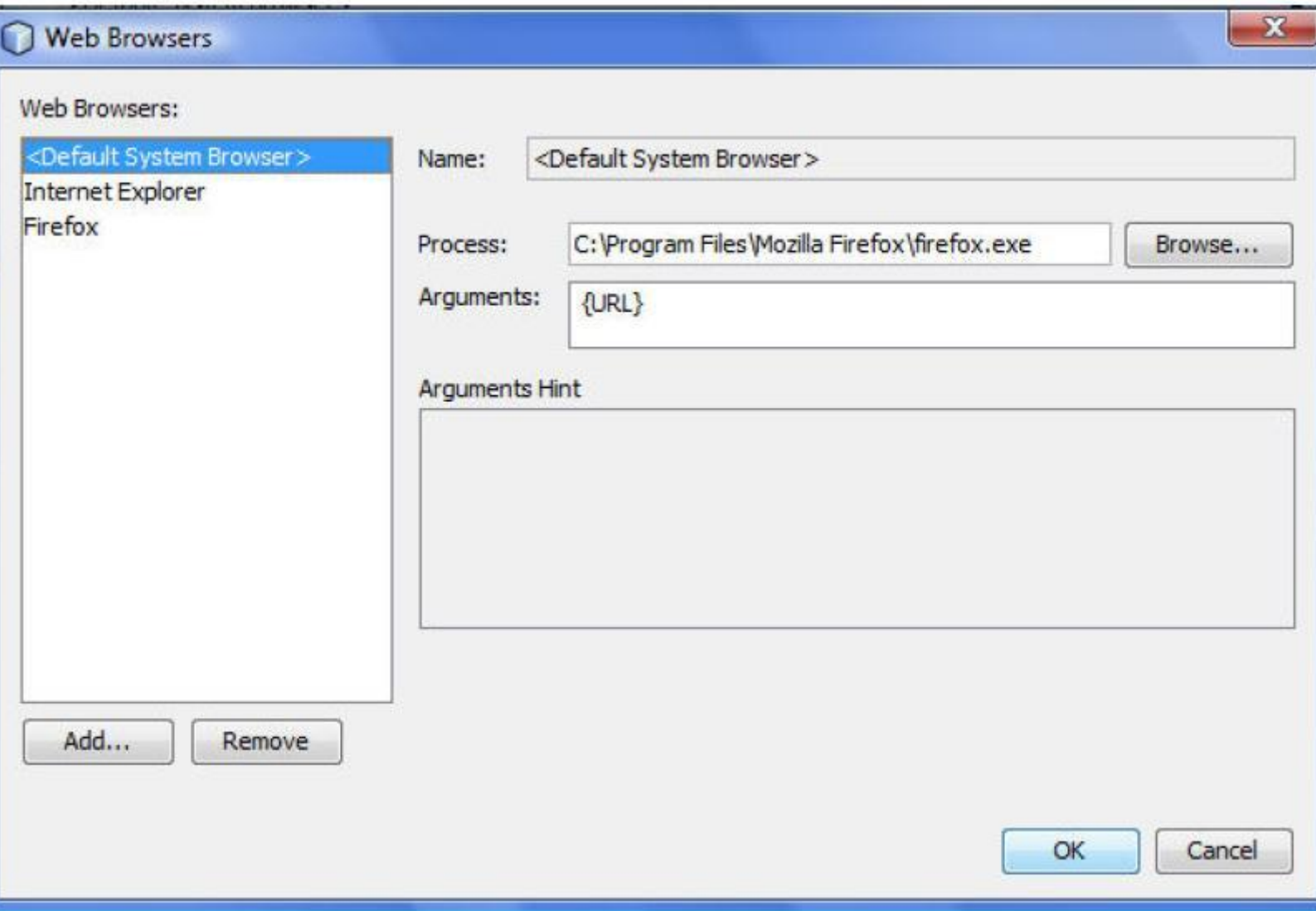


Figure 6-14: Options du browser

On constate qu'il est notamment possible d'ajouter les définitions d'autres browsers.

LLVI-C - Base de données

Le noeud « Databases » de l'onglet « Services » répertorie les serveurs de base de données et les connexions:

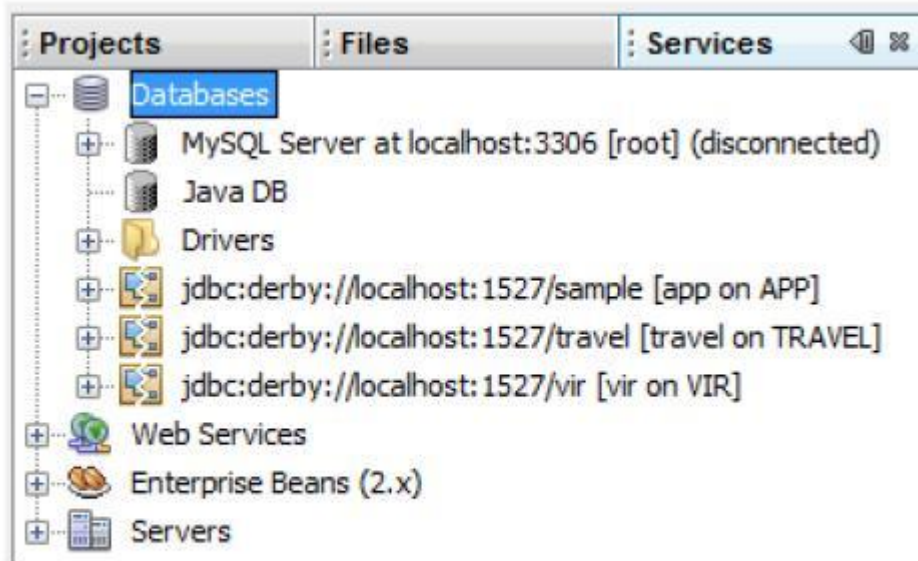


Figure 6-15: Databases sous NetBeans

LLVI-C-1 - Arrêt et démarrage du serveur de base de données

Vous avons vu que le démarrage du serveur "Petstore" démarrait automatiquement le serveur de base de données Java DB.

Il peut être utile de démarrer et stopper manuellement le serveur Java DB.

Ces opérations sont accessibles au travers du menu contextuel lié au noeud Java DB :

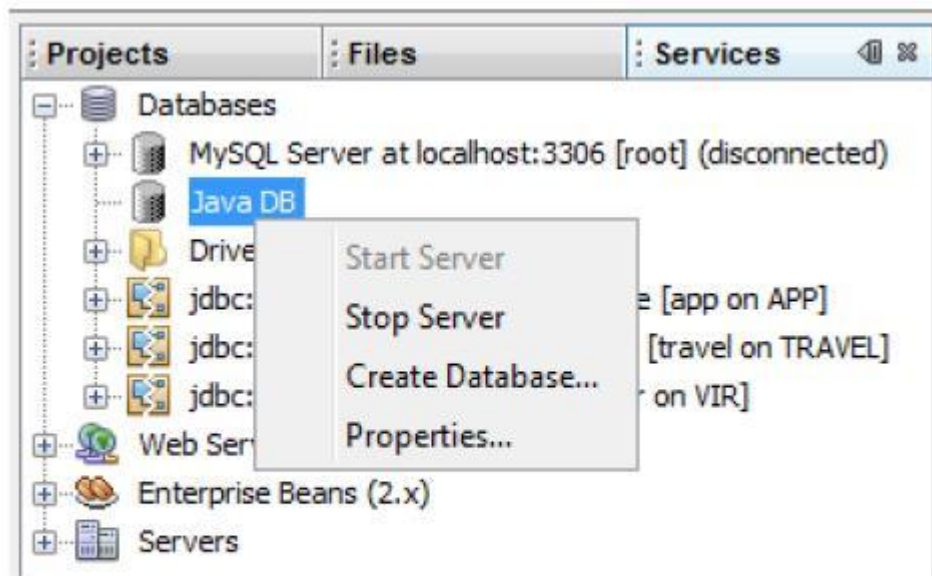


Figure 6-16: Menu contextuel de Java DB

LLVI-C-2 - Vérifier les propriétés

Pour vérifier les propriétés définies sur un serveur de base de données, sélectionnez l'option « Properties » du menu contextuel :

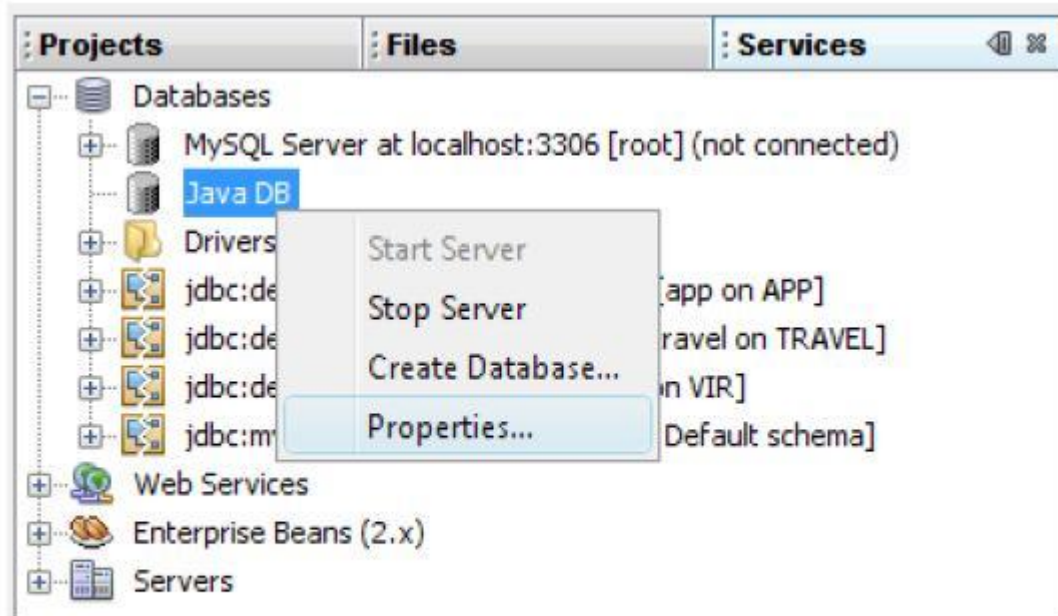


Figure 6-17: Accès aux propriétés de Java DB

Le menu « Properties » permet d'accéder aux propriétés définies par NetBeans pour le serveur de base de données.

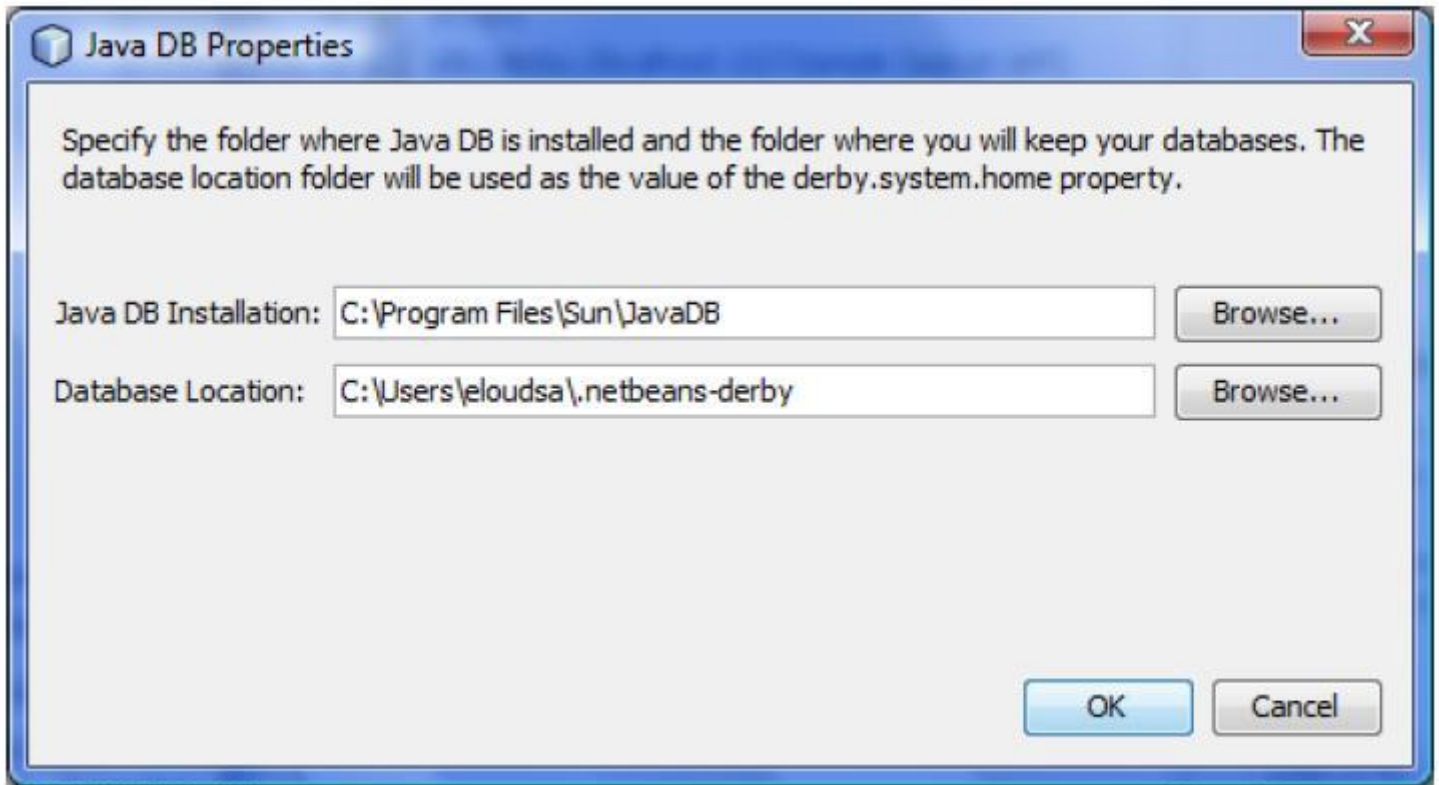


Figure 6-18: Propriétés de Java DB

Les propriétés de Java DB vous apprennent que NetBeans utilise des répertoires autres que ceux attendus par notre configuration.

Pour être conforme à notre déploiement (voir [ici](#)), vous devez vérifier et modifier le cas échéant les répertoires suivants :

- Java DB Installation : **%GLASSFISH_HOME%\javadb**
- Database Location : **%GLASSFISH_HOME%\javadb**

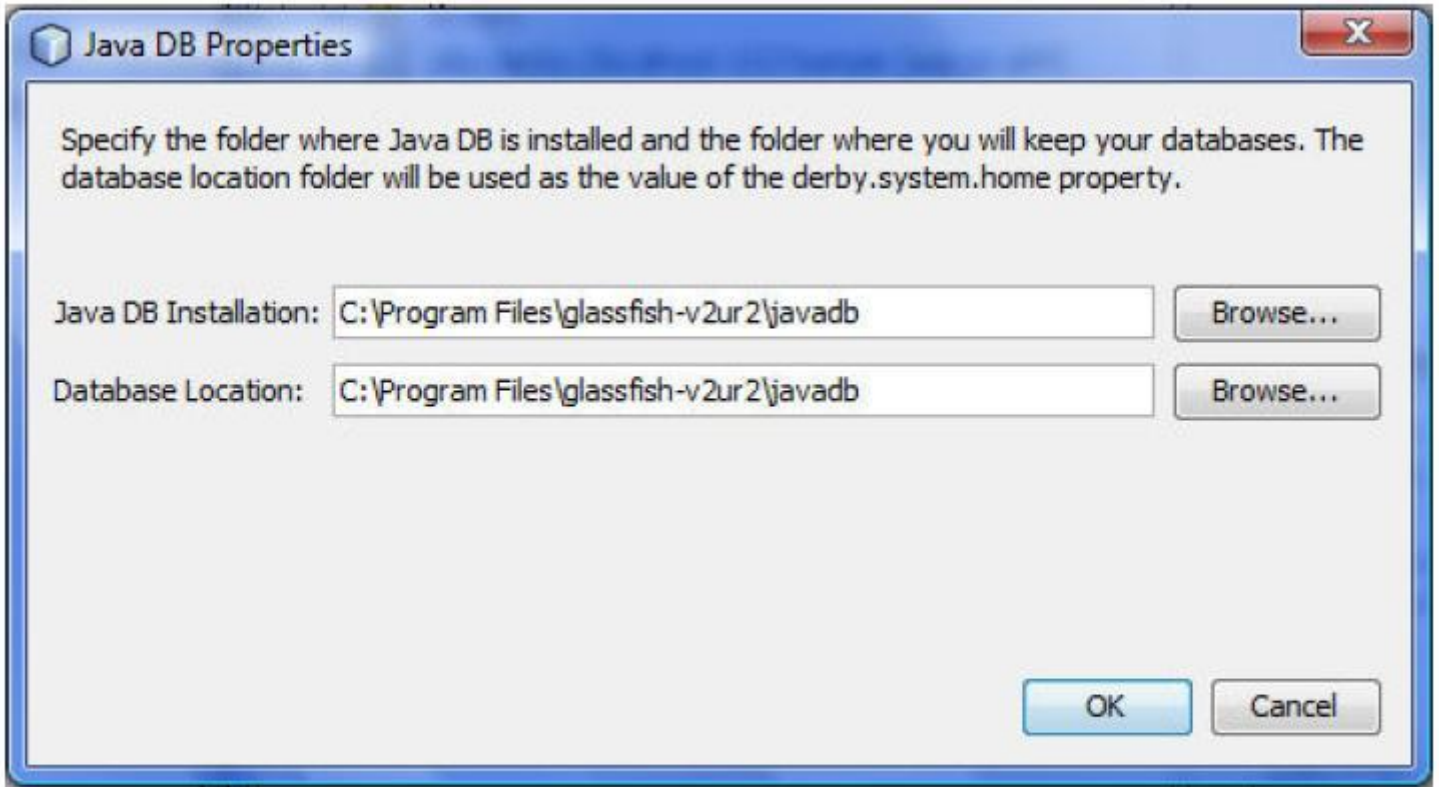


Figure 6-19: Modification des propriétés

Appuyez sur le bouton "OK" pour valider la modification.

Modifications de propriétés de Java DB

La modification des propriétés de Java DB entraîne automatique l'arrêt du serveur de base de données:

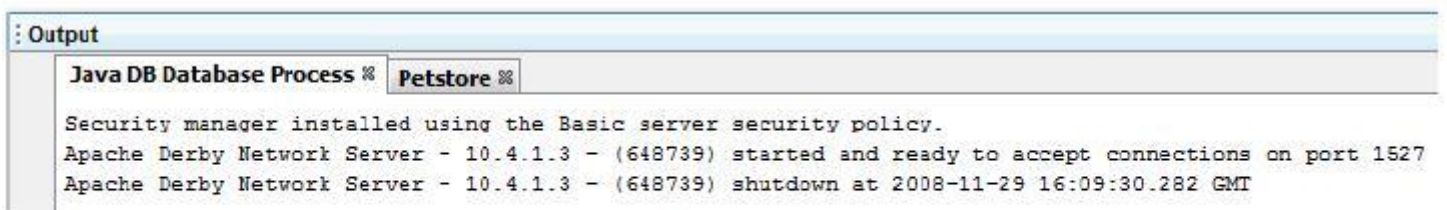


Figure 6-20: Arrêt automatique de Java DB

Le serveur de base de données devra donc être redémarré :

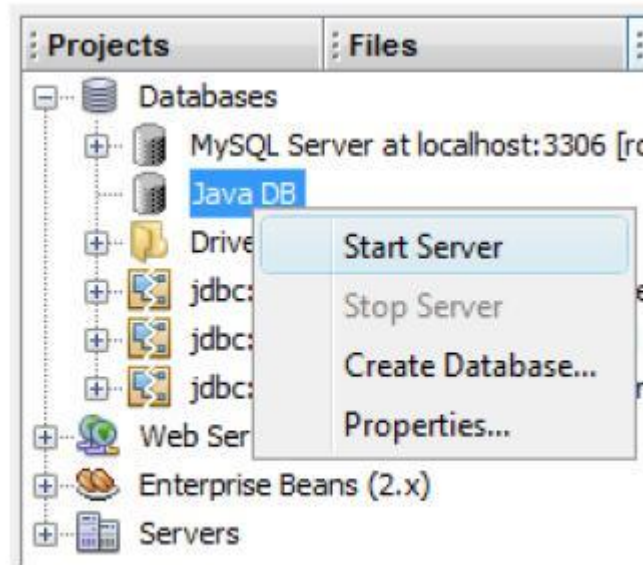


Figure 6-21: Démarrer Java DB

LLVI-C-3 - Connexion vers la base de données

Bien que la data source et le schéma de la base soit déjà créée, il est possible d'utiliser NetBeans pour consulter et manipuler les données du schéma de la base « petstoreDB ».

Mais à ce stade, le schéma de base de données n'a pas encore été créé.

Cette opération sera réalisée plus tard lorsque les directives JPA attachées aux Entity auront généré les différentes tables de l'application Yaps.

LLVI-D - Conclusion

NetBeans est configuré et prêt à accueillir les projets de l'application Yaps.

Dans la suite du document, nous allons décrire le processus d'importation des projets suivants :

- BarkBank
- PetEx
- Yaps-Server
- Yaps-Client

VII - Importer le projet BarkBank

Le projet BarkBank est une application Web (« Web Applications ») exposant le Web Service « ValidationService » chargé de simuler le processus de validation des cartes de crédit.

Nous allons décrire les différentes étapes nécessaires pour importer le projet BarkBank de l'application Yaps vers NetBeans.

L'ensemble des opérations qui suivent est réalisé sous NetBeans.

LLVII-A - Créer un nouveau projet

LLVII-A-1 - Fermeture des projets

Vous allez créer un nouveau de groupe de projets qui sera spécifique à BarkBank.

Si c'est le cas, il y a d'abord lieu de fermer l'ensemble des projets actuellement ouverts sous NetBeans.

Sélectionnez l'onglet « Projects ».

Un clic droit fait apparaître le menu contextuel. L'option « Project Group (none) » permet de fermer l'ensemble des projets ouverts :

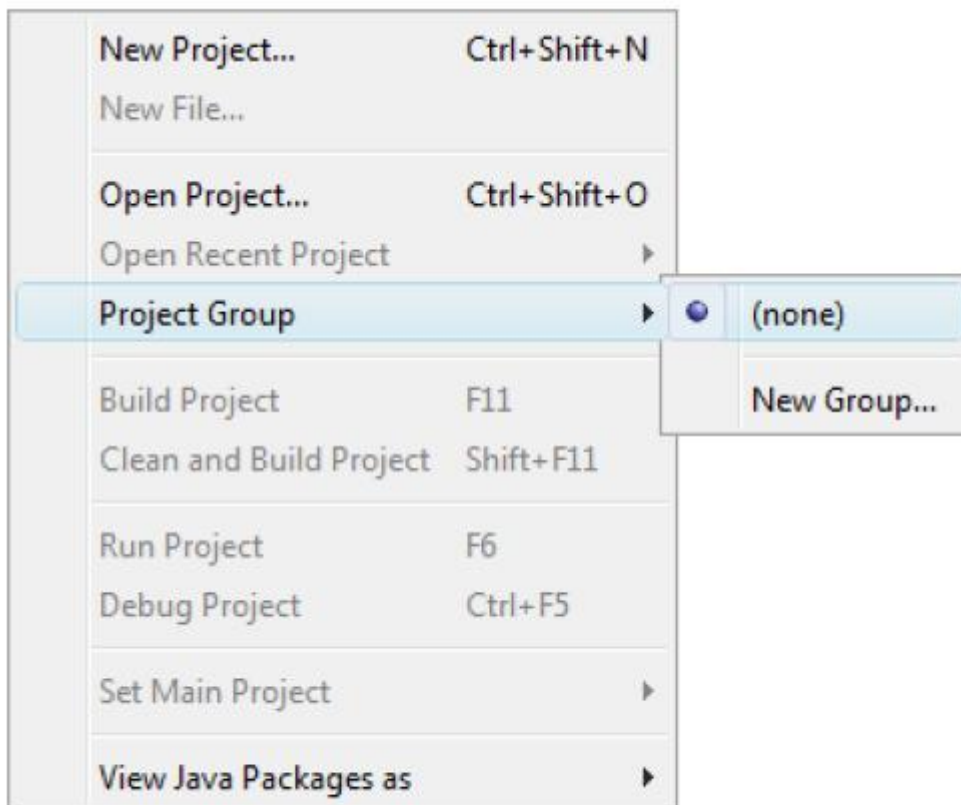


Figure 7-1: Fermer les projets ouverts

LLVII-A-2 - Type de projet

Créez un nouveau projet de type « Web Application ».

Pour cela, sélectionnez l'onglet « Project » et créez un nouveau projet via le menu « File New Project ».

Le projet que vous allez créer est de type « Java Web Web Application » :

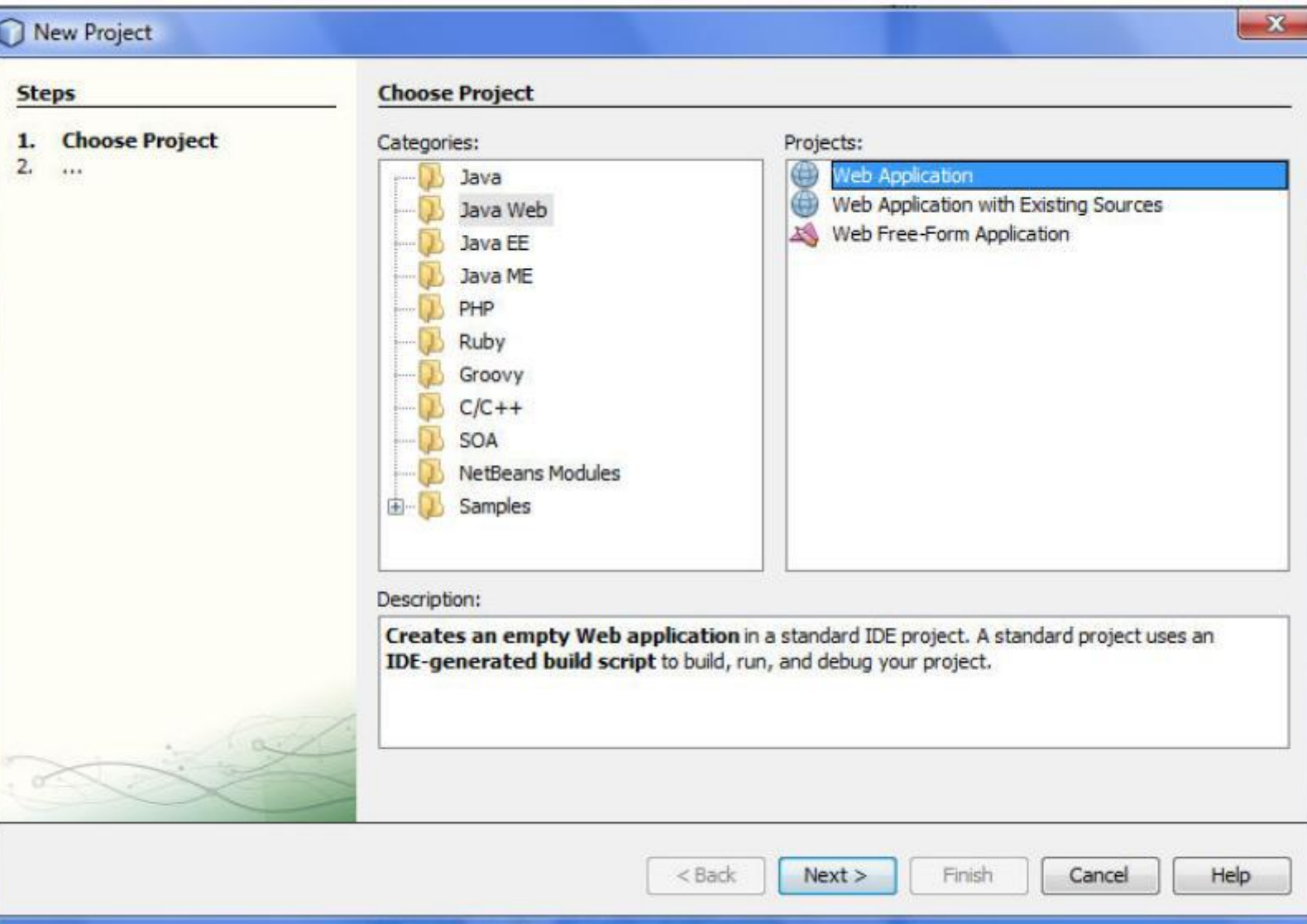


Figure 7-2: Nouveau projet Web Application

Cliquez sur le bouton « Next » pour continuer.

LLVII-A-3 - Nom et répertoires

L'écran suivant vous permet de nommer le projet, et de définir le dossier cible:

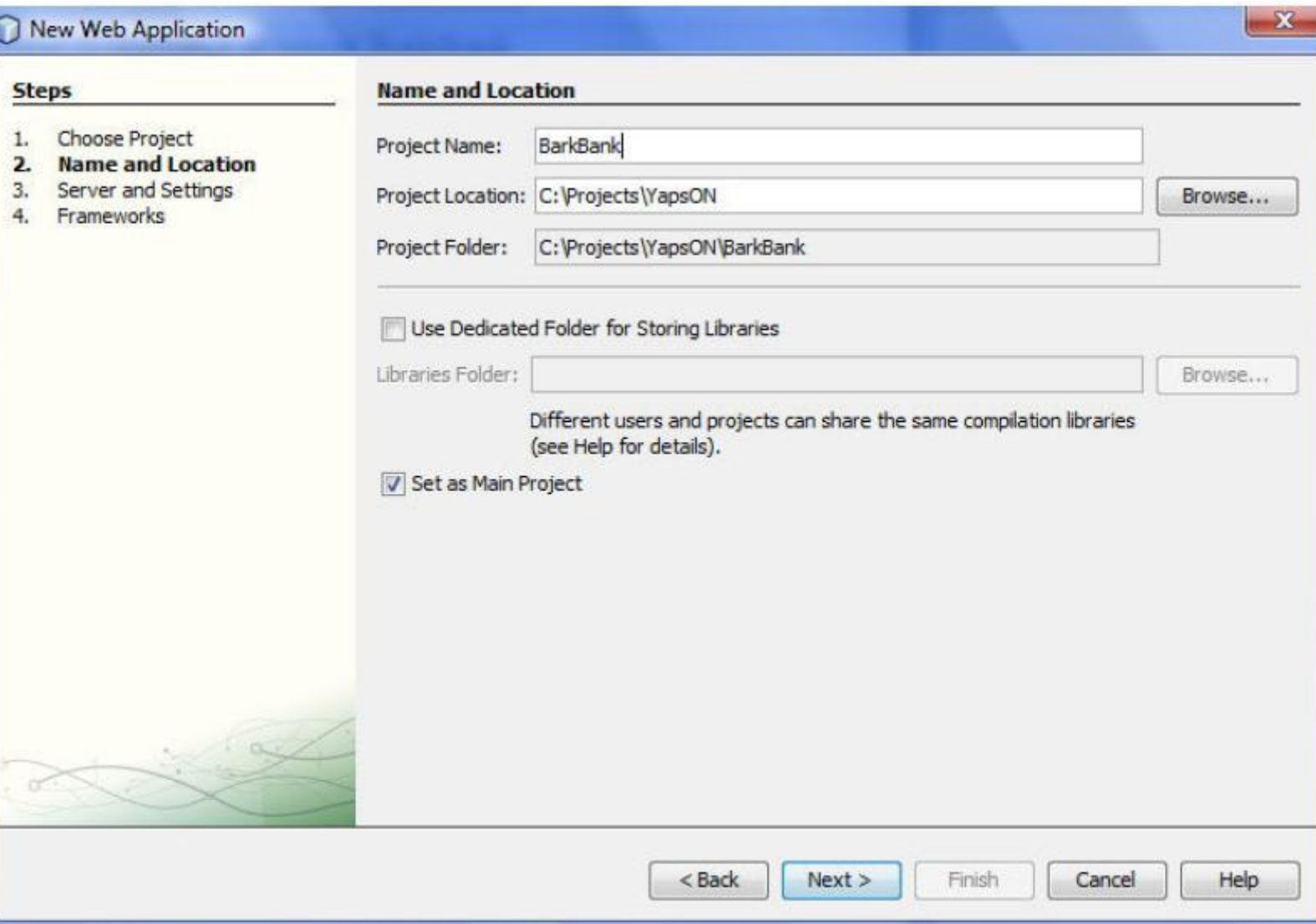


Figure 7-3: Nom du projet et répertoire

Les informations à introduire sont les suivantes:

| | |
|------------------------|-----------------------------|
| Project Name: | BarkBank |
| Project Folder: | C:\Projects\YapsON\BarkBank |

Project Folder

Le répertoire du projet est donné ici à titre d'exemple et doit correspondre au répertoire où vous avez décompressé le fichier d'archives de l'application YapsON (voir [ici](#)).

Cliquez sur le bouton « Next » pour continuer.

LLVII-A-4 - Configuration du serveur

Cette étape vous permet d'identifier le serveur d'applications « PetStore » (voir [ici](#)) sur lequel sera déployée l'application BarkBank :

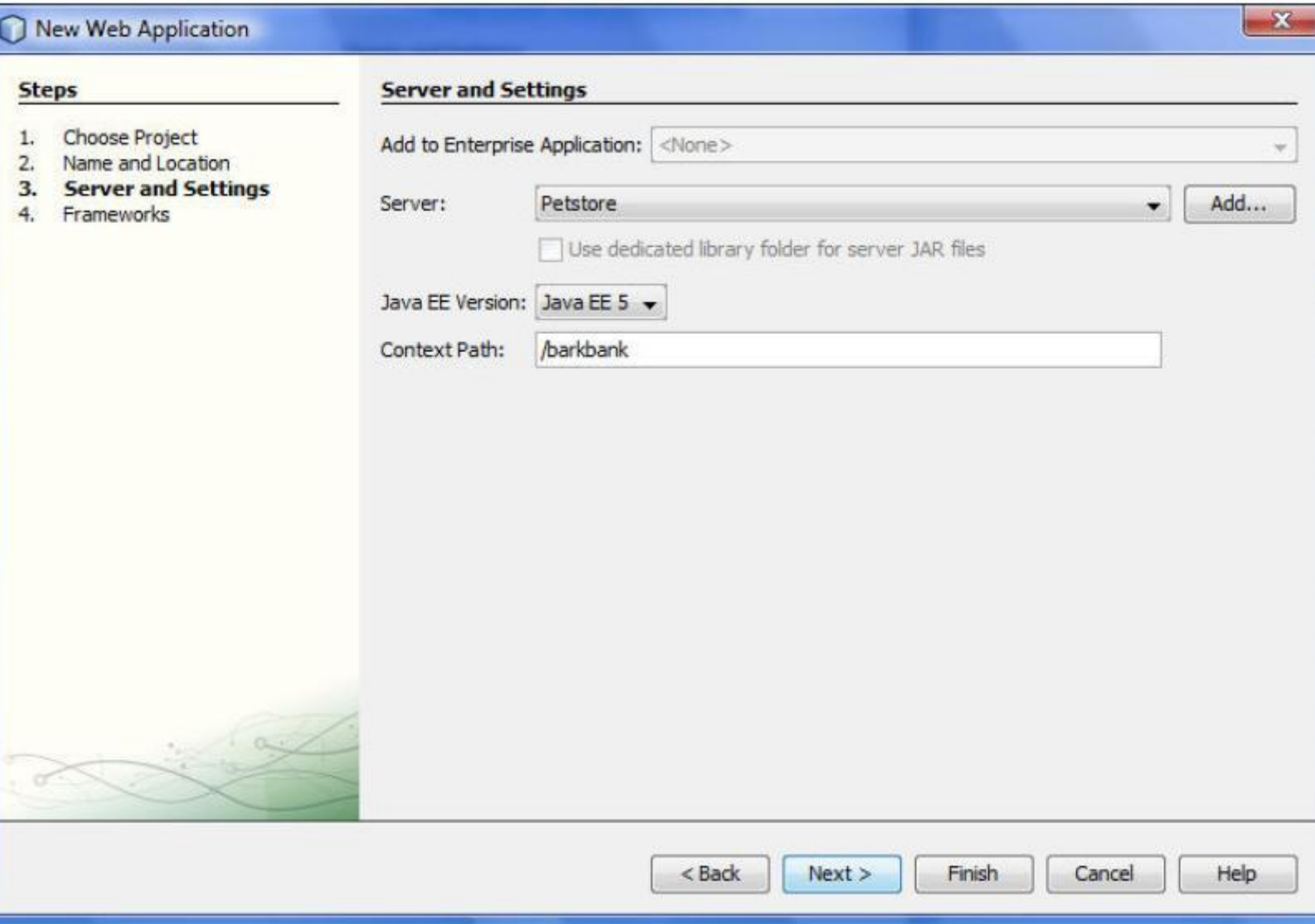


Figure 7-4: Configuration du serveur

La configuration utilisée pour le déploiement est la suivante :

| | |
|----------------------|-----------|
| Server: | Petstore |
| Context Path: | /barkbank |

Cliquez sur le bouton « Next » pour continuer.

LLVII-A-5 - Les frameworks

La dernière étape consiste à spécifier les frameworks à utiliser.

Dans le cadre du projet « BarkBank », aucun framework spécifique n'est employé :

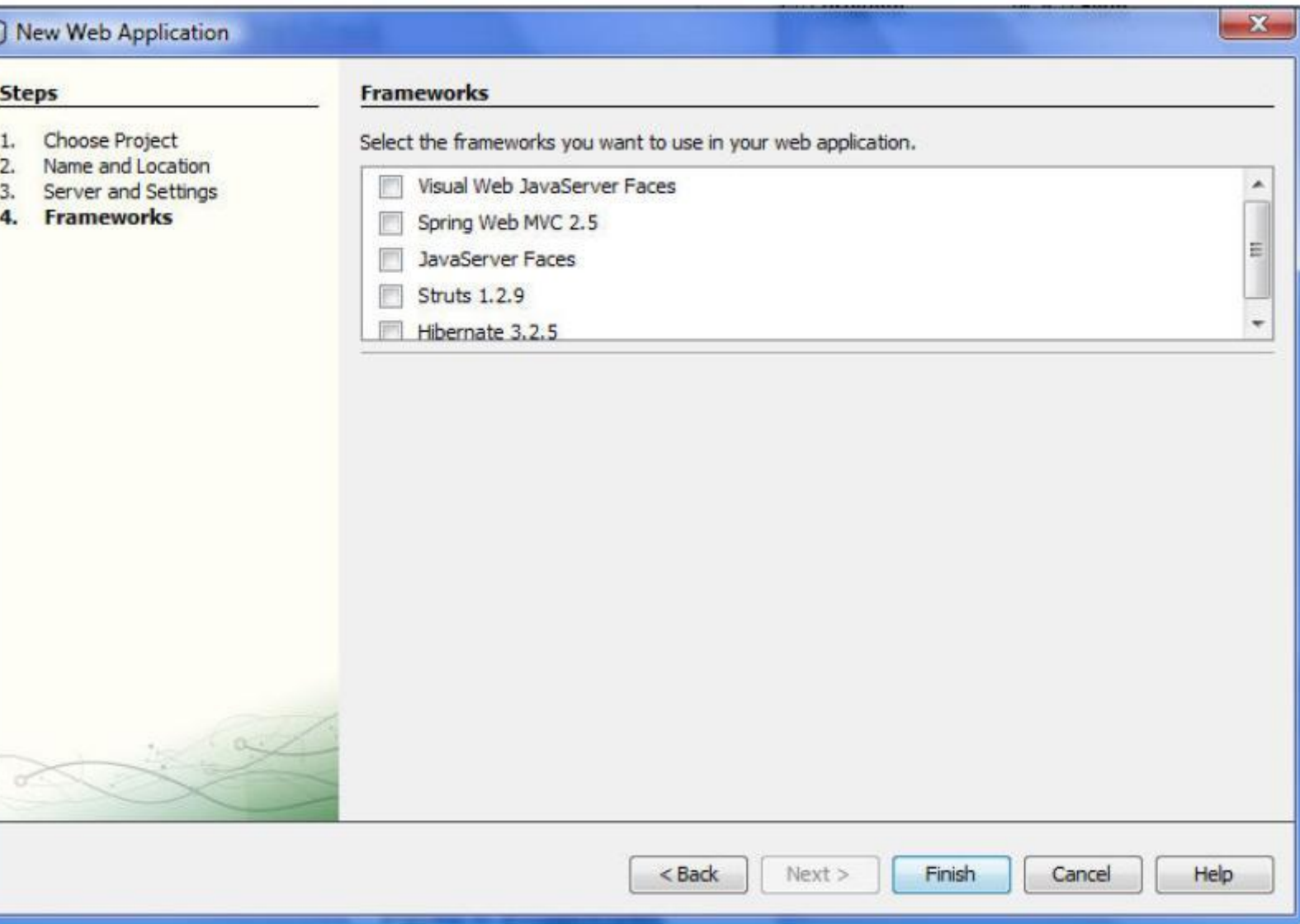


Figure 7-5: Frameworks

Appuyez sur le bouton « Finish » pour terminer le processus de création du projet.

LLVII-A-6 - Génération du projet

Le projet est maintenant généré et présente l'ensemble des dossiers suivants :

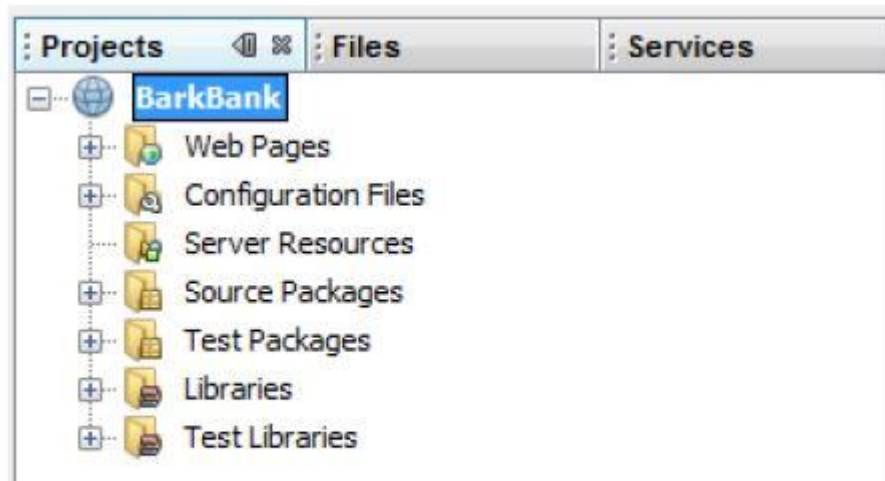


Figure 7-6: Dossiers du projet

Vous êtes prêt à importer le code source du projet BarkBank de l'application Yaps vers l'arborescence de NetBeans.

LLVII-B - Transférer le projet sous NetBeans

L'opération de transfert du projet BarkBank consiste à copier les fichiers de l'application Yaps vers l'application YapsON.

Pour faciliter cette opération, le script Ant « **yapson.xml** » (décrit dans la section 3.5) automatise la copie des fichiers.

Pour faciliter cette opération, le script Ant « yapson.xml » (décrit dans la section 3.5) automatise la copie des fichiers.

LLVII-B-1 - Positionner les variables d'environnement

Les commandes que vous allez exécuter nécessitent de positionner les variables d'environnement pour l'exécution des tâches Ant.

Rendez-vous dans le répertoire de l'application YapsON et exécutez le fichier de commande « setenv-yaps.cmd » :

```
C:> cd {yapson_home}
{yapson_home}> setenv-yaps.cmd
```

LLVII-B-2 - Transférer le projet

À partir de la ligne de commande, exécutez la tâche « **transferBarkBank** » chargée de transférer le projet sous NetBeans :

```
{yapson_home}> ant -f yapson.xml transferBarkBank
Buildfile: yapson.xml
transferBarkBank:
[echo] Transfer the BarkBank projet to NetBeans
...
```

L'exécution du script Ant commence par demander une confirmation avant de procéder au remplacement des fichiers :

```
...  
[input] The target source files will be overwritten by the original YAPS project. Do you  
want to continue? (y, n)  
y  
...
```

Dans l'affirmative, la tâche Ant copie respectivement les codes sources Java, les fichiers de ressources Web et les descripteurs :

```
...  
[echo] === Copy Java Sources ===  
[copy] Copying 1 file to c:\Projects\YapsON\BarkBank\src\java  
[echo] === Copy Web Resources ===  
[copy] Copying 2 files to c:\Projects\YapsON\BarkBank\web  
[echo] === Copy Web Descriptors ===  
[copy] Copying 1 file to c:\Projects\YapsON\BarkBank\web\WEB-INF  
BUILD SUCCESSFUL
```

NetBeans rafraîchit automatiquement le projet BarkBank :

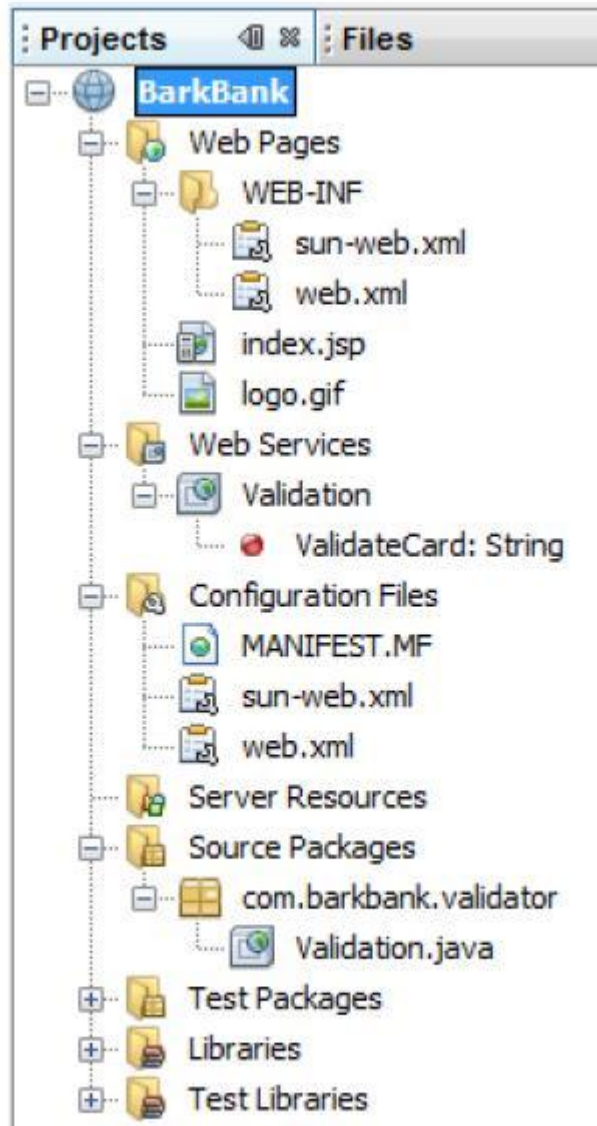


Figure 7-7: Projet BarkBank après le transfert

LLVII-B-3 - Créer le groupe de projet

NetBeans offre la possibilité de créer des groupes de projets pour faciliter le passage d'un projet à l'autre sans avoir à redémarrer l'IDE.

À partir du menu « File » ou via un clic droit dans la fenêtre « Projects », NetBeans offre la capacité de créer le nouveau groupe de projet:

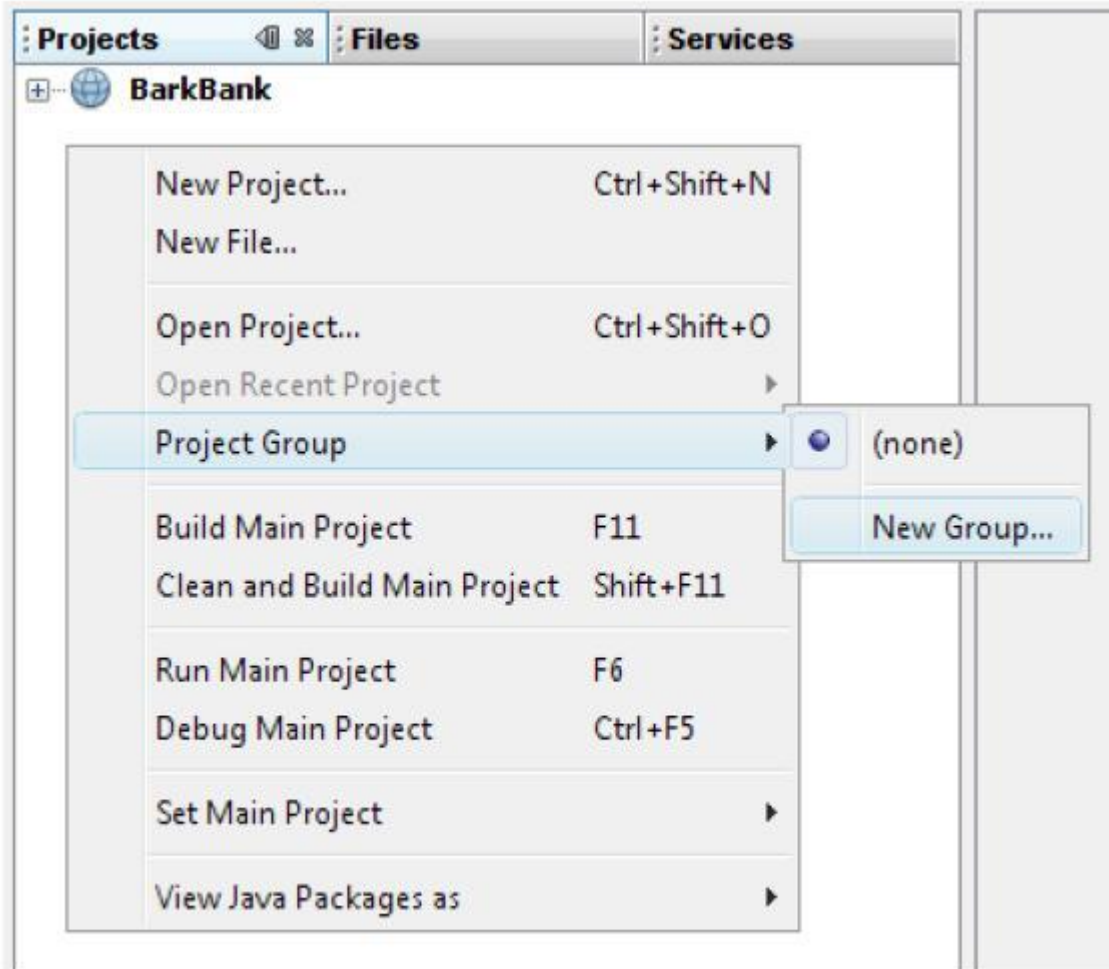


Figure 7-8: Créer un groupe de projet

Sélectionnez l'option « New group ».

La fenêtre suivante permet de nommer le groupe de projet qui est dans notre cas BarkBank :

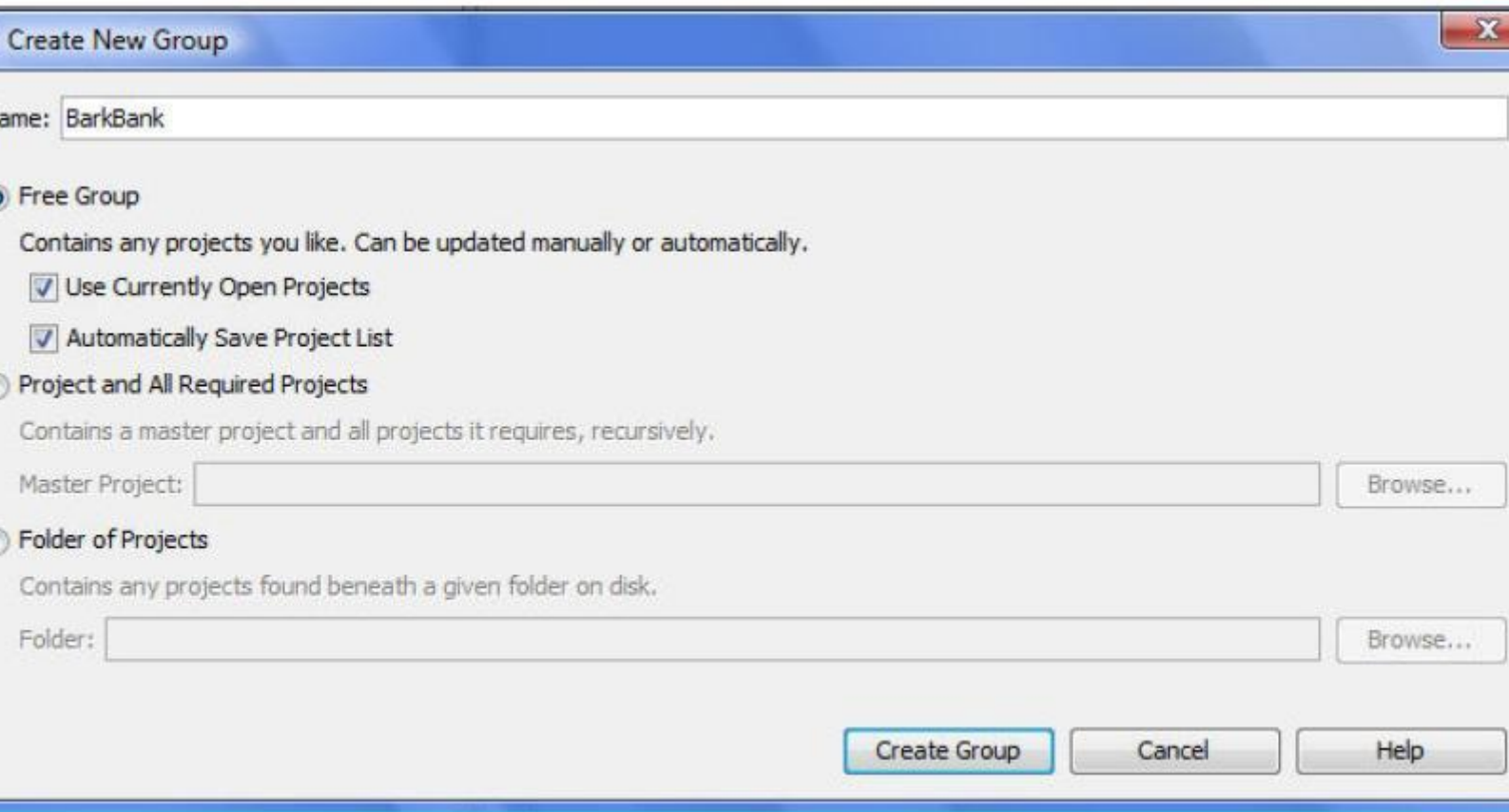


Figure 7-9: Nouveau groupe de projet

Les options par défaut indiquent que les projets en cours de chargement dans l'IDE seront automatiquement ajoutés au groupe de projet BarkBank.

Cliquez sur le bouton « **Create Group** ».

LLVII-C - Compiler le projet

Un clic droit sur le projet fait apparaître le menu contextuel proposant les options pour compiler le projet (« Build » ou « Clean and Build ») :

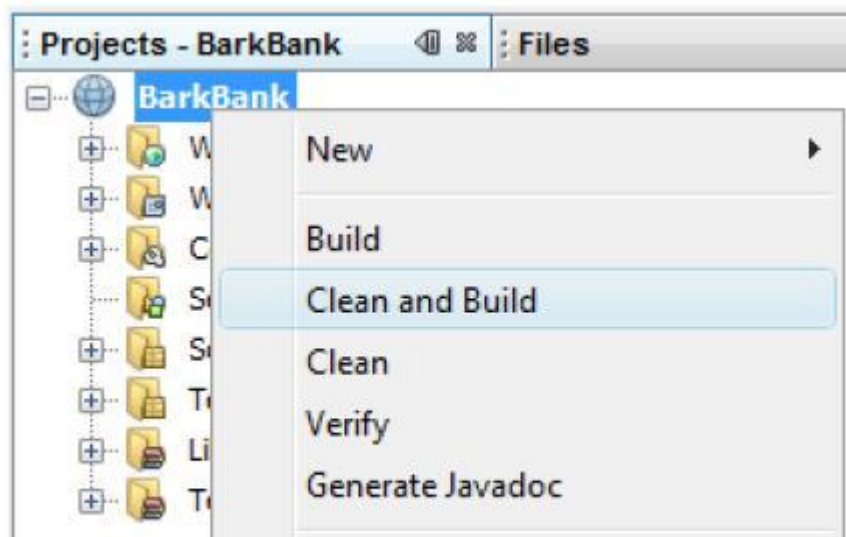


Figure 7-10: Compiler le projet

Cliquez sur l'option « Clean and Build ».

Le fenêtre « Output » affiche les traces de la compilation :

```

Output
Java DB Database Process  Petstore  BarkBank (clean,dist)
-----
deps-ear-jar:
deps-jar:
Created dir: C:\Projects\YapsON\BarkBank\build\web\WEB-INF\classes
Created dir: C:\Projects\YapsON\BarkBank\build\web\META-INF
Copying 1 file to C:\Projects\YapsON\BarkBank\build\web\META-INF
Copying 4 files to C:\Projects\YapsON\BarkBank\build\web
library-inclusion-in-archive:
library-inclusion-in-manifest:
Compiling 1 source file to C:\Projects\YapsON\BarkBank\build\web\WEB-INF\classes
C:\Projects\YapsON\BarkBank\src\java\com\barkbank\validator\Validation.java:86: warning: unmappable character for encoding UTF-8
    // =          Methodes Priv?es          =
1 warning
compile:
compile-jsp:
Created dir: C:\Projects\YapsON\BarkBank\dist
Building jar: C:\Projects\YapsON\BarkBank\dist\BarkBank.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 2 seconds)

```

Figure 7-11: Résultat de la compilation

Les messages intéressants sont les suivants :

```

...
compile:
compile-jsp:
Created dir: C:\Projects\YapsON\BarkBank\dist
Building jar: C:\Projects\YapsON\BarkBank\dist\BarkBank.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 2 seconds)

```

Ces traces indiquent que le projet a été correctement compilé (« BUILD SUCCESSFUL ») et archivé dans le fichier « BarkBank.war ».

LLVII-4 - Déployer le projet

Vous allez maintenant procéder au déploiement du projet sous GlassFish.

Un clic droit sur le projet fait apparaître le menu contextuel proposant l'option de déploiement (« Deploy ») :

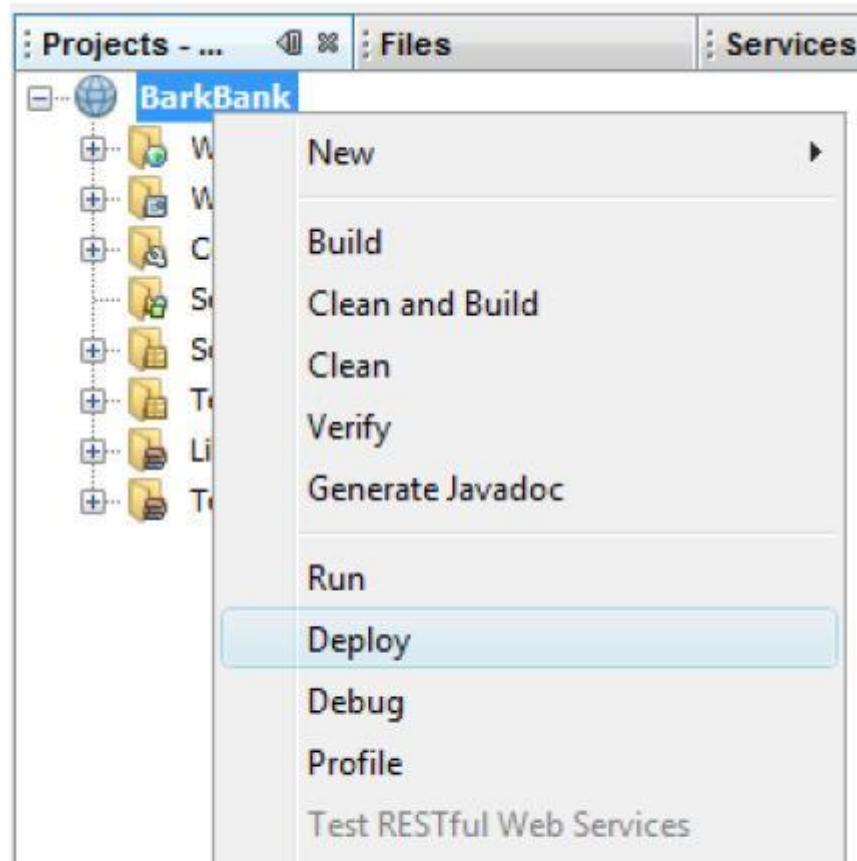


Figure 7-12: Déployer le projet

Cliquez sur l'option « Deploy ».

Si nécessaire, NetBeans se charge de démarrer le serveur d'applications « PetStore ».

Si l'application BarkBank était déjà déployée sur le serveur, NetBeans se charge de la supprimer (Undeploying) pour la remplacer par la nouvelle version.

La fenêtre « Output » trace les messages suivants :

```
...
deployment started : 0%
deployment finished : 100%
Deploying application in domain completed successfully
...
All operations completed successfully
run-deploy:
BUILD SUCCESSFUL (total time: 4 seconds)
```

Le déploiement terminé, vous pouvez explorer le contenu de l'application BarkBank sur le serveur PetStore.

Rendez-vous sur l'onglet "Services" et sélectionnez le serveur "Petstore".

Le noeud « Applications -> Web Applications -> BarkBank » affiche les ressources et services exposés par l'application :

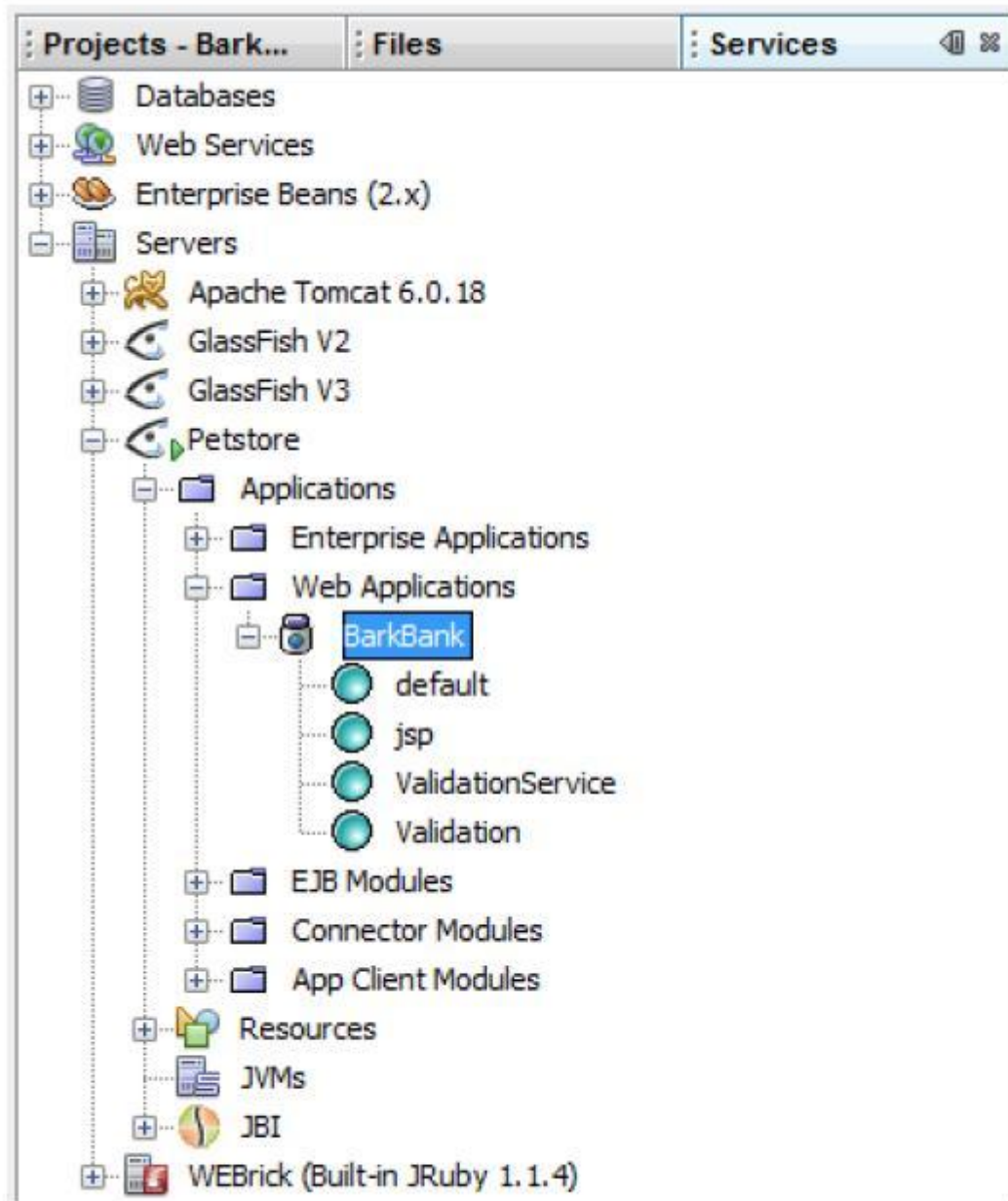


Figure 7-13: Application BarkBank sous GlassFish

L'application BarkBank est accessible à l'adresse <http://localhost:8080/barkbank> et se présente comme suit:

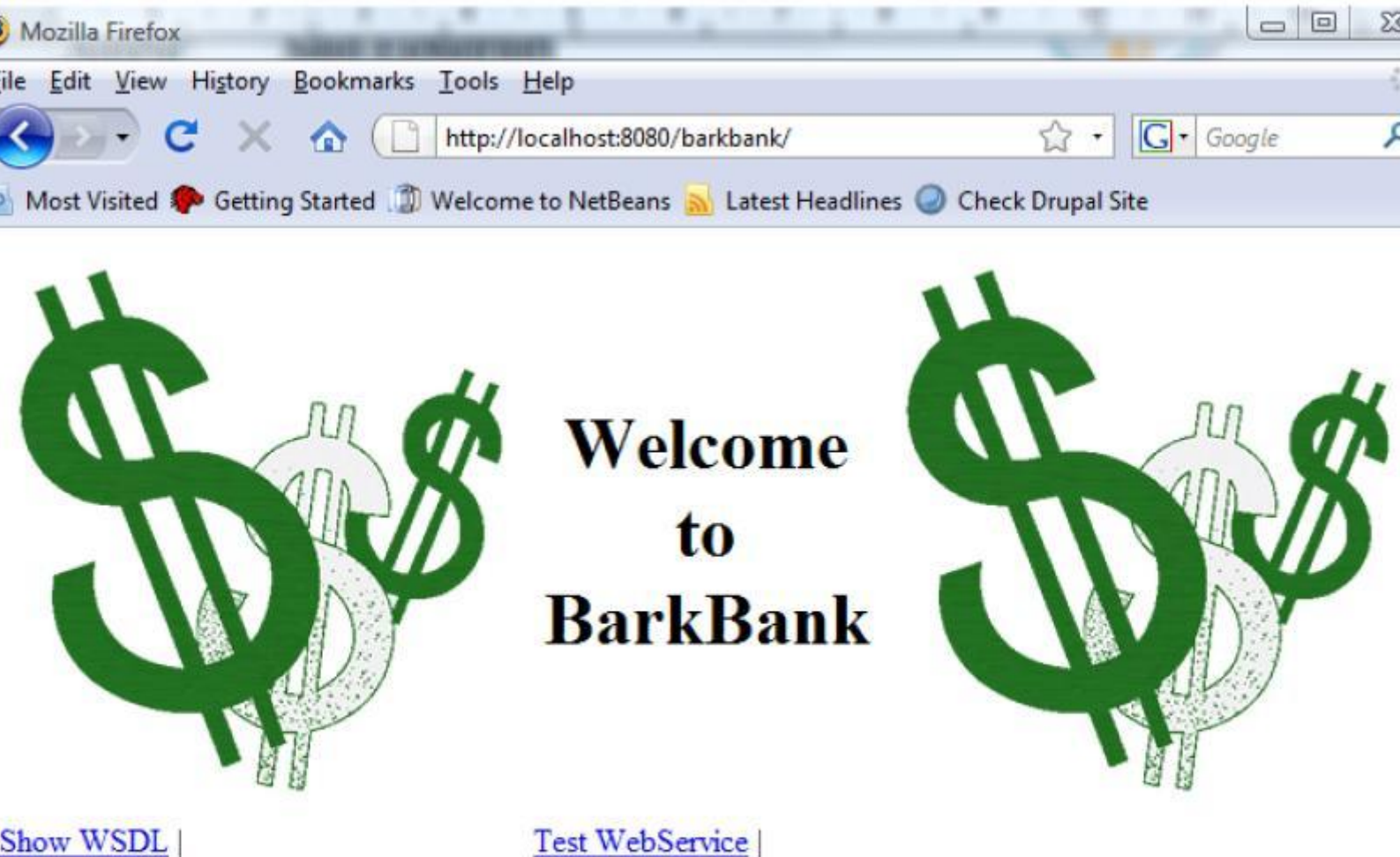


Figure 7-14: Page d'accueil de BarkBank

À ce stade, vous pouvez inspecter le WSDL ou tester le service tel que décrit dans le livre de référence.

LLVII-E - Tester le projet

Vous êtes maintenant en mesure de tester l'application BarkBank en invoquant le Web Service « ValidationService ».

Un Web Service peut être testé de différentes manières. Nous allons ici nous contenter de quelques unes d'entre elles disponibles sous NetBeans.

Client de test SOAP

Nous vous recommandons d'évaluer la solution **soapUI** (<http://www.soapui.org>) qui est un outil de test de Web Service.

soapUI est proposé sous deux licences. La version standard est gratuite et distribuée sous licence Open Source. La version Pro est payante.

soapUI peut-être intégré sous forme de plug-in tant sous Eclipse que sous NetBeans.

Mais nous avons toutefois constaté des problèmes de stabilités avec le plug-in pour NetBeans contrairement à la stabilité affichée par la version standalone.

LLVII-E-1 - Méthode 1 : Web Service Tester de GlassFish

Au travers de la fenêtre « Projects », un clic droit sur le Web Service « Validation » propose le menu contextuel suivant:

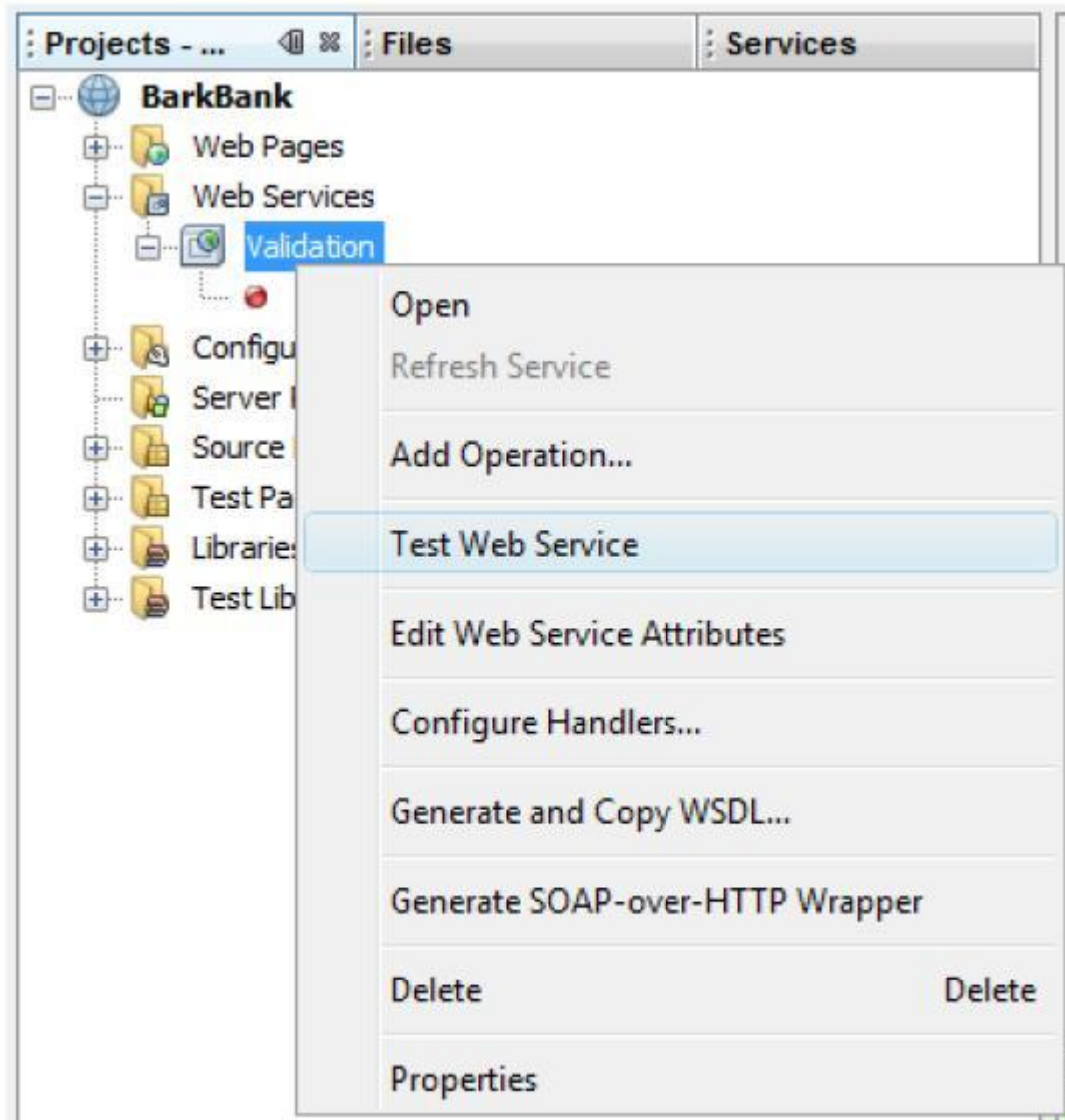


Figure 7-15: Tester le Web Service

L'option « **Test Web Service** » invoque le mécanisme de tests de Web Service de GlassFish qui a son tour invoque le Web Service « ValidationService » au travers de l'URL suivante:

<http://localhost:8080/barkbank/ValidationService?Tester>

Ce mécanisme est largement décrit dans le livre de référence.

LLVII-E-2 - Méthode 2 : Web Service Tester de NetBeans

Cette méthode consiste à enregistrer le Web Service sous NetBeans afin de bénéficier d'un outil de test un peu plus évolué que celui offert par GlassFish.

Les Web Services sont répertoriés dans la fenêtre « Services » sous le noeud « Web Service » :

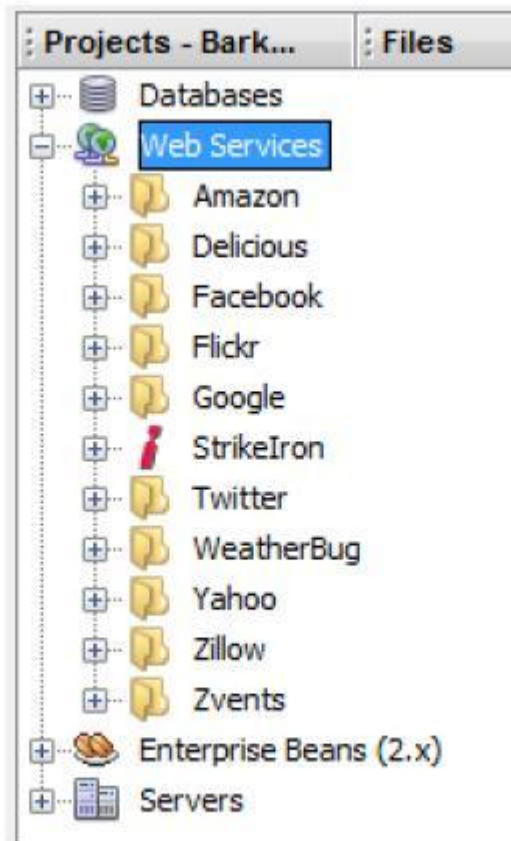


Figure 7-16: Liste des Web Services sous NetBeans

Pour ajouter un Web Service, procédez comme suit.

Vous devez avant tout créer un groupe dans lequel seront stockés les Web Service du projet YapsON.

Un click droit sur le noeud « Web Services » propose un menu contextuel dans lequel vous allez créer le groupe « YapsON » par l'option « Create Group » :

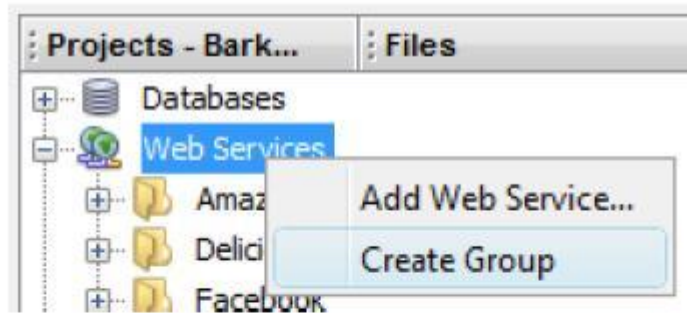


Figure 7-17: Groupe de Web Services

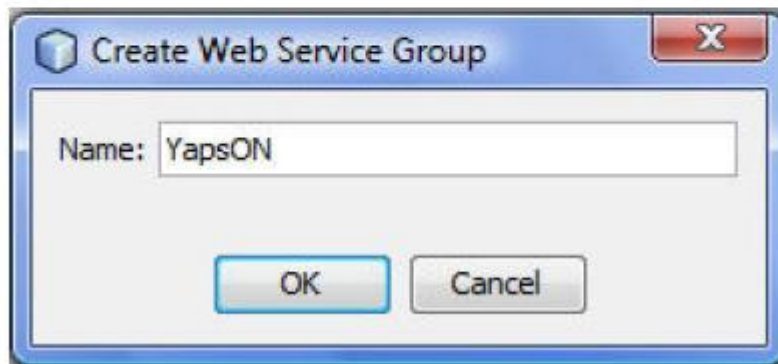


Figure 7-18: Ajout du groupe de Web Service

Ensuite, un clic droit sur le groupe « YapsON » offre un ensemble d'options parmi lesquelles celle qui permet d'ajouter un Web Service par l'option « Add Web Service » :

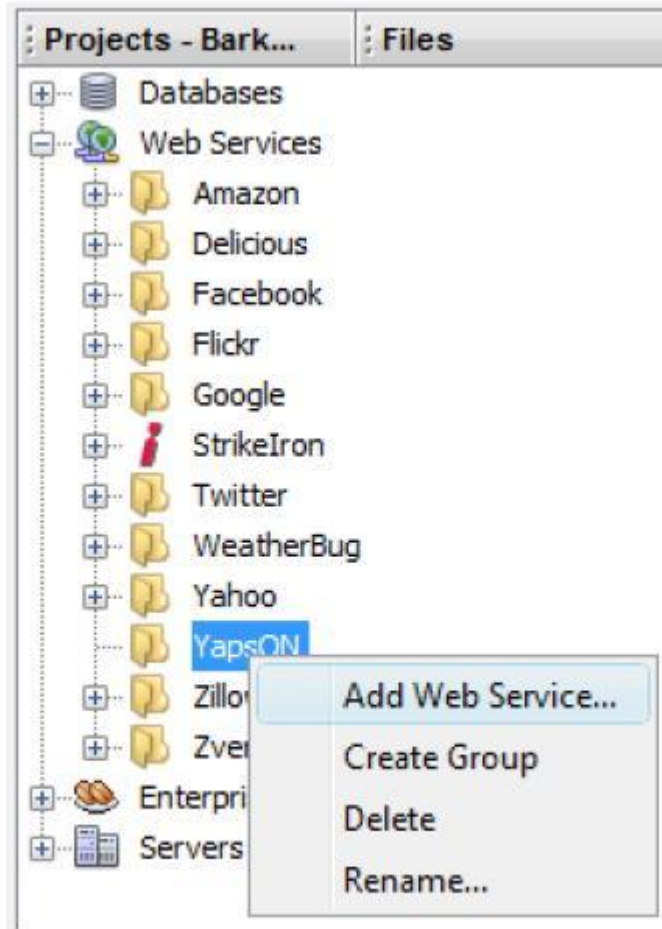


Figure 7-19: Ajout d'un Web Service

Le formulaire permet d'identifier le Web Service à partir de son WSDL:

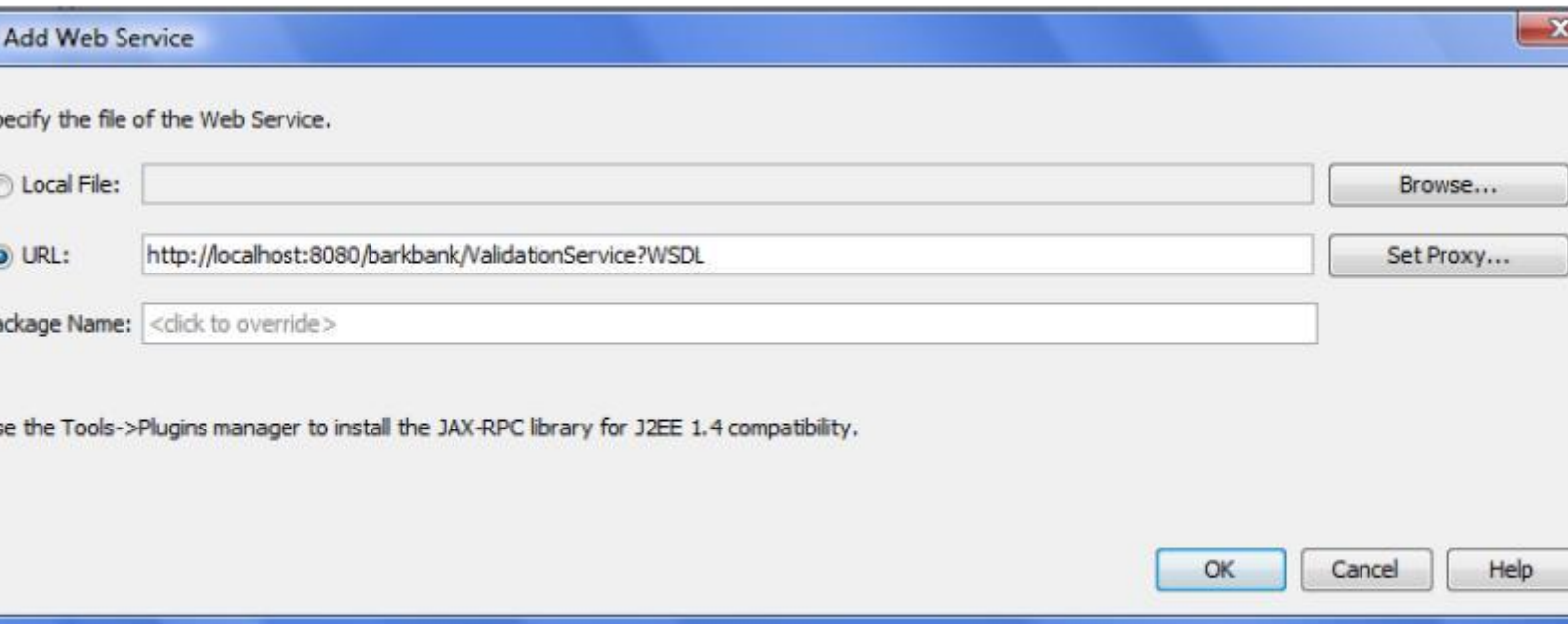


Figure 7-20: Localiser le WSDL

Veuillez localiser le WSDL du Web Service à partir de son URL :

<http://localhost:8080/barkbank/ValidationService?WSDL>

Cliquez sur le bouton "OK" pour ajouter la référence du Web Service sous NetBeans.

Le Web Service « **ValidationService** » est maintenant ajouté à la liste des Web Service :

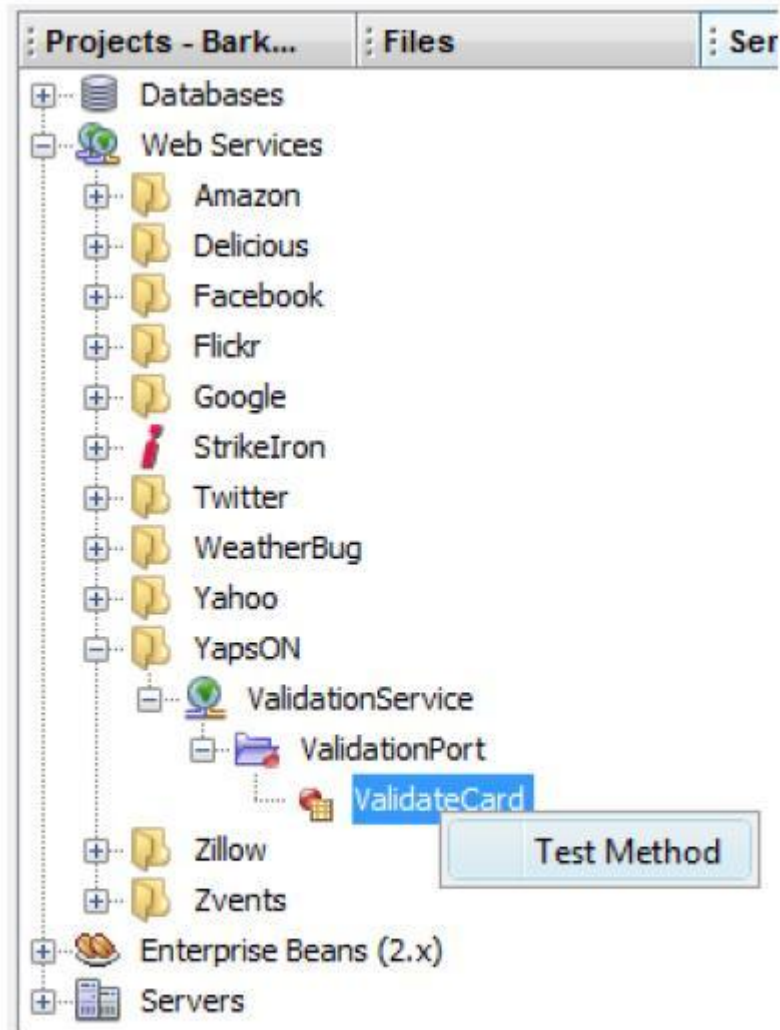


Figure 7-21: Référence de ValidationService

Un clic droit sur l'opération « ValidateCard » permet de tester la méthode au travers de l'écran suivant :

Test Web Service Method
X

Test Web Service Method:

Enter the values for the parameters and press the "Submit" button.

| Type | Name | Value |
|--------------------|------------------|-------|
| ● java.lang.String | creditCardNumber | |
| ● java.lang.String | creditCardType | |
| ● java.lang.String | expiryDate | |

Results:

Figure 7-22: Écran de test du Web Service

La partie supérieure concerne la requête à émettre tandis que la partie inférieure se charge d'afficher le message de retour. Le bouton « Submit » permet d'envoyer la requête au Web Service.

Testez le Web Service avec les données suivantes:

| | |
|--------------------------|------------|
| creditCardNumber: | 1234567890 |
| creditCardType: | MC |
| expiryDate: | 11/09 |

Test Web Service Method: validateCard
Enter the values for the parameters and press the "Submit" button.

| Type | Name | Value |
|--------------------|------------------|------------|
| ● java.lang.String | creditCardNumber | 1234567890 |
| ● java.lang.String | creditCardType | MC |
| ● java.lang.String | expiryDate | 11/09 |

Figure 7-23: Requête de test

Cliquez sur le bouton « **Submit** » pour envoyer la requête au Web Service « **ValidationService** ».

Les données sont considérées comme valides par le service et le résultat est un message « OK » :

Results:

| Type | Value |
|--------------------|-------|
| ● java.lang.String | OK |

Figure 7-24: Message de retour

Si l'on omet par exemple la date d'expiration, le Web Service retourne un message d'erreur explicite composé par l'application BarkBank :

sults:

| type | Value |
|--------------------|----------------------|
| ● java.lang.String | Expiry date is empty |

Figure 7-25: Message d'erreur

LLVII-F - Conclusion

Les sources de l'application BarkBank ont été transférées sous NetBeans dans une arborescence conforme au format préconisé par le Java Blueprints.

Les tests que nous avons réalisés attestent que le Web Service de l'application BarkBank a été correctement compilé et est déployé sur le serveur.

VIII - Importer le projet PetEx

Le projet PetEx est une application Web (« Web Applications ») exposant le Web Service « DeliveryService » chargé de simuler le processus de transport des marchandises achetées en ligne.

Nous allons décrire les différentes étapes nécessaires pour importer le projet PetEx de l'application Yaps vers NetBeans. Vous noterez que le principe est en tout point similaire au processus d'importation du projet BarkBank.

L'ensemble des opérations qui suivent est réalisé sous NetBeans.

LLVIII-A - Créer un nouveau projet

LLVIII-A-1 - Fermeture des projets

Vous allez créer un nouveau de groupe de projets qui sera spécifique à PetEx.

Si c'est le cas, il y a d'abord lieu de fermer l'ensemble des projets actuellement ouverts sous NetBeans.

Sélectionnez l'onglet « Projects ».

Un clic droit fait apparaître le menu contextuel. L'option « Project Group (none) » permet de fermer l'ensemble des projets ouverts :

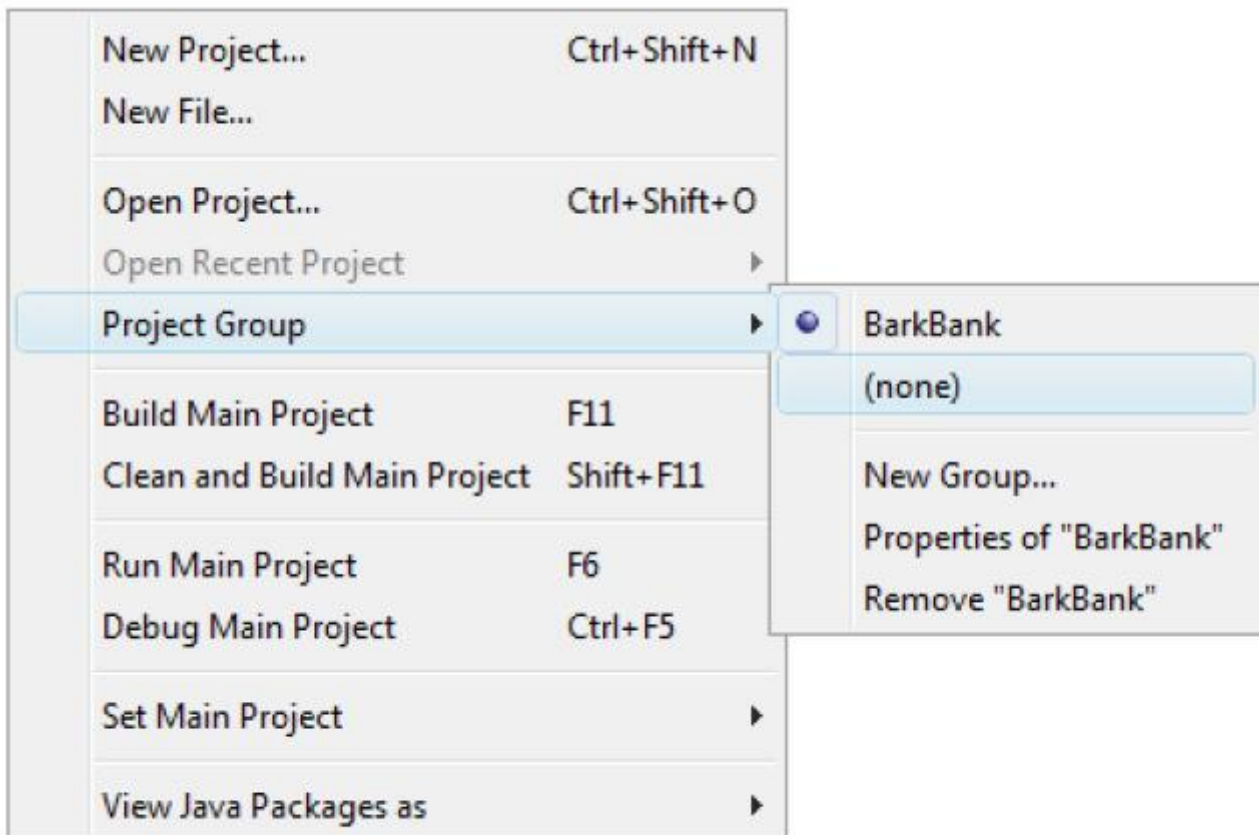


Figure 8-1: Fermer les projets ouverts

LLVIII-A-2 - Type de projet

Créez un nouveau projet de type « Web Application ».

Pour cela, sélectionnez l'onglet « Project » et créez un nouveau projet via le menu « File New Project ».

Le projet que vous allez créer est de type « Java Web -> Web Application » :

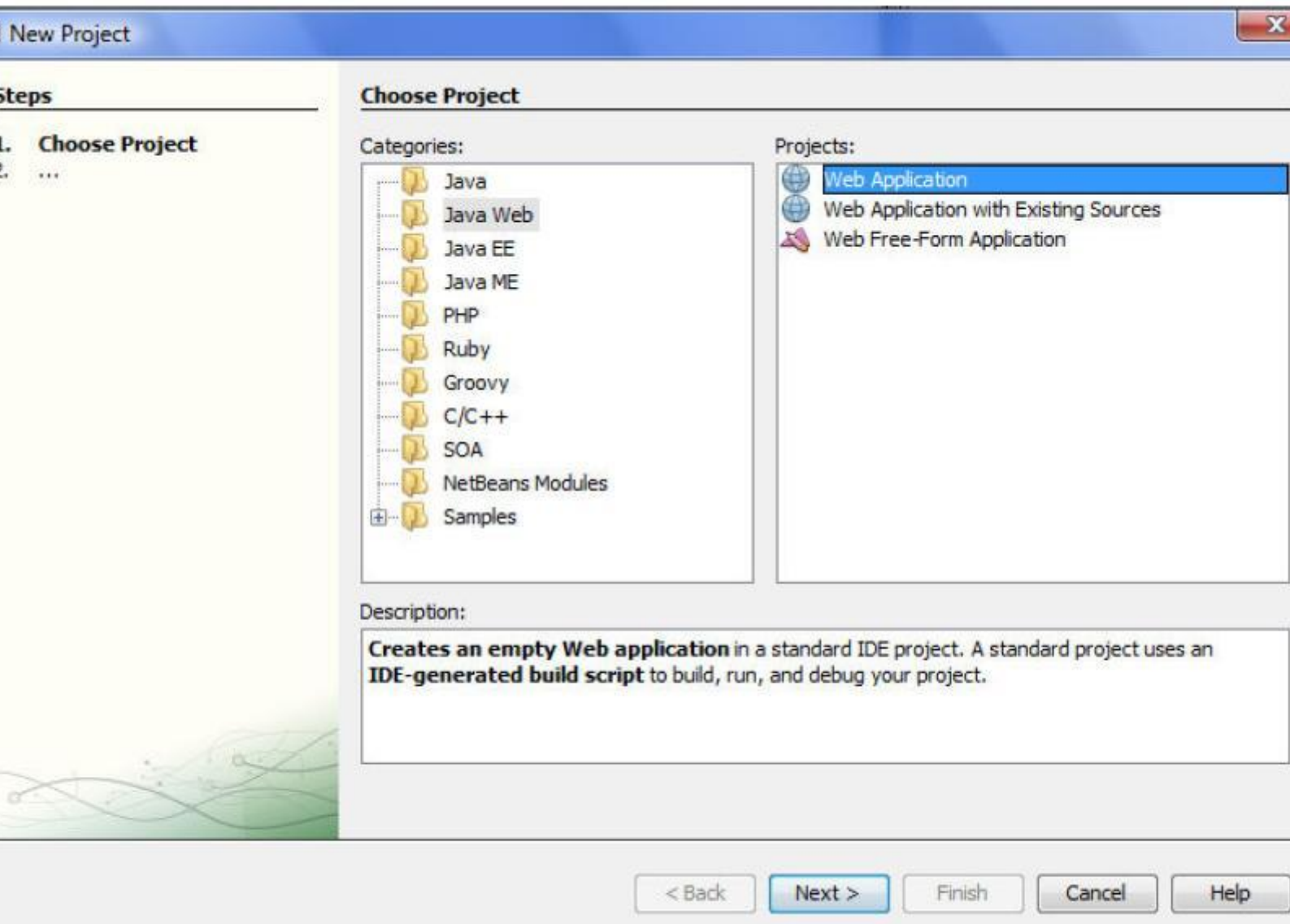


Figure 8-2: Nouveau projet Web Application

Cliquez sur le bouton « Next » pour continuer.

LLVIII-A-3 - Nom et répertoires

L'écran suivant vous permet de nommer le projet, et de définir le dossier cible:

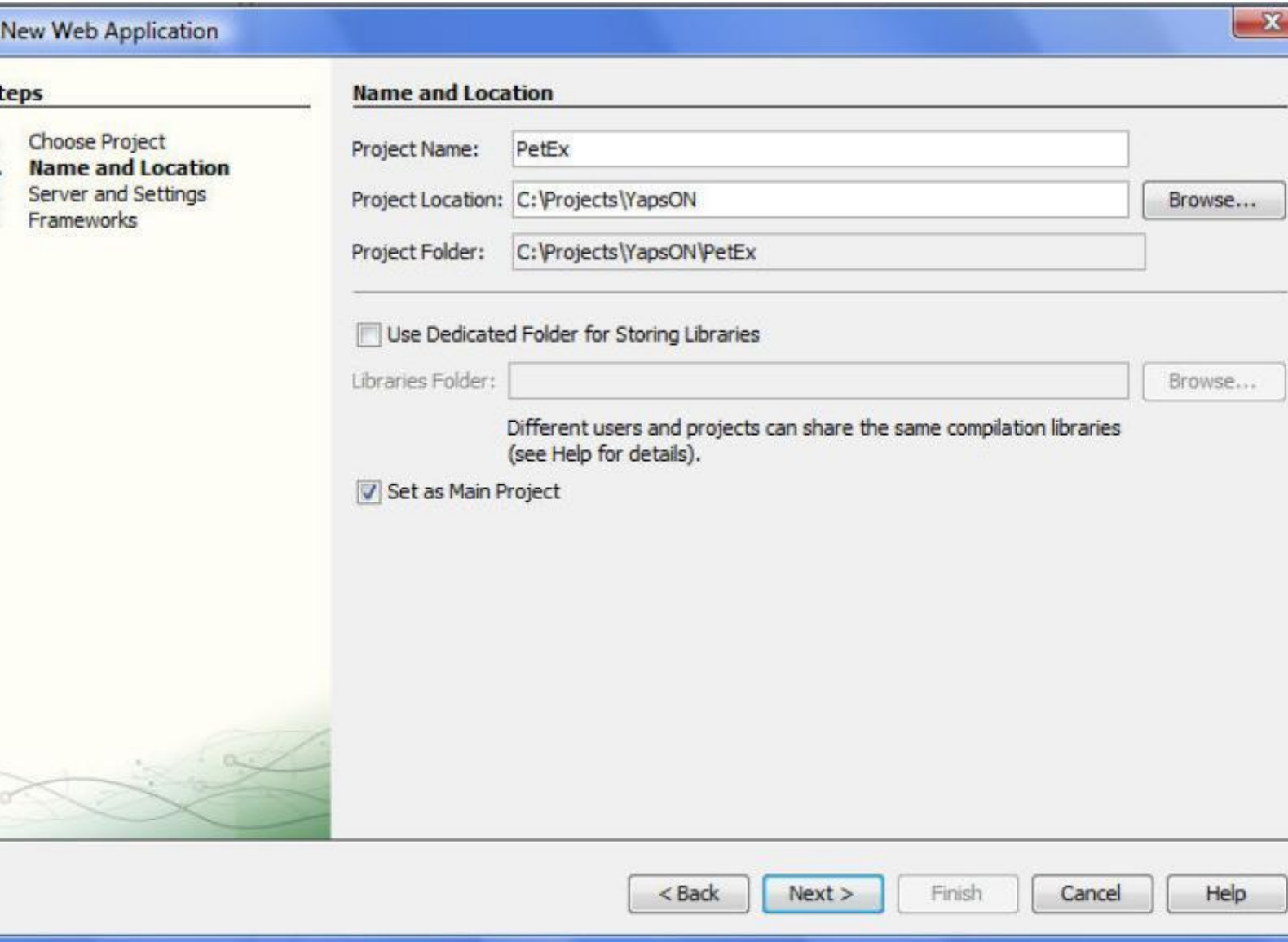


Figure 8-3: Nom du projet et répertoire

Les informations à introduire sont les suivantes:

| | |
|------------------------|--------------------------|
| Project Name: | PetEx |
| Project Folder: | C:\Projects\YapsON\PetEx |

Project Folder

Le répertoire du projet est donné ici à titre d'exemple et doit correspondre au répertoire où vous avez décompressé le fichier d'archives de l'application YapsON (voir [ici](#)).

Cliquez sur le bouton « Next » pour continuer.

LLVIII-A-4 - Configuration du serveur

Cette étape vous permet d'identifier le serveur d'applications « PetStore » (voir [ici](#)) sur lequel sera déployée l'application PetEx :

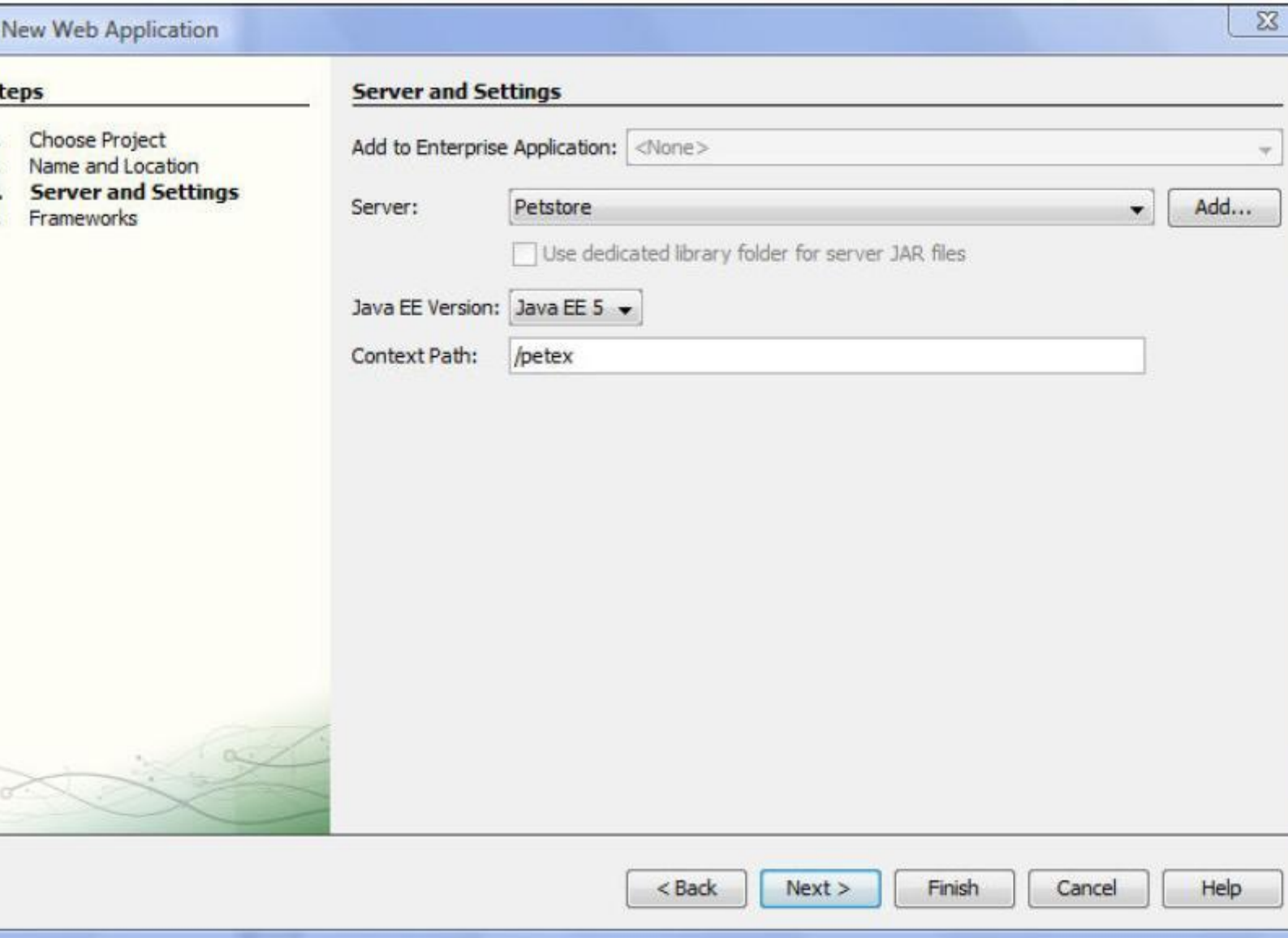


Figure 8-4: Configuration du serveur

La configuration utilisée pour le déploiement est la suivante :

| | |
|----------------------|----------|
| Server: | Petstore |
| Context Path: | /petex |

Cliquez sur le bouton « Next » pour continuer.

LLVIII-A-5 - Les frameworks

La dernière étape consiste à spécifier les frameworks à utiliser.

Dans le cadre du projet « PetEx », aucun framework spécifique n'est employé :

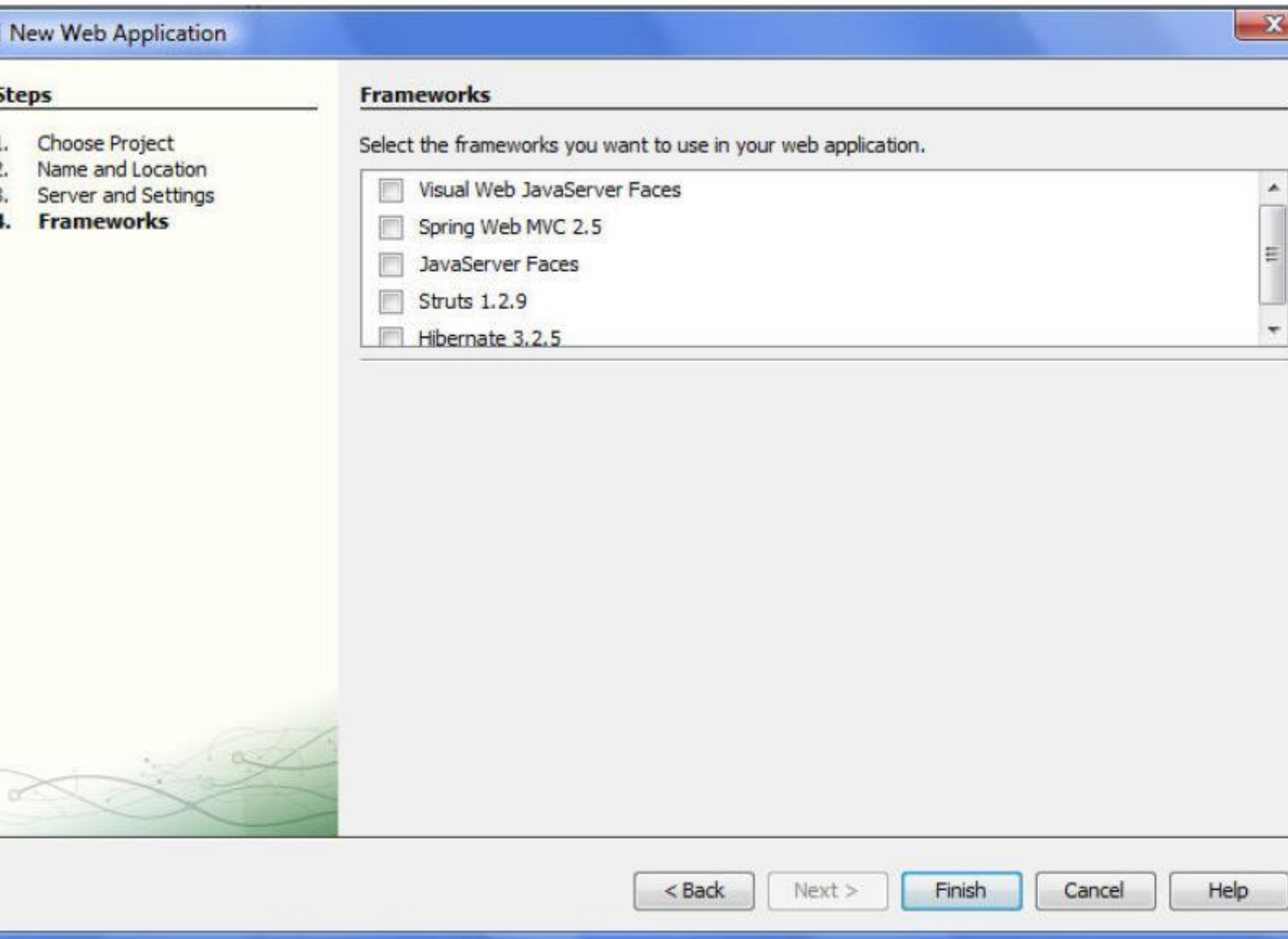


Figure 8-5: Frameworks

Appuyez sur le bouton « Finish » pour terminer le processus de création du projet.

LLVIII-A-6 - Génération du projet

Le projet est maintenant généré et présente l'ensemble des dossiers suivants :

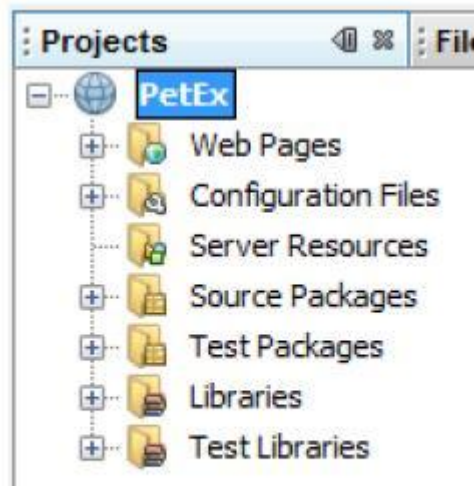


Figure 8-6: Dossiers du projet

Vous êtes prêt à importer le code source du projet PetEx de l'application Yaps vers l'arborescence de NetBeans.

LLVIII-B - Transférer le projet sous NetBeans

L'opération de transfert du projet PetEx consiste à copier les fichiers de l'application Yaps vers l'application YapsON.

Pour faciliter cette opération, le script Ant « **yapson.xml** » (décrit dans la section 3.5) automatise la copie des fichiers.

Assurez-vous d'avoir configuré le fichier « yapson.properties » tel que décrit dans la section 4.

LLVIII-B-1 - Positionner les variables d'environnement

Les commandes que vous allez exécuter nécessitent de positionner les variables d'environnement pour l'exécution des tâches Ant.

Rendez-vous dans le répertoire de l'application YapsON et exécutez le fichier de commande « setenv-yaps.cmd » :

```
C:> cd {yapson_home}
{yapson_home}> setenv-yaps.cmd
```

LLVIII-B-2 - Transférer le projet

À partir de la ligne de commande, exécutez la tâche Ant « **transferPetEx** » chargée de transférer le projet sous NetBeans :

```
%YAPSON-HOME%> ant -f yapson.xml transferPetEx
Buildfile: yapson.xml
transferPetEx:
[echo] Transfer the PetEx projet to NetBeans
...
```

L'exécution du script Ant commence par demander une confirmation à l'utilisateur avant de procéder au remplacement des fichiers :

```
...  
[input] The target source files will be overwritten by the original YAPS project. Do you  
want to continue? (y, n)  
y  
...
```

Dans l'affirmative, la tâche Ant copie respectivement les codes sources Java, les fichiers de ressources Web et les descripteurs :

```
...  
[echo] === Copy Java Sources ===  
[copy] Copying 1 file to c:\Projects\YapsON\PetEx\src\java  
[echo] === Copy Web Resources ===  
[copy] Copying 2 files to c:\Projects\YapsON\PetEx\web  
[echo] === Copy Web Descriptors ===  
[copy] Copying 1 file to c:\Projects\YapsON\PetEx\web\WEB-INF  
BUILD SUCCESSFUL
```

NetBeans rafraîchit automatiquement le projet PetEx :

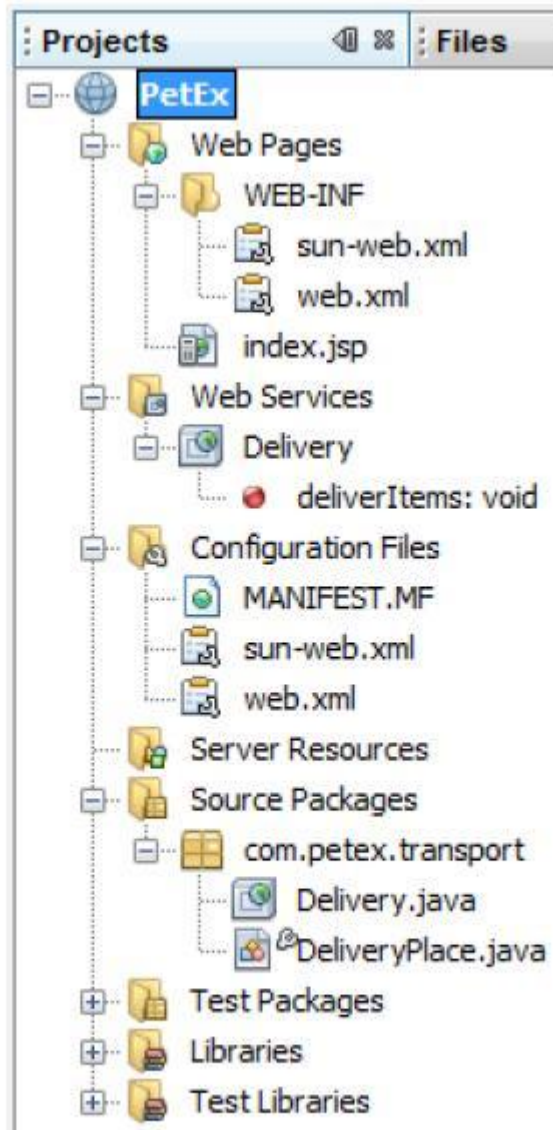


Figure 8-7: Projet PetEx après le transfert

LLVIII-B-3 - Créer le groupe de projet

À partir du menu « File » ou via un clic droit dans la fenêtre « Projects », NetBeans offre la capacité de créer le nouveau groupe de projet:

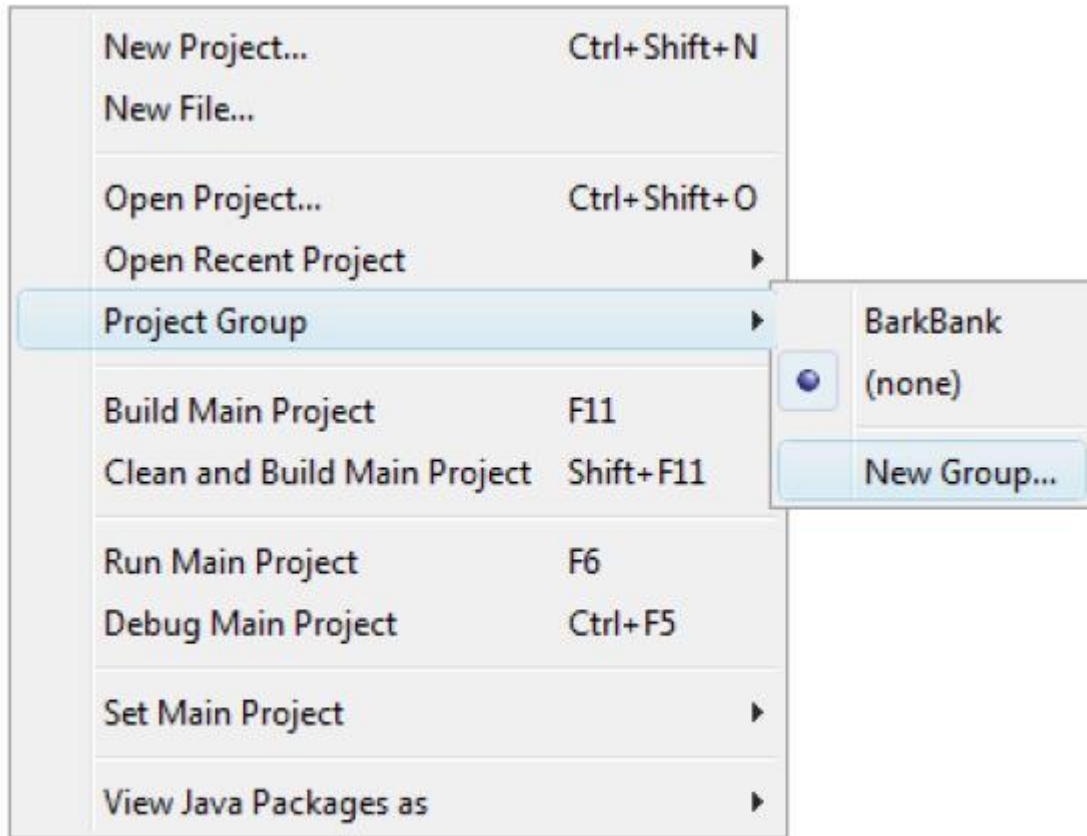


Figure 8-8: Créer un groupe de projet

Sélectionnez l'option « New group ».

La fenêtre suivante permet de nommer le groupe de projet qui est dans notre cas **PetEx** :

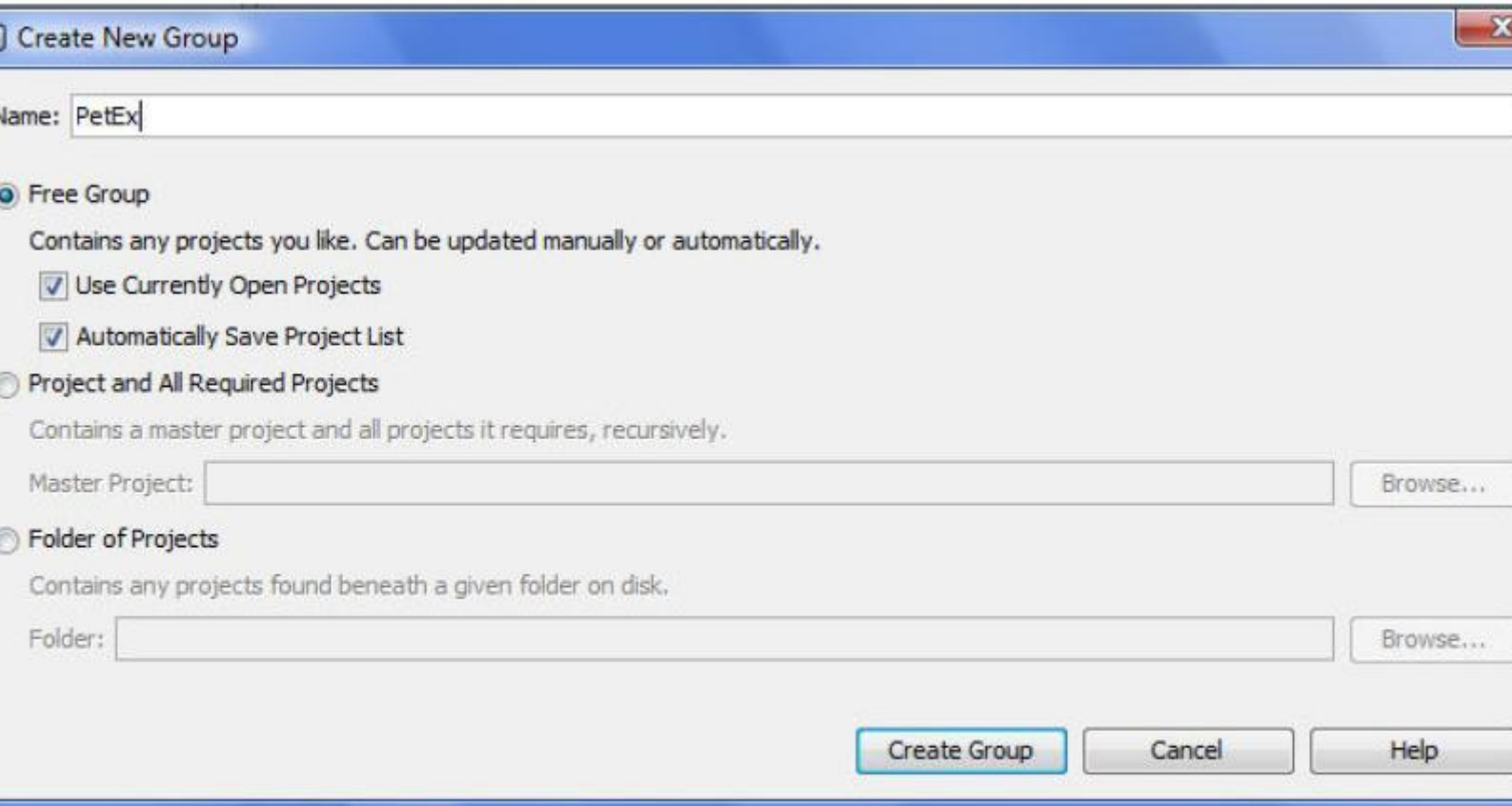


Figure 8-9: Nouveau groupe de projet

Les options par défaut indiquent que les projets en cours de chargement dans l'IDE seront automatiquement ajoutés au groupe de projet PetEx.

Cliquez sur le bouton « **Create Group** ».

LLVIII-C - Compiler le projet

Un clic droit sur le projet fait apparaître le menu contextuel proposant les options pour compiler le projet (« Build » ou « Clean and Build ») :

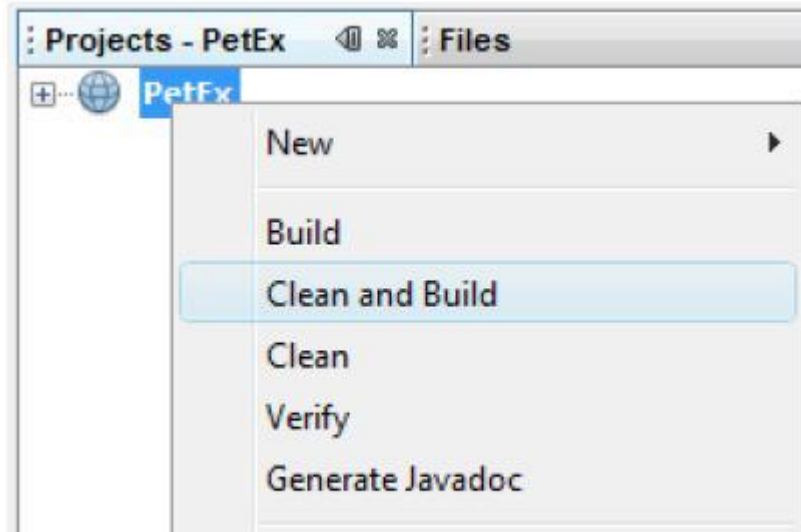


Figure 8-10: Compiler le projet

Cliquez sur l'option « Clean and Build ».

La fenêtre « Output » affiche les traces de la compilation :

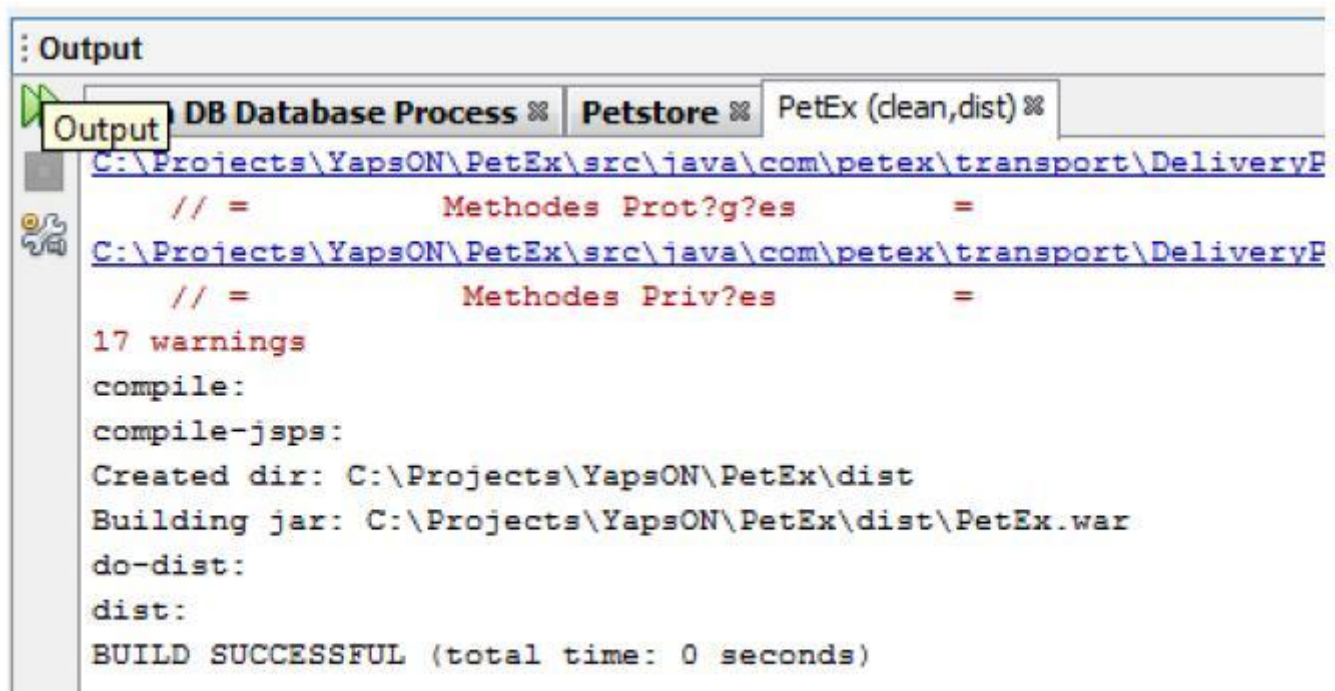


Figure 8-11: Résultat de la compilation

Les messages intéressants sont les suivants :

```

...
compile:
compile-jsp:
Created dir: C:\Projects\YapsON\PetEx\dist
Building jar: C:\Projects\YapsON\PetEx\dist\PetEx.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 2 seconds)

```

Ces traces indiquent que le projet a été correctement compilé (« BUILD SUCCESSFUL ») et archivé dans le fichier « PetEx.war ».

LLVIII-D - Déployer le projet

Vous allez maintenant procéder au déploiement du projet sous GlassFish.

Un clic droit sur le projet fait apparaître le menu contextuel proposant l'option de déploiement (« Deploy ») :

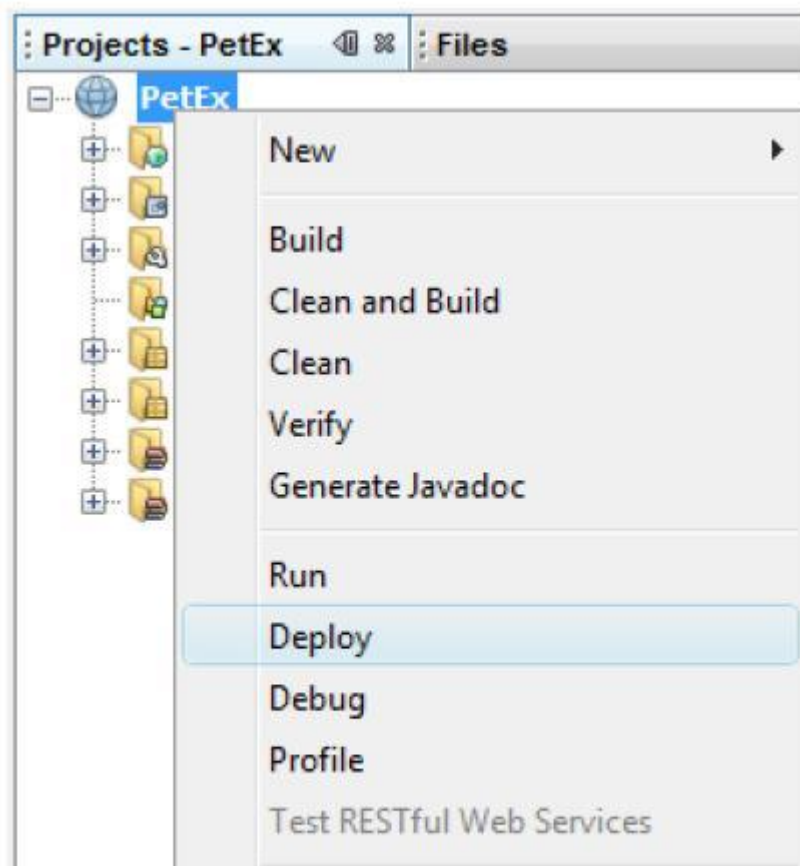


Figure 8-12: Déployer le projet

Cliquez sur l'option « Deploy ».

Si nécessaire, NetBeans se charge de démarrer le serveur d'applications « PetStore ».

Si l'application PetEx était déjà déployée sur le serveur, NetBeans se charge de la supprimer (Undeploying) pour la remplacer par la nouvelle version.

La fenêtre « Output » trace les messages suivants :

```
...
deployment started : 0%
deployment finished : 100%
Deploying application in domain completed successfully
...
All operations completed successfully
run-deploy:
```

BUILD SUCCESSFUL (total time: 4 seconds)

Le déploiement terminé, vous pouvez explorer le contenu de l'application PetEx sur le serveur PetStore.

Rendez-vous sur l'onglet "Services" et sélectionnez le serveur "Petstore".

Le noeud « Applications -> Web Applications -> PetEx» affiche les ressources et services exposés par l'application :

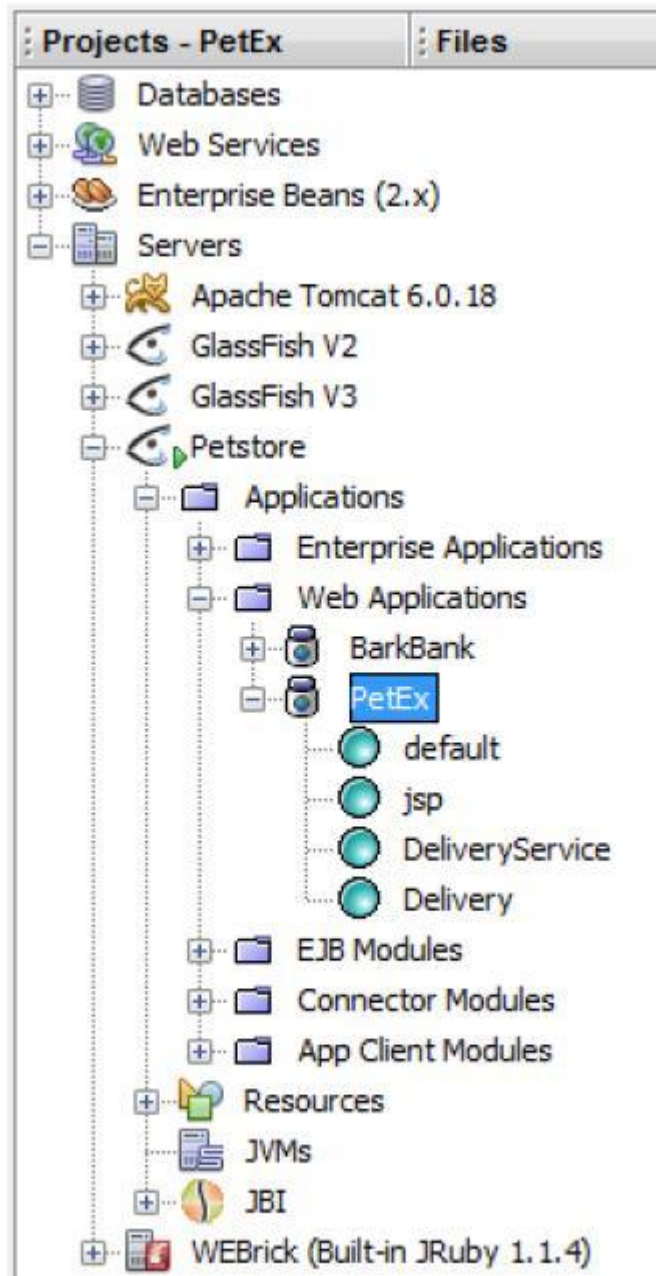


Figure 8-13: Application PetEx sous GlassFish

L'application PetEx est accessible à l'adresse <http://localhost:8080/petex> et se présente comme suit:



Figure 8-14: Page d'accueil de PetEx

À ce stade, vous pouvez inspecter le WSDL ou tester le service tel que décrit dans le livre de référence.

LLVIII-E - Tester le projet

Nous sommes maintenant en mesure de tester l'application PetEx en invoquant le Web Service « DeliveryService ».

Un Web Service peut être testé de différentes manières. Nous allons ici nous contenter de quelques unes d'entre elles disponibles sous NetBeans.

LLVIII-E-1 - Méthode 1: Web Service Tester de GlassFish

Au travers de la fenêtre « Projects », un clic droit sur le Web Service « Delivery » propose le menu contextuel suivant:

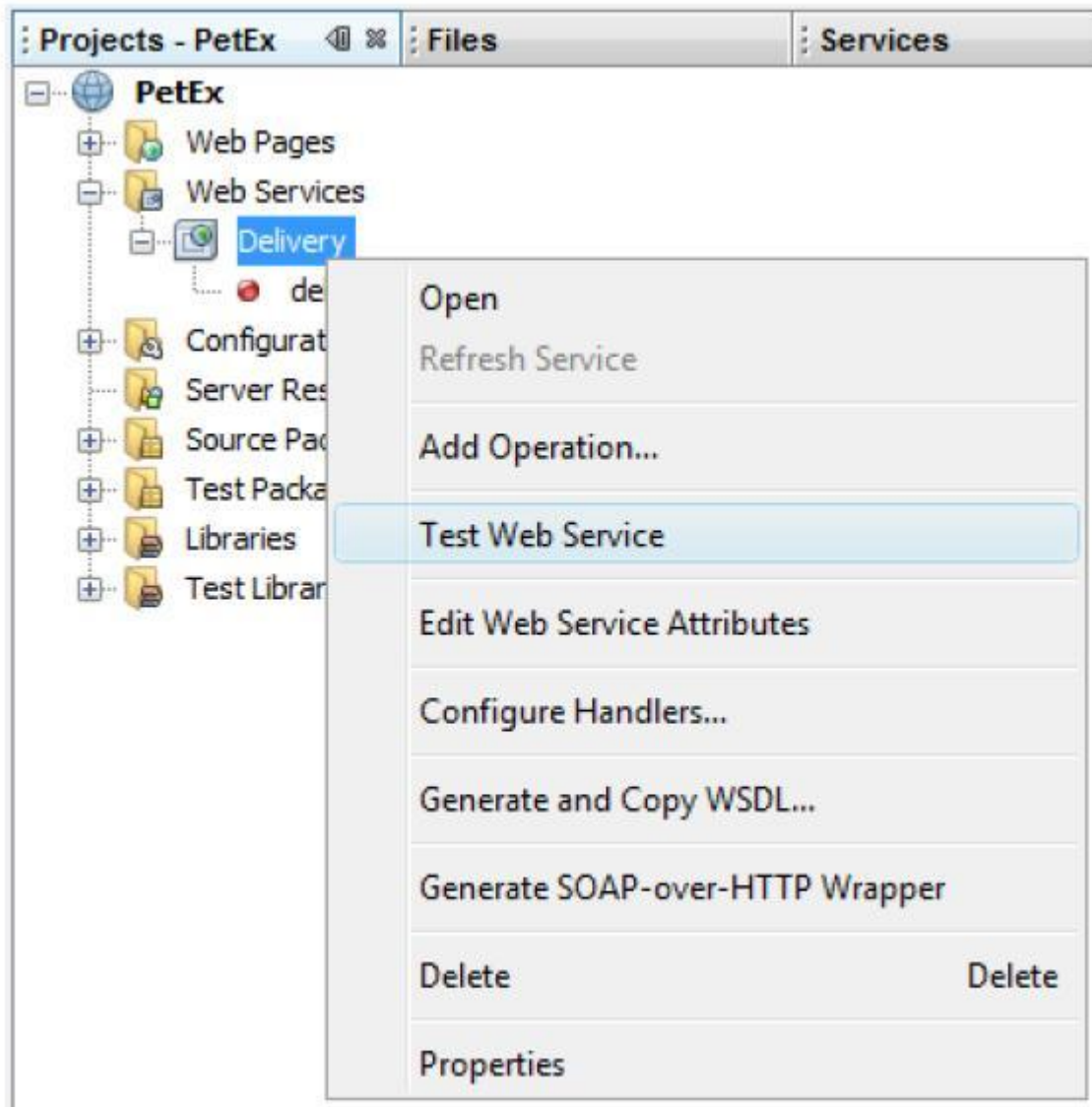


Figure 8-15: Tester le Web Service

L'option « **Test Web Service** » invoque le mécanisme de tests de Web Service de GlassFish qui a son tour invoque le Web Service « DeliveryService » au travers de l'URL suivante:

<http://localhost:8080/petex/DeliveryService?Tester>

Cette mécanique de test n'est pas vraiment appropriée pour tester le Web Service « DeliveryService ». La raison est que la requête à transmettre contient un message complexe.

LLVIII-E-2 - Méthode 2: Web Service Tester de NetBeans

Cette méthode consiste à enregistrer le Web Service sous NetBeans afin de bénéficier d'un outil de test un peu plus évolué que celui offert par GlassFish.

Les Web Services sont répertoriés dans la fenêtre « Services » sous le noeud « Web Service » :

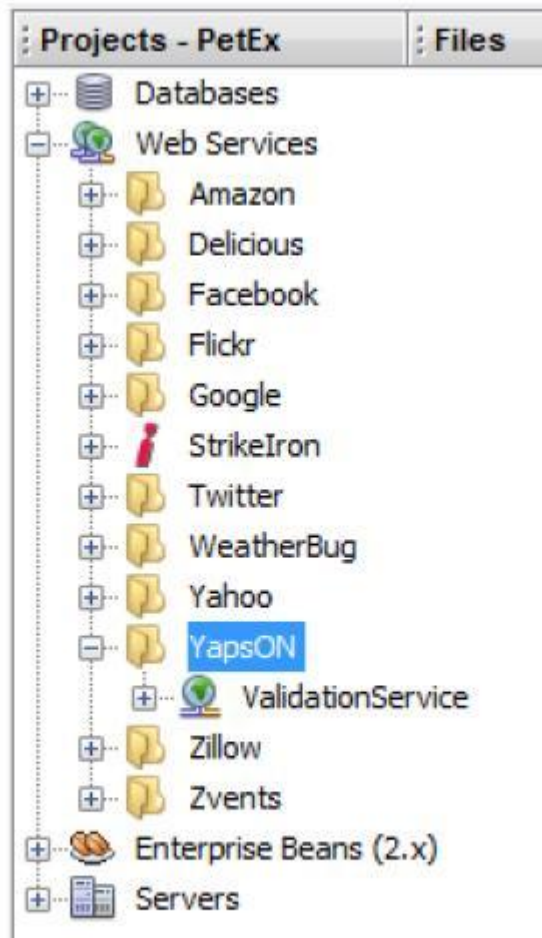


Figure 8-16: Liste des Web Services sous NetBeans

Le groupe de Web Service « YapsON » a déjà été créé précédemment.

Nous allons nous concentrer sur l'ajout de la référence du Web Service de PetEx.

Un clic droit sur le groupe « YapsON » offre un ensemble d'options parmi lesquelles celles permettant d'ajouter un Web Service:

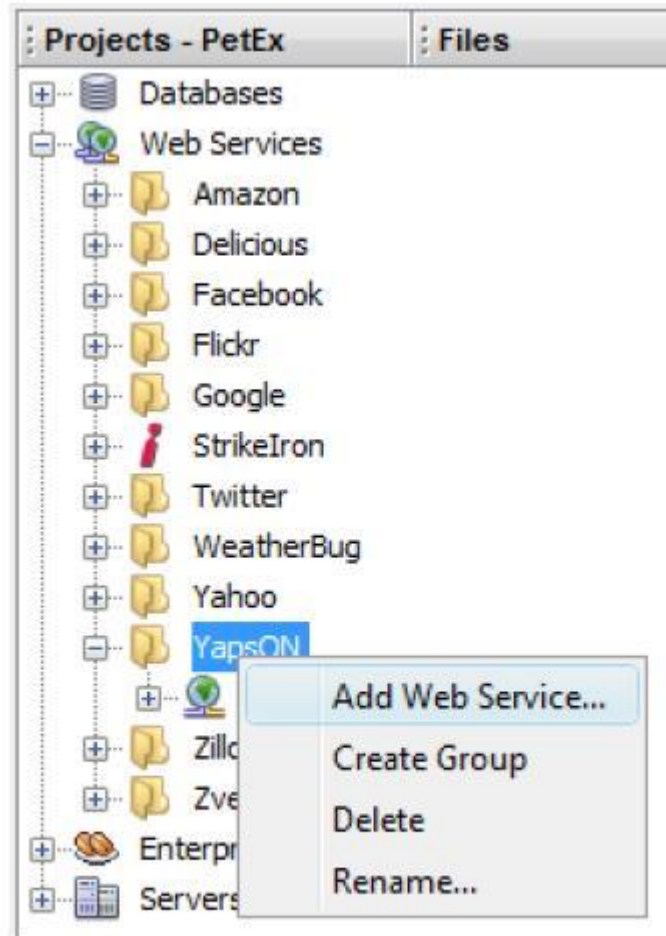


Figure 8-17: Ajout d'un Web Service

Le formulaire permet d'identifier le Web Service à partir de son WSDL:

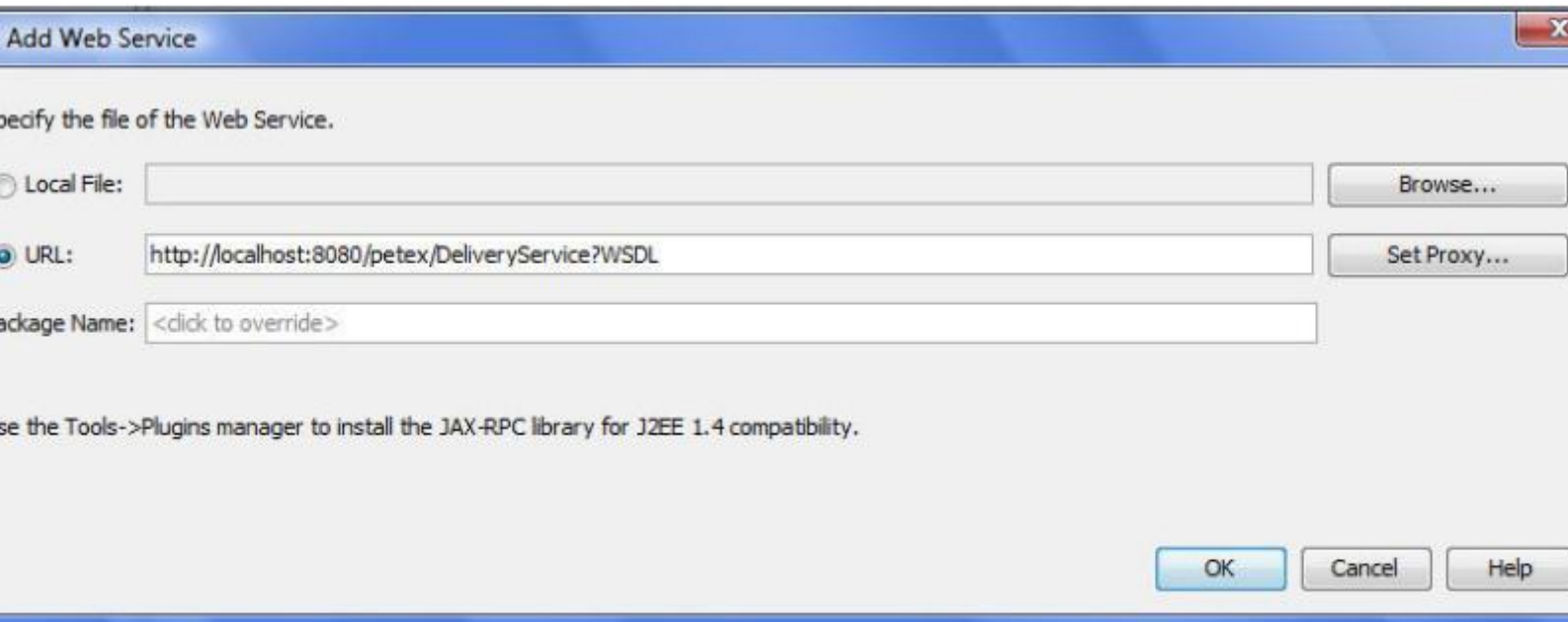


Figure 8-18: Localiser le WSDL

Veuillez localiser le WSDL du Web Service à partir de son URL :

<http://localhost:8080/petex/DeliveryService?WSDL>

Cliquer sur le bouton "OK" pour ajouter la référence au Web Service sous NetBeans.

Le Web Service « **DeliveryService** » est maintenant ajouté à la liste des Web Service :

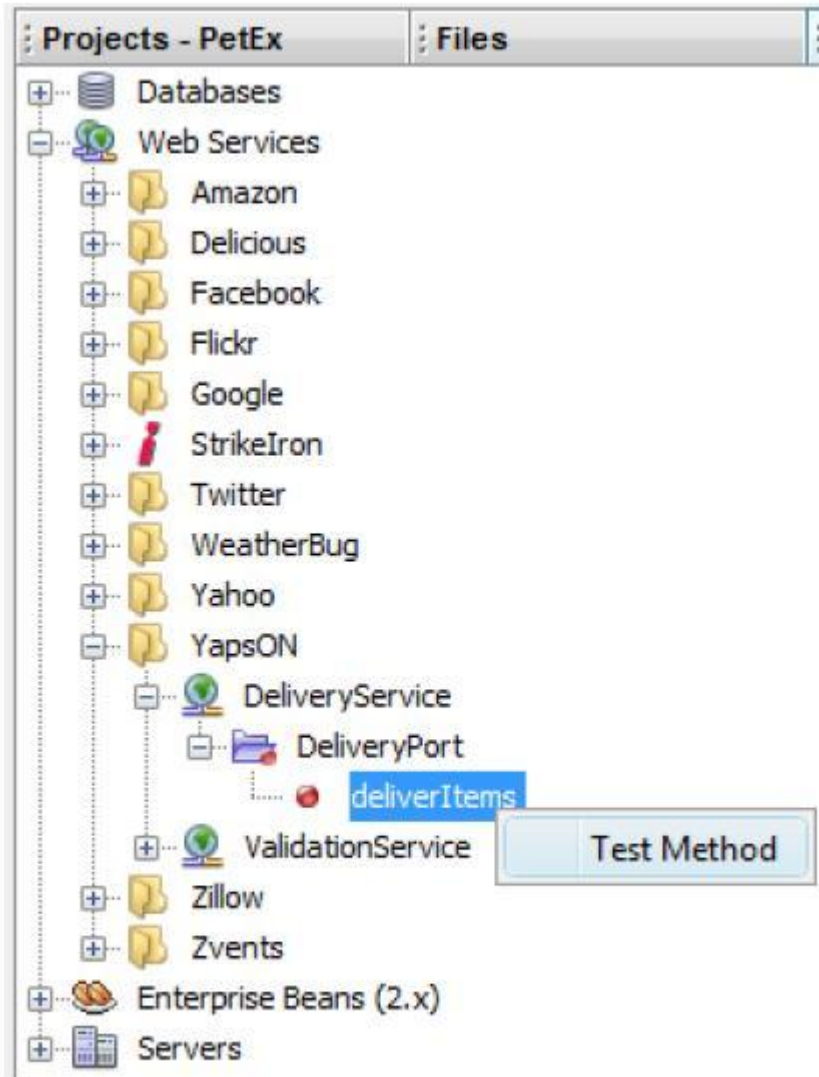


Figure 8-19: Référence de DeliveryService

Un clic droit sur l'opération « deliverItems » permet de tester la méthode au travers de l'écran suivant :

Test Web Service Method

Test Web Service Method: deliverItems
Enter the values for the parameters and press the "Submit" button.

| Type | Name | Value |
|-----------------------------------|------|---|
| com.petex.transport.DeliveryPlace | arg0 | com.petex.transport.DeliveryPlace@13f5694 |
| com.petex.transport.DeliveryPlace | arg1 | com.petex.transport.DeliveryPlace@2d07cc |
| java.lang.String | arg2 | |

Results:

Figure 8-20: Écran de test du Web Service

La partie supérieure concerne la requête à émettre tandis que la partie inférieure se charge d'afficher le message de retour. Le bouton « Submit » permet d'envoyer la requête au Web Service.

Testez le Web Service avec les données suivantes:

| | | |
|----------------------|---------|-----------------|
| Delivery From | City | Liège |
| | Contact | Said Eloudrhiri |
| | Country | BE |
| | State | None |
| | Street | My street |
| | Zipcode | 1234 |

| | | |
|--------------------|---------|-------------------|
| Delivery To | City | Paris |
| | Contact | Antonio Goncalves |
| | Country | FR |
| | State | None |
| | Street | Your street |
| | Zipcode | 5678 |

| | | |
|------------------|--------|----------|
| Reference | Number | 12345678 |
|------------------|--------|----------|

Test Web Service Method: deliverItems

Enter the values for the parameters and press the "Submit" button.

| Type | Name | Value |
|-----------------------------------|---------|--|
| com.petex.transport.DeliveryPlace | arg0 | com.petex.transport.DeliveryPlace@9539b0 |
| ● java.lang.String | city | Liège |
| ● java.lang.String | contact | Said Eloudrhiri |
| ● java.lang.String | country | BE |
| ● java.lang.String | state | None |
| ● java.lang.String | street | My street |
| ● java.lang.String | zipcode | 1234 |
| com.petex.transport.DeliveryPlace | arg1 | com.petex.transport.DeliveryPlace@b30577 |
| ● java.lang.String | city | Paris |
| ● java.lang.String | contact | Antonio Goncalves |
| ● java.lang.String | country | FR |
| ● java.lang.String | state | None |
| ● java.lang.String | street | Your street |
| ● java.lang.String | zipcode | 5678 |
| ● java.lang.String | arg2 | 12345678 |

Submit

Figure 8-21: Requête de test

Cliquez sur le bouton « **Submit** » pour envoyer la requête au Web Service « DeliveryService ».

La réponse retournée par le Web Service est vide :

sults:

| type | Value |
|------|-------|
| void | |

Figure 8-22: Message de réponse

En effet, le code de retour du Web Service « DeliveryService » est de type « void ».

Par contre, le Web Service trace la requête en utilisant le « logger » applicatif défini pour « com.petex.transport ».

Si l'on se reporte sur la fenêtre « Output » et en particulier sur les traces générées par le serveur « Petstore », on retrouvera les traces suivantes :

```

..
ENTRY
Delivery Order Received
Deliver from DeliveryPlace{contact='Saïd Eloudrhiri', street='My street', city='Liège',
state='None', zipcode='1234', country='BE'}
Deliver to DeliveryPlace{contact='Antonio Goncalves', street='Your street', city='Paris',
state='None', zipcode='5678', country='FR'}
Reference n° 12345678
RETURN

```

Ces traces montrent que le Web Service a bien reçu le message de requête.

LLVIII-F - Conclusion

Les sources de l'application « PetEx » ont été transférées sous NetBeans dans une arborescence conforme au format préconisé par le Java Blueprints.

Les tests que nous avons réalisés attestent que le Web Service de l'application PetEx a été correctement compilé et est déployé sur le serveur.

IX - Importer le projet Yaps - côté serveur

Le projet Yaps-Server est une application de type « Enterprise Applications » qui a pour fonction d'héberger la logique serveur de l'application Yaps.

Le projet Yaps-Server est composé des modules suivants :

- **Module EJB** : Entity, Stateless, Statefull, Message Driver Bean
- **Module Web** : JSF

L'ensemble des opérations qui suivent est réalisé sous NetBeans.

LLIX-A - Créer un nouveau projet

LLIX-A-1 - Fermeture des projets

Vous allez créer un nouveau groupe de projets qui sera spécifique à Yaps-Server.

Si c'est le cas, il y a d'abord lieu de fermer l'ensemble des projets actuellement ouverts sous NetBeans.

Sélectionnez l'onglet « Projects ».

Un clic droit fait apparaître le menu contextuel. L'option « Project Group (none) » permet de fermer l'ensemble des projets ouverts :

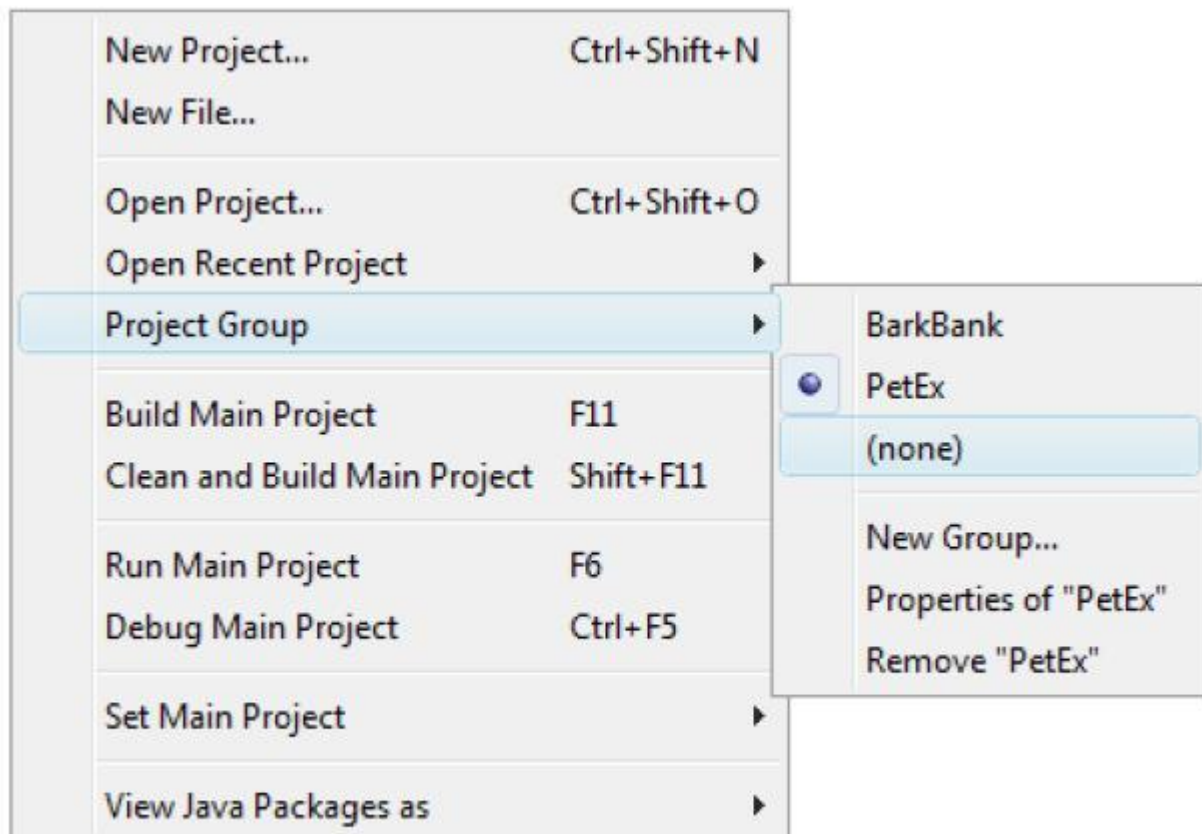


Figure 9-1: Fermer les projets ouverts

LLIX-A-2 - Type de projet

Créez un nouveau projet de type Enterprise Application ».

Pour cela, sélectionnez l'onglet « Project » et créez un nouveau projet via le menu « File -> New Project ».

Le projet que vous allez créer est de type « Java EE > Enterprise Application » :

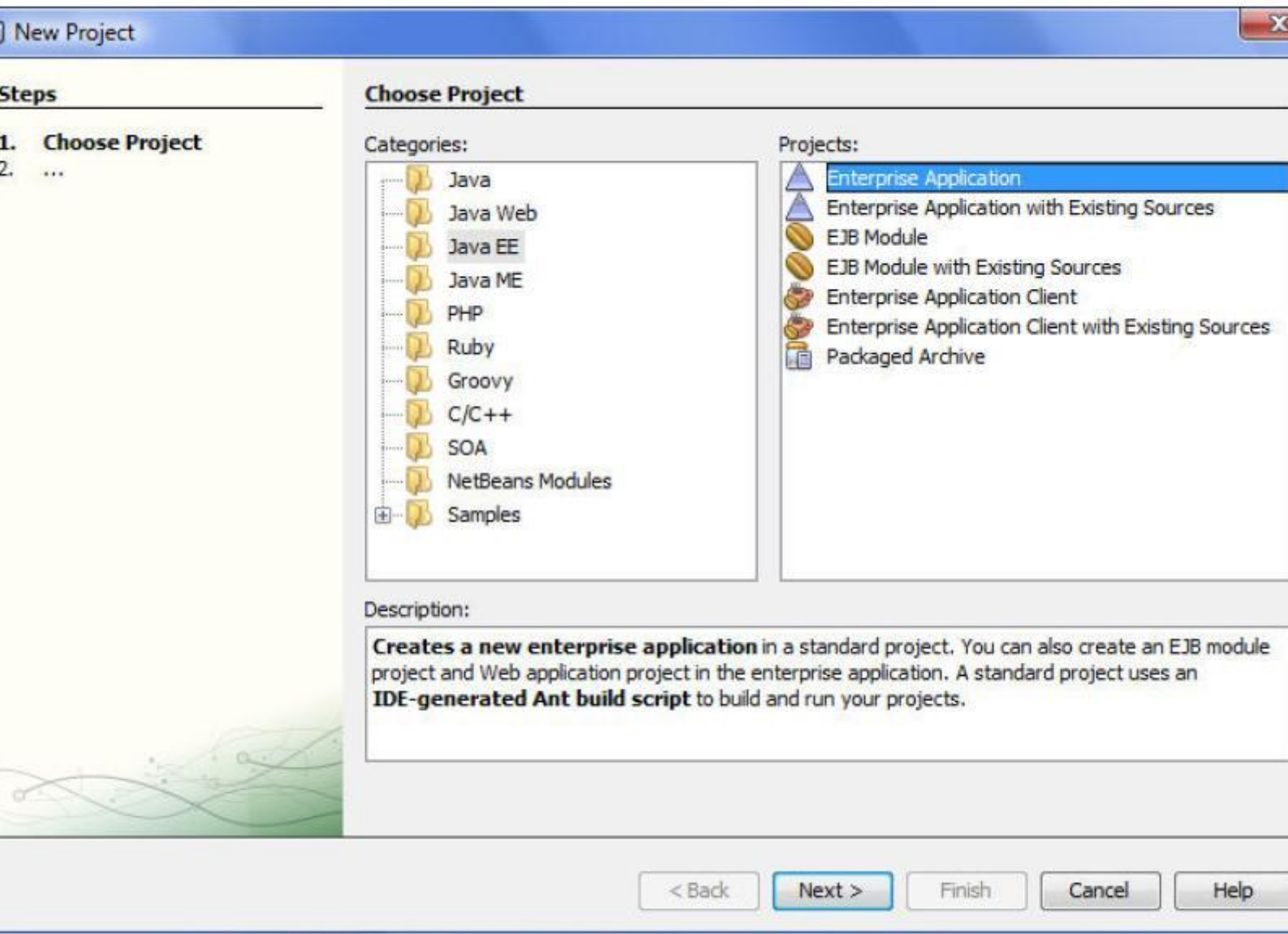


Figure 9-2: Nouveau projet Enterprise Application

Cliquez sur le bouton « Next » pour continuer.

LLIX-A-3 - Nom et répertoires

L'écran suivant vous permet de nommer le projet, et de définir le dossier cible:

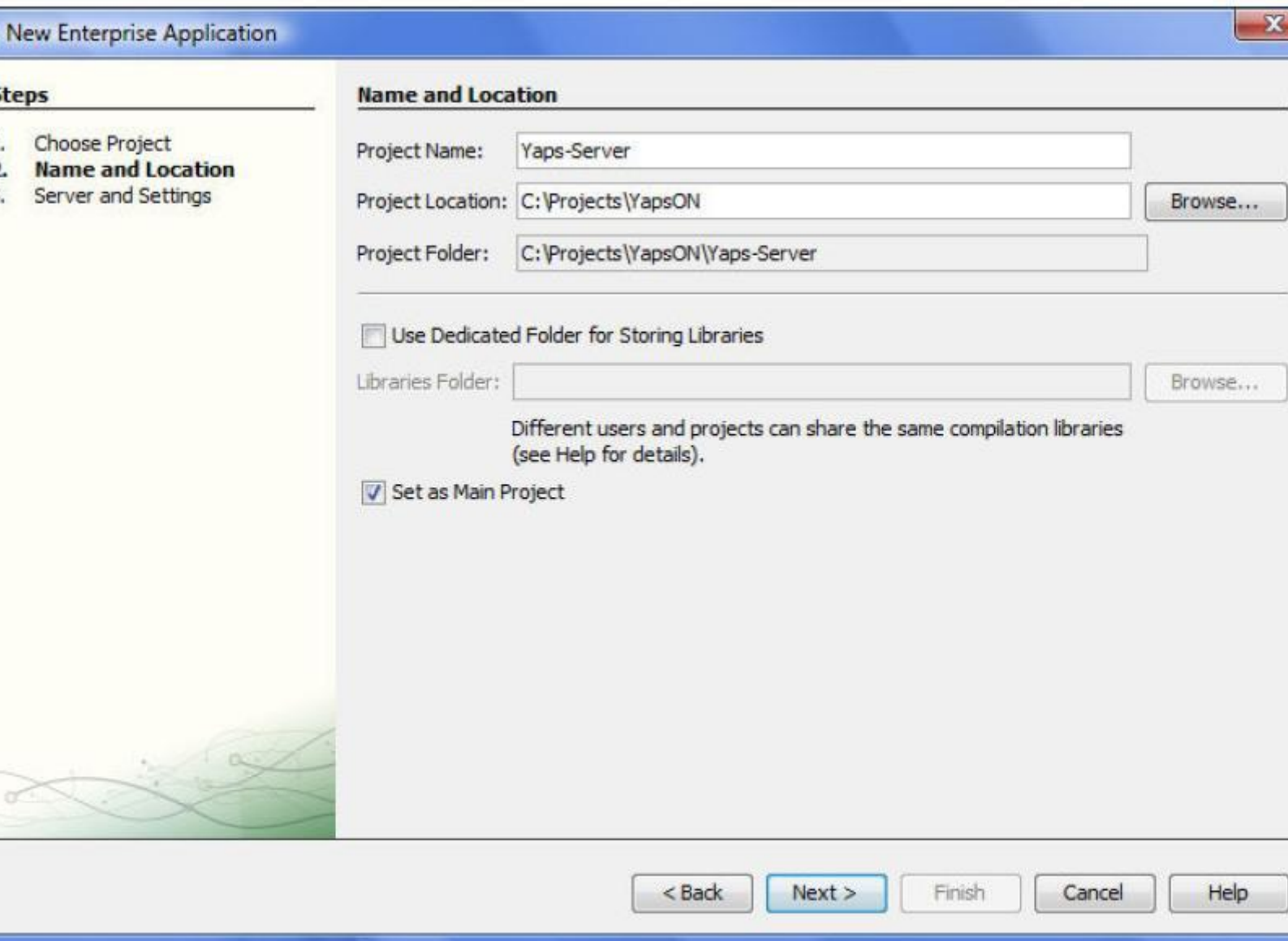


Figure 9-3: Nom du projet et répertoire

Les informations à introduire sont les suivantes:

| | |
|------------------------|--------------------------------|
| Project Name: | Yaps-Server |
| Project Folder: | C:\Projects\YapsON\Yaps-Server |

Project Folder

Le répertoire du projet est donné ici à titre d'exemple et doit correspondre au répertoire où vous avez décompressé le fichier d'archives de l'application YapsON (voir [ici](#)).

Cliquez sur le bouton « Next » pour continuer.

LLIX-A-4 - Configuration du serveur

Cette étape vous permet d'identifier le serveur d'applications « PetStore » (voir [ici](#)) sur lequel sera déployée l'application PetEx ainsi que les modules (EJB et Web) associés au projet:

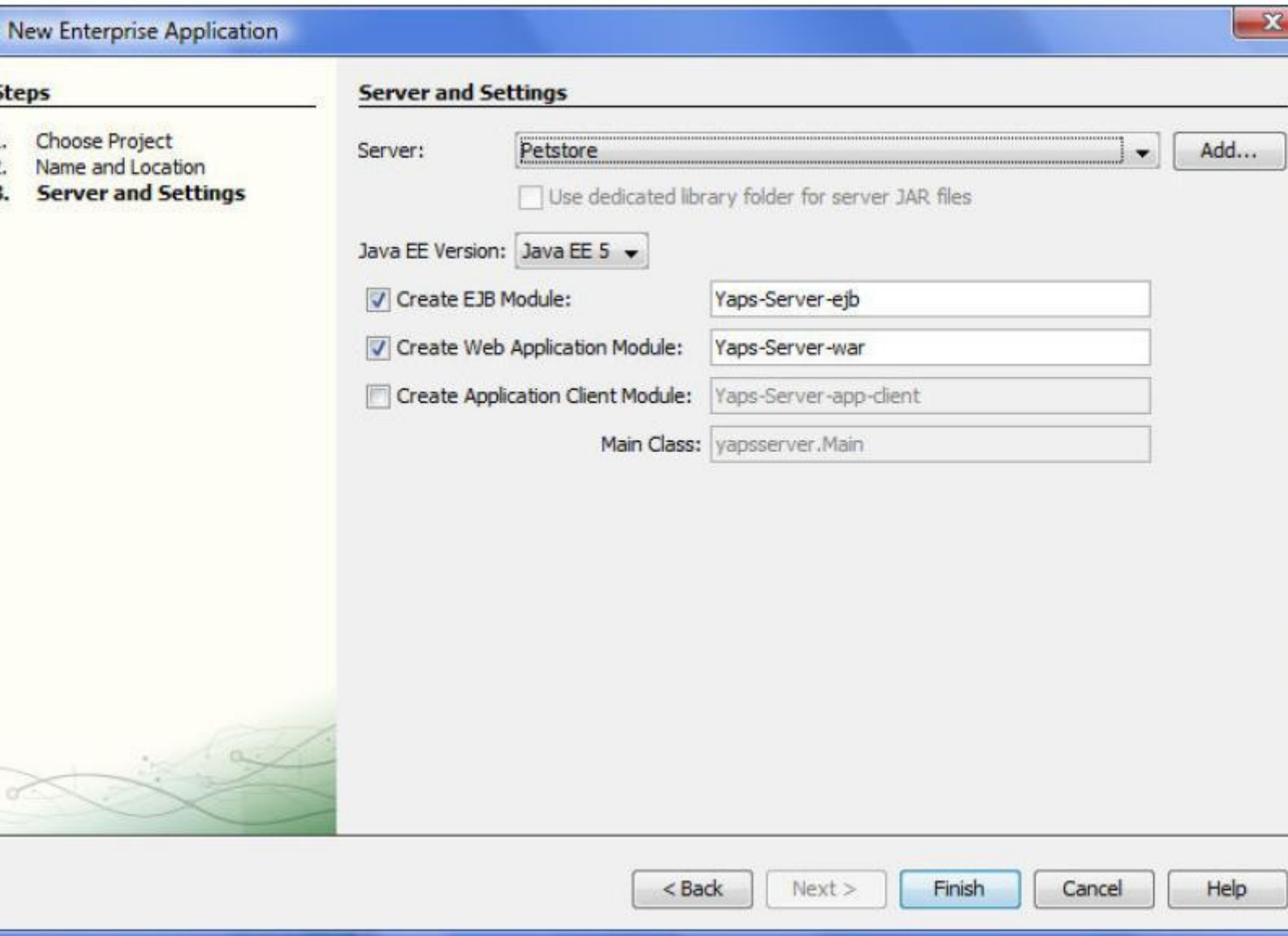


Figure 9-4: Server settings

La configuration utilisée pour le déploiement est la suivante :

| | |
|--------------------------------|-----------------|
| Server: | Petstore |
| EJB Module: | Yaps-Server-ejb |
| Web Application Module: | Yaps-Server-war |

Cliquez sur le bouton « Finish » pour terminer le processus de création du projet.

LLIX-A-5 - Génération du projet

Le projet est maintenant généré et présente l'ensemble des dossiers suivants :

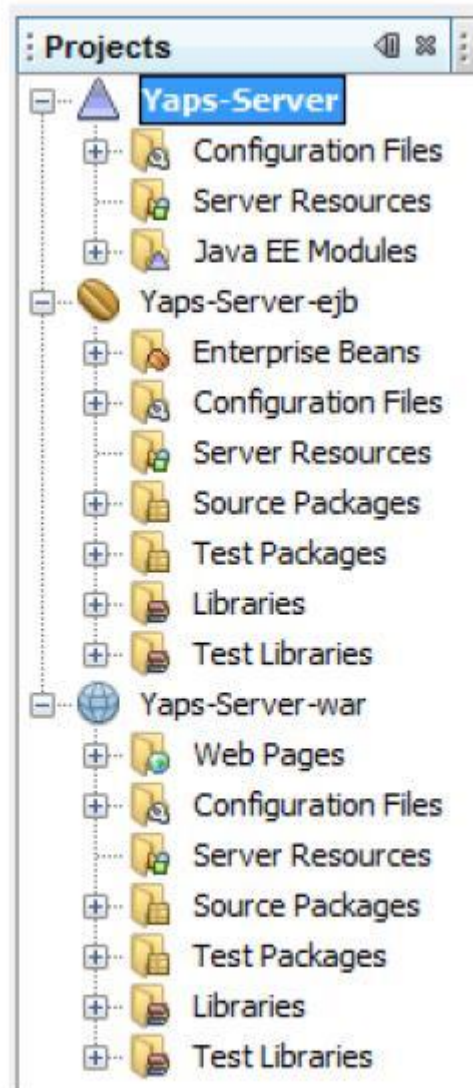


Figure 9-5: Dossiers du projet

Vous êtes prêt à importer le code source du projet Yaps (côté serveur) de l'application Yaps vers l'arborescence de NetBeans.

LLIX-B - Transférer le projet sous NetBeans

L'opération de transfert du projet Yaps consiste à copier les fichiers de la partie serveur de l'application Yaps vers l'application YapsON.

Pour faciliter cette opération, le script Ant « **yapson.xml** » (décrit dans la section 3.5) automatise la copie des fichiers.

Assurez-vous d'avoir configuré le fichier « yapson.properties » tel que décrit dans la section 4.

LLIX-B-1 - Positionner les variables d'environnement

Les commandes que vous allez exécuter nécessitent de positionner les variables d'environnement pour l'exécution des tâches Ant.

Rendez-vous dans le répertoire de l'application YapsON et exécutez le fichier de commande « setenv-yaps.cmd » :

```
C:> cd {yapson_home}
{yapson_home}> setenv-yaps.cmd
```

LLIX-B-2 - Transférer le projet

À partir de la ligne de commande, exécutez la tâche Ant « **transferYapsServer** » chargée de transférer le projet sous NetBeans :

```
%YAPSON-HOME%> ant -f yapson.xml transferYapsServer
Buildfile: yapson.xml
transferYapsServer:
[echo] Transfer to NetBeans the server part of the Yaps project
...
```

L'exécution du script Ant commence par demander une confirmation à l'utilisateur avant de procéder au remplacement des fichiers :

```
...
[input] The target source files will be overwritten by the original YAPS project. Do you
want to continue? (y, n)
y
...
```

La tâche Ant copie ensuite les éléments requis pour le module « Yaps-Server-ejb » :

```
...
[echo] >>>>> Module Yaps-Server-ejb >>>>>
[echo] === Copy Java Sources ===
[echo] entity ...
[copy] Copying 8 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverejb\
src\java\com\yaps\petstore\entity
[echo] exception ...
[copy] Copying 2 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverejb\
src\java\com\yaps\petstore\exception
[echo] mdb ...
[copy] Copying 2 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverejb\
src\java\com\yaps\petstore\mdb
[echo] stateless ...
[copy] Copying 9 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverejb\
src\java\com\yaps\petstore\stateless
[echo] statefull ...
[copy] Copying 3 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverejb\
src\java\com\yaps\petstore\stateful
[echo] util ...
[copy] Copying 2 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverejb\
src\java\com\yaps\petstore\util
[echo] === Copy Persistence Unit ===
[copy] Copying 1 file to c:\Projects\YapsON\Yaps-Server\Yaps-Server-ejb\src\conf
...
```

Ensuite, les ressources nécessaires pour le module "Yaps-Server-war" sont transférées :

```
...
[echo] >>>>> Module Yaps-Server-war >>>>>
[echo] === Copy Web Sources ===
[copy] Copying 4 files to c:\Projects\YapsON\Yaps-Server\Yaps-Serverwar\
src\java\com\yaps\petstore\jsf
[echo] === Copy Web Resources ===
[copy] Copying 39 files to c:\Projects\YapsON\Yaps-Server\Yaps-Server-war\web
[echo] === Copy Web Descriptors ===
[copy] Copying 2 files to c:\Projects\YapsON\Yaps-Server\Yaps-Server-war\web\WEB-INF
...
```

Pour terminer, les ressources utilitaires sont transférées (dans notre cas le fichier SQL contenant les données de test) :

```
...
[echo] >>>>> Tools >>>>>
[echo] === Copy Test data file ===
[copy] Copying 1 file to c:\Projects\Yaps-Server
BUILD SUCCESSFUL
```

NetBeans rafraîchit automatiquement le projet Yaps-Server :

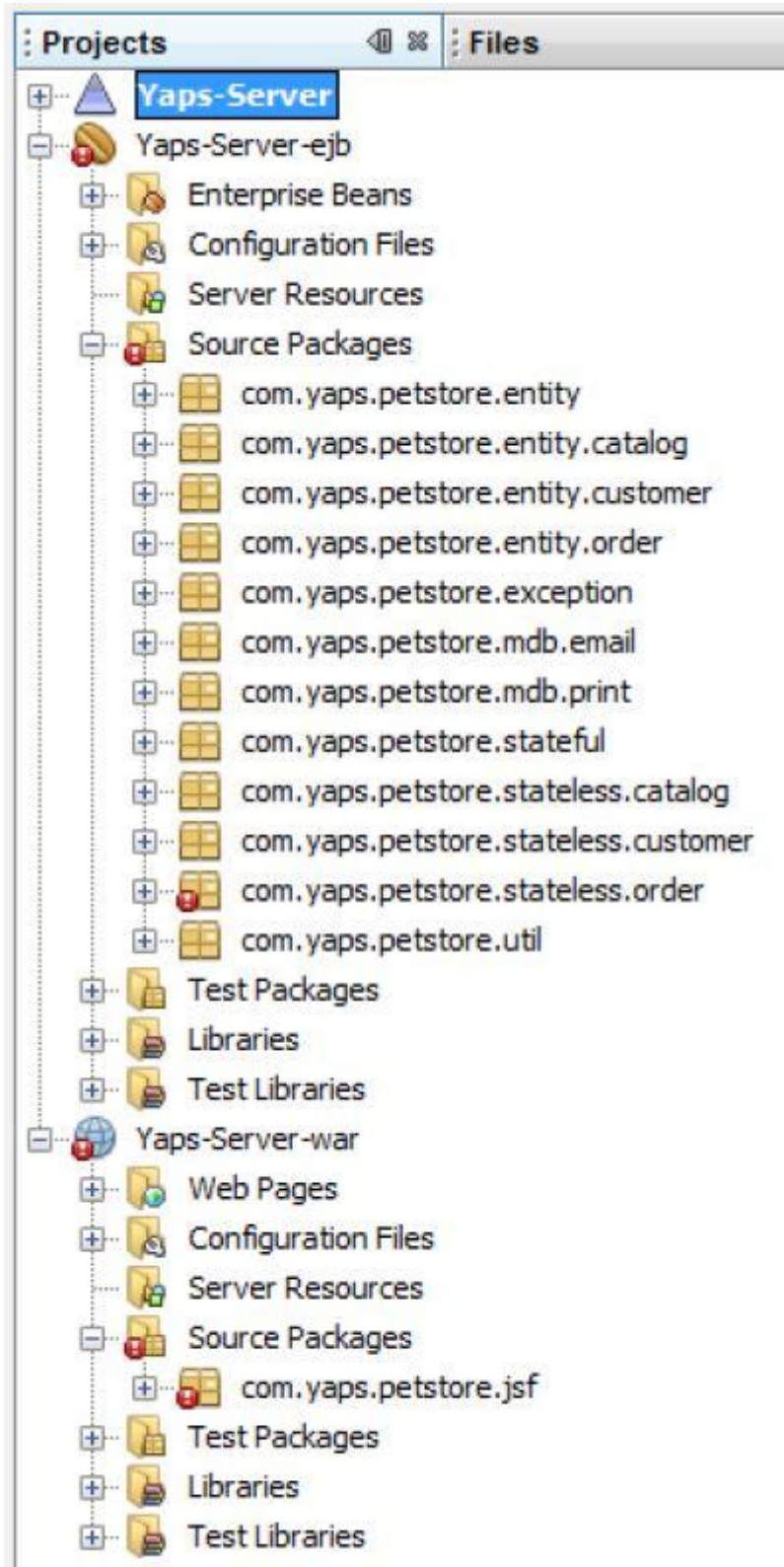


Figure 9-6: Projet Yaps-Server après le transfert

Ouverture des modules

Il se peut que les modules du projet Yaps-Server soient fermés :

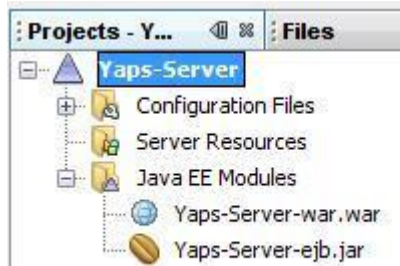


Figure 9-7: Modules fermés

Un clic droit sur le module vous permet d'ouvrir le sous-projet dans l'IDE :

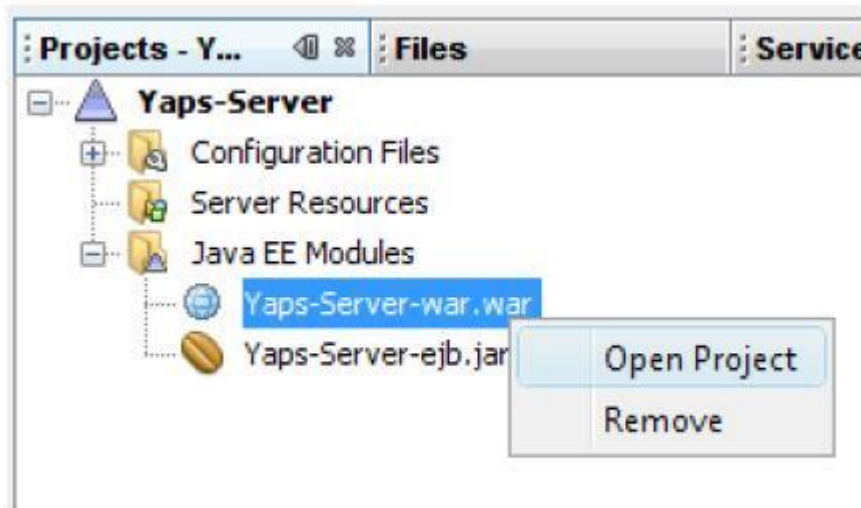


Figure 9-8: Ouverture d'un module

LLIX-B-3 - Référencer les Web Services

Des erreurs apparaissent dans les modules « Yaps-Server-ejb » et « Yaps-Server-war ».

La cause de ces erreurs est que les références des Web Service des projets « BarkBank » et « PetEx » sont manquantes.

LLIX-B-3-a - Référencer DeliveryService

Dans le module « Yaps-Server-ejb », la classe « OrderBean » injecte la référence du Web Service « DeliveryService » issue du projet « PetEx ».

Pour corriger l'erreur, vous allez devoir ajouter le Proxy client du Web Service.

Un clic droit sur le module « Yaps-Server-ejb » fait apparaître le menu contextuel dans vous devez sélectionner l'option « New -> Web Service Client » :

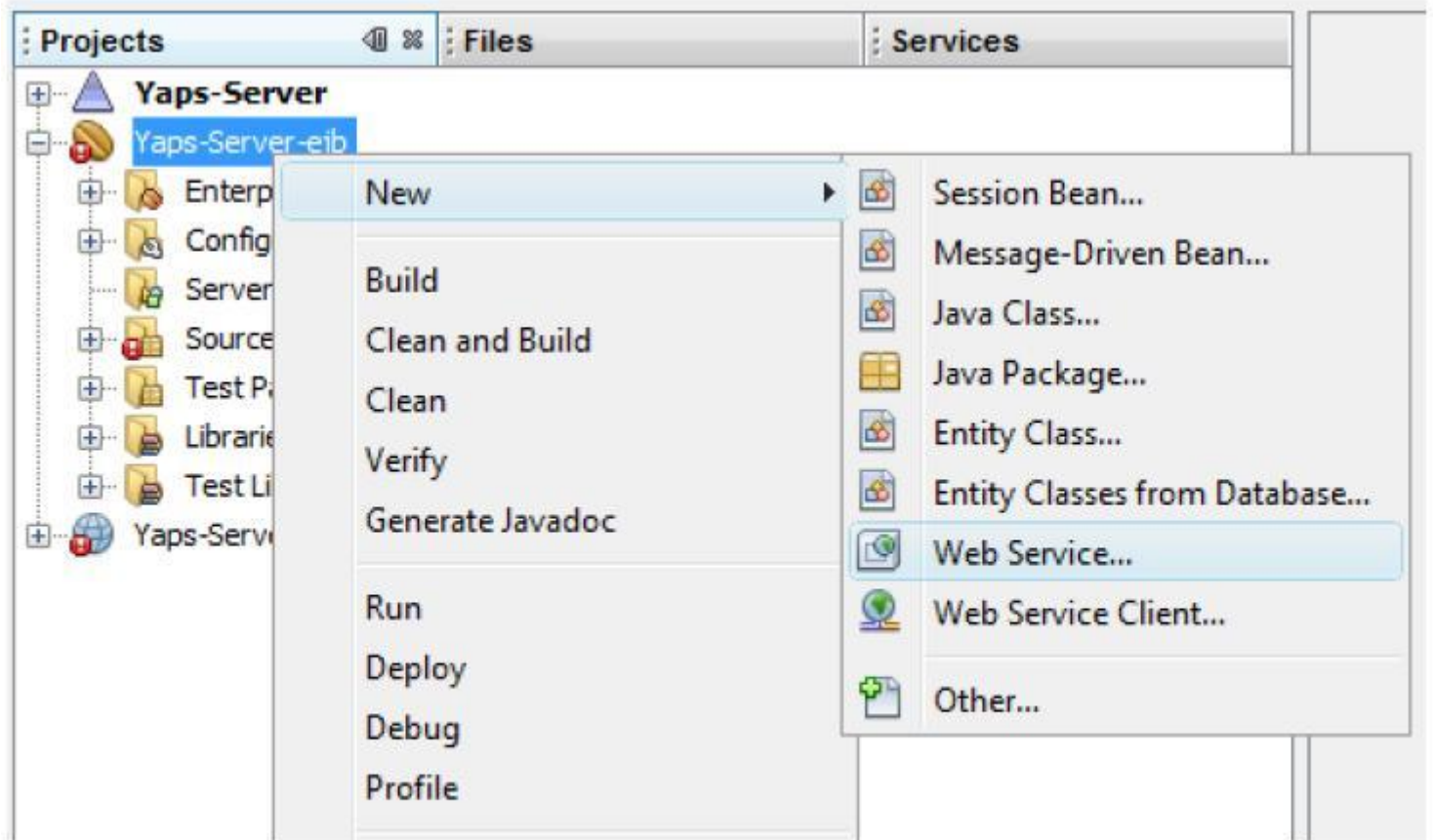


Figure 9-9: Ajout d'un client Web Service

L'écran suivant vous demande de localiser le WSDL du Web Service :

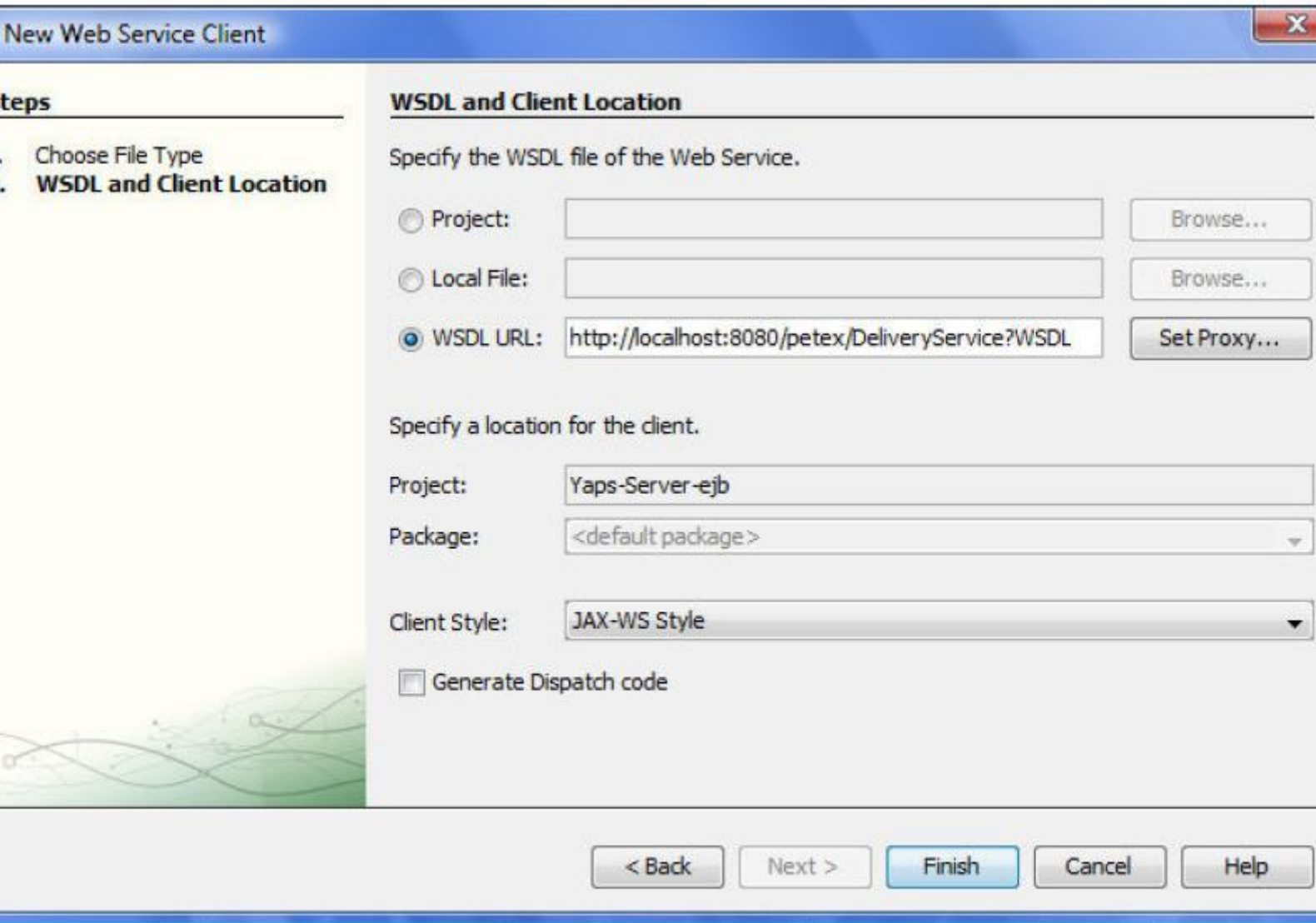


Figure 9-10: Localisation du WSDL

Dans notre cas, vous devez fournir l'URL du WSDL relatif à l'application PetEx:

<http://localhost:8080/petex/DeliveryService?WSDL>

Démarrage du serveur PetStore

Assurez-vous que le serveur « Petstore » est démarré, car dans le cas contraire la tâche de création du client Web Service ne pourra localiser l'URL du WSDL.

Cliquer sur « Finish » pour générer le code du Proxy client du Web Service.

Localisation du WSDL

Nous aurions pu localiser le WSDL soit par projet ou par fichier.

Par **Projet**, cela aurait impliqué que le projet PetEx soit également ouvert dans l'IDE.

Par **Fichier**, cela nécessite de localiser le fichier WSDL généré dans le projet PetEx.

Nous avons préféré illustrer le mode URL pour simuler la situation où le Web Service serait déployé sur un serveur distant ?

À la fin du processus de génération, le dossier « Web Service References » est ajouté au module « Yaps-Server-ejb » et contient le Proxy client du Web Service :

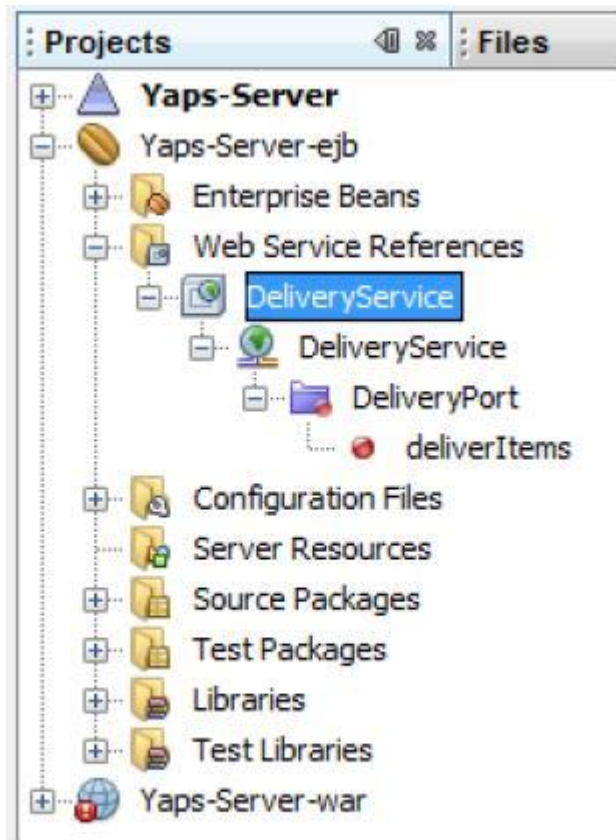


Figure 9-11: Proxy client de DeliveryService

Code source du Proxy client

Le code source Java du Proxy client est généré sous le répertoire suivant :

```
{yapson_home}/Yaps-Server/Yaps-Server-ejb/build/generated/wsimport/client
```

NetBeans va ensuite automatiquement rafraîchir le projet et faire disparaître les erreurs de références repérées dans la classe « OrderBean ».

LLIX-B-3-b - Référencer ValidationService

Dans le module « Yaps-Server-war », la classe « ShoppingCartController » injecte la référence du Web Service « ValidationService » issue du projet « BarkBank ».

Pour corriger l'erreur, vous allez devoir ajouter le Proxy client du Web Service.

Un clic droit sur le module « Yaps-Server-war » fait apparaître le menu contextuel dans vous devez sélectionner l'option « New -> Web Service Client » :

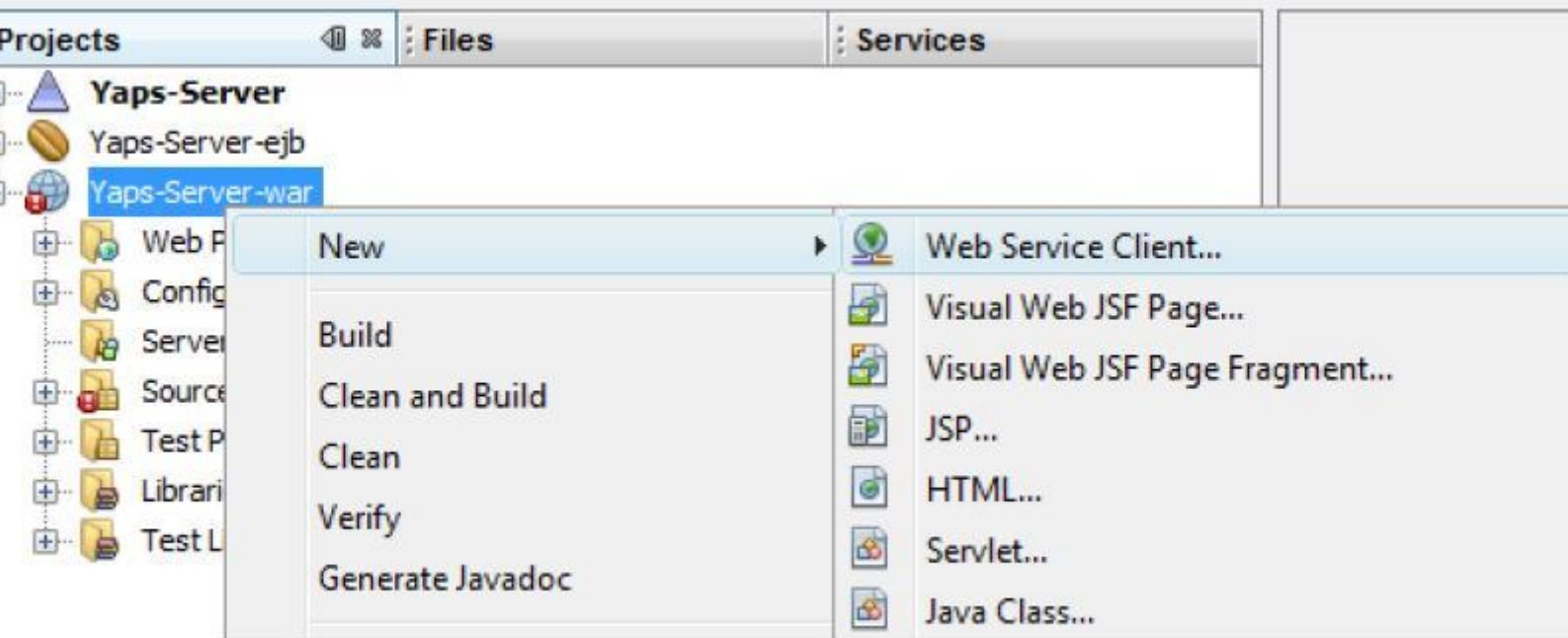


Figure 9-12: Ajout d'un client Web Service

L'écran suivant vous demande de localiser le WSDL du Web Service :

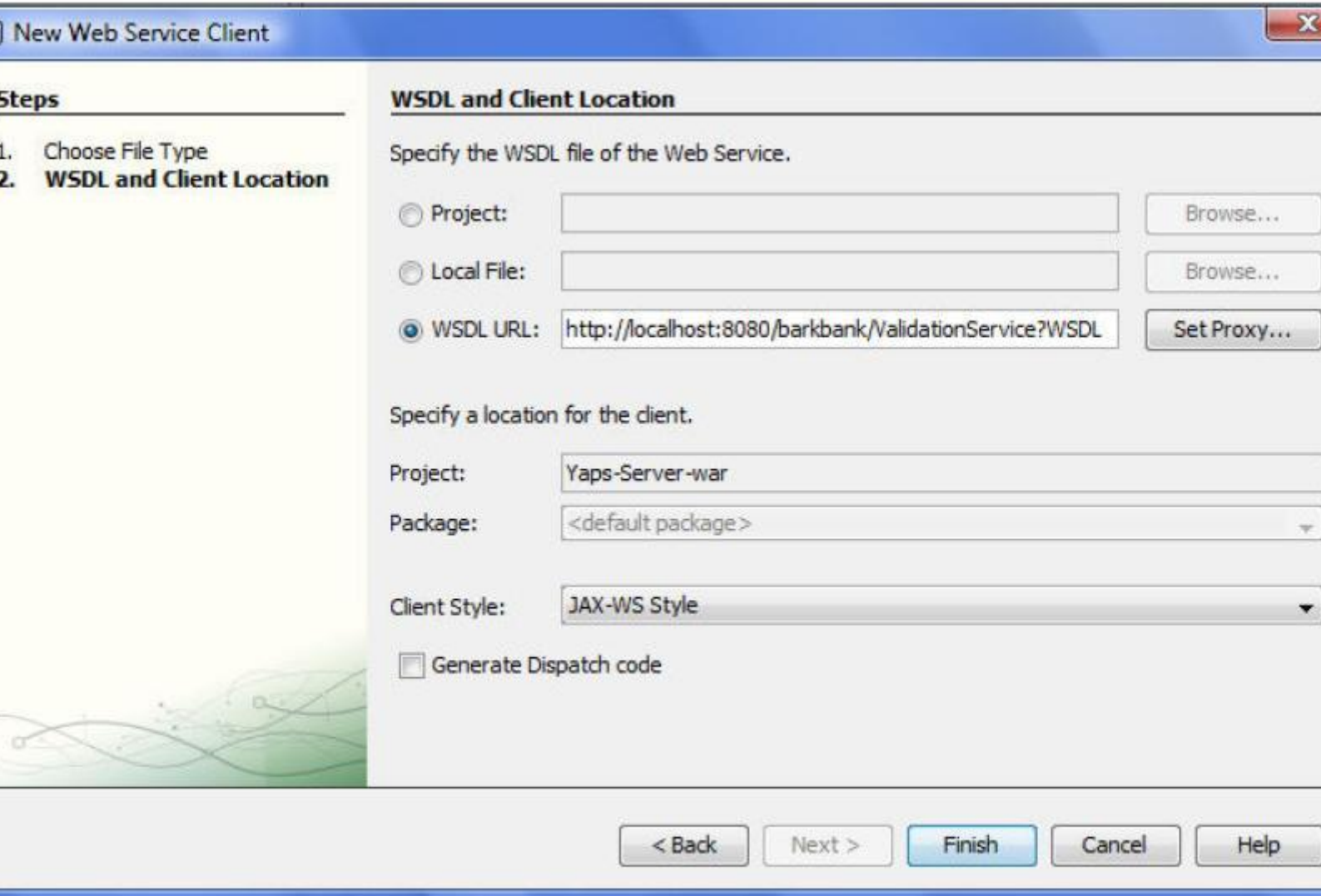


Figure 9-13: Localisation du WSDL

Dans notre cas, vous devez fournir l'URL du WSDL relatif à l'application BarkBank:

<http://localhost:8080/barkbank/ValidationService?WSDL>

Cliquer sur « Finish » pour générer le code du Proxy client du Web Service.

À la fin du processus de génération, le dossier « Web Service References » est ajouté au module « Yaps-Server-war » et contient le Proxy client du Web Service :

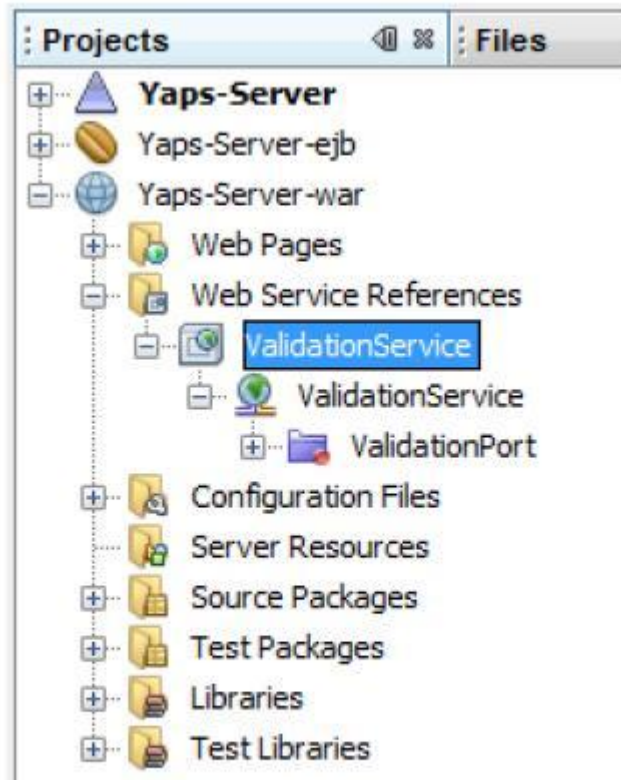


Figure 9-14: Proxy client de ValidationService

Code source du Proxy client

Le code source Java du Proxy client est généré sous le répertoire suivant :

```
{yapson_home}/Yaps-Server/Yaps-Server-war/build/generated/wsimport/client
```

NetBeans va ensuite automatiquement rafraîchir le projet et faire disparaître les erreurs de références repérées dans la classe « ShoppingCartController ».

LLIX-B-4 - Créer le groupe de projet

À partir du menu « File » ou via un clic droit dans la fenêtre « Projects », NetBeans offre la capacité de créer le nouveau groupe de projet:

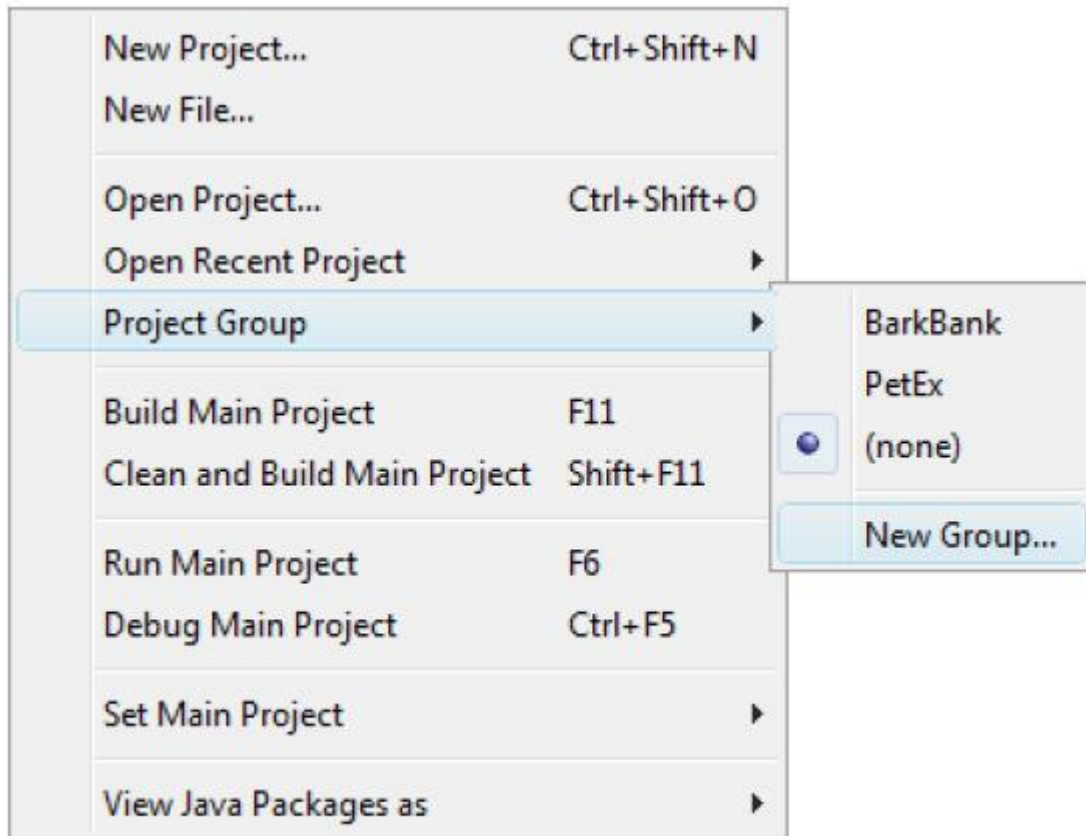


Figure 9-15: Créer un groupe de projet

Sélectionnez l'option « New group ».

La fenêtre suivante permet de nommer le groupe de projet qui est dans notre cas **Yaps-Server** :

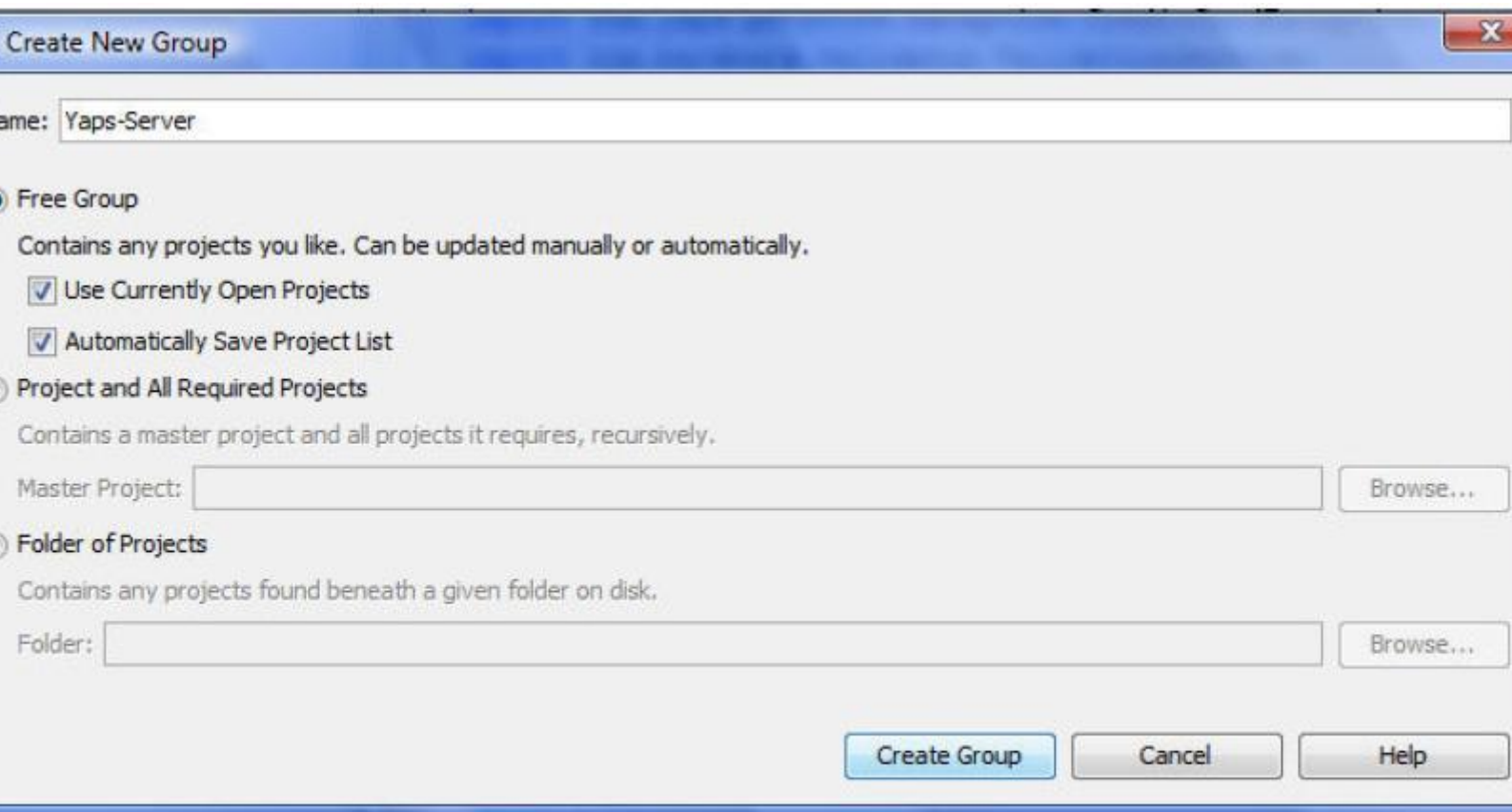


Figure 9-16: Nouveau groupe de projet

Les options par défaut indiquent que les projets en cours de chargement dans l'IDE seront automatiquement ajoutés au groupe de projet.

Cliquez sur le bouton « **Create Group** ».

LLIX-C - Changer le Context Root

Pour être conforme au livre de référence, vous allez changer le **Context Root** de l'application Web qui a été défini par défaut dans le module « Yaps-Server-war ».

Le Context Root est défini dans le fichier de déploiement « sun-web.xml » qui se trouve sous :

Yaps-Server-war -> Configuration Files -> sun-web.xml

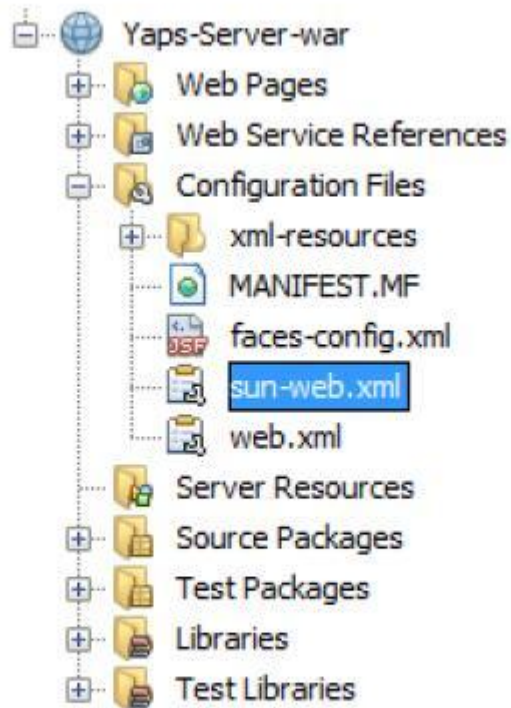


Figure 9-17: Fichier de déploiement « sun-web.xml »

Le fichier « sun-web.xml » est le descripteur de déploiement Web pour SUN GlassFish.

Éditez ce fichier et modifiez l'entrée « Context Root » pour y placer la valeur « /petstore » :

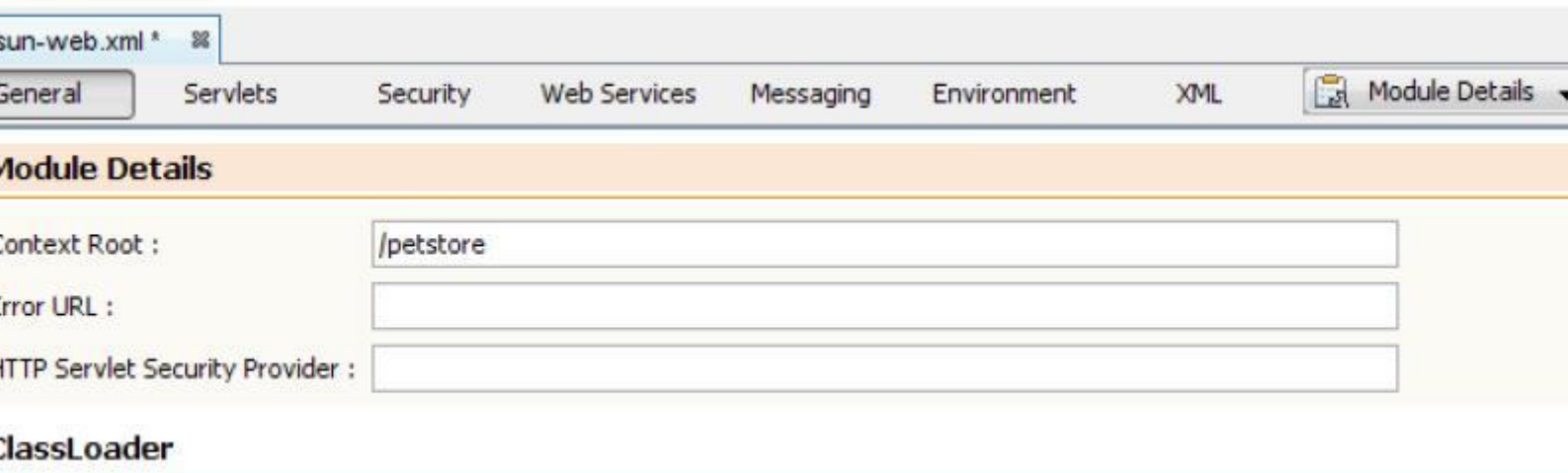


Figure 9-18: Modification du context root

LLIX-D - Créer le fichier de déploiement standard

Avant de déployer l'application, il vous reste à générer le fichier de déploiement standard « application.xml ».

Pour cela, un clic droit sur le projet « Yaps-Server » propose le menu contextuel suivant dans lequel vous allez pouvoir ajouter le descripteur de déploiement standard (Standard Deployment Descriptor) :

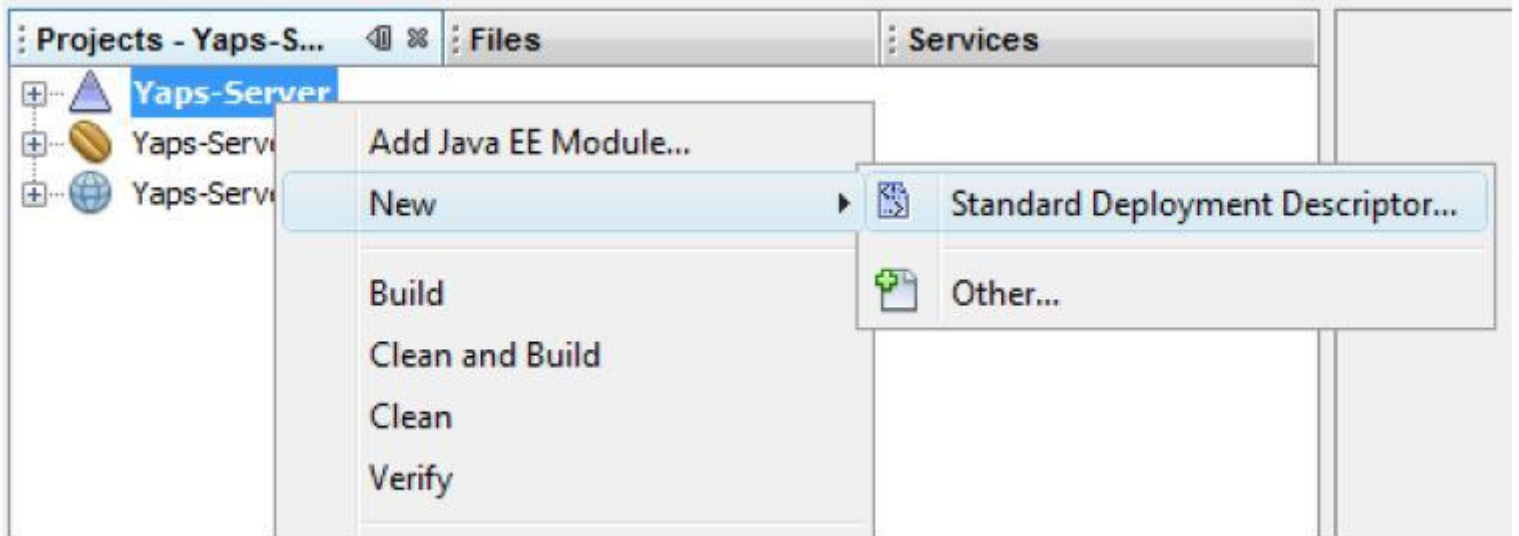


Figure 9-19: Ajouter le fichier « application.xml »

L'écran suivant apparaît précisant où résidera le fichier de déploiement :

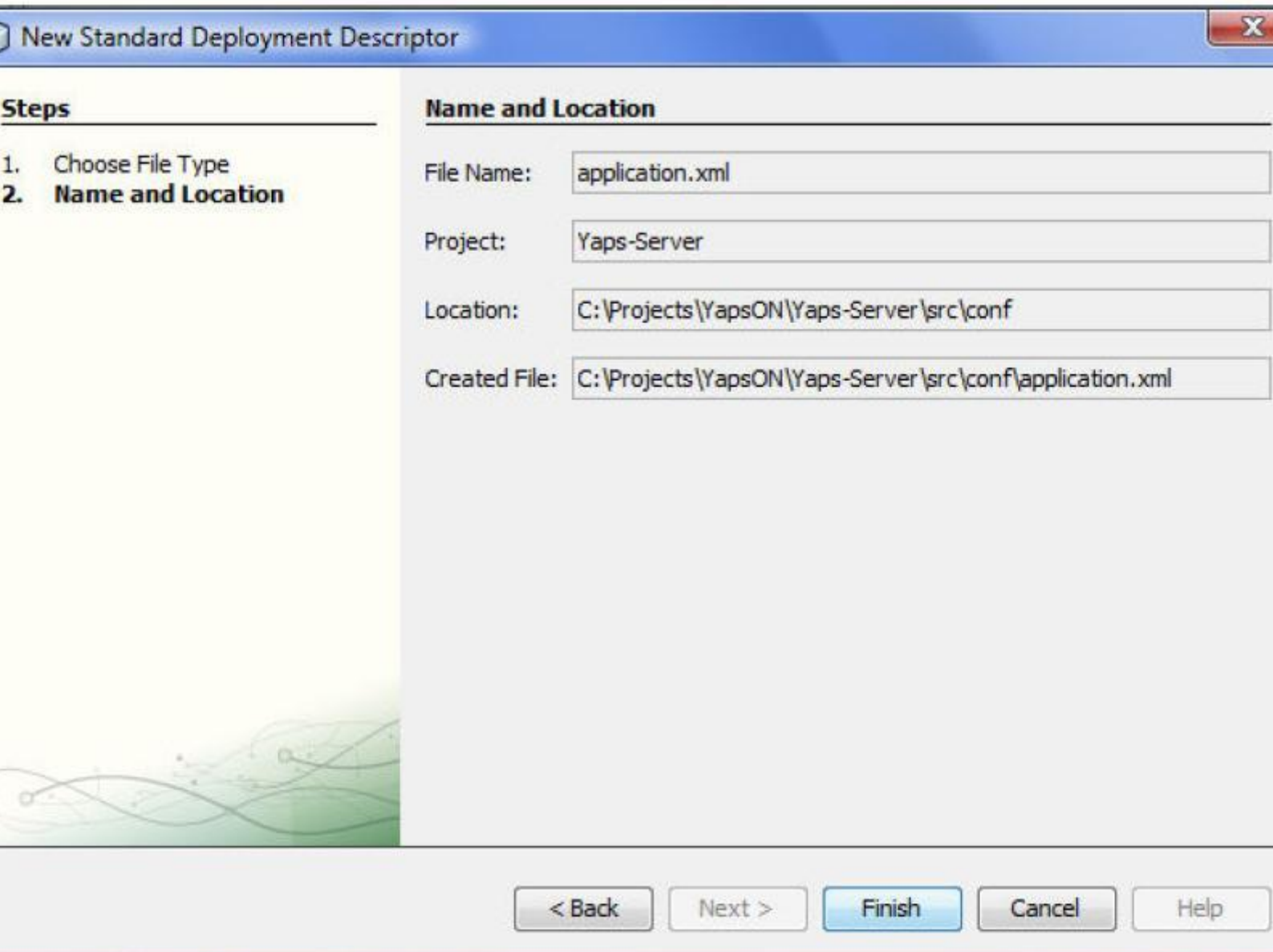


Figure 9-20: Localisation du fichier « application.xml »

Cliquez sur « Finish » pour générer le fichier de déploiement.

Le fichier « application.xml » est généré et associé au projet :

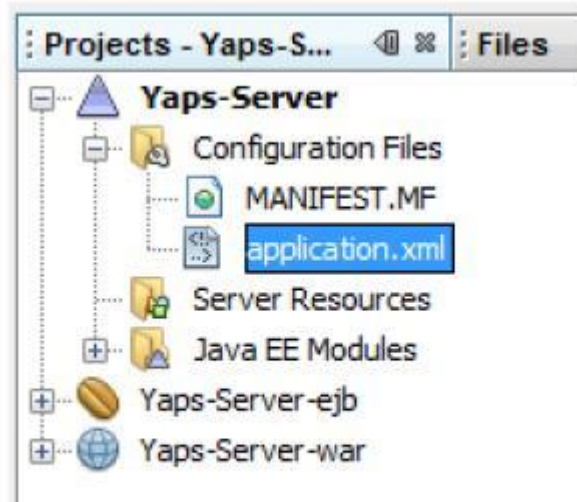


Figure 9-21: Fichier « application.xml »

Si l'on consulte le fichier « application.xml », on constate que la section « web » du fichier met en relation l'URI du projet Web (« Yaps-Server-war.war ») avec le Context Root « /petstore » que vous avez défini précédemment.

```

...
<web>
<web-uri>Yaps-Server-war.war</web-uri>
<context-root>/petstore</context-root>
</web>
...

```

LLIX-E - Compiler le projet

Un clic droit sur le projet fait apparaître le menu contextuel proposant les options pour compiler le projet (« Build » ou « Clean and Build ») :

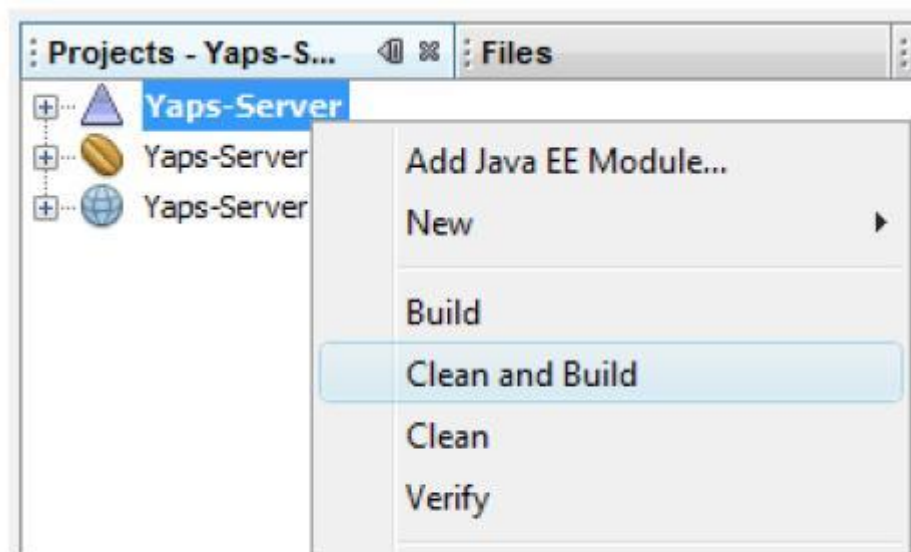
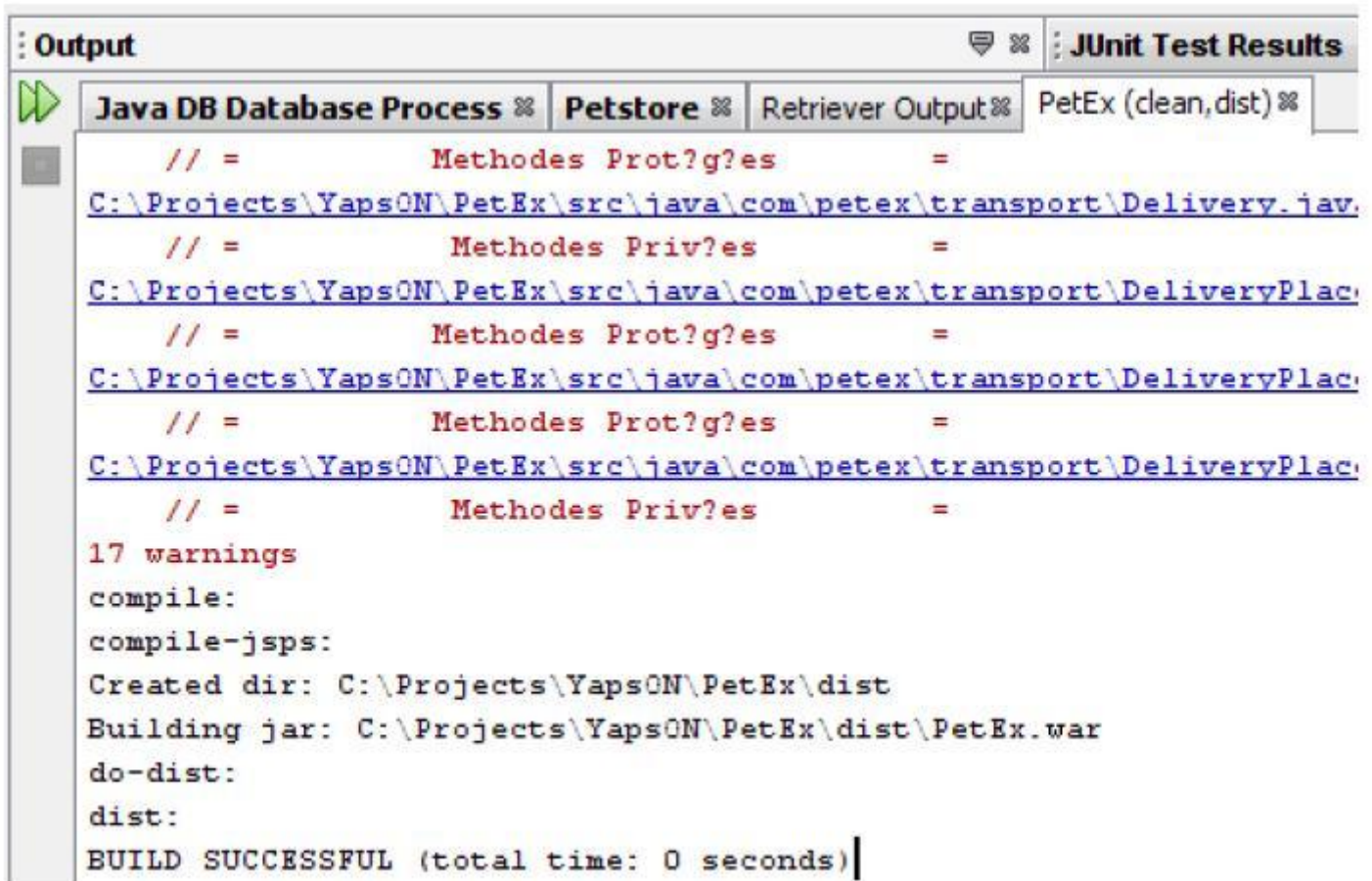


Figure 9-22: Compiler le projet

Cliquez sur l'option « Clean and Build ».
 La fenêtre « Output » affiche les traces de la compilation :



```

Output
Java DB Database Process
Petstore
Retriever Output
PetEx (clean, dist)

// =          Methodes Prot?g?es          =
C:\Projects\YapsON\PetEx\src\java\com\petex\transport\Delivery.jav
// =          Methodes Priv?es            =
C:\Projects\YapsON\PetEx\src\java\com\petex\transport\DeliveryPlac
// =          Methodes Prot?g?es          =
C:\Projects\YapsON\PetEx\src\java\com\petex\transport\DeliveryPlac
// =          Methodes Prot?g?es          =
C:\Projects\YapsON\PetEx\src\java\com\petex\transport\DeliveryPlac
// =          Methodes Priv?es            =

17 warnings
compile:
compile-jsp:
Created dir: C:\Projects\YapsON\PetEx\dist
Building jar: C:\Projects\YapsON\PetEx\dist\PetEx.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 0 seconds)
    
```

Figure 9-23: Résultat de la compilation

Les messages intéressants sont les suivants :

```

...
Building jar: C:\Projects\YapsON\Yaps-Server\Yaps-Server-ejb\dist\Yaps-Server-ejb.jar
...
Building jar: C:\Projects\YapsON\Yaps-Server\Yaps-Server-war\dist\Yaps-Server-war.war
...
Building jar: C:\Projects\YapsON\Yaps-Server\dist\Yaps-Server.ear
...
BUILD SUCCESSFUL (total time: 7 seconds)
    
```

Ces traces indiquent que les modules ont été correctement compilés de même que l'archive de type EAR.

LLIX-F - Déployer le projet

Vous allez maintenant procéder au déploiement du projet sous GlassFish.
 Un clic droit sur le projet fait apparaître le menu contextuel proposant l'option de déploiement (« Deploy ») :

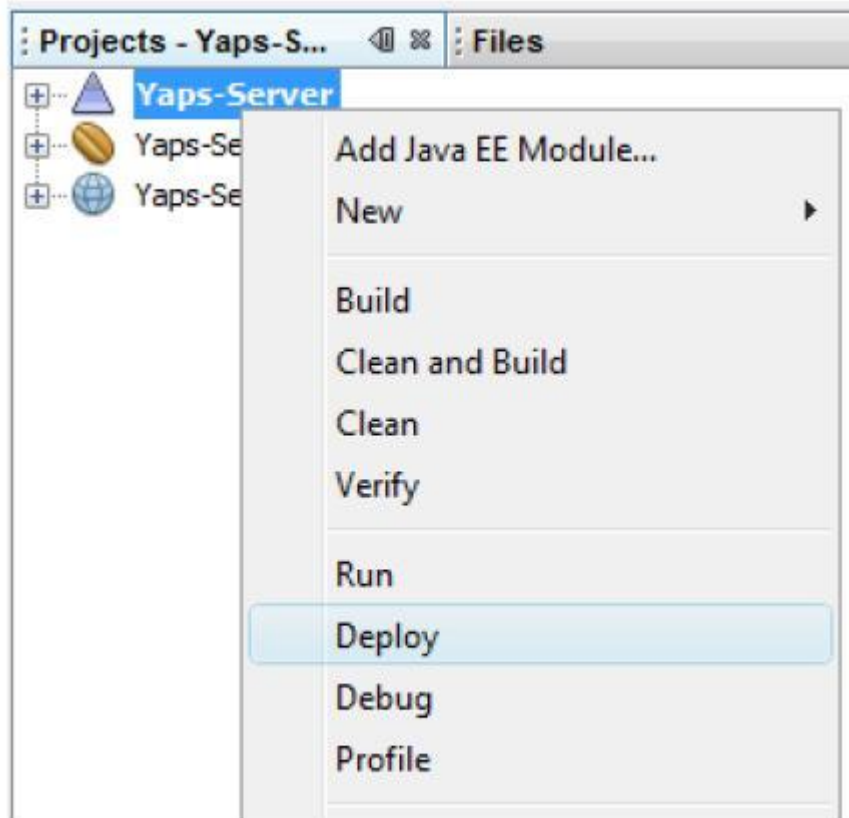


Figure 9-24: Déployer le projet

Cliquez sur l'option « Deploy ».

Si nécessaire, NetBeans se charge de démarrer le serveur d'applications « PetStore ».

Si l'application Yaps-Server était déjà déployée sur le serveur, NetBeans se charge de la supprimer (Undeploying) pour la remplacer par la nouvelle version.

La fenêtre « Output » trace les messages suivants :

```

...
deployment started : 0%
deployment finished : 100%
Deploying application in domain completed successfully
...
All operations completed successfully
post-run-deploy:
run-deploy:
BUILD SUCCESSFUL (total time: 10 seconds)
    
```

Le déploiement terminé, vous pouvez explorer le contenu de l'application Yaps-Server sur le serveur PetStore.

Rendez-vous sur l'onglet "Services" et sélectionnez le serveur "Petstore".

Le noeud « Applications -> Enterprise Applications -> Yaps-Server » affiche les ressources et services exposés par l'application :

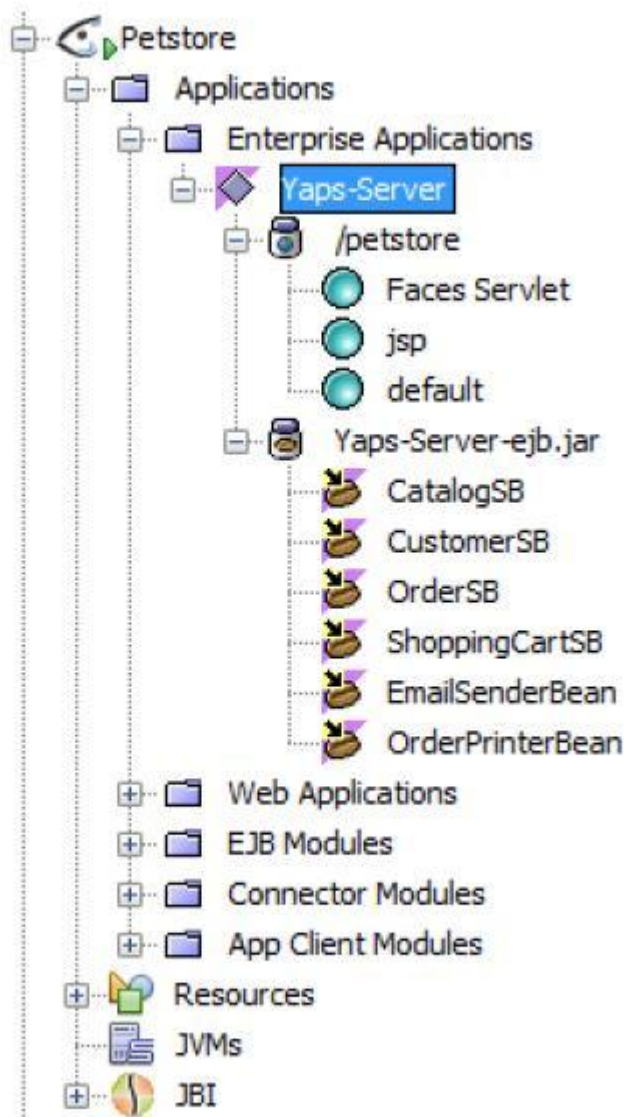


Figure 9-25: Application Yaps-Server sous GlassFish

LLIX-G - Base de données

Le déploiement de l'application Yaps-Server a eu pour effet de supprimer et de créer les tables de la base de données « petstoreDB ».

Ces opérations sont réalisées par le biais d'annotations JPA apposées dans les différentes Entity.

Dans cette section, nous allons montrer comment - au travers de NetBeans - accéder à la structure et aux données de la base de données « petstoreDB ».

LLIX-G-1 - Créer une connexion

La première étape consiste à créer une connexion vers la base de données.

Sélectionnez l'onglet « Services » et ensuite un clic droit sur « Databases ».

Le menu contextuel suivant apparaît offrant la possibilité de créer une nouvelle connexion :

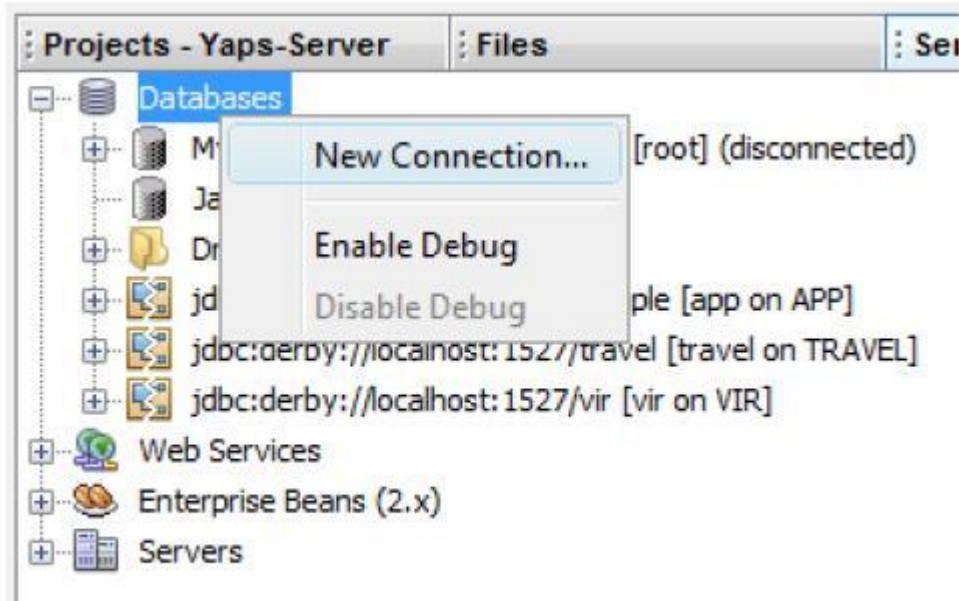


Figure 9-26: Nouvelle connexion DB

Cliquez sur « New Connection ».

L'écran suivant apparaît dans lequel vous allez identifier les propriétés de la base de données « petstoreDB » :

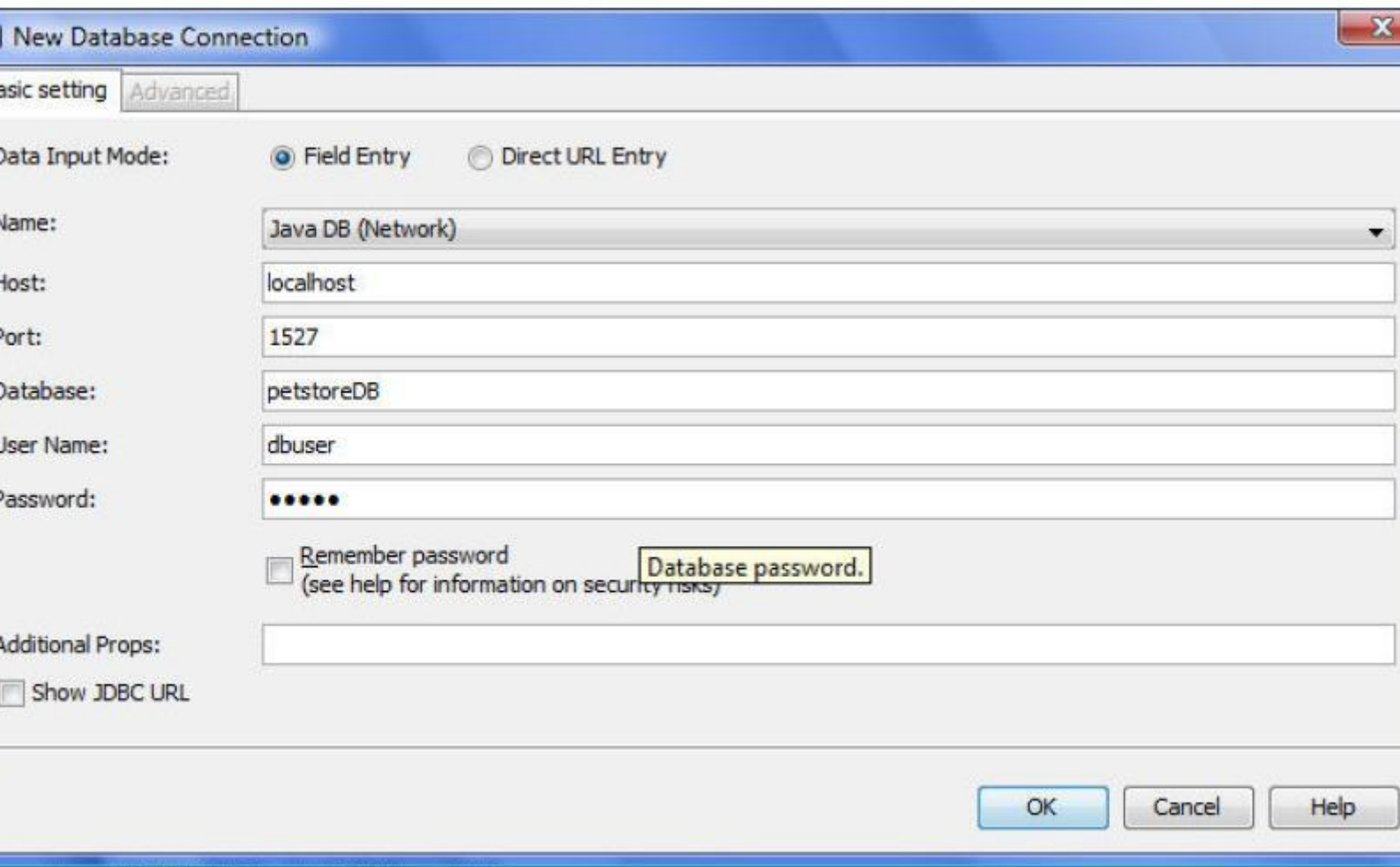


Figure 9-27: Nouvelle connexion DB

Dans le formulaire, introduisez les informations suivantes :

| | |
|-------------------|-------------------|
| Name: | Java DB (Network) |
| Host: | localhost |
| Port: | 1527 |
| Database: | petstoreDB |
| User Name: | dbuser |
| Password: | dbpwd |

Propriétés de la source de données

Les informations de connexion proviennent des propriétés définies durant la création du Connection Pool (voir [ici](#)).

Cliquez sur le bouton « OK » pour créer la connexion.

Avant de fermer la fenêtre, NetBeans peut vous demander de préciser le schéma de la base de données :

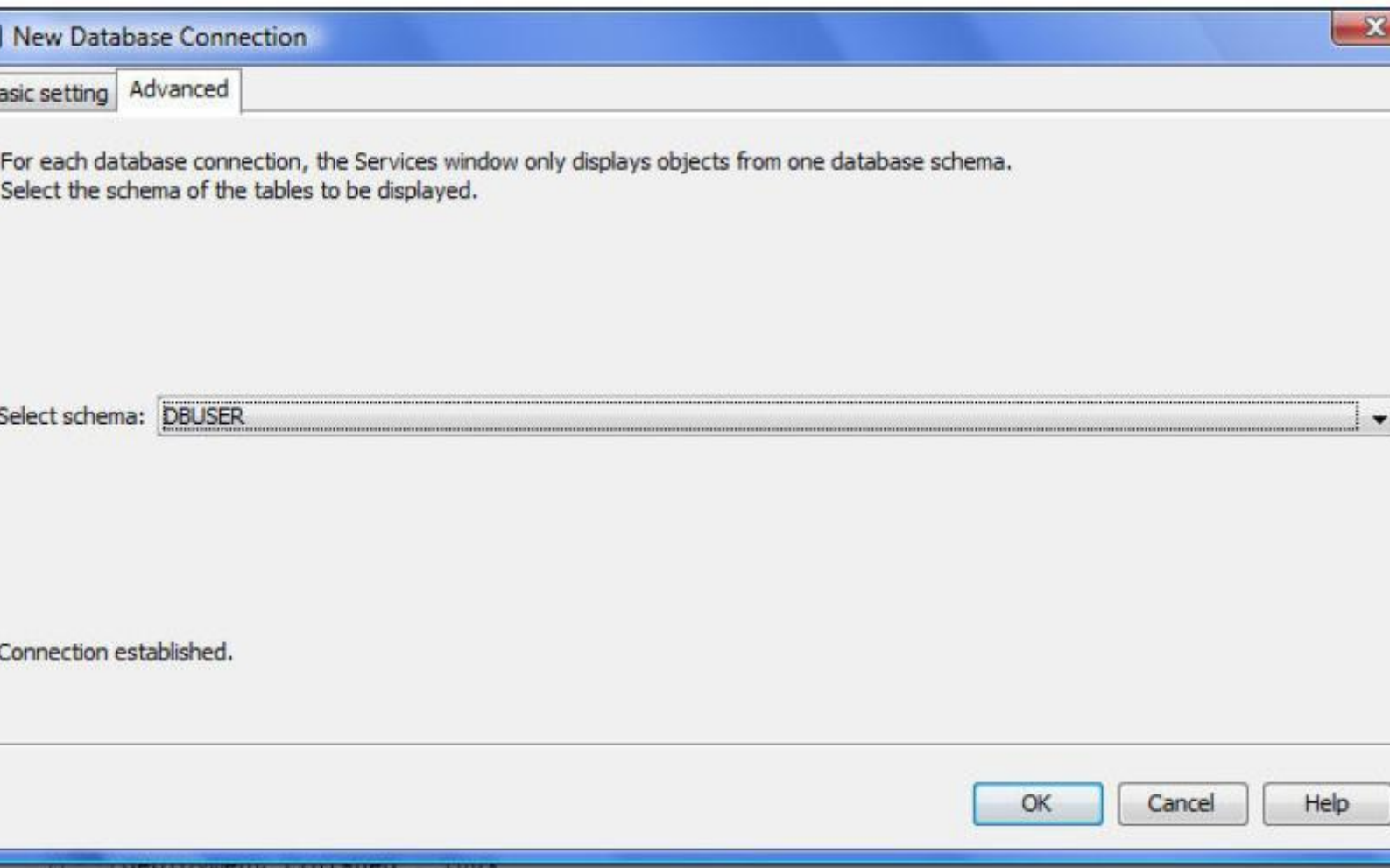


Figure 9-28: Choix du schéma de la DB

Veuillez préciser « DBUSER » et cliquez ensuite sur « OK ».

Le noeud « Databases » présente la nouvelle connexion

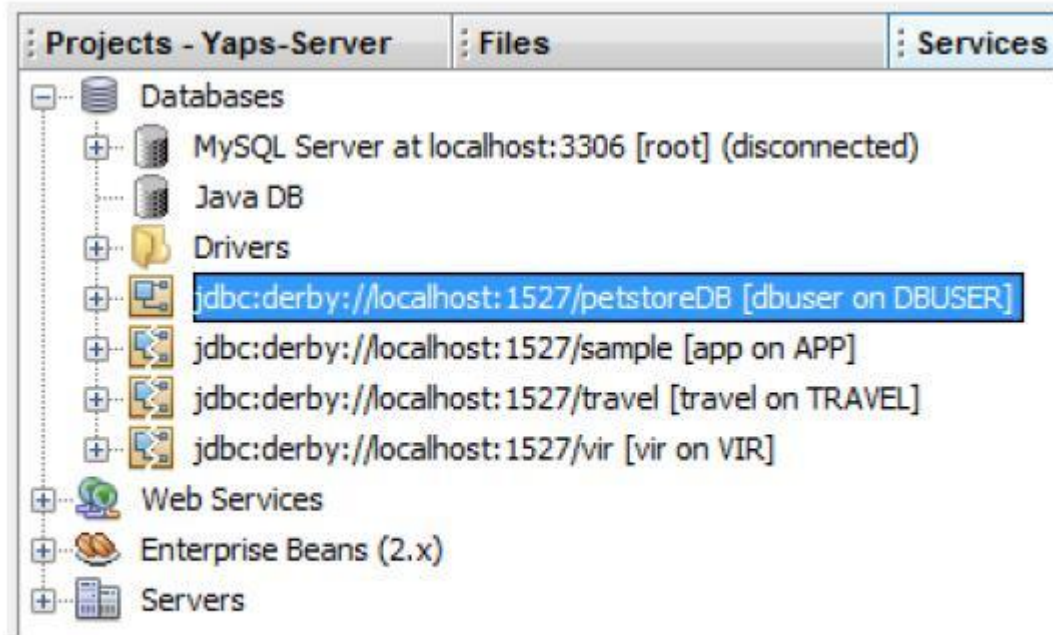


Figure 9-29: Connexion vers petstoreDB

Chaque connexion est préfixée par une icône qui précise un statut :

- Une connexion DB est en cours



- Pas de connexion DB



LLIX-G-2 - Connexion à la DB

Un clic droit sur la connexion fait apparaître le menu contextuel par lequel vous pouvez vous connecter à la base de données :

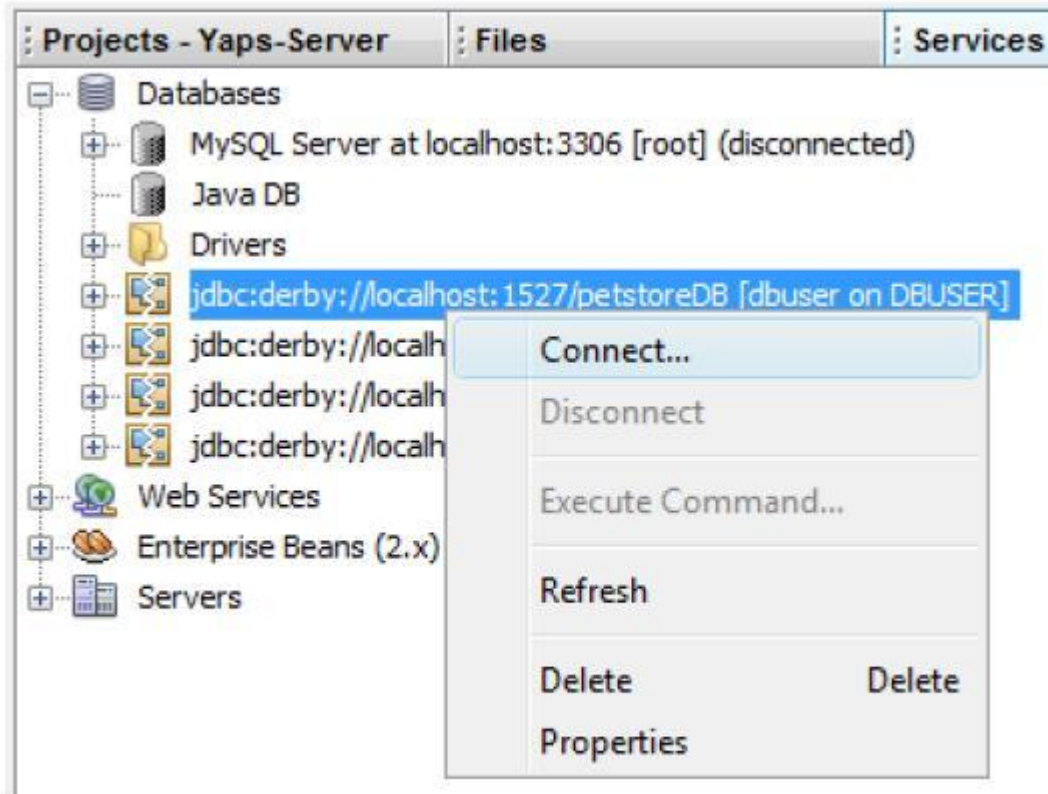


Figure 9-30: Connexion à la DB

Cliquez sur « Connect » pour vous connecter à la base de données.

Le système vous demande d'introduire le mot de passe :



Figure 9-31: mode de passe de la DB

En cochant l'option « Remember password », le mot de passe ne vous sera plus demandé. Vous êtes maintenant connecté à la base de données :

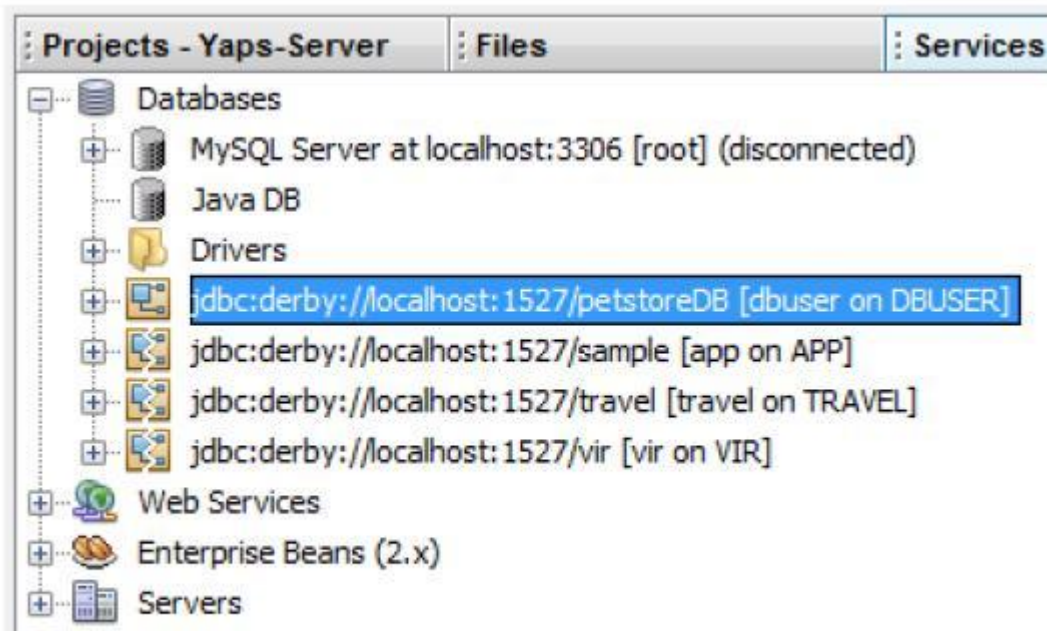


Figure 9-32: Base de données connectée à NetBeans

LLIX-G-3 - Explorer la DB

Les noeuds de la connexion peuvent être ouverts pour explorer la structure de la base de données « petstoreDB » :

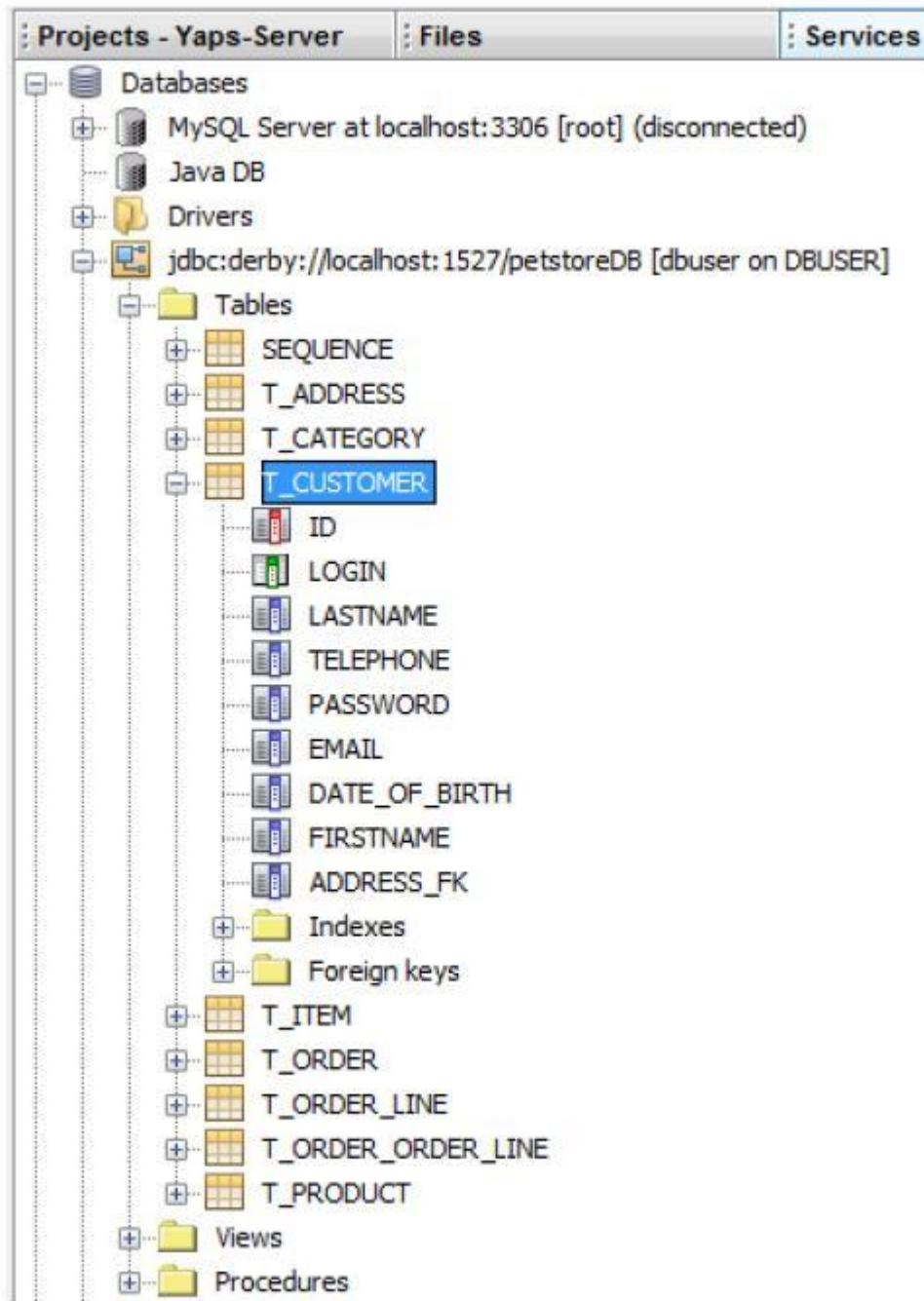


Figure 9-33: Structure de petstoreDB

Un clic droit sur une table fait apparaître le menu contextuel qui permet de visualiser le contenu de la table ou d'altérer sa structure :

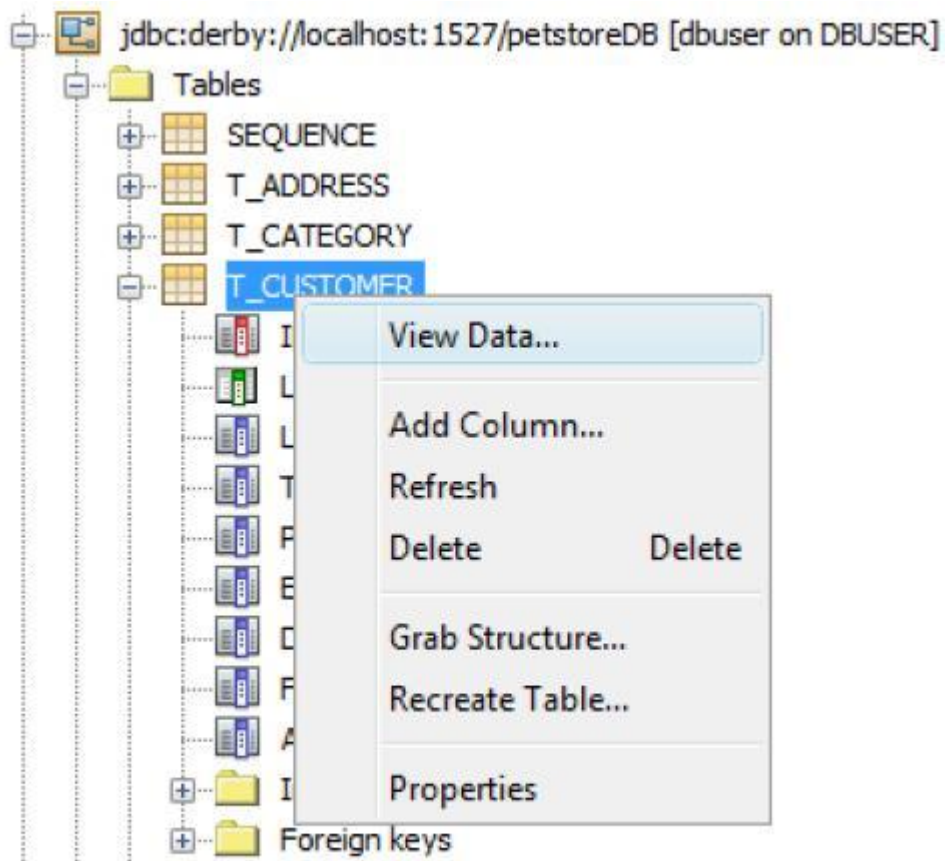


Figure 9-34: Opérations sur la table

Si vous cliquez sur l'option « View Data », une fenêtre apparaît et vous donnez la possibilité d'exécuter des requêtes de type SQL et obtenir le résultat de la requête :

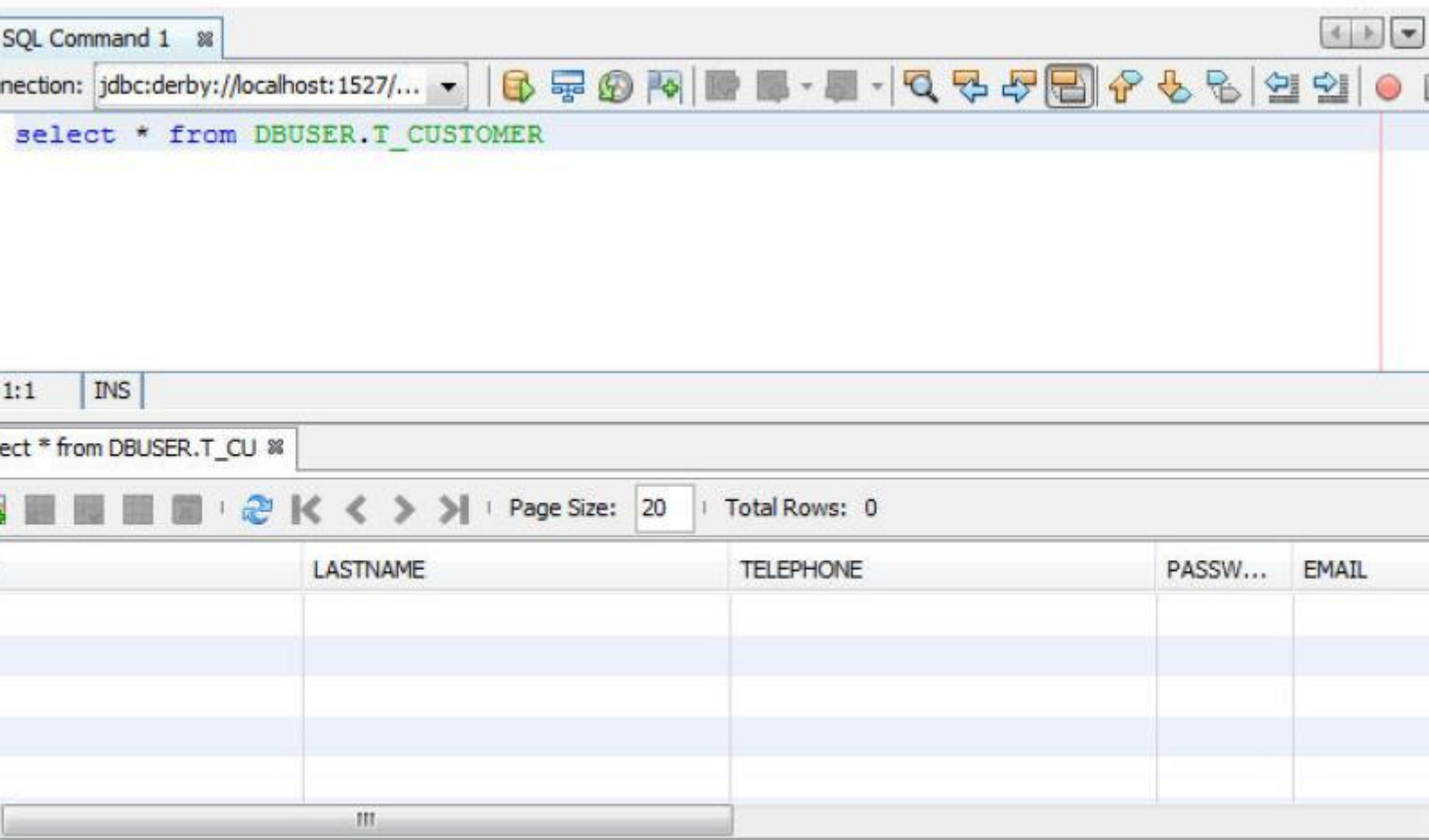


Figure 9-35: Option "View Data"

Vous constaterez que les tables de la base de données sont vides.

Nous allons maintenant présenter les méthodes pour charger les données de test de livrées avec le projet Yaps original.

LLIX-G-4 - Données de test

Dans le fichier « build.xml » du projet original Yaps, la tâche « **db-insert-data** » insère des données de test dans la base de données.

Ce processus d'insertion des données de test est exécuté à chaque déploiement de l'application.

Durant la phase de transfert (voir **ici**), les données de test ont été transférées à l'emplacement suivant :

```
{yapson_home}\Yaps-Server\data.sql
```

Nous allons montrer deux méthodes que l'on peut utiliser sous NetBeans pour charger ces données de test.

LLIX-G-4-a - Méthode 1 : Chargement manuel des données de test

Cette méthode décrit comment charger manuellement les données de test dans la base de données en utilisant les opérations offertes dans le service « Database ».

Un clic droit sur le noeud « Table » de la base de données « petstoreDB » présente le menu contextuel suivant :

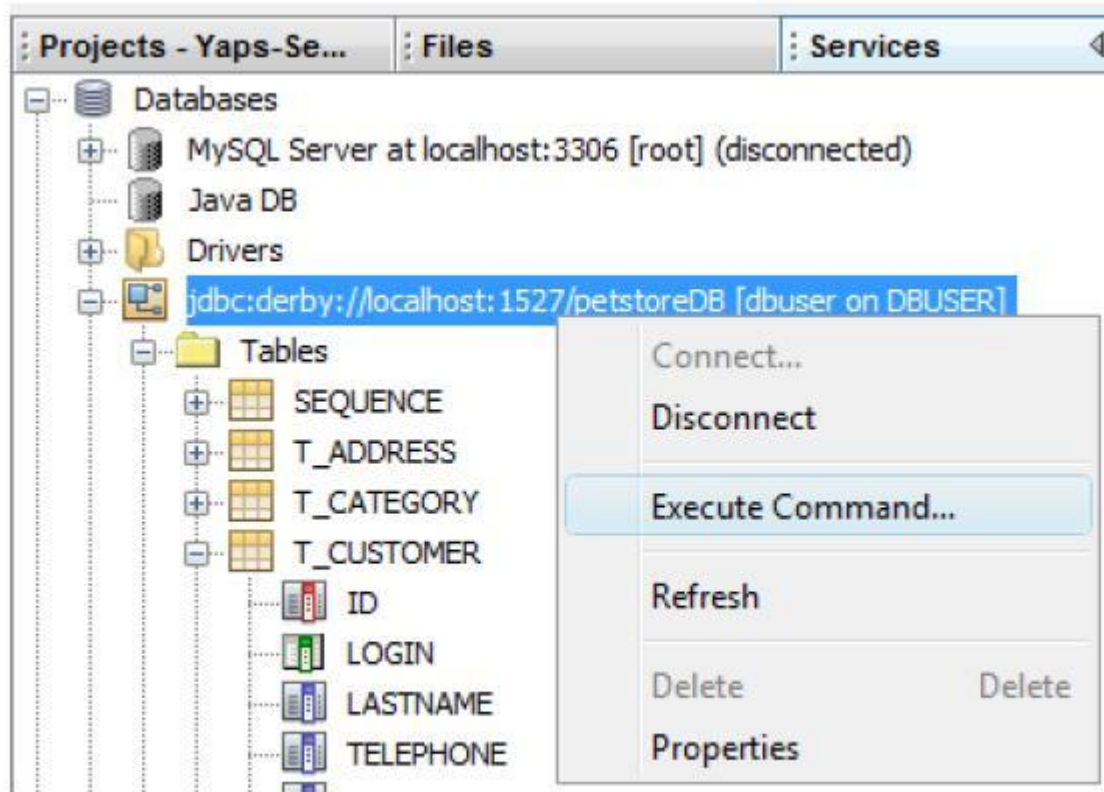


Figure 9-36: Exécuter une commande SQL

Sélectionnez l'option « Execute Command ».

Copiez le contenu du fichier « **data.sql** » pour le coller dans la fenêtre de commande :

```

SQL Command 1
Connection: jdbc:derby://localhost:1527/...

-- Delete
DELETE FROM T_ORDER_ORDER_LINE;
DELETE FROM T_ORDER_LINE;
DELETE FROM T_ORDER;
DELETE FROM T_ITEM;
DELETE FROM T_PRODUCT;
DELETE FROM T_CATEGORY;
DELETE FROM T_CUSTOMER;
DELETE FROM T_ADDRESS;

-- Load
INSERT INTO T_CATEGORY (id, name, description) VALUES (1, 'Fish', 'Any of numerous
INSERT INTO T_CATEGORY (id, name, description) VALUES (2, 'Dogs', 'A domesticated
INSERT INTO T_CATEGORY (id, name, description) VALUES (3, 'Reptiles', 'Any of var:
INSERT INTO T_CATEGORY (id, name, description) VALUES (4, 'Cats', ' Small carnivor
INSERT INTO T_CATEGORY (id, name, description) VALUES (5, 'Birds', 'Any of the cla
14:59  INS
  
```

Figure 9-37: Fenêtre de commande SQL

Exécutez les requêtes SQL en cliquant sur l'icône de la toolbar.

La fenêtre « Output » trace le résultat de chaque requête SQL.

Vous pouvez maintenant visualiser le contenu de chaque table en invoquant l'option « View Data » décrite plus haut.

Par exemple, nous obtenons le résultat suivant pour la table « CUSTOMER » :

SQL Command 1

Connection: jdbc:derby://localhost:1527/petstoreDB ...

select * from DBUSER.T_CUSTOMER

l:1 | INS

ect * from DBUSER.T_CU

Page Size: 20 | Total Rows: 4 (Page 1 of 1)

| LASTNAME | TELEPHONE | PASSWORD | EMAIL | LOGIN | DATE_OF_BIRTH | FIRSTNAME | ADDRESS_FK |
|-----------|------------|----------|-----------------------------|----------|---------------|-----------|------------|
| Fleury | 545 123 45 | marc | marc@jboss.org | marc | May 29, 1971 | Marc | 1 |
| Gates | 654 046 12 | bill | bill.gates@microsoft.com | bill | Feb 16, 1965 | Bill | 2 |
| Jobs | 548 157 15 | job | steve.jobs@apple.com | job | Nov 30, 1964 | Steve | 3 |
| Goncalves | 487 487 21 | antoncal | antonio.goncalves@gmail.com | antoncal | May 29, 1971 | Antonio | 4 |

Figure 9-38: Contenu de la table "CUSTOMER"

Exécuter une sélection de commandes SQL

Il vous est également loisible d'exécuter un ensemble de commande.
 Dans la fenêtre de commande SQL, sélectionnez les commandes à exécuter.
 Un clic droit fait apparaître le menu contextuel avec l'option « Run Selection » :

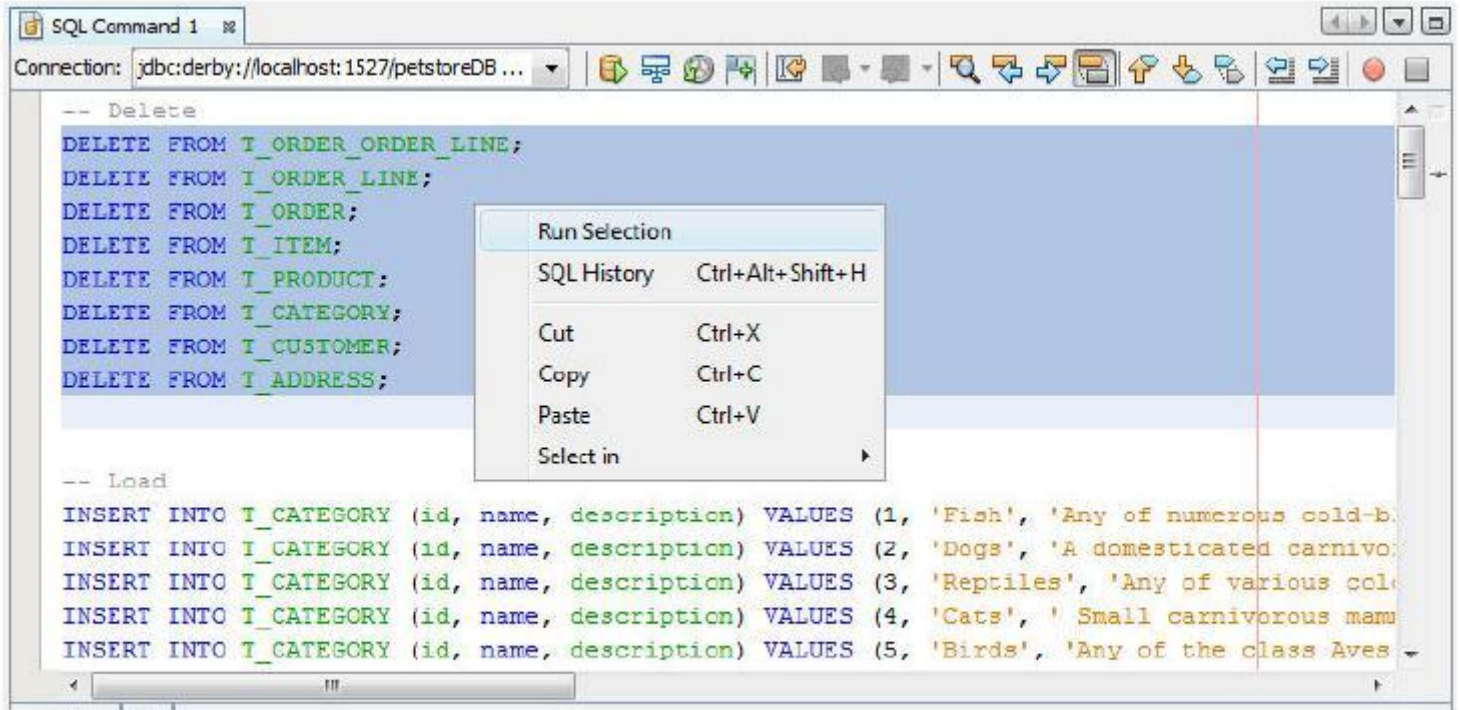


Figure 9-39: Exécuter une sélection de commandes SQL

On se rend vite compte que le chargement manuel est quelque peu contraignant, car il vous oblige à répéter les opérations de chargement après chaque déploiement de l'application. Nous allons montrer une autre méthode qui automatise le processus de chargement des données de test.

LLIX-G-4-b - Méthode 2 : Chargement automatique des données de test

Cette seconde méthode est beaucoup plus élégante, car elle automatise le chargement des données de test conformément au projet original Yaps.

En inspectant les fichiers de projet Netbeans, vous aurez compris que NetBeans utilise Ant pour compiler et déployer les applications.

NetBeans offre la possibilité de personnaliser le processus de construction et de déploiement des applications en offrant au développeur la liberté de modifier le fichier « **build.xml** » du projet afin d'exploiter les targets Ant suivantes :

| | |
|------------------------|--|
| pre-init | Called before initialization of project properties |
| post-init | Called after initialization of project properties |
| pre-compile | Called before javac compilation |
| post-compile | Called after javac compilation |
| pre-dist | Called before archive building |
| post-dist | Called after archive building |
| post-clean | Called after cleaning build products |
| pre-run-deploy | Called before deploying |
| post-run-deploy | Called after deploying |

Nous allons enrichir le processus de construction de l'application Yaps-Server afin d'y insérer une tâche pour ajouter les données de test dans la base de données petstoreDB.

Nous allons montrer comment enrichir le fichier « build.xml » afin d'insérer les données de test après chaque déploiement de l'application (**post-run-deploy**).

Sélectionnez l'onglet « Files » et ouvrez le noeud du projet Yaps-Server :

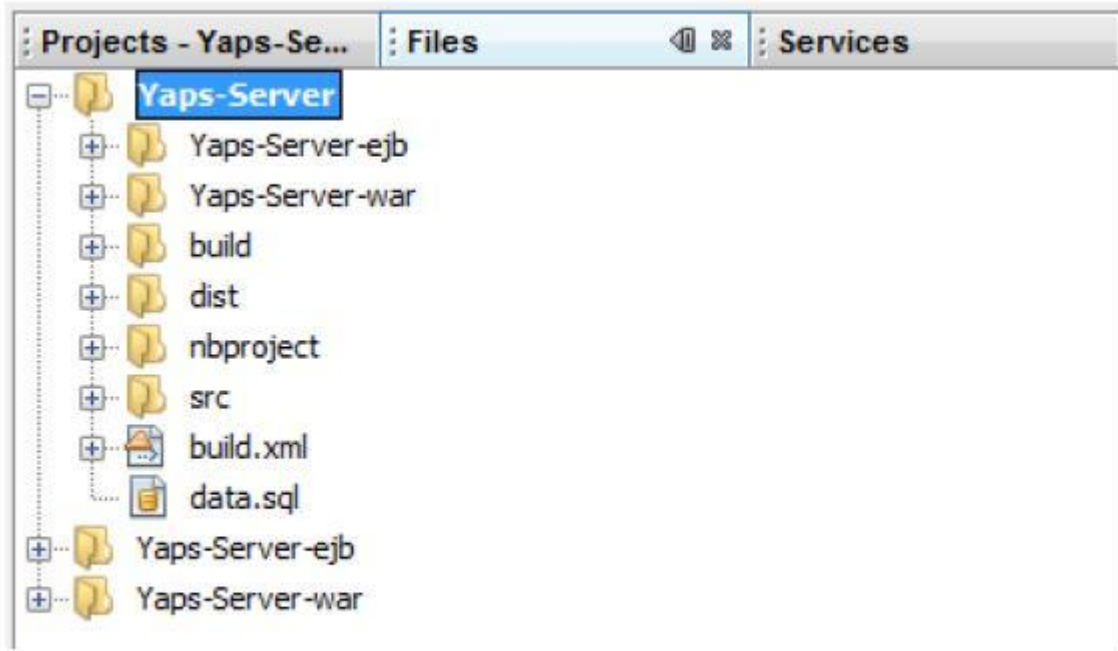


Figure 9-40: Fichiers du projet Yaps-Server

Vous constatez la présence du fichier « **data.sql** » qui a été transféré du projet original. Vous allez créer le script Ant « **yaps-data.xml** » qui sera utilisé pour charger les données de test.

Fichier « yaps-data.xml »

L'archive de l'application YapsON est livrée avec le fichier « yaps-data.xml ».

Il n'est donc pas nécessaire de créer ce fichier. Vous êtes libres de le déplacer du répertoire principal vers le répertoire du projet « Yaps-Server » :

```
C:\> cd {yapson_home}
{yapson_home}> copy yaps-data.xml Yaps-Server
```

Sélectionnez le dossier « yaps-Server ».

Un clic droit affiche le menu contextuel suivant dans lequel vous allez sélectionner l'option « New -> Other » :

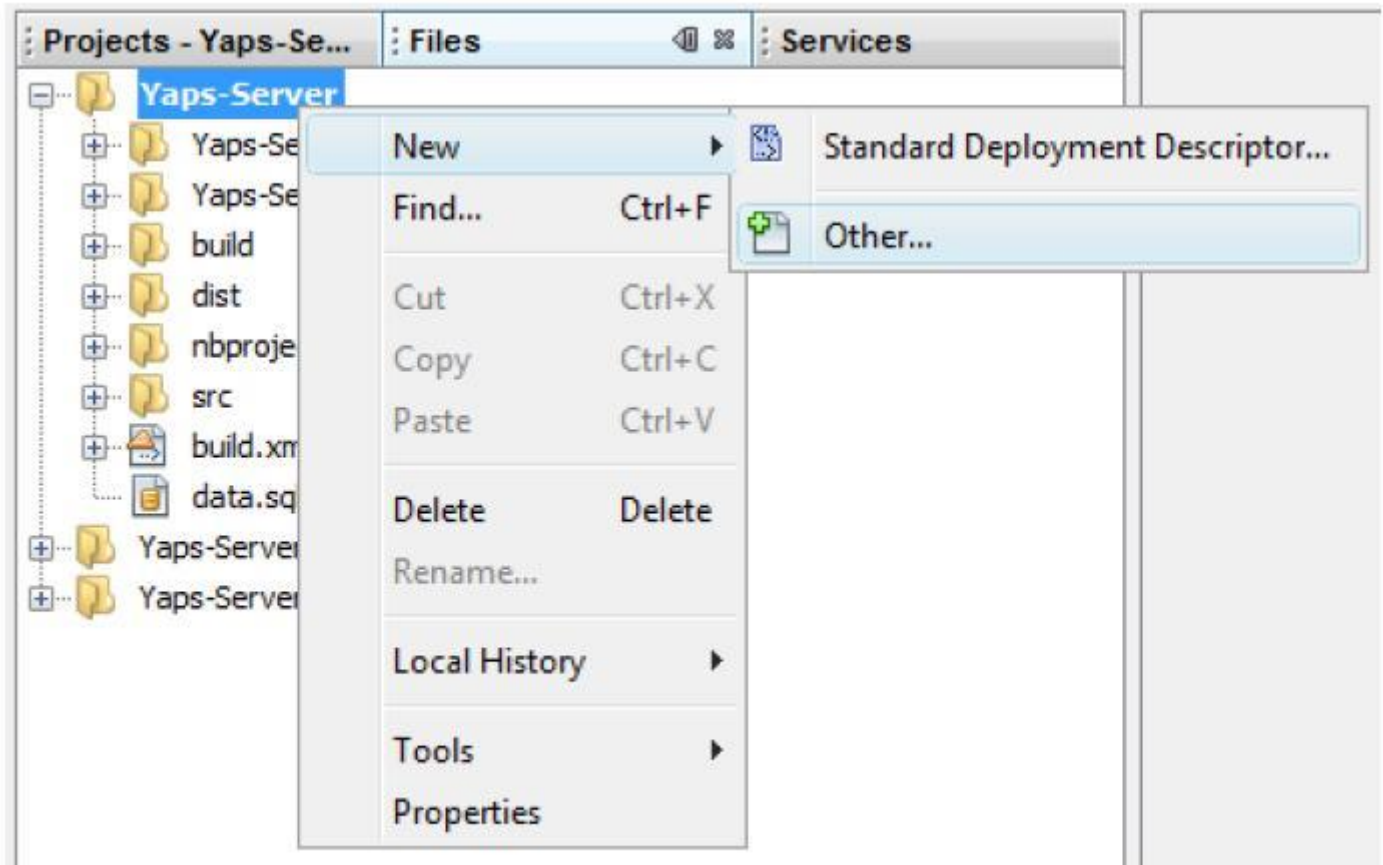


Figure 9-41; Création d'un nouveau fichier

Pour créer un fichier de type Ant, vous allez choisir l'option « Other -> Ant Build Script» :

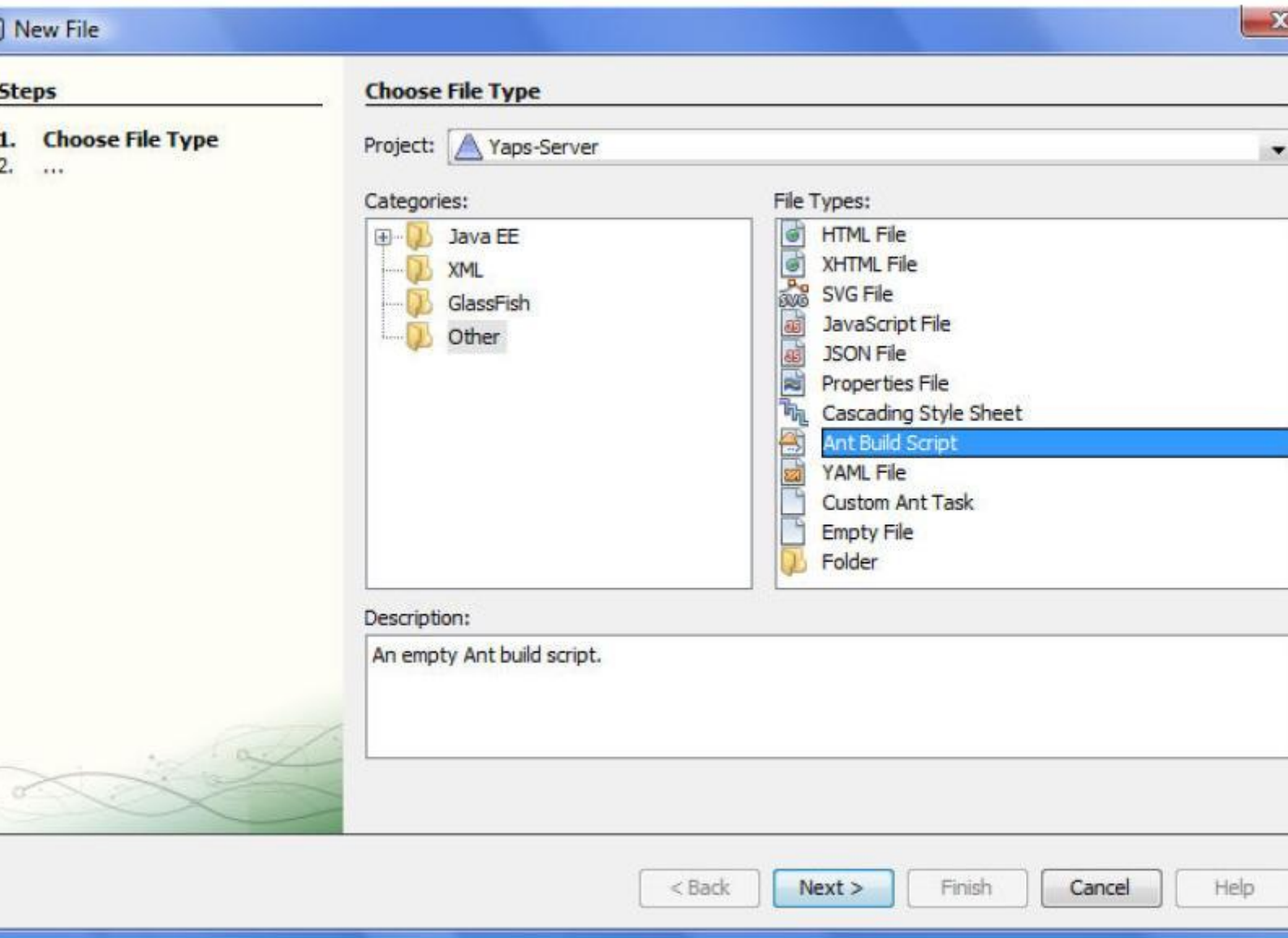


Figure 9-42: Type de fichier « Ant Build Script »

Cliquez sur « Next » pour poursuivre.

Ensuite, entrez le nom du fichier qui sera dans notre cas « **yaps-data.xml** » :

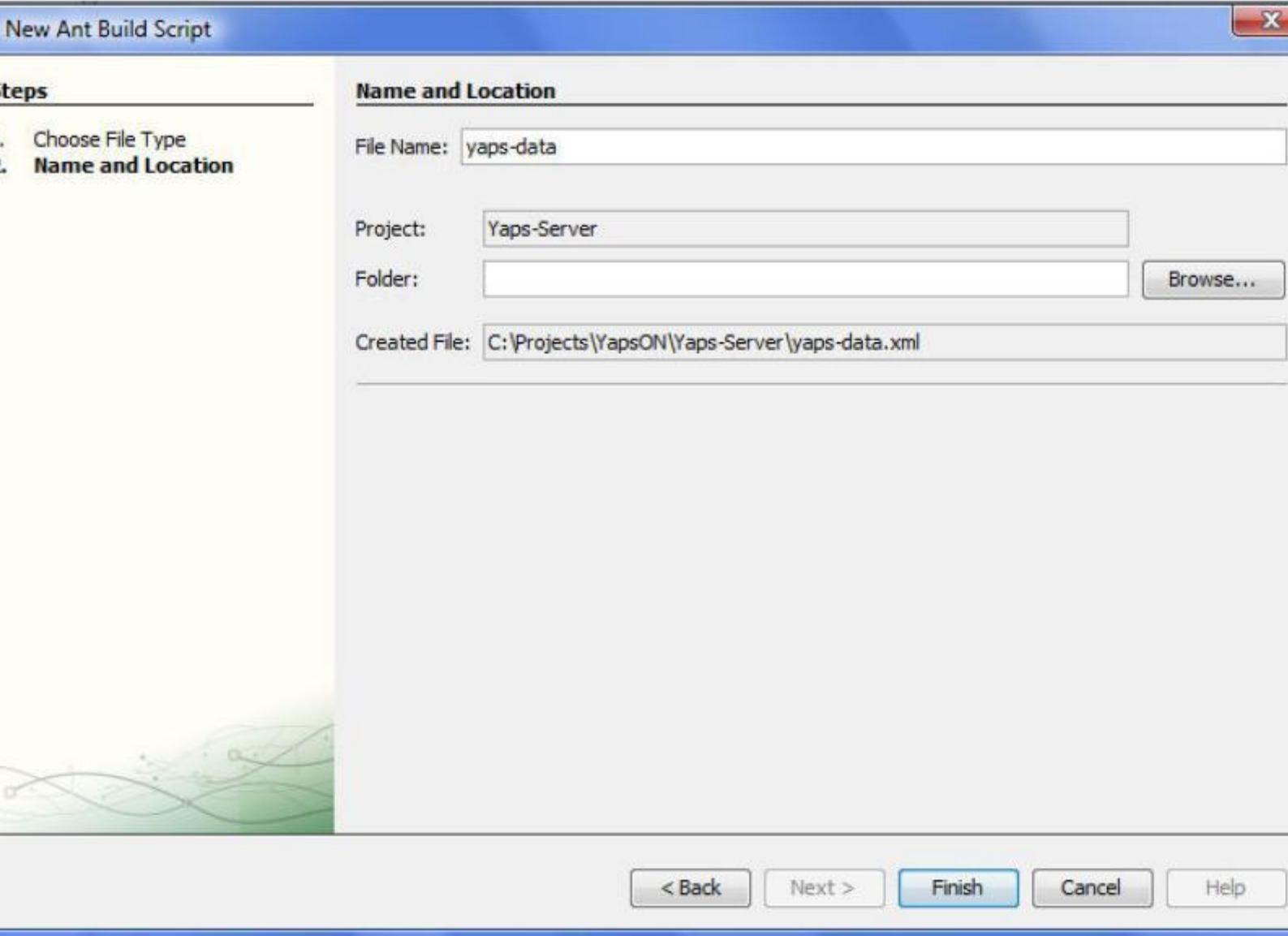


Figure 9-43: Nom du fichier Ant

Cliquez sur « Finish » pour terminer le processus.

Le fichier « **yaps-data.xml** » s'ouvre maintenant dans l'éditeur de NetBeans pour vous permettre de le compléter :

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="changeme" default="all" basedir=".">
  <target name="all">
    <!-- TODO define script. See Help menu or http://ant.apache.org/ -->
  </target>
</project>
```

Figure 9-44: Fichier yaps-data.xml

Nous fournissons ci-dessous le contenu à appliquer au fichier « yaps-data.xml » :

```
<?xml version="1.0"?>
<!-- This Ant script is used to load test data on the Yaps Pet Store Database -->
<project name="Yaps-Data" default="load-data" basedir=".">
<!-- Database properties -->
<property name="db-url" value="jdbc:derby://localhost:1527/petstoreDB" />
<property name="db-driver" value="org.apache.derby.jdbc.ClientDriver" />
<property name="db-userid" value="dbuser" />
<property name="db-password" value="dbpwd" />
<!-- Derby Library location -->
<property name="derby.home" value="C:/Program Files/glassfish-v2ur2/javaDB"/>
<property name="derby.lib" value="${derby.home}/lib"/>
<!-- Classpath -->
<path id="classpath">
<!-- Derby classes -->
<pathelement location="${derby.lib}/derbyclient.jar"/>
</path>
<!-- Ant task used to load data on the Yaps Pet Store Database -->
<target name="load-data">
<sql driver="${db-driver}"
url="${db-url}"
userid="${db-userid}"
password="${db-password}"
autocommit="yes"
onerror="continue"
caching="yes"
src="data.sql">
<classpath refid="classpath"/>
</sql>
</target>
</project>
```

Nous allons décrire le fichier Ant.

Le script Ant « Yaps-Data » commence par définir un ensemble de propriétés pour identifier :

- **Database properties** : les propriétés de la base de données « petstoreDB ».
- **Derby Library location**: le chemin vers les bibliothèques de Derby.
- **Classpath** : la bibliothèque Derby nécessaire pour importer les données.

Ensuite, la cible « load-data » fournit la tâche SQL utilisée pour charger les données dans la base de données « petstoreDB » à partir du fichier « data.sql ».

Sauvez les modifications et vérifiez la structure XML du fichier.

Pour cela, un clic droit dans le fichier fait apparaître le menu contextuel offrant l'option « Check XML » :

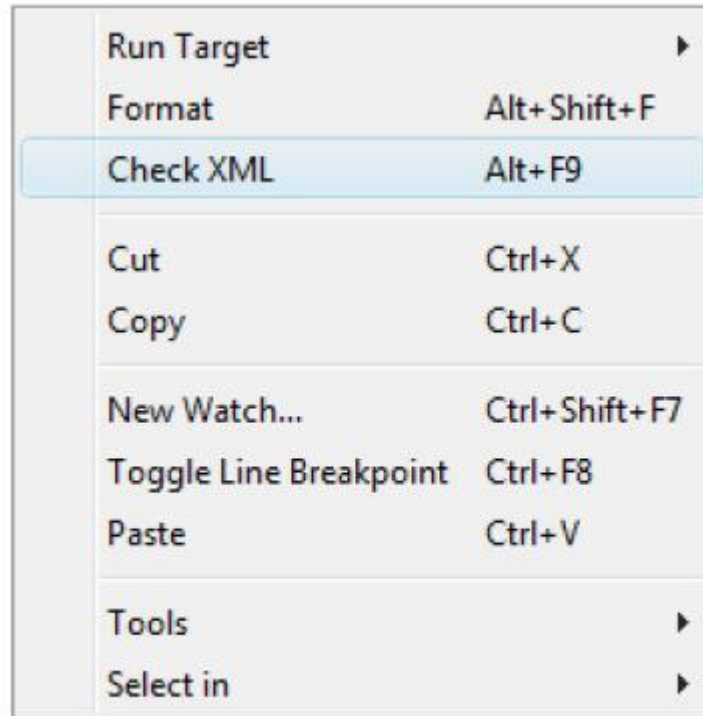


Figure 9-45: Vérifier le fichier XML

Maintenant, vous allez modifier le fichier « build.xml » du projet « Yaps-Server » pour invoquer le script Ant « yaps-data.xml » après la phase de déploiement du projet.

Toujours à partir de l'onglet « Files », ouvrez le fichier « build.xml » du projet « Yaps- Server » :

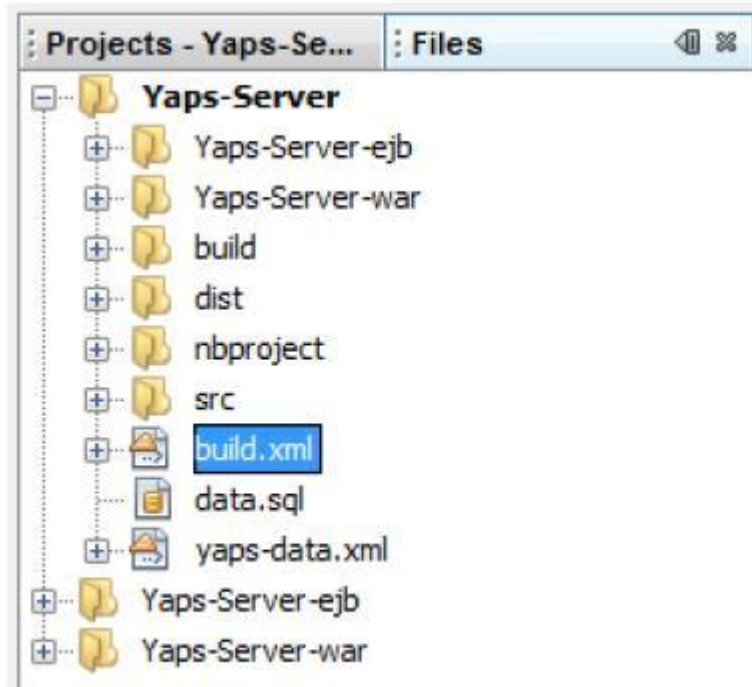


Figure 9-46: Fichier « build.xml » de Yaps-Server

L'objectif étant d'insérer les données de test après le déploiement du projet, vous allez ajouter la target Ant « **post-run-deploy** » dans le fichier « build.xml » :

```
...
<!-- After deploying the application, we populate the database with test data -->
<target name="post-run-deploy">
<echo>Load Data on the Yaps Pet Store Database ... </echo>
...
<!-- After deploying the application, we populate the database with test data -->
<target name="post-run-deploy">
<echo>Load Data on the Yaps Pet Store Database ... </echo>
```

La target « **post-run-deploy** » se charge d'invoquer la cible « load-data » du fichier « yaps-data.xml ».

Pour vérifier le bon déroulement du processus, l'instruction « **echo** » permet d'afficher une trace sur la console.

Vérifiez la structure du fichier en invoquant l'option « Check XML » comme décrite précédemment.

Sauvez vos modifications.

Si vous déployez le projet comme décrit plus haut (voir [ici](#)), vous verrez apparaître la trace suivante sur la console :

```
...
post-run-deploy:
Load Data on the Yaps Pet Store Database ...
load-data:
Executing resource: C:\Projects\YapsON\Yaps-Server\data.sql
91 of 91 SQL statements executed successfully
...
```

Ces informations indiquent que 91 requêtes SQL du fichier « data.sql » ont été exécutées durant la phase « post-run-deploy ».

Avec cette méthode, les données de tests seront automatiquement injectées dans la base de données à chaque déploiement du projet « Yaps-Server ».

LLIX-H - Tester le projet

Le test du projet est largement décrit dans le livre de référence.
On rappellera que l'application Web est accessible à l'adresse suivante :

<http://localhost:8080/petstore>



Figure 9-47: Page web de l'application

N'oubliez pas de créer un compte utilisateur afin de pouvoir valider le comportement de l'application :

- Ajout d'articles dans le panier.
- Création et envoi d'un bon de commande.
- Réception d'un email de confirmation.

- Messages de trace produits sur le serveur et mentionnant la réception des messages du topic JMS.

L'analyse des traces du serveur Petstore vous permet également de suivre le cheminement des messages et de constater que les Web Services (**DeliveryService** et **ValidationService**) sont bien invoqués.

Configuration du serveur SMTP

La classe « **EmailSenderBean** » chargée d'envoyer les emails utilise des constantes pour identifier le serveur SMTP:

```
private static final String SMTP_HOST = "smtp.free.fr";
private static final String USER = "yaps.petstore";
private static final String PASSWORD = "yapspwd";
```

Il est nécessaire d'adapter les valeurs de ces constantes en fonction de votre environnement. Après la modification, n'oubliez de recompiler et de déployer l'application.

Bug JMS

En consultant les traces du serveur Petstore, on peut rencontrer le message d'erreur suivant:

```
DirectConsumer:Caught Exception delivering messagecom.sun.messaging.jmq.io.Packet
cannot be cast to com.sun.messaging.jms.ra.DirectPacket
```

Ce problème est identifié comme un bug connu par SUN:

http://bugs.sun.com/view_bug.do?bug_id=6650996

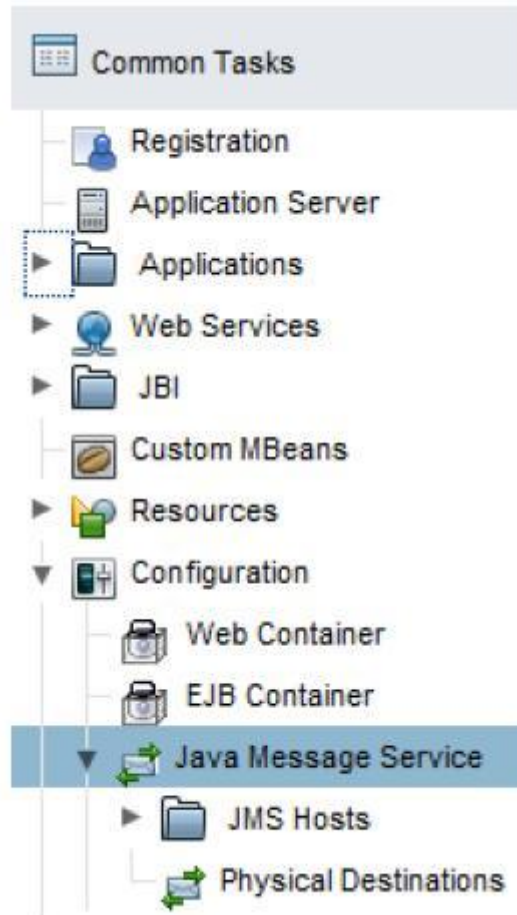
Le problème est lié à la manière dont le JMS Provider et l'Application Server sont liés.

Par défaut, la relation est de type « EMBEDDED ». Cela signifie que le serveur d'applications (GlassFish) et le JMS Provider (Sun Java System Message Queue software) fonctionnent sous la même JVM.

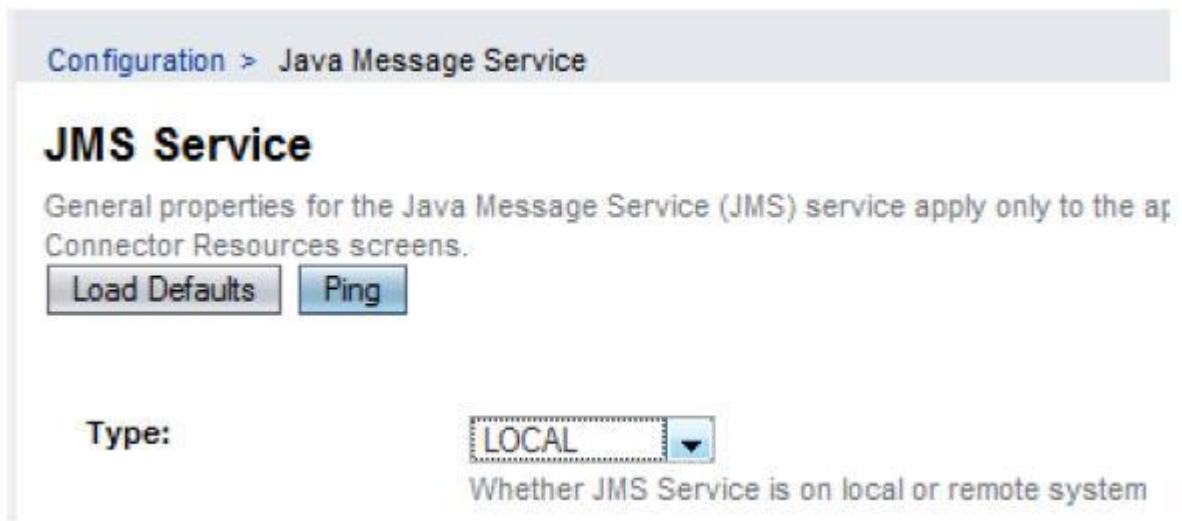
Pour contourner le problème, il est recommandé de changer le type de relation en « LOCAL ». Ce mode stipule que le JMS Provider est démarré dans sa propre JVM au moment où le serveur d'applications est lui-même démarré.

Pour modifier la configuration du service JMS, procédez comme suit :

- Démarrer la console d'administration du serveur Petstore sur le site suivant : <http://localhost:8282/>
- Dans la partie de gauche, sélectionnez la tâche « Configuration > Java Message Service » :



- Changer le type de relation de « EMBEDDED » à « LOCAL » :



- Sauver vos modifications.
- Cette modification nécessite un redémarrage du serveur.
- Redémarrez le serveur Petstore.

Les différents types de relation sont exposés dans la documentation de GlassFish (Sun Java System Application Server 9.1) disponible à l'adresse suivante :

<http://docs.sun.com/app/docs/doc/819-3672/beaob?a=view>

LLIX-I - Conclusion

Les sources du serveur de l'application « Yaps» ont été transférées sous NetBeans dans une arborescence conforme au format préconisé par le Java Blueprints.

Les tests que nous avons réalisés attestent que l'application Web, la logique métier et les Web Services communiquent correctement ensemble.

X - Importer le projet Yaps - côté client

Le projet YAPS Pet Store est composé de deux interfaces clientes. L'une dite légère et composée d'une interface Web (JSF) et l'autre dite lourde (Java Swing).

Comme nous l'avons vu (voir [ici](#)), le client « léger » de type Web est déployé avec les composants du serveur de l'application Yaps.

Nous allons décrire le processus d'importation du client « lourd » (Java Swing) sous le projet de type « Java Applications » baptisé Yaps-Client.

L'ensemble des opérations qui suivent est réalisé sous NetBeans.

LLX-A - Créer un nouveau projet

LLX-A-1-a - Fermer les projets

Vous allez créer un nouveau de groupe de projets qui sera spécifique à Yaps-Client.

Si c'est le cas, il y a d'abord lieu de fermer l'ensemble des projets actuellement ouverts sous NetBeans.

Sélectionnez l'onglet « Projects ».

Un clic droit fait apparaître le menu contextuel. L'option « Project Group (none) » permet de fermer l'ensemble des projets ouverts :

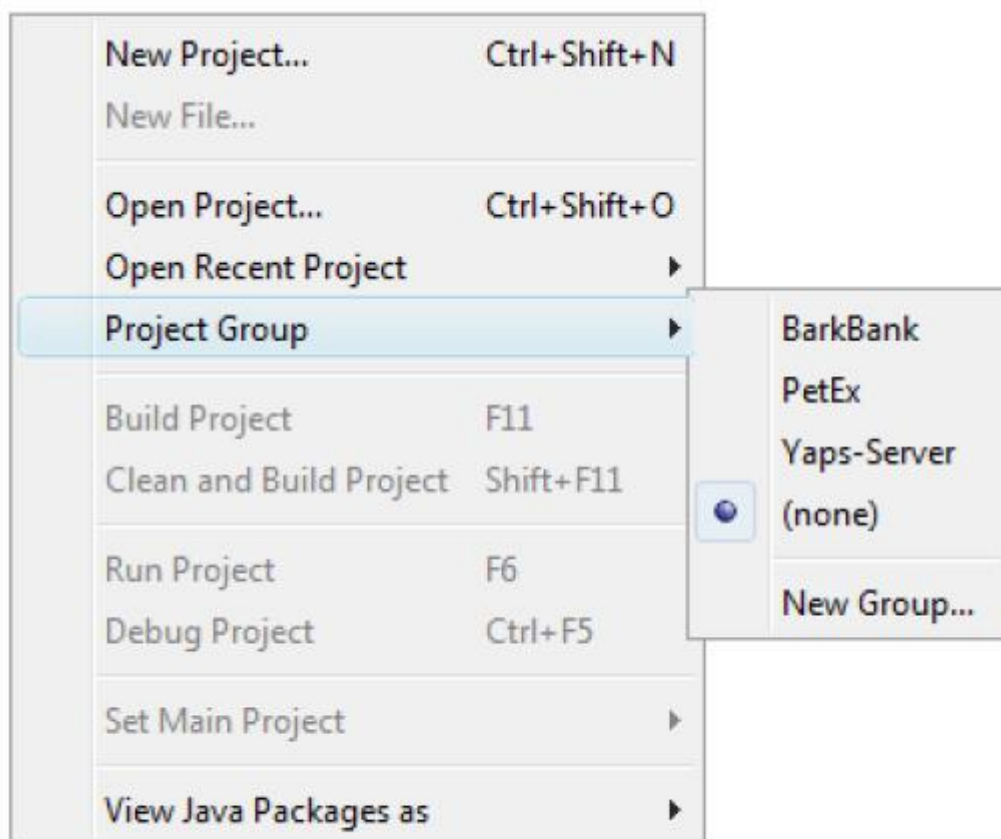


Figure 10-1: Fermer les projets ouverts

LLX-A-1-b - Type de projet

Créez un nouveau projet de type « Java Application ».

Pour cela, sélectionnez l'onglet « Project » et créez un nouveau projet via le menu « File -> New Project ». Le projet que vous allez créer est de type « Java > Java Application » :

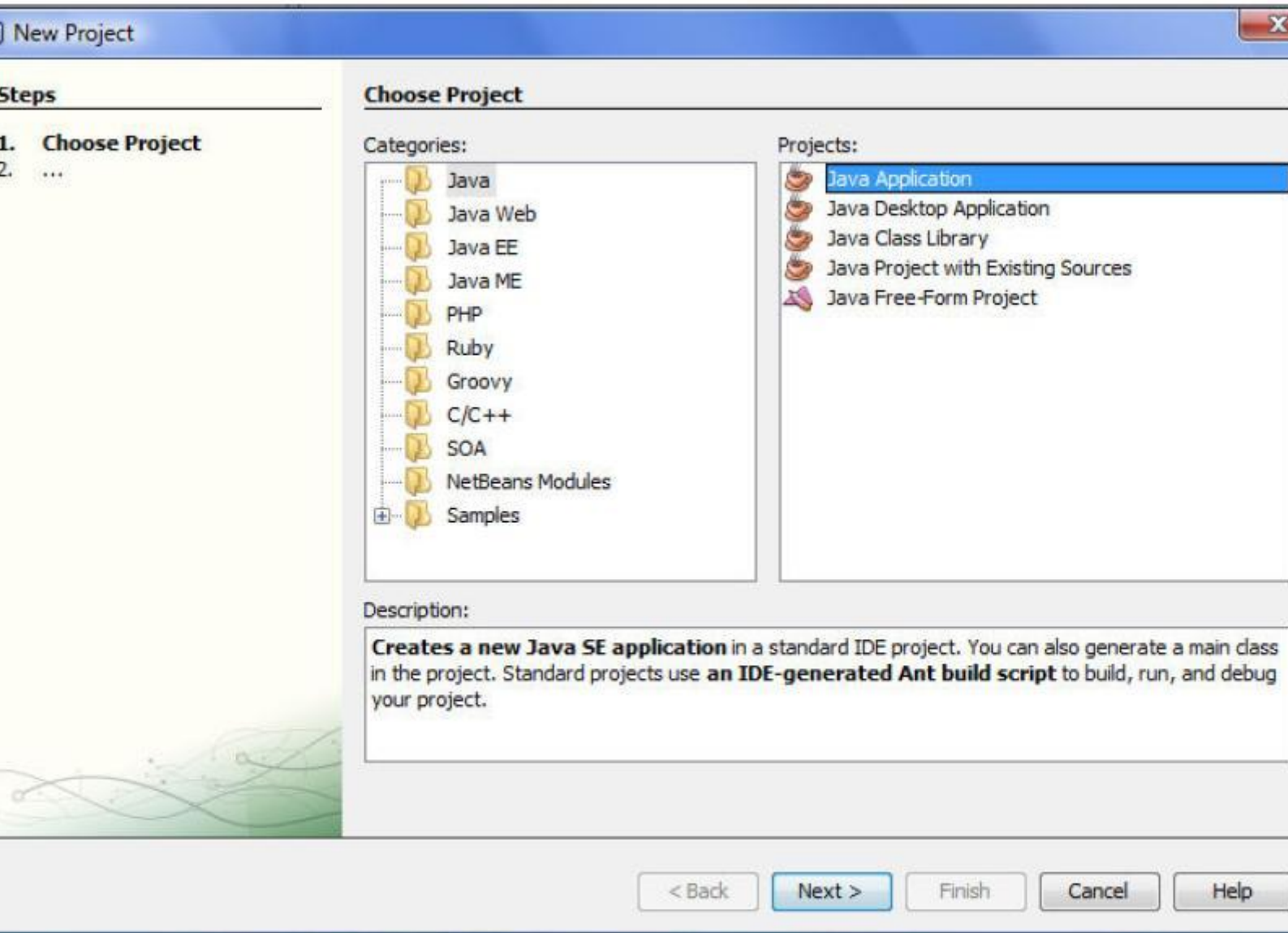


Figure 10-2: Nouveau projet Java Application

Cliquez sur le bouton « Next » pour continuer.

Java Desktop Application ?

Nous aurions tendance à opter pour un projet de type « Java Desktop Application » préconisé pour le développement d'applications de type Swing.

L'application client de Yaps Pet Store est certes de type Swing mais son fonctionnement a été enrichi par l'apport d'un framework baptisé XSwing.

Nous avons ainsi préféré rester dans le contexte d'une application Java standard pour rester dans l'esprit de l'application originale.

LLX-A-1-c - Nom et répertoires

L'écran suivant nous permet de nommer le projet, et de définir le dossier cible :

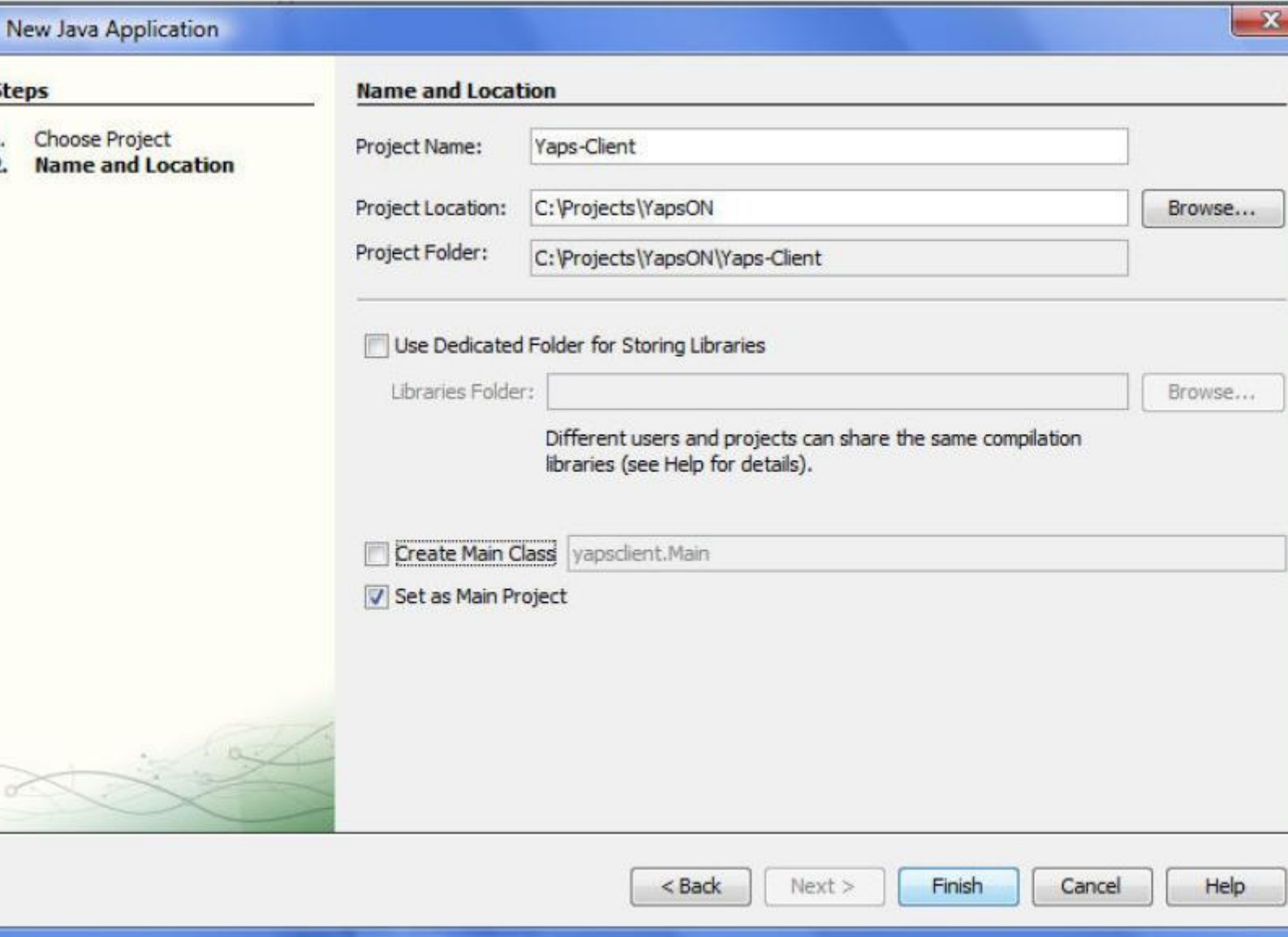


Figure 10-3: Nom du projet et répertoire

Les informations à introduire sont les suivantes :

| | |
|-----------------|--------------------------------|
| Project Name: | Yaps-Client |
| Project Folder: | C:\Projects\YapsON\Yaps-Client |

Il n'est pas nécessaire de cocher l'option « Create Main Class », car nous allons importer nos propres codes sources.

Project Folder

Le répertoire du projet est donné ici à titre d'exemple et doit correspondre au répertoire où vous avez décompressé le fichier d'archives de l'application YapsON (voir [ici](#)).

Cliquez sur le bouton « Finish » pour terminer le processus.

LLX-A-1-d - Génération du projet

Le projet est maintenant généré et présente l'ensemble des dossiers suivants :

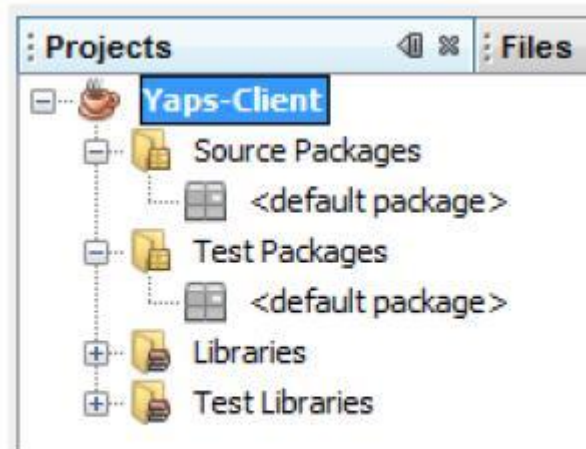


Figure 10-4: Dossiers du projet

Vous êtes prêt à importer le code source du projet Yaps (côté client) de l'application Yaps vers l'arborescence de NetBeans.

LLX-B - Transférer le projet sous NetBeans

L'opération de transfert du projet Yaps consiste à copier les fichiers de la partie serveur de l'application Yaps vers l'application YapsON.

Pour faciliter cette opération, le script Ant « **yapson.xml** » (décrit dans la section 3.5) automatise la copie des fichiers.

Assurez-vous d'avoir configuré le fichier « yapson.properties » tel que décrit dans la section 4.

LLX-B-1 - Positionner les variables d'environnement

Les commandes que vous allez exécuter nécessitent de positionner les variables d'environnement pour l'exécution des tâches Ant.

Rendez-vous dans le répertoire de l'application YapsON et exécutez le fichier de commande « setenv-yaps.cmd » :

```
C:> cd {yapson_home}
{yapson_home}> setenv-yaps.cmd
```

LLX-B-2 - Transférer le projet

À partir de la ligne de commande, exécutez la tâche Ant « **transferYapsClient** » chargée de transférer le projet sous NetBeans :

```
%YAPSON-HOME%> ant -f yapson.xml transferYapsClient
```

```
Buildfile: yapson.xml
transferYapsClient:
[echo] Transfer to NetBeans the client's side of the Yaps project
...
```

L'exécution du script ANT commence par demander une confirmation à l'utilisateur avant de procéder au remplacement des fichiers :

```
...
[input] The target source files will be overwritten by the original YAPS project. Do you
want to continue? (y, n)
y
...
```

La tâche Ant copie code source Java du projet :

```
...
[echo] === Copy Java Sources ===
[copy] Copying 82 files to c:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client
...
```

Pour terminer, les bibliothèques sont copiées dans un répertoire spécifique :

```
...
[echo] === Copy Libraries ===
[copy] Copying 2 files to c:\Projects\YapsON\Tools\Lib
BUILD SUCCESSFUL
```

NetBeans rafraîchit automatiquement le projet Yaps-Client :

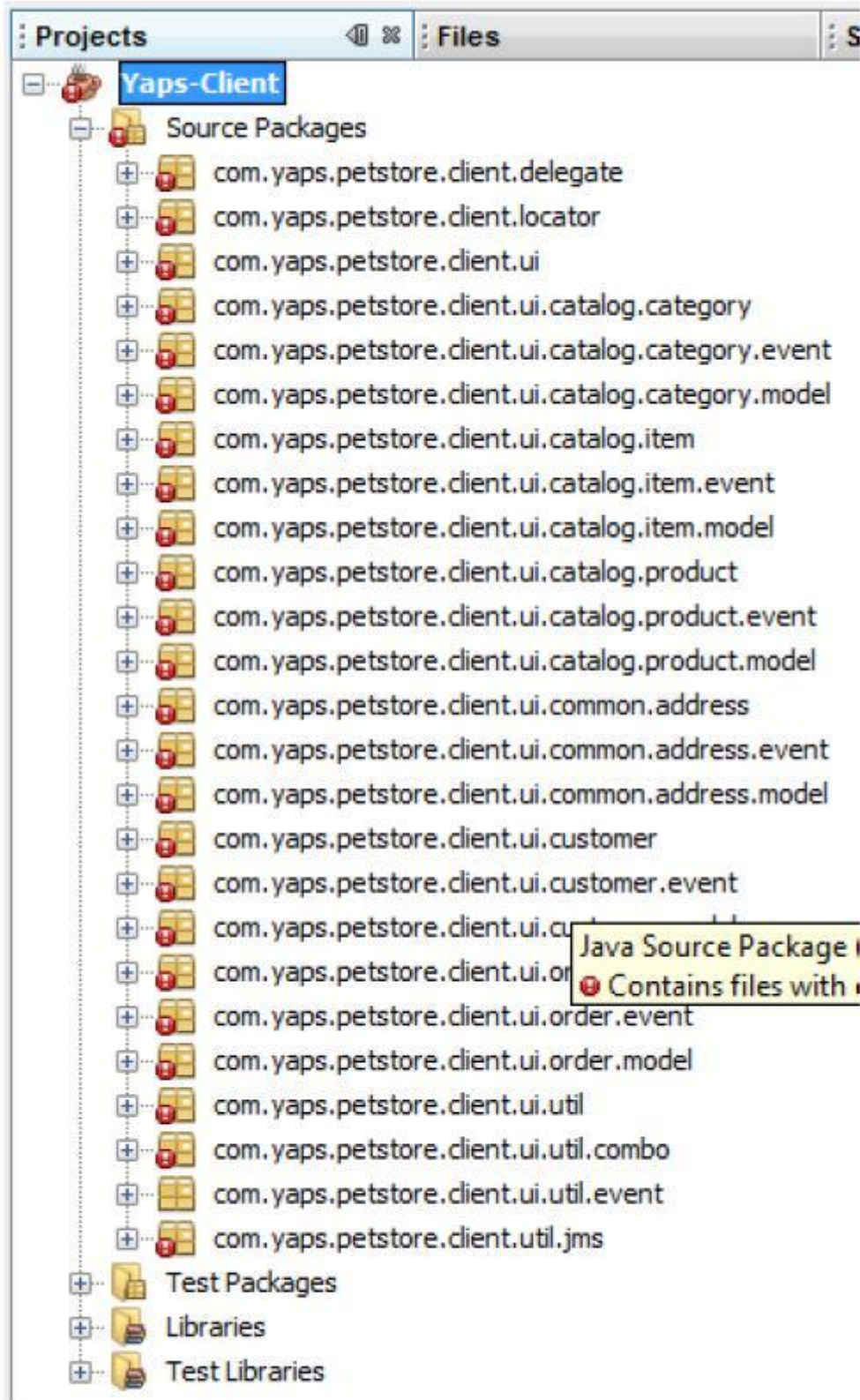


Figure 10-5: Projet Yaps-Client après le transfert

LLX-B-3 - Référencer les librairies

Comme on peut le constater, des erreurs apparaissent dans le code source du projet.

Ces erreurs sont causées par l'absence de bibliothèques requises par le projet :

- Les classes des EJBs du projet Yaps-Server
- Les bibliothèques utilitaires (Apache Commons, Framework XSwing)
- Les bibliothèques JMS
- Les bibliothèques de TopLink

Les références à ces bibliothèques se réalisent en modifiant les propriétés du projet.

Nous allons décrire comment référencer chacune de ces bibliothèques ainsi que la manière de créer certaines d'entre elles sous NetBeans.

LLX-B-3-a - Référencer les classes de Yaps-Server

L'application *Yaps-Client* référence un ensemble de classes définies dans le projet Yaps-Server : entity, stateless, util, exception.

Vous allez ajouter la référence à la bibliothèque contenant les classes construites dans le cadre de l'application *Yaps-Server* et plus spécifiquement les classes du module *Yaps-Server-ejb*.

Un clic droit sur le projet Yaps-Client fait apparaître le menu contextuel dans lequel vous allez activer l'option *Properties* :

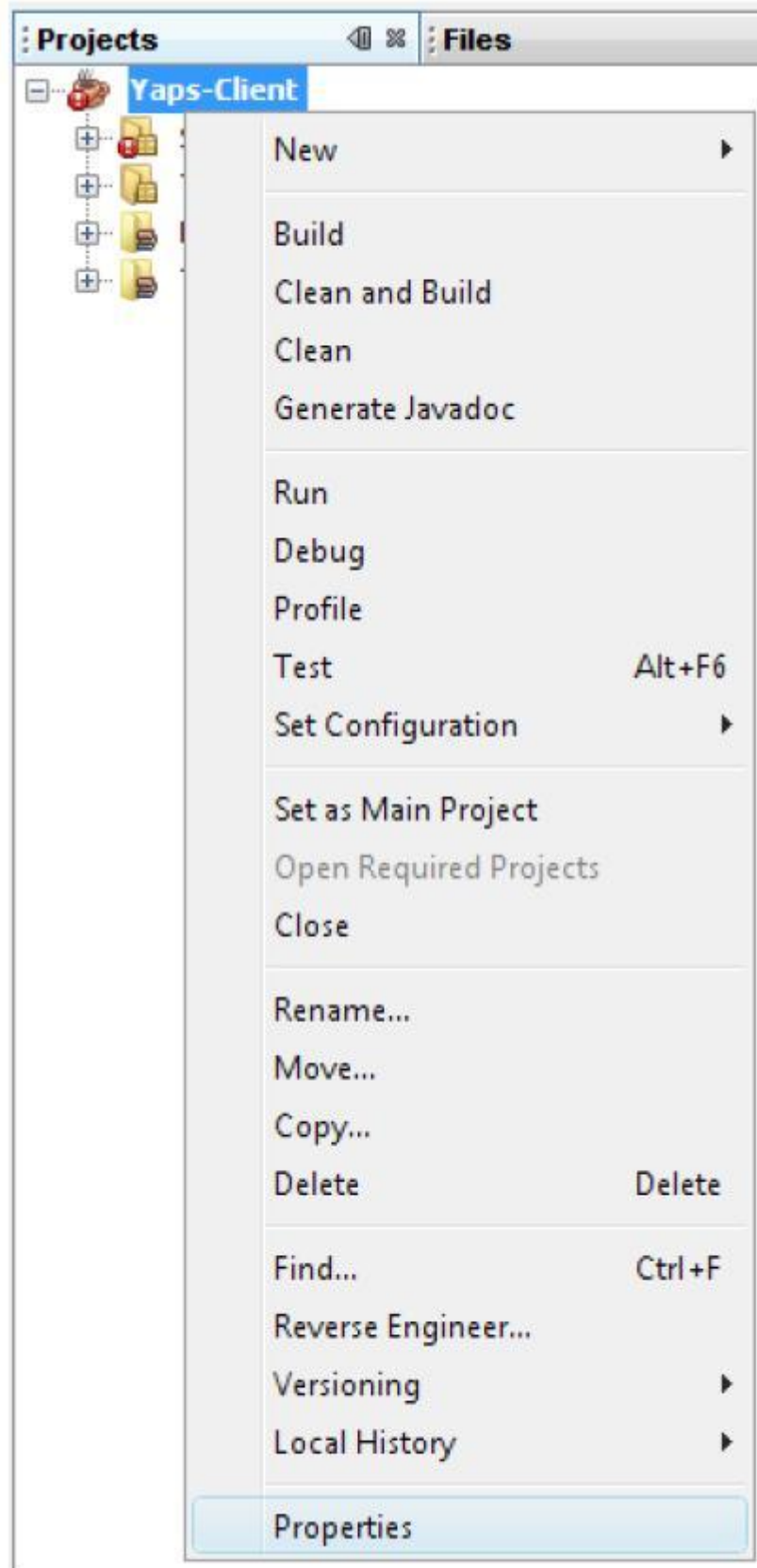


Figure 10-6: Menu propriétés

Dans l'écran des propriétés du projet, sélectionnez la catégorie *Libraries* :

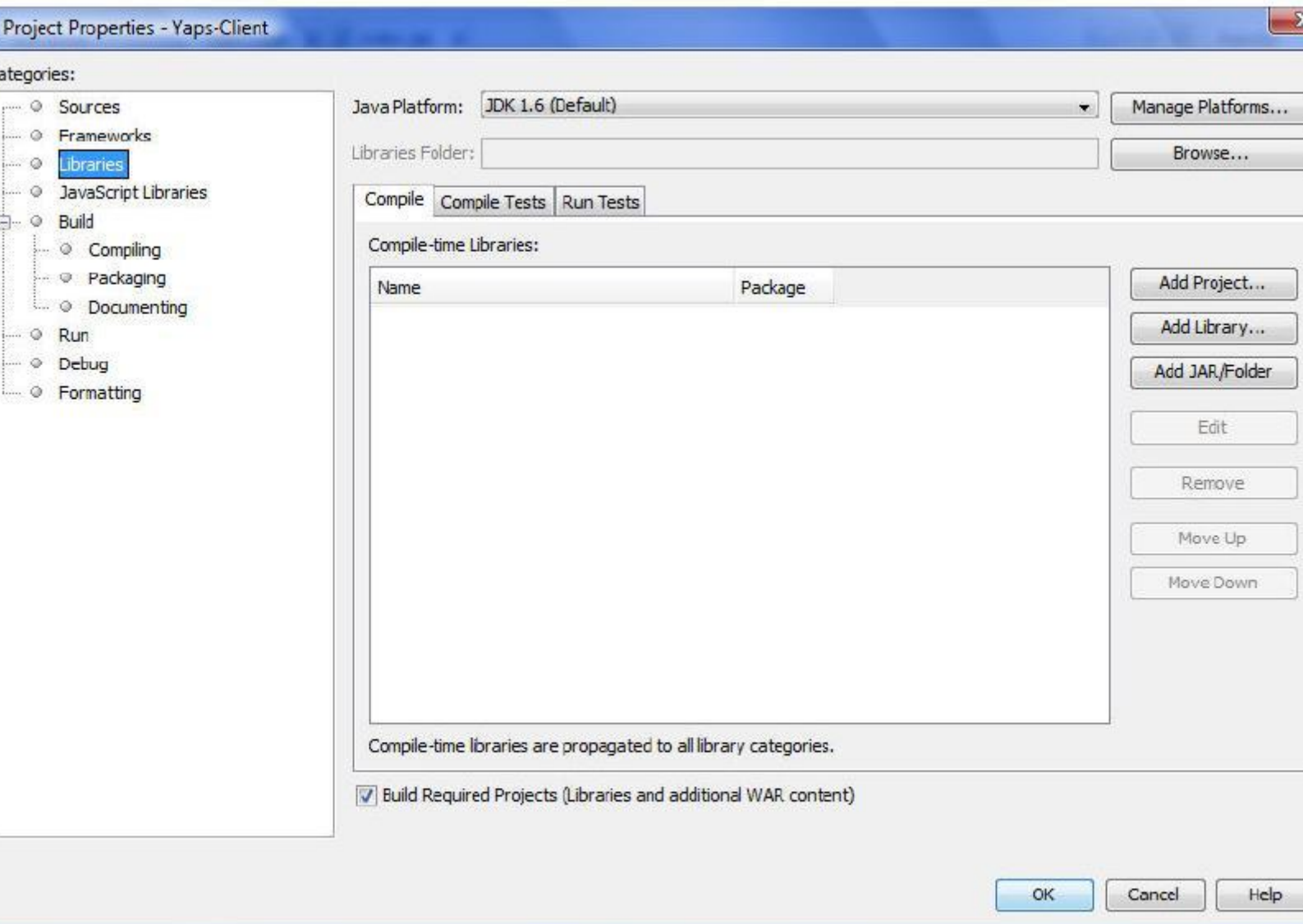


Figure 10-7: Librairies

Dans la partie de droite, cliquez sur le bouton « Add JAR/Folder ». La boîte de dialogue permet de sélectionner la librairie à référencer dans le projet. Dans notre cas, vous devez retrouver la librairie *Yaps-Srvr-ejb.jar* qui se trouve sous le répertoire *Build* du projet *Yaps-Server* :

{yapson_home}\build\Yaps-Server-ejb.jar

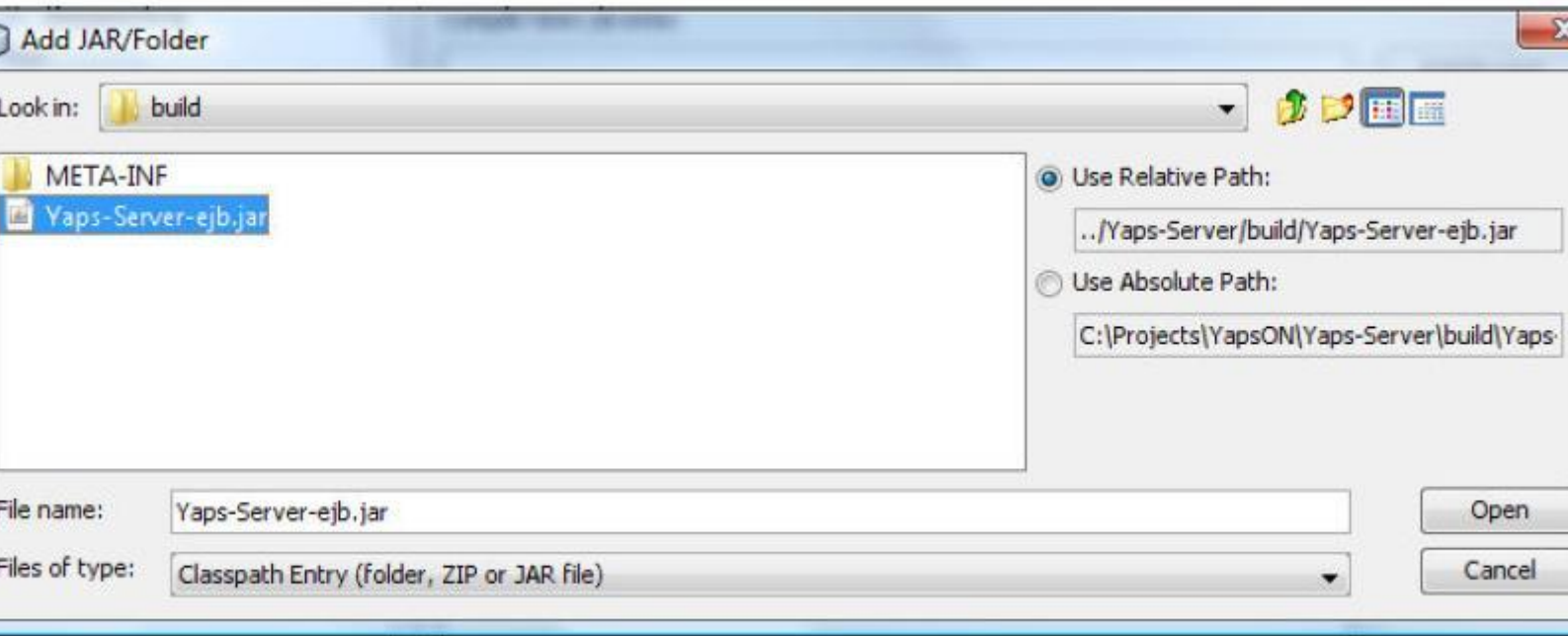


Figure 10-8: Librairie de Yaps-Server-ejb

À ce stade, les librairies référencées dans le projet sont les suivantes :

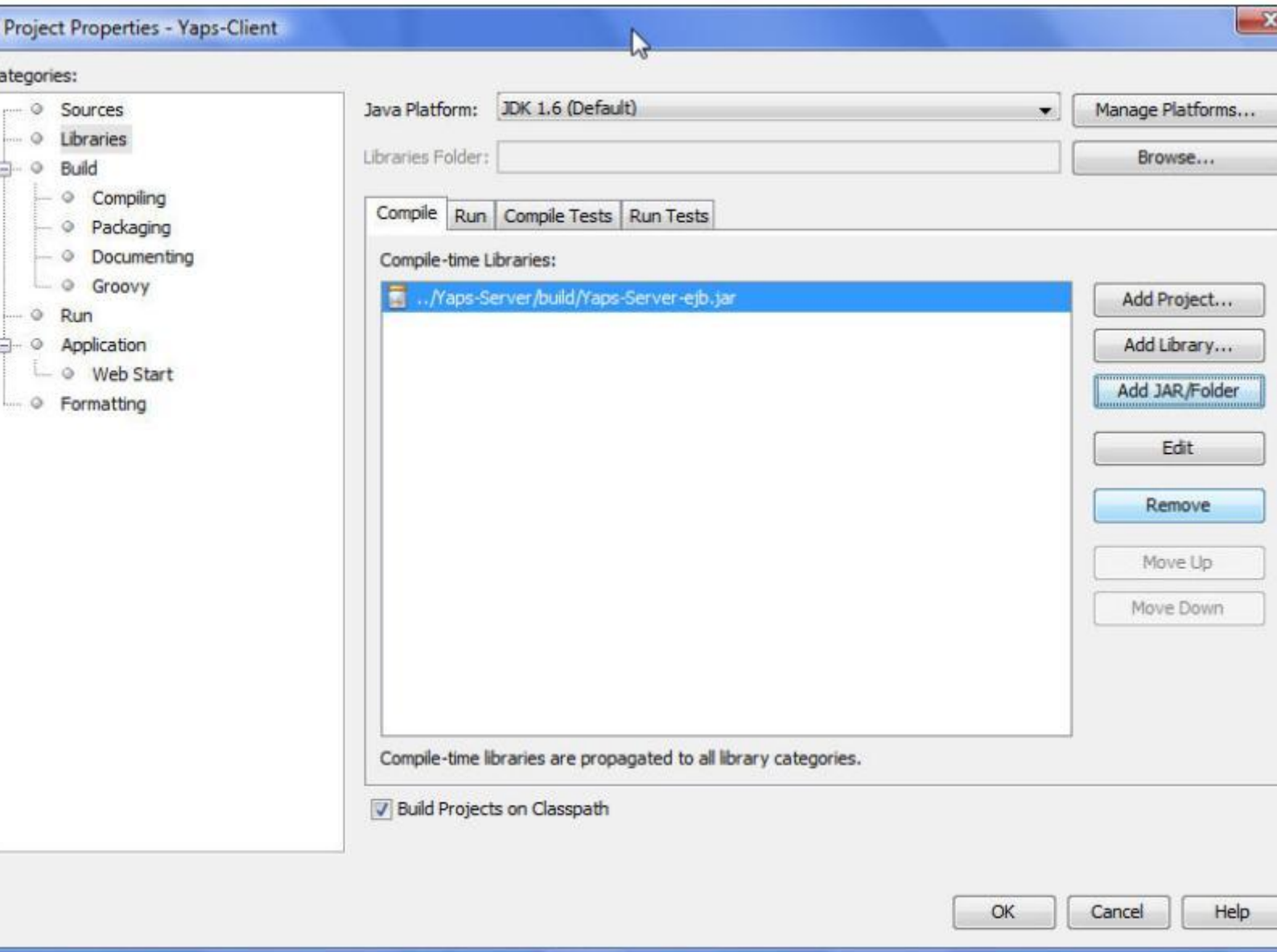


Figure 10-9: Liste des libraires

Cliquez sur « OK » afin d'ajouter la référence de la librairie au projet.

En consultant le projet, on constate la disparition des erreurs relatives au référencement des classes du projet *Yaps-Server*.

LLX-B-3-b - Référencer les librairies utilitaires

En utilisant la méthode décrite précédemment, vous allez référencer les deux librairies utilitaires transférées du projet *Yaps*.

Ces deux librairies sont localisées sous le répertoire *suivant* :

```
{yapson_home}\Tools\Libs\commons-lang-2.3.jar
{yapson_home}\Tools\Libs\vstm-xcore-0.5.0.jar
```

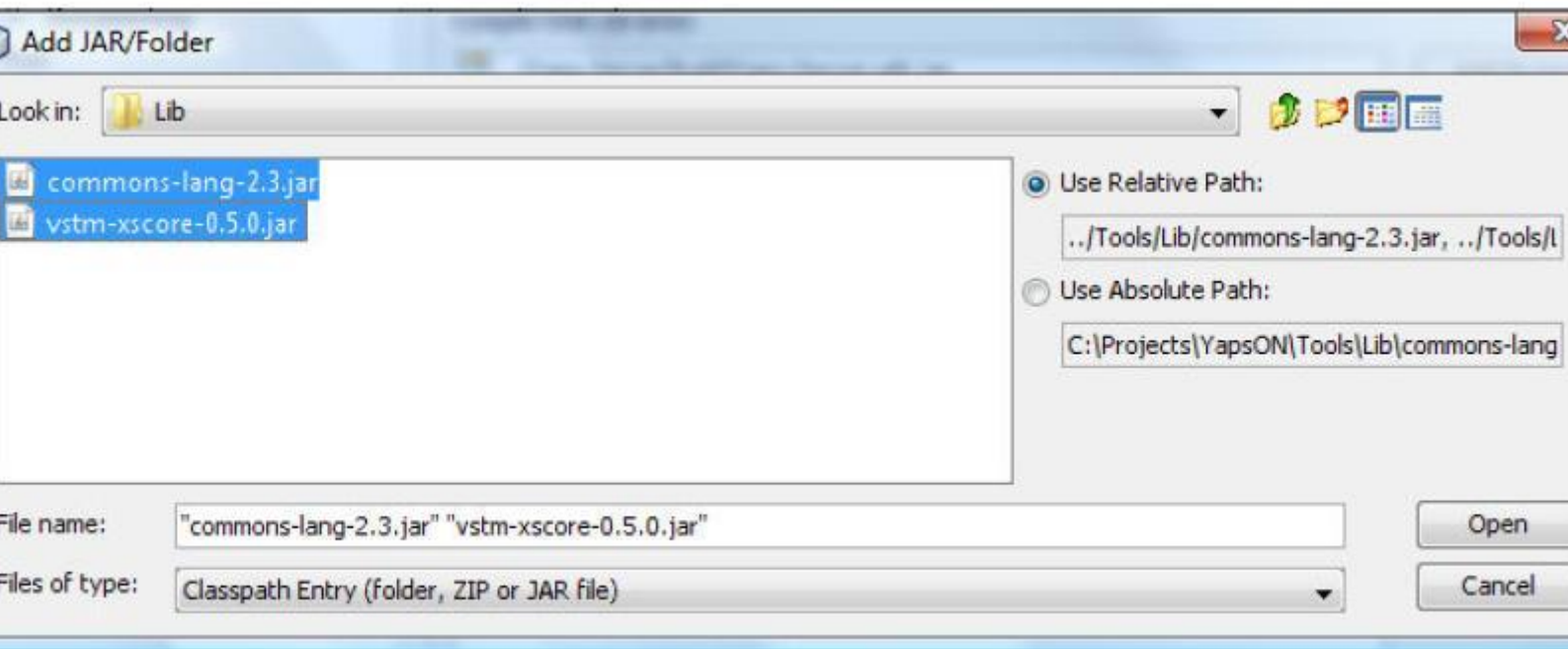


Figure 10-10: Librairies utilitaires

À ce stade, les librairies référencées dans le projet sont les suivantes:

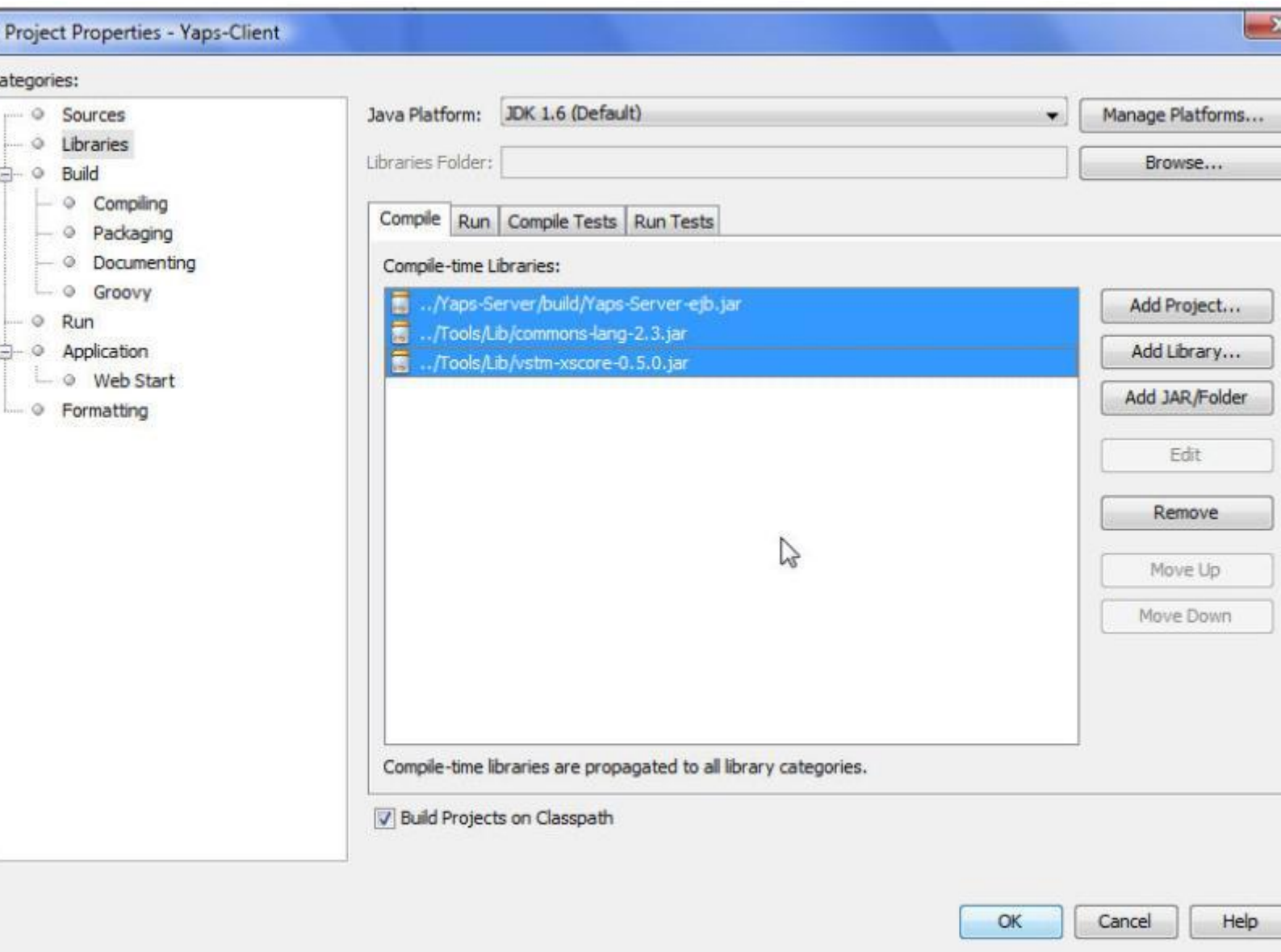


Figure 10-11: Liste des librairies

Cliquez sur « OK » afin d'ajouter la référence des libraires au projet.

En consultant le projet, on constate la disparition des erreurs relatives au référencement des classes d'*Apache Commons* et du framework *XSwing*.

LLX-B-3-c - Référencer les librairies JMS

L'application cliente met en oeuvre un *MessageListener* JMS pour être à l'écoute de la file de messages.

Il est donc nécessaire de référencer l'ensemble des classes JMS pour permettre à l'application de se connecter au topic JMS, de souscrire aux messages et de recevoir lesdits messages.

Le fournisseur JMS utilisé par l'application YAPS est celui fourni en standard avec Sun GlassFish à savoir Sun Java System Message Queue software.

Vous allez ajouter les références aux libraires JMS livrées avec Sun GlassFish.

Sous NetBeans, sélectionnez le menu « *Tools > Libraries* » :

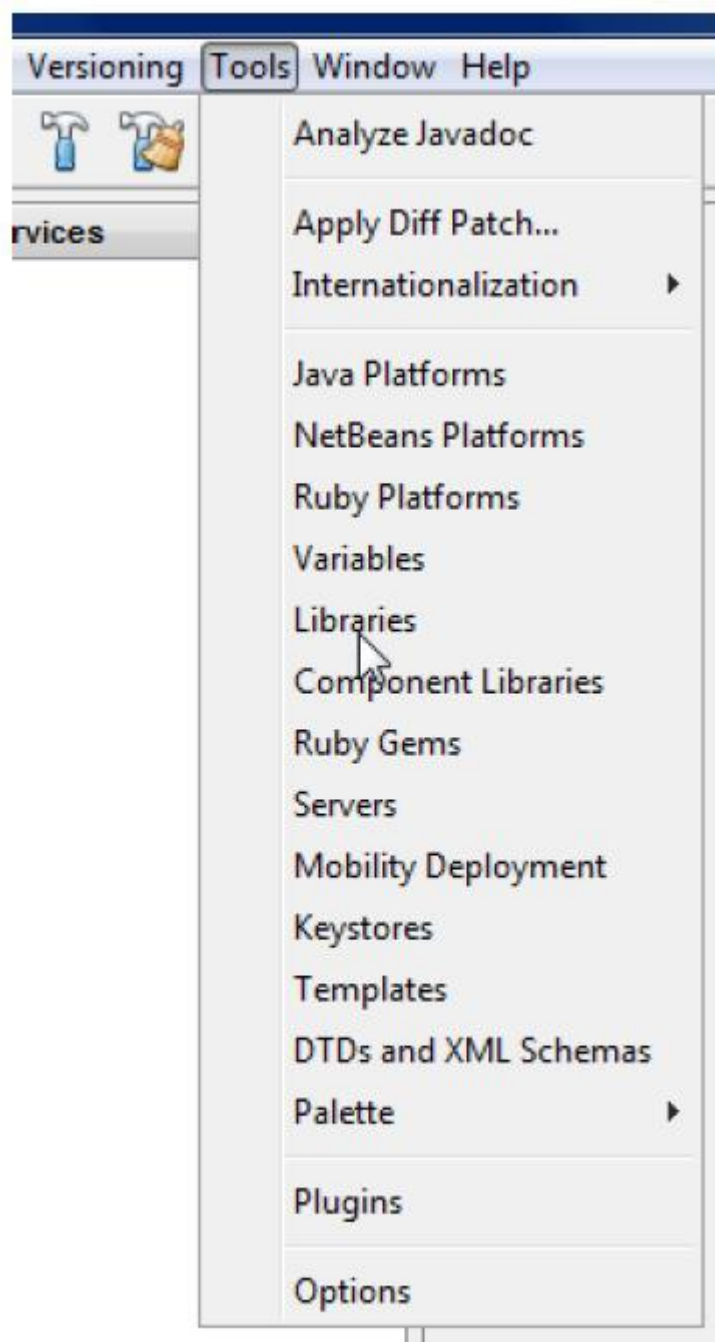


Figure 10-12: Menu Libraries

L'écran suivant montre l'ensemble des librairies définies sous NetBeans :

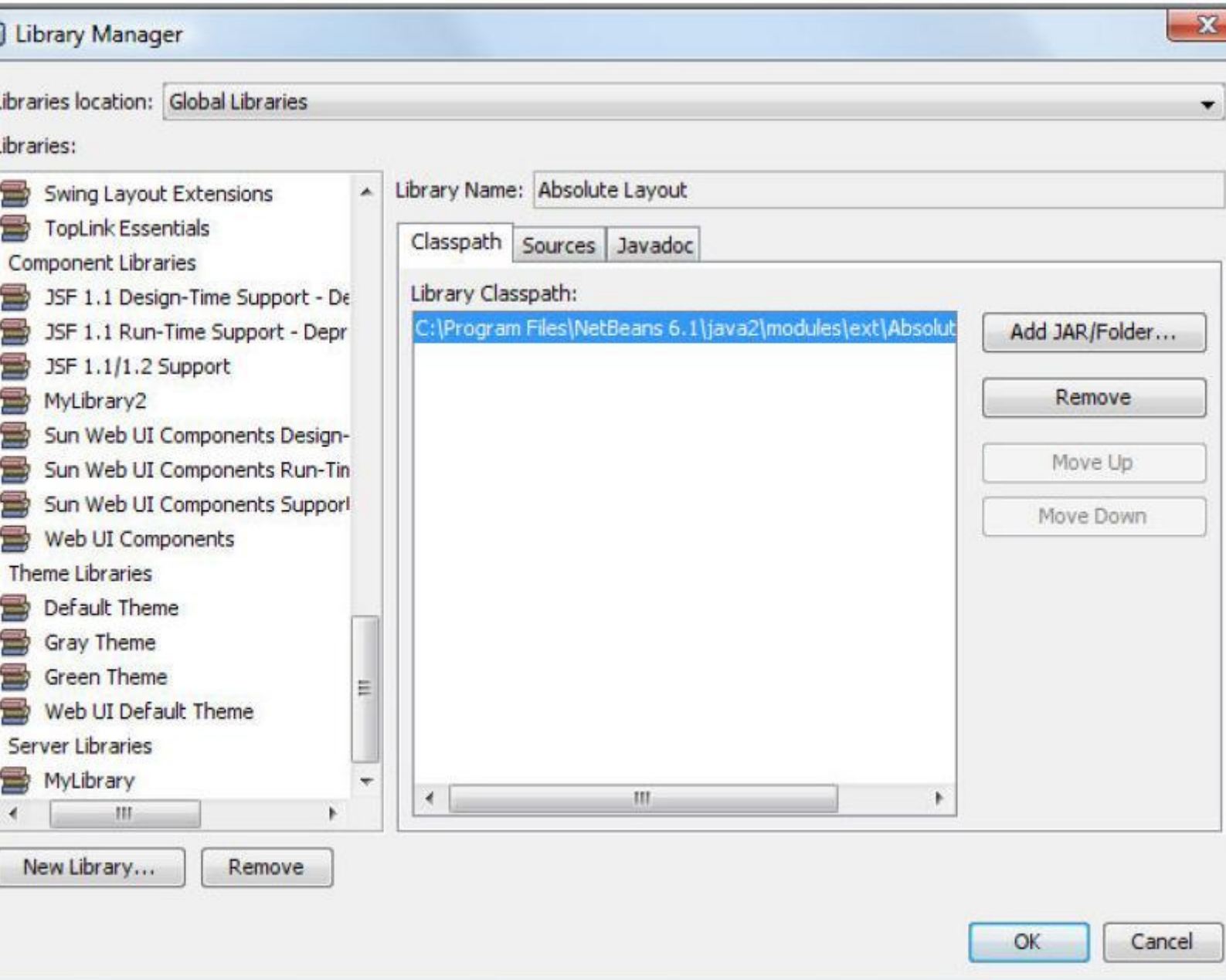
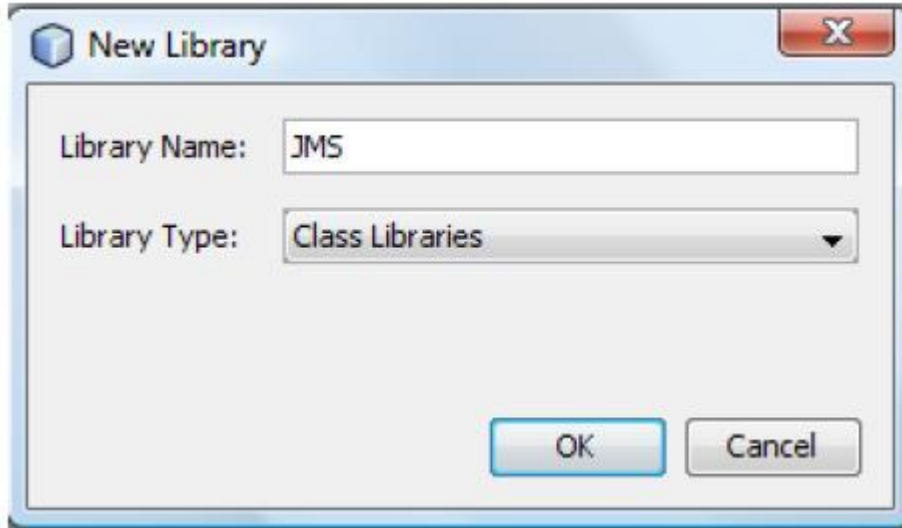


Figure 10-13: Bibliothèques sous NetBeans

Vous allez ajouter une nouvelle bibliothèque.

Cliquez sur le bouton « **New Library** » pour ajouter une nouvelle bibliothèque de type « Class Libraries » que nous allons nommer « **JMS** » :



Cliquez sur « OK » pour poursuivre.
À ce stade, la librairie JMS est sélectionnée.

En utilisant le bouton « Add Jar/Folder », vous allez ajouter l'ensemble des fichiers suivants localisés sous le répertoire de Sun GlassFish :

- 1 {glassfish-home}/lib/install/applications/jmsra/imqjmsra.jar
- 2 {glassfish-home}/lib/appserv-admin.jar
- 3 {glassfish-home}/imq/lib/imq.jar
- 4 {glassfish-home}/lib/appserv-rt.jar
- 5 {glassfish-home}/lib/javaee.jar

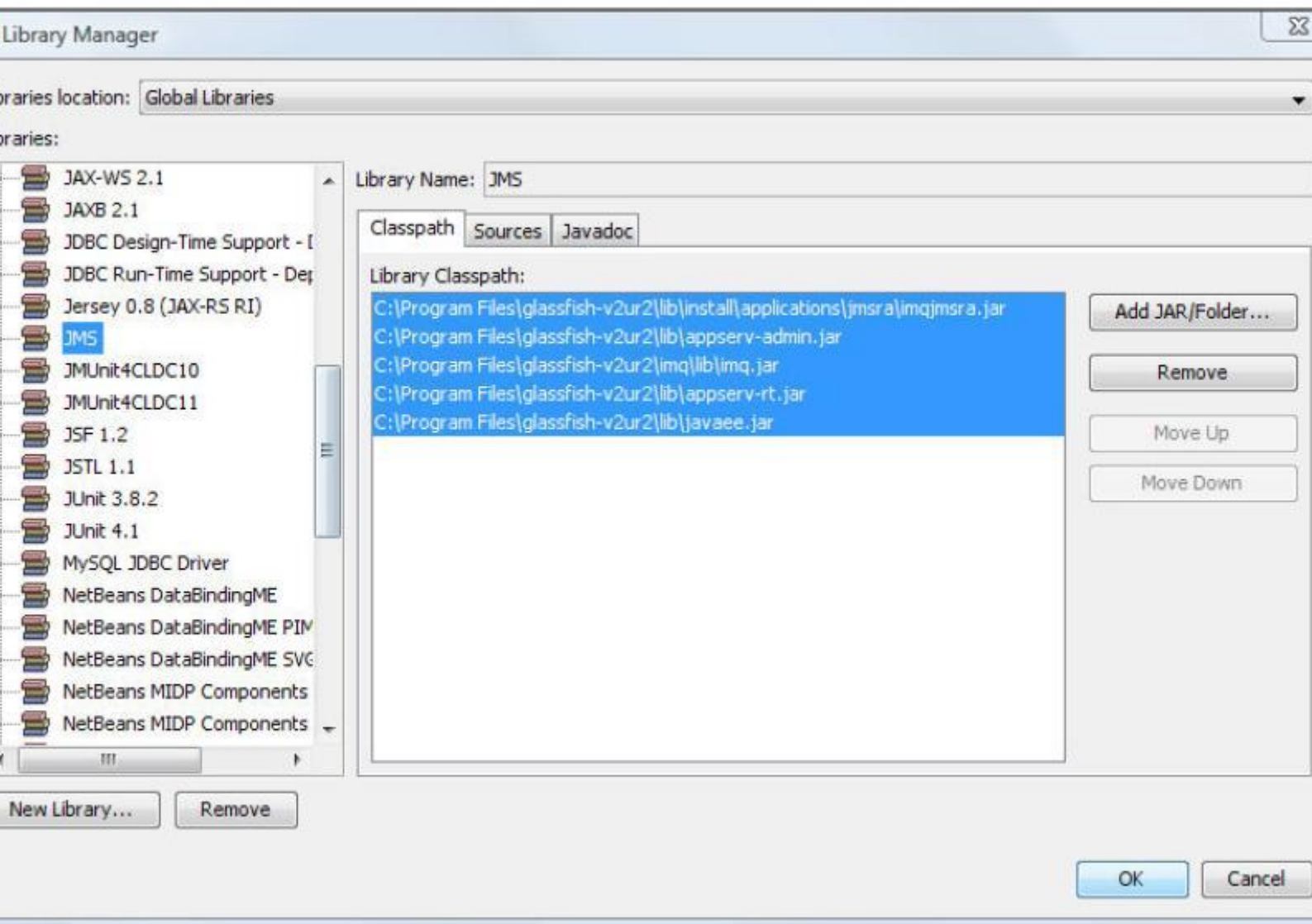


Figure 10-14: Liste des fichiers pour JMS

Les boutons « Move Up » et « Move Down » permettent d'agencer l'ordre des fichiers tel que nous l'avons décrit plus haut.

Cliquez sur le bouton OK pour appliquer vos modifications.

Maintenant que la librairie JMS est prête à l'emploi, vous allez ajouter sa référence au projet Yaps-Client.

Ouvrez les propriétés du projet Yaps-Client et sélectionnez la catégorie « Libraries » :

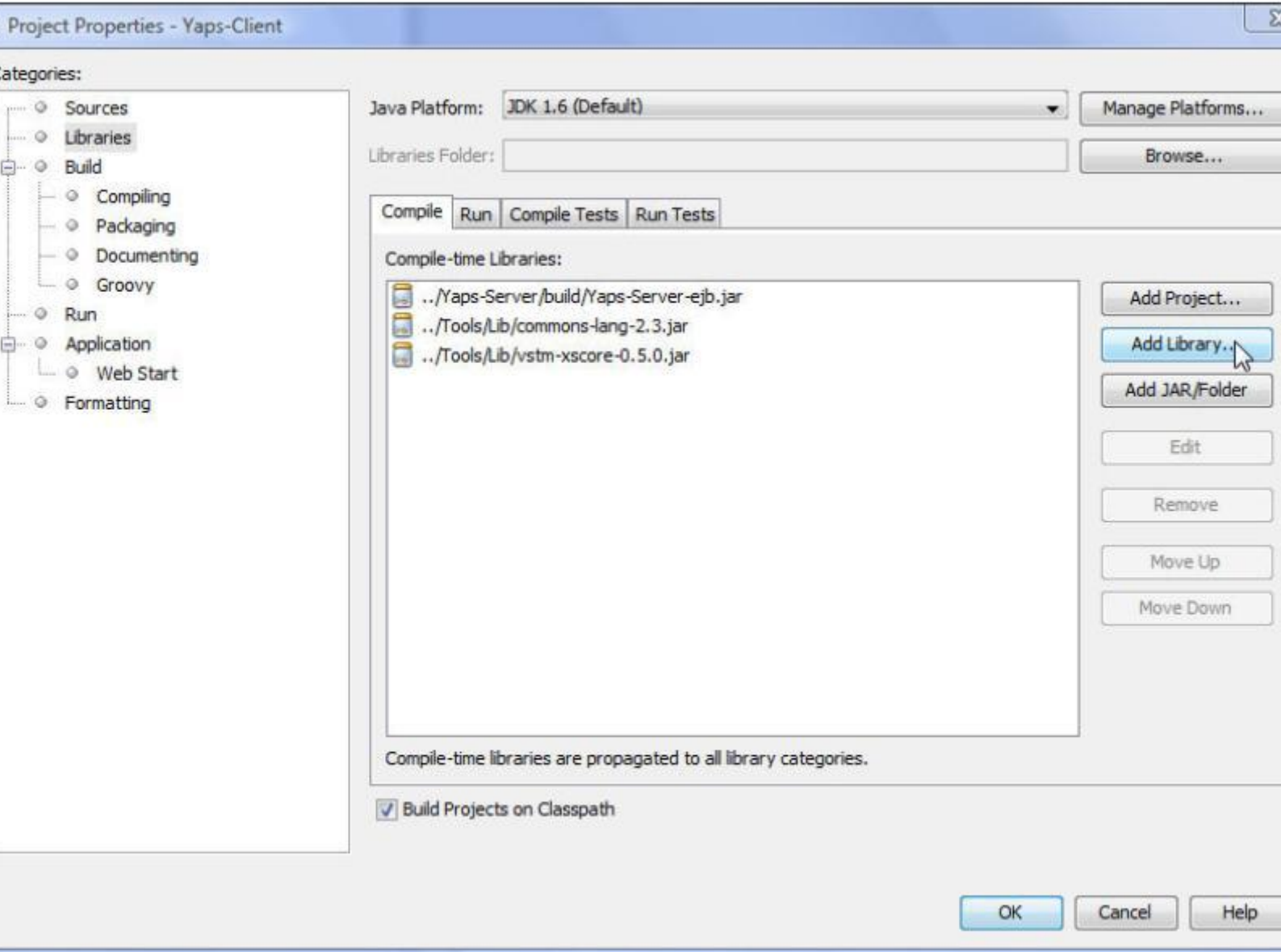


Figure 10-15: Propriétés du projet

Cliquez sur le bouton « Add Library » :

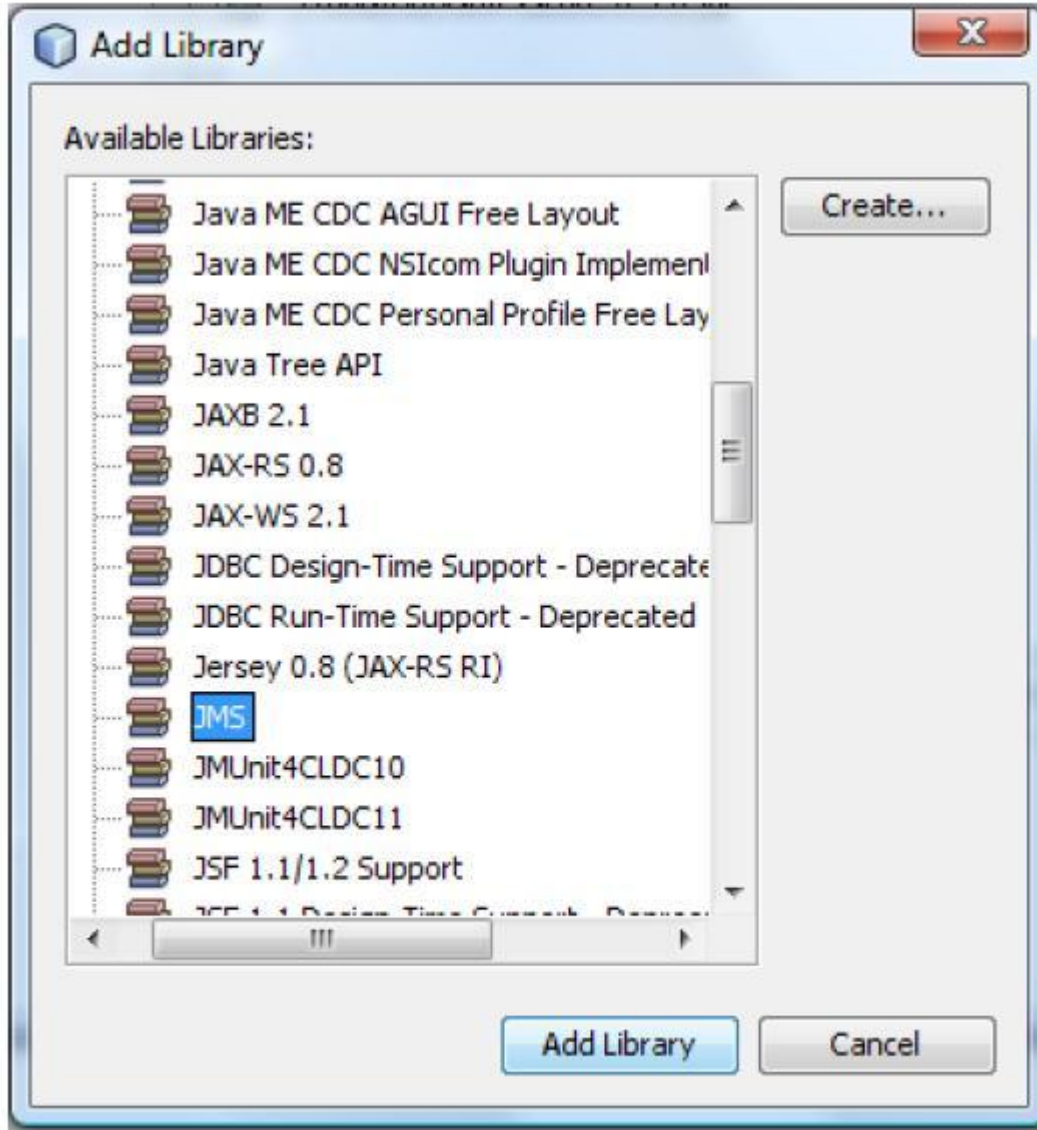


Figure 10-16: Référence à la librairie JMS

Dans la liste des librairies, sélectionnez « JMS » et cliquez sur le bouton « Add Library » :

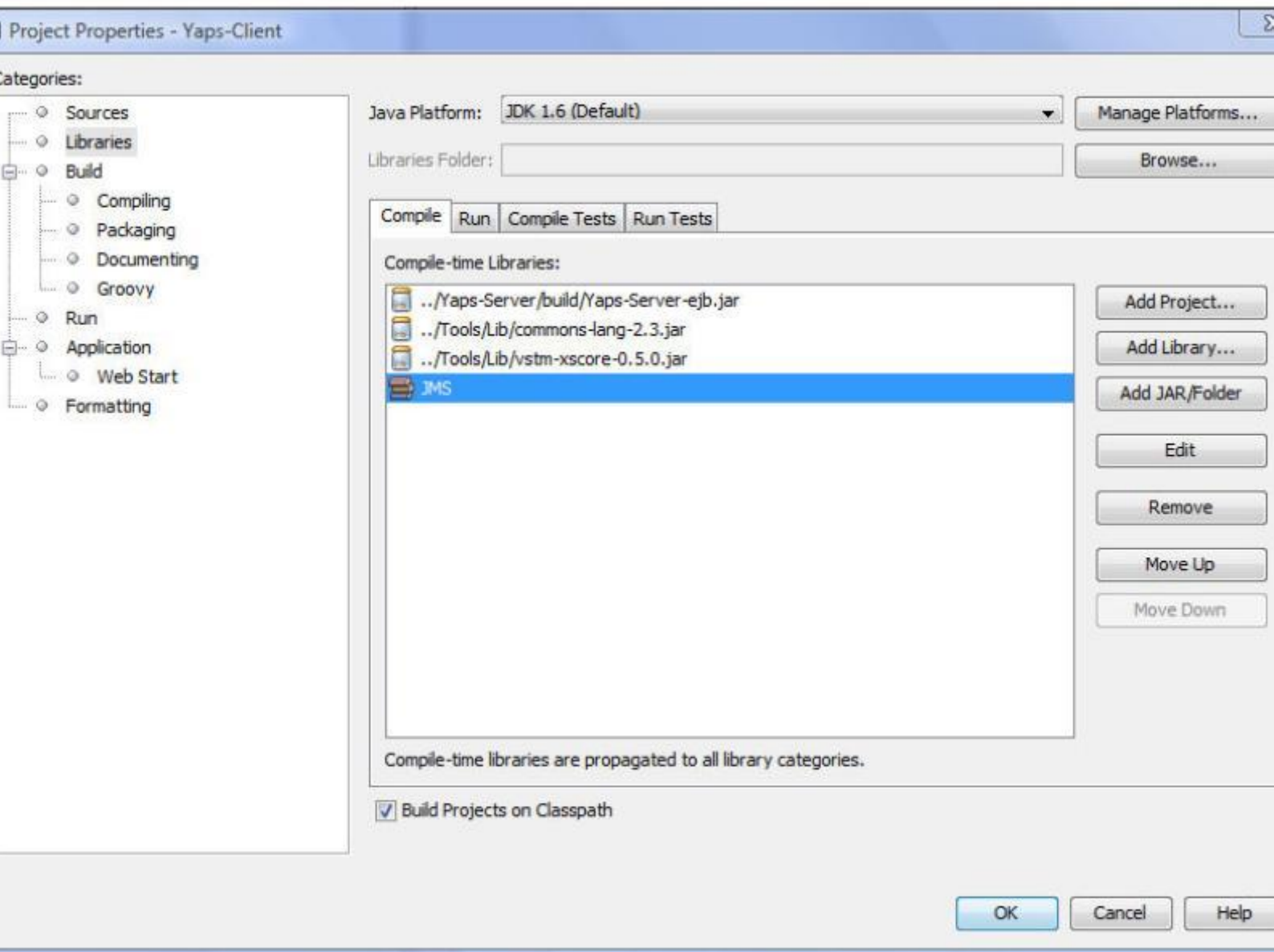


Figure 10-17: Librairie JMS dans le projet

Cliquez sur « OK ».

La librairie JMS est maintenant référencée dans le projet Yaps-Client.

NetBeans va automatiquement compiler le projet et résoudre les références manquantes.

À ce stade, plus aucune erreur ne doit être détectée dans le projet :

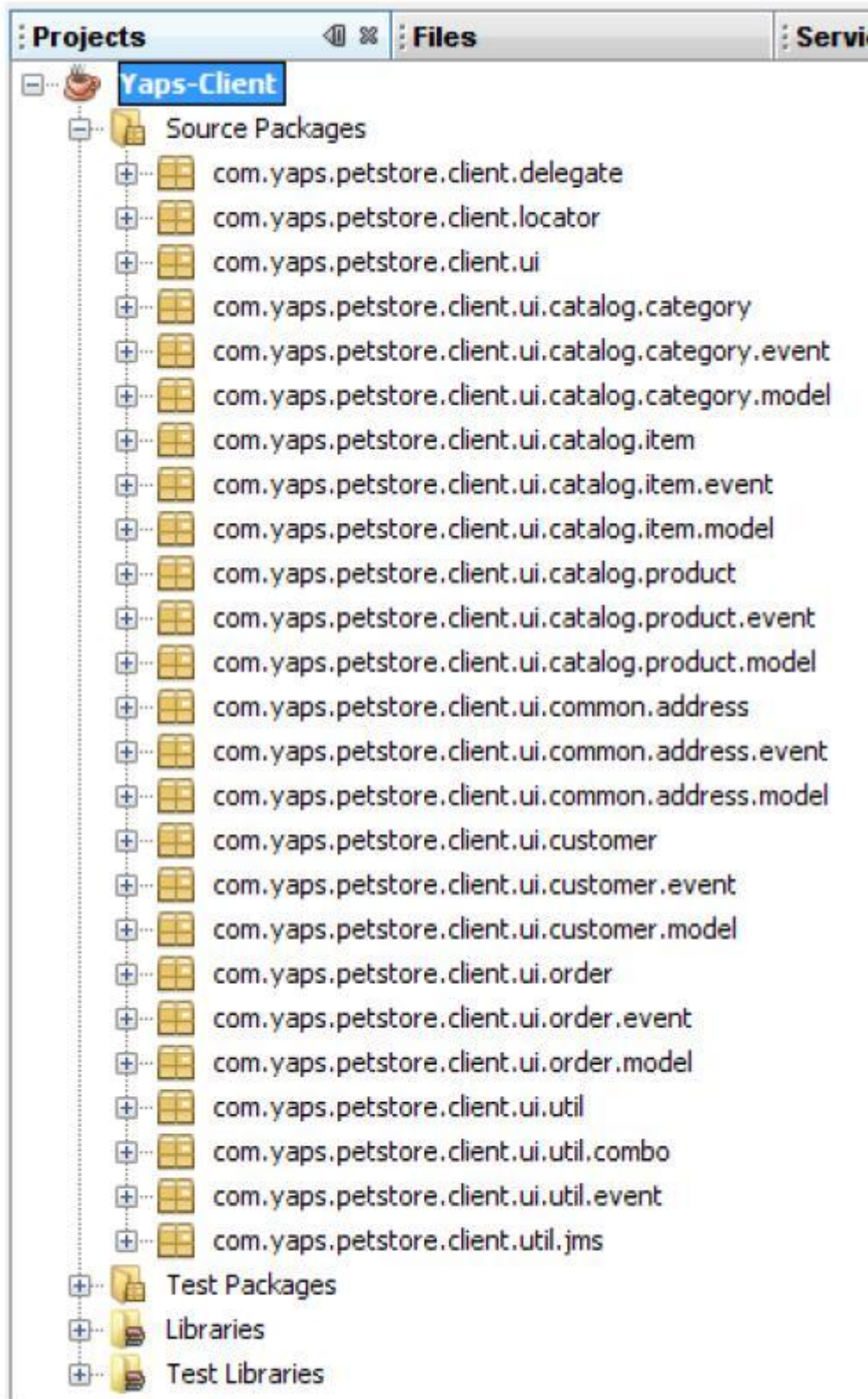


Figure 10-18: Résolution des problèmes

LLX-B-3-d - Référencer les librairies TopLink

Durant la phase d'exécution de l'application Yaps-Client, des objets de type Entity sont manipulés pour lesquelles les classes du framework TopLink sont requises.

Vous allez référencer les classes du framework TopLink pour éviter les erreurs d'exécution.

Ouvrez les propriétés du projet et sélectionnez la catégorie « Libraries » :

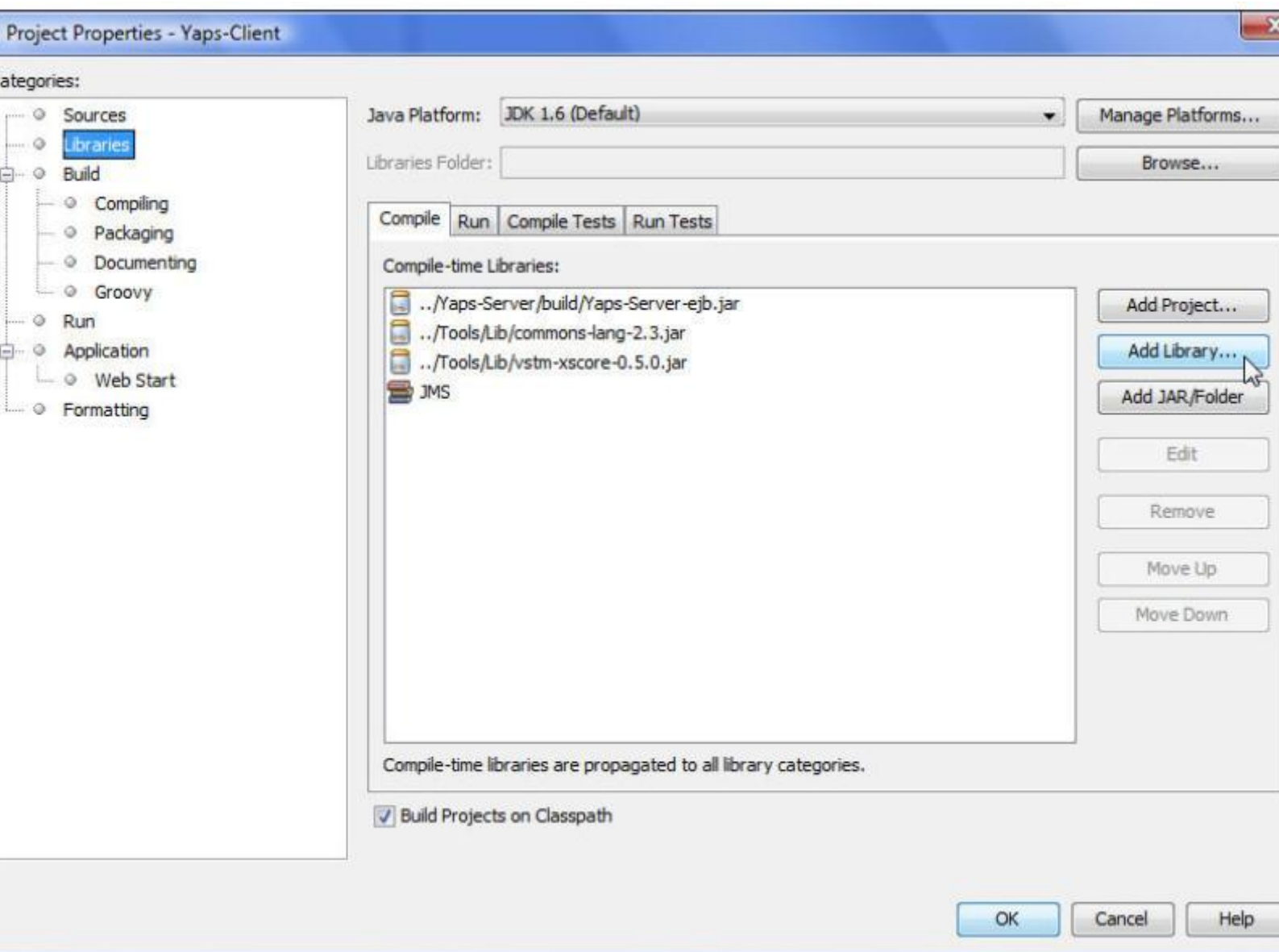


Figure 10-19: Librairies du projet

Cliquez sur le bouton "Add Library" et sélectionnez la librairie *TopLink Essentials* :

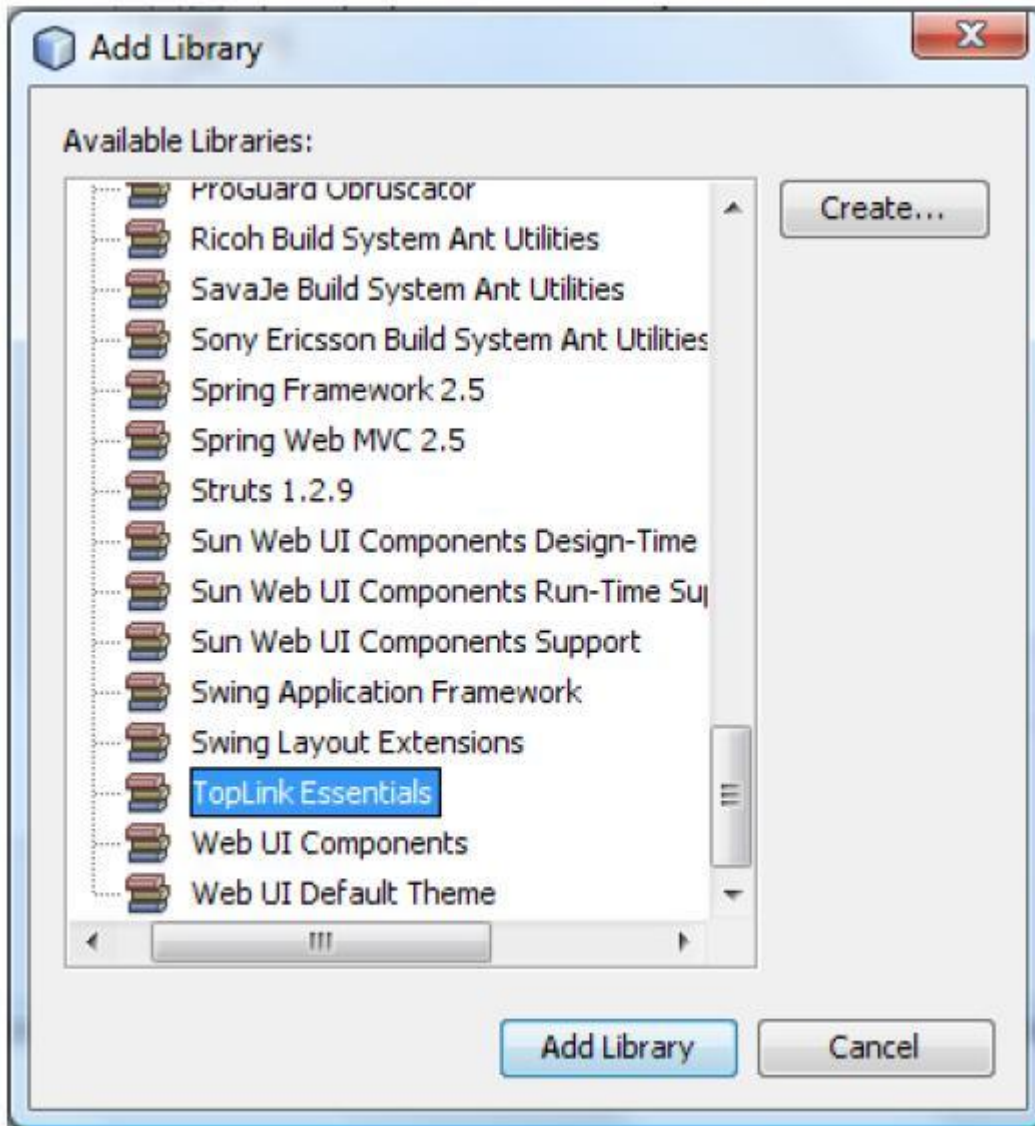


Figure 10-20: Libraire TopLink Essentials

Cliquez sur « Add Library ».

Cliquez ensuite sur OK afin d'ajouter la référence de la librairie TopLink :

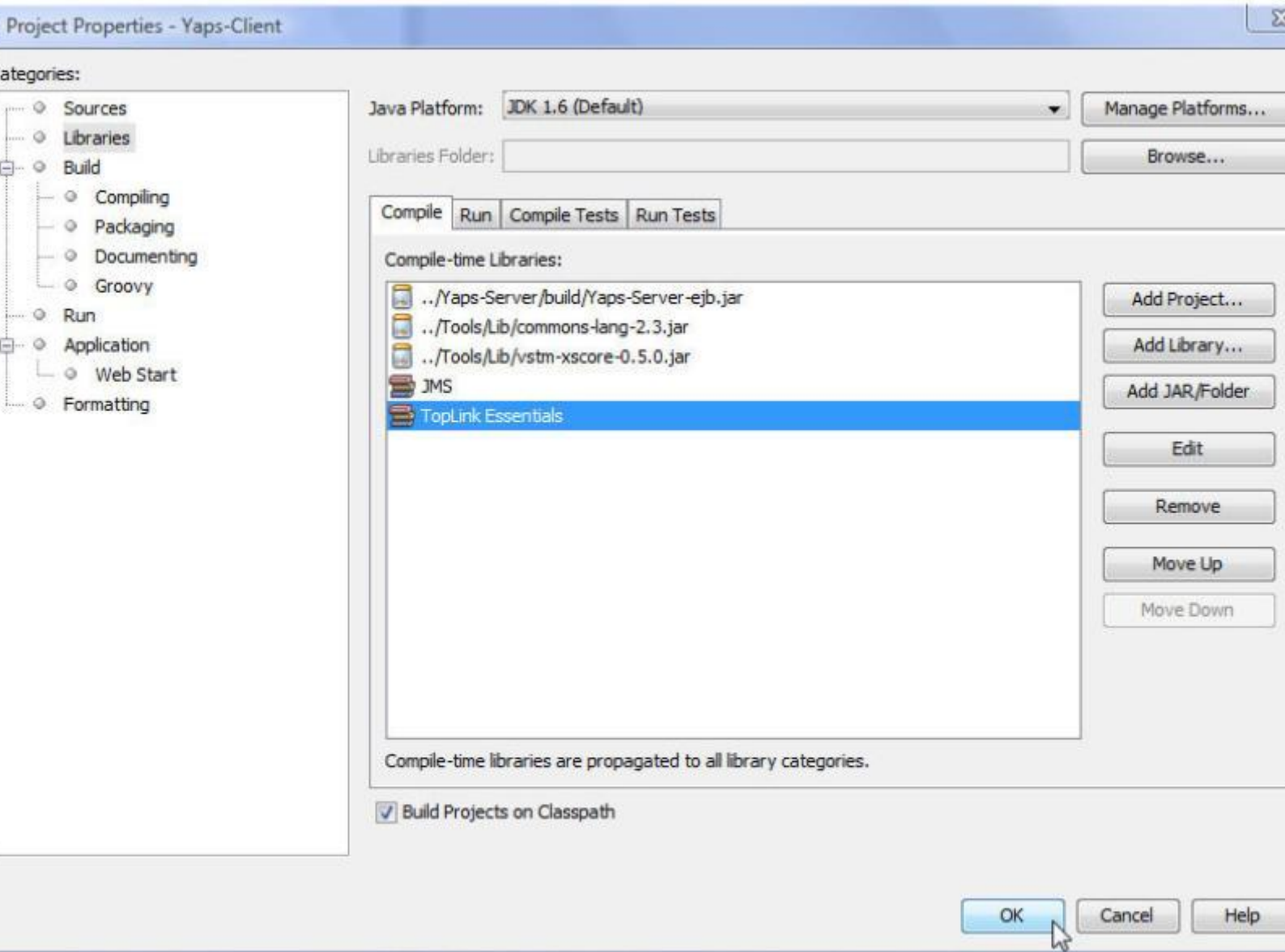


Figure 10-21: Référence à TopLink Essentials

LLX-B-4 - Identifier la classe principale

Avant de tester l'application, il vous reste à localiser la classe principale « Main Class » qui sera invoquée à l'exécution.

Ouvrez les propriétés du projet et rendez-vous sur la catégorie « Run » :

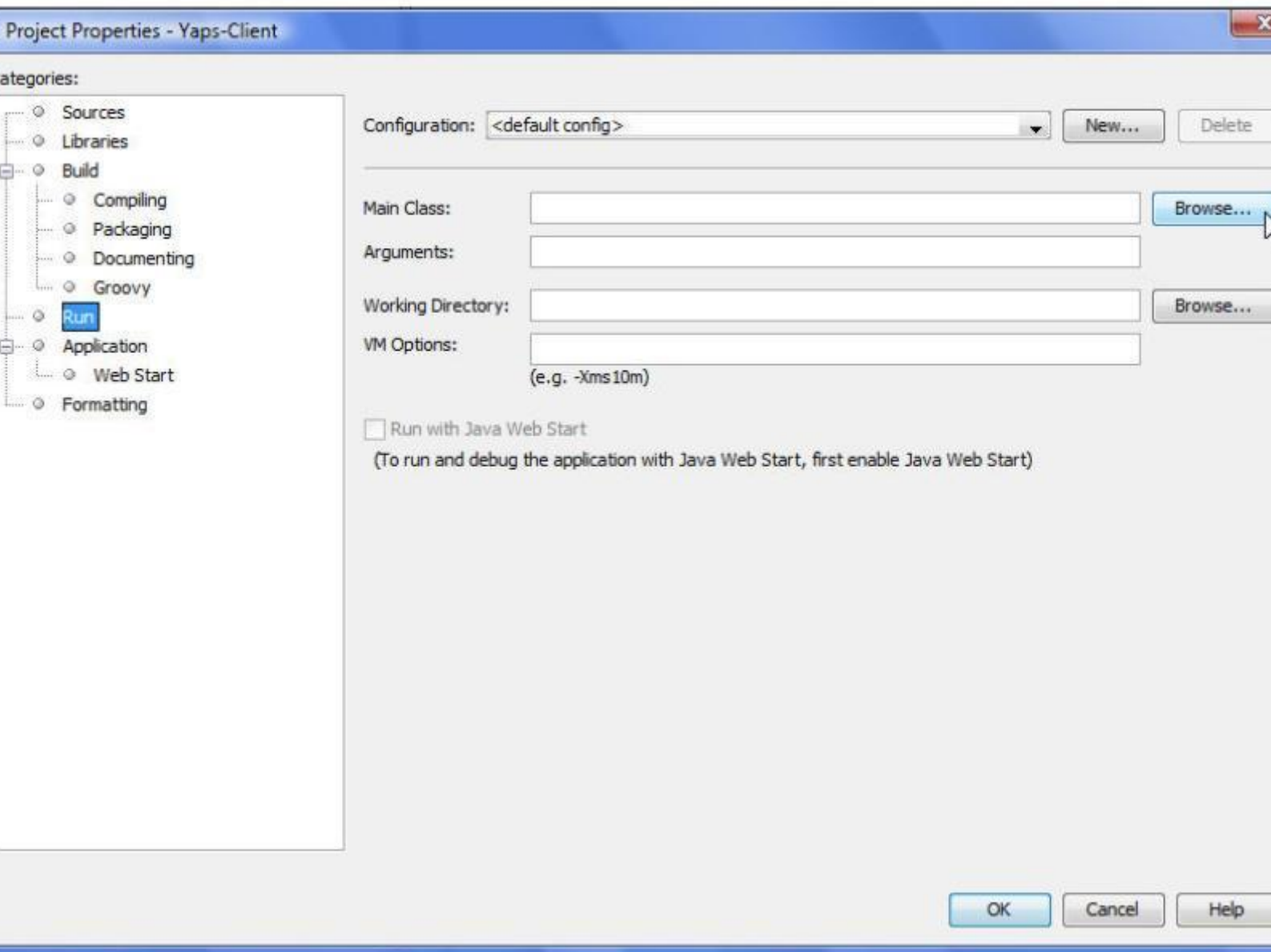


Figure 10-22: Propriétés du projet (Run)

Le champ « Main Class » permet d'identifier la classe principale.

Cliquez sur le bouton « Browse » pour sélectionner la classe principale « com.yaps.petstore.client.ui.PetstoreFrame » :

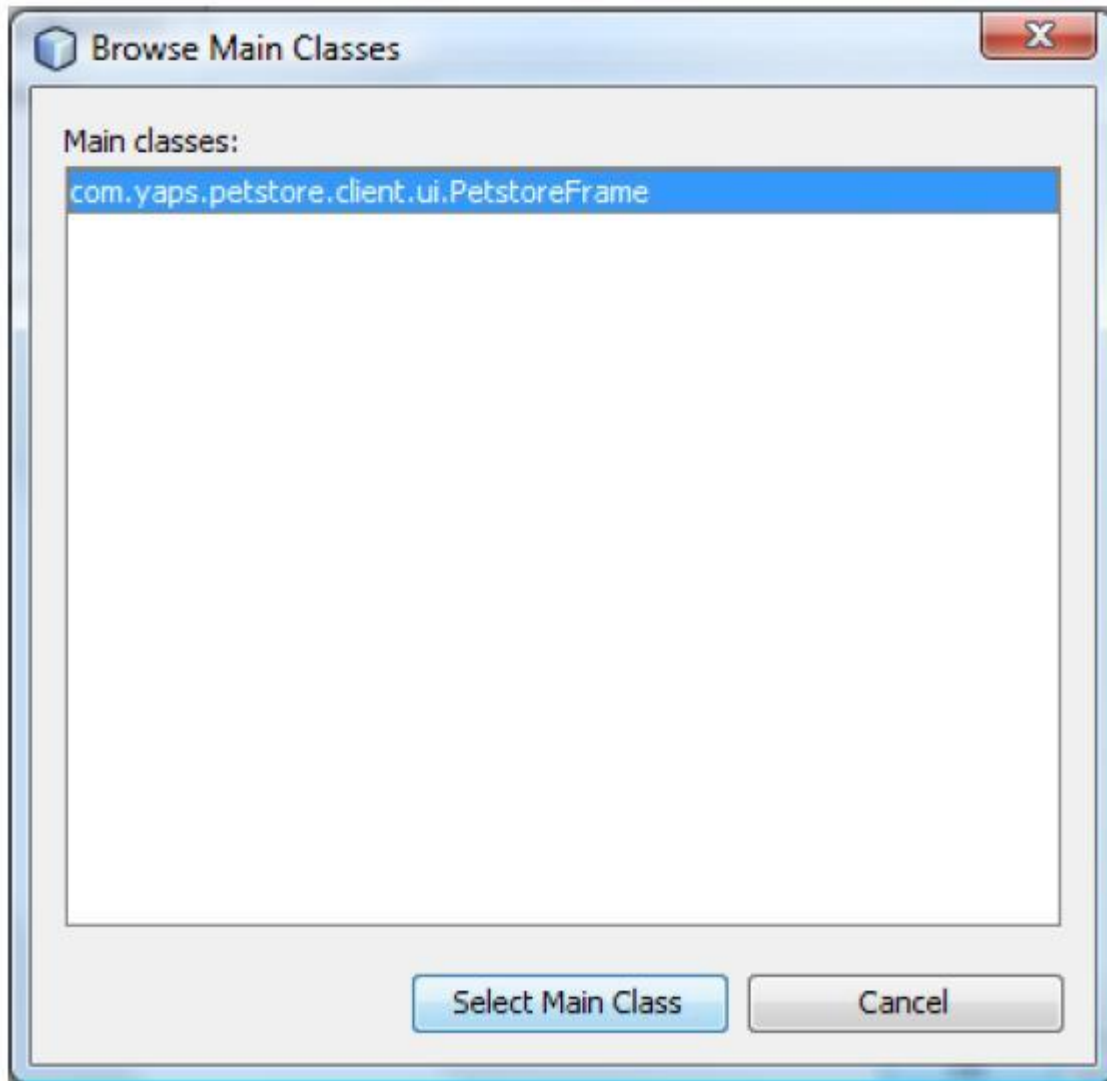


Figure 10-23: Sélection de la classe principale

Cliquez sur le bouton « Select Main Class ».

Cliquez ensuite sur OK afin de sauver vos modifications dans les propriétés du projet :

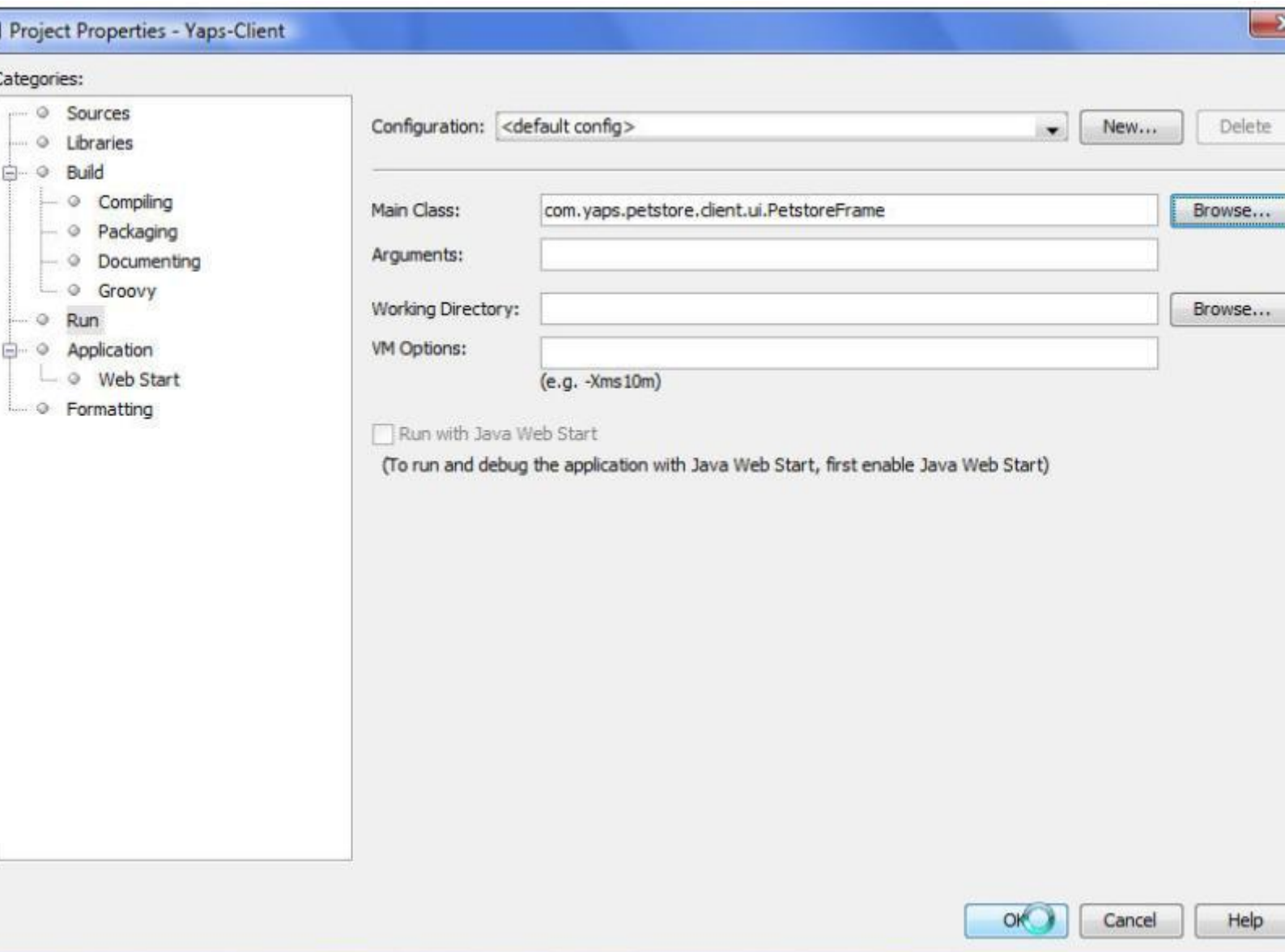


Figure 10-24: Classe principale identifiée

LLX-B-5 - Créer le groupe de projet

À partir du menu « File » ou via un clic droit dans la fenêtre « Projects », NetBeans offre la capacité de créer le nouveau groupe de projet :

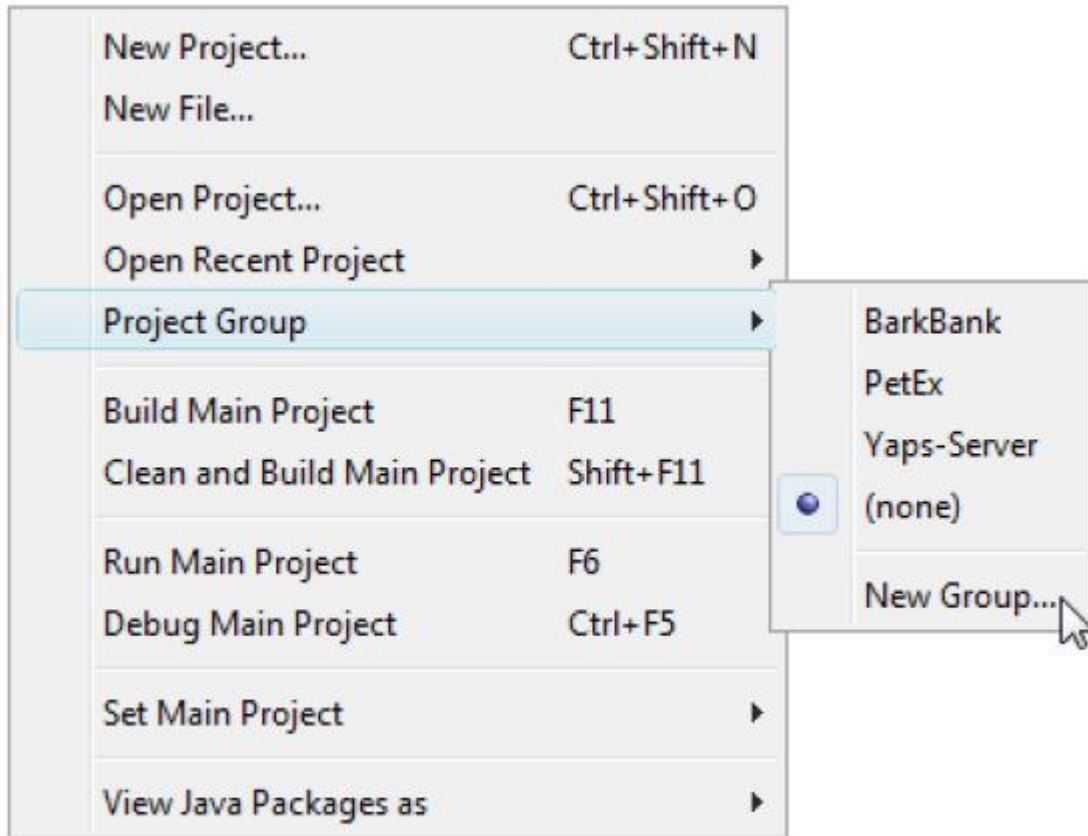


Figure 10-25: Créer un groupe de projet

Sélectionnez l'option « New group ».

La fenêtre suivante permet de nommer le groupe de projet qui est dans notre cas « **Yaps - Client** » :

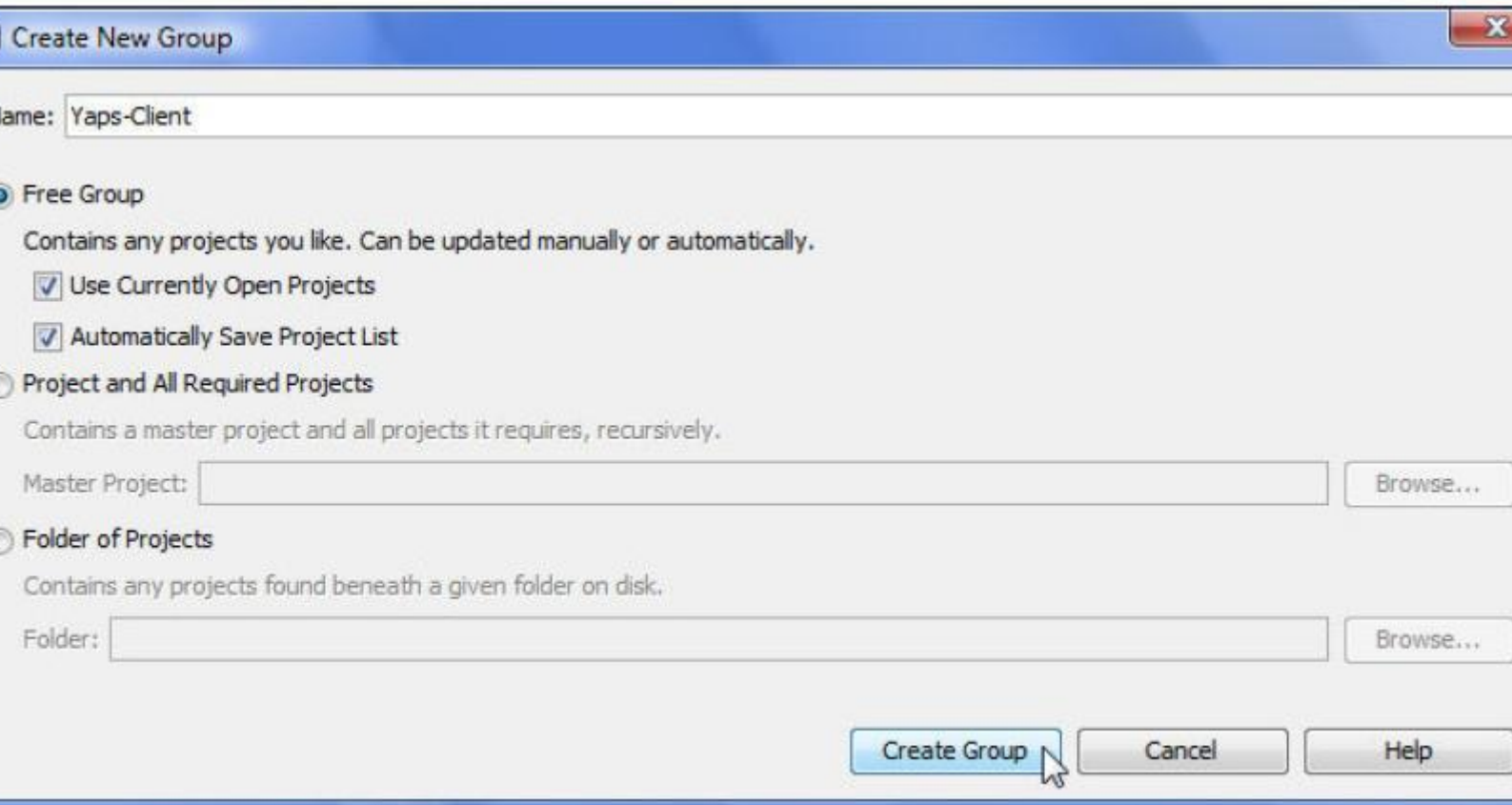


Figure 10-26: Nouveau groupe de projet

Les options par défaut indiquent que les projets en cours de chargement dans l'IDE seront automatiquement ajoutés au groupe de projet.

Cliquez sur le bouton « **Create Group** ».

LLX-C - Compiler le projet

Un clic droit sur le projet fait apparaître le menu contextuel proposant les options pour compiler le projet (« Build » ou « Clean and Build ») :

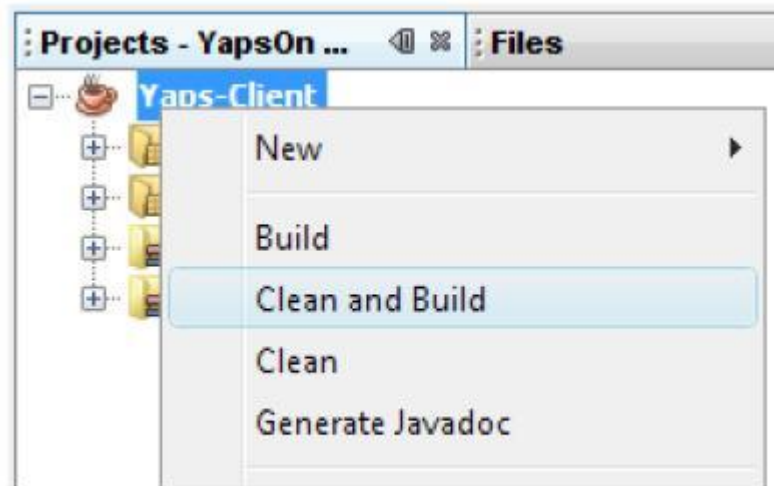


Figure 10-27: Compiler le projet

Cliquez sur l'option « Clean and Build ».

La fenêtre « Output » affiche les traces de la compilation :

```

Output - Yaps-Client (clean.jar)
JUnit Test Results
C:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client\util\jms\Jm
* Lib?re toutes les ressources JMS utilis?es.
C:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client\util\jms\Jm
* Lib?re toutes les ressources JMS utilis?es.
C:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client\util\jms\Jm
* D?marre la r?ception des messages.
C:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client\util\jms\Jm
* D?marre la r?ception des messages.
C:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client\util\jms\Jm
* Arr?te la r?ception des messages.
C:\Projects\YapsON\Yaps-Client\src\com\yaps\petstore\client\util\jms\Jm
* Arr?te la r?ception des messages.
26 warnings
Copying 1 file to C:\Projects\YapsON\Yaps-Client\build\classes
compile:
Created dir: C:\Projects\YapsON\Yaps-Client\dist
Building jar: C:\Projects\YapsON\Yaps-Client\dist\Yaps-Client.jar
Copy libraries to C:\Projects\YapsON\Yaps-Client\dist\lib.
To run this application from the command line without Ant, try:
java -jar "C:\Projects\YapsON\Yaps-Client\dist\Yaps-Client.jar"
jar:
BUILD SUCCESSFUL (total time: 6 seconds)

```

Figure 10-28: Résultat de la compilation

En analysant en détail les traces de la compilation, on relève les messages suivants :

```

...
Building jar: C:\Projects\YapsON\Yaps-Client\dist\Yaps-Client.jar
...
To run this application from the command line without Ant, try:
java -jar "C:\Projects\YapsON\Yaps-Client\dist\Yaps-Client.jar"
...
BUILD SUCCESSFUL (total time: 6 seconds)

```

Ces traces indiquent que l'application Yaps-Client a été correctement compilée et qu'il est possible de l'exécuter à partir de la ligne de commande en invoquant la commande suivante :

```

java -jar {yapson_home}\Yaps-Client\dist\Yaps-Client.jar

```

LLX-D - Tester le projet

Nous allons décrire les spécificités de NetBeans en matière d'exécutions d'applications Java.

LLX-D-1 - Démarrer le serveur PetStore

Avant de démarrer l'application Yaps-Client, il est nécessaire de démarrer le serveur « PetStore ».

Rendez-vous sur l'onglet « Services » et démarrez le serveur PetStore :

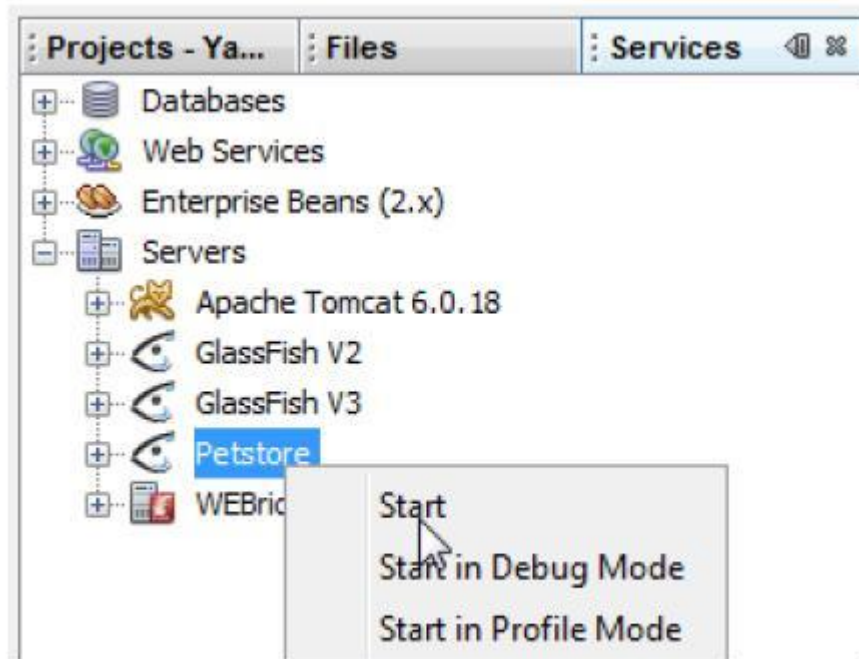
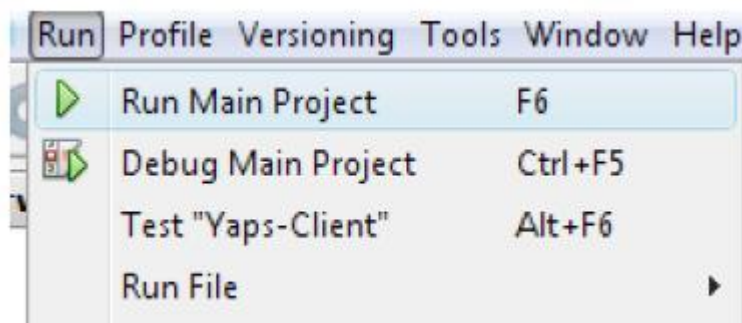


Figure 10-29: Démarrez le serveur PetStore

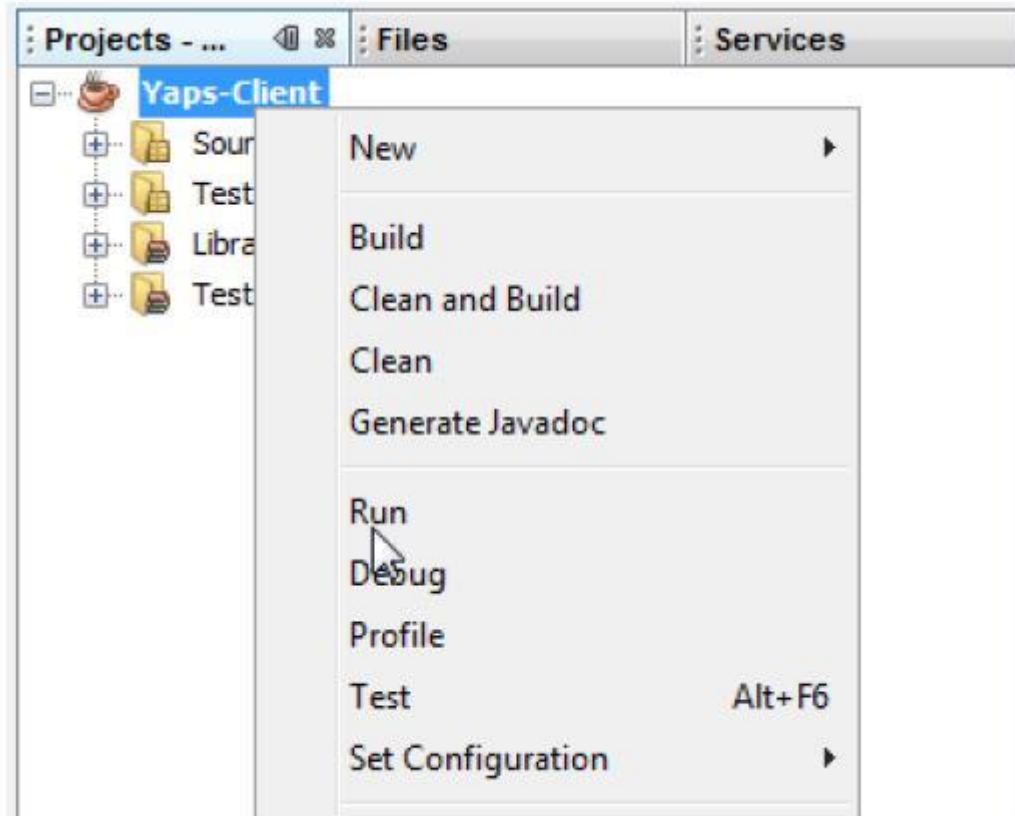
LLX-D-2 - Exécuter l'application Yaps-Client

Sous NetBeans, l'application Yaps-Client peut-être démarrée de différentes façons :

- En appuyant sur la touche « F6 »
- À partir du menu « Run -> Run Main project » :



- À partir de l'option « Run » du menu contextuel du projet :



- À partir de la barre d'outils en cliquant sur l'icône suivant :



L'application apparaît à l'écran :

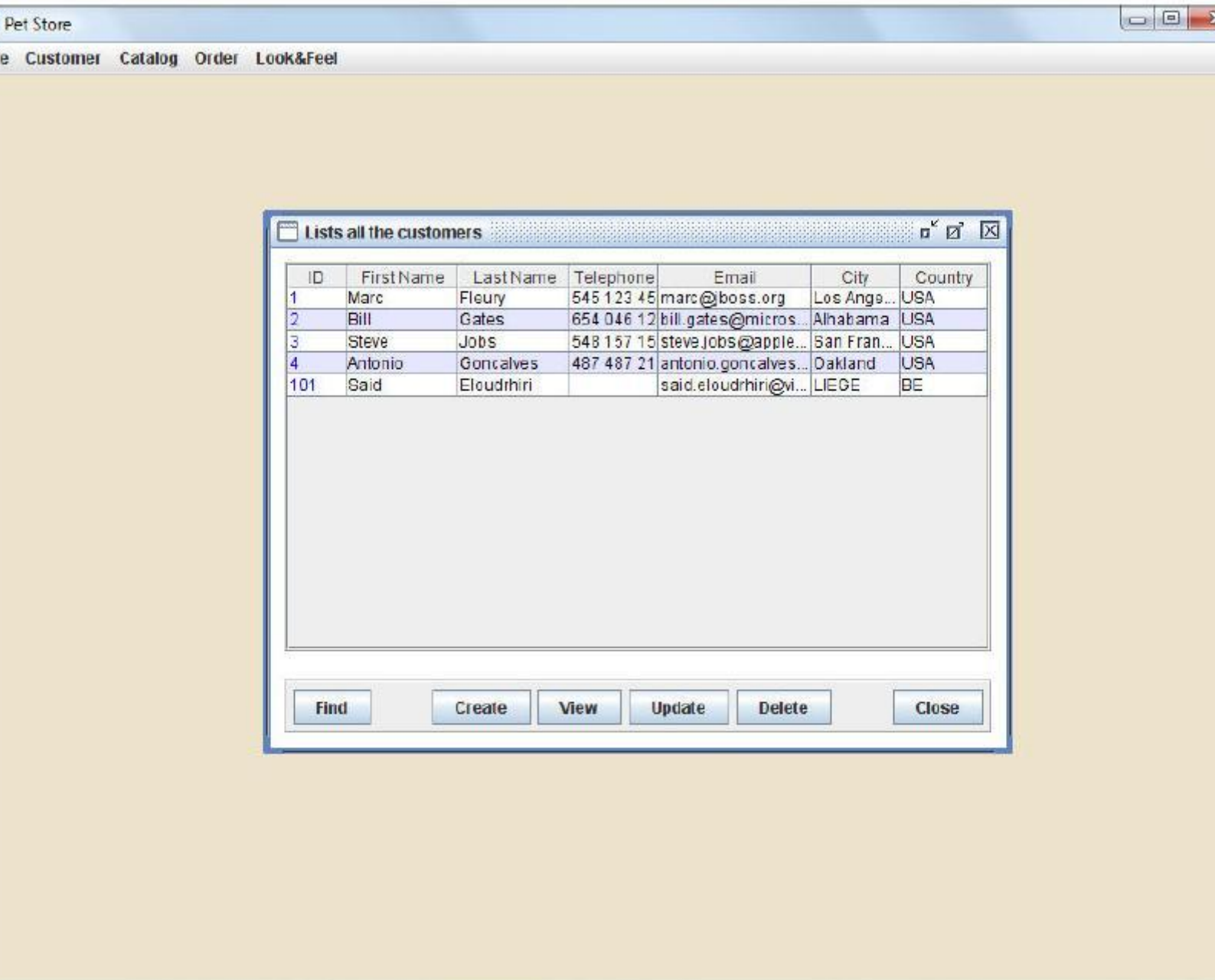


Figure 10-30: Application Yaps-Client

Reportez-vous au livre de référence qui décrit les fonctionnalités de l'application client.

S'abonner au topic JMS

Pour recevoir les messages postés dans la file d'attente JMS (les articles de type « reptiles »), n'oubliez pas de vous abonner au topic JMS en ouvrant la fenêtre « Watch orders » disponible sous le menu « Order ».

L'application cliente ne gère pas la notion de durabilité des messages JMS.

La barre de statut de NetBeans (en bas et à droite) affiche le statut de l'application en cours d'exécution :



Pour stopper l'exécution de l'application, un clic droit sur la barre de statut propose un menu contextuel permettant de stopper l'application :

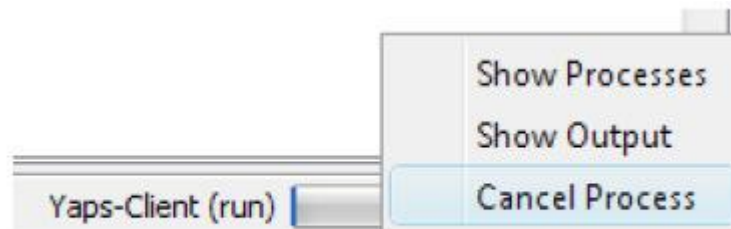


Figure 10-31: Annuler l'exécution

LLX-E - Conclusion

Les sources du client de l'application « Yaps» ont été transférées sous NetBeans dans une arborescence conforme au format préconisé par le Java Blueprints.

Les tests que nous avons réalisés attestent que l'application cliente Swing communique correctement avec les composants déployés sur le serveur.

XI - Conclusion

Nous voilà arrivés au terme de ce document.

Vous êtes maintenant en mesure d'utiliser l'application Yaps Pet Store sous NetBeans et d'approfondir vos connaissances de JEE5 en bénéficiant de la richesse et de la souplesse de NetBeans 6.5.

Couplé au livre de référence, vous êtes paré à vous lancer dans l'élaboration d'application JEE5 ou de réaliser des POC (Proof Of Concept) nettement plus riches qu'un simple « Hello World ». L'application Yaps Pet Store peut servir d'excellent point de départ pour vos expérimentations.

Bien entendu, nous n'avons qu'effleuré les différentes facettes de NetBeans. Le site www.netbeans.org vous sera d'un grand secours tant les informations qui s'y trouvent sont nombreuses.

Et maintenant, à vous de jouer !

XII - Références

| | |
|------|--|
| [R1] | Le livre « Les Cahiers du Programmeur Java EE5 - Seconde édition » de Antonio Goncalves parut aux éditions Eyrolles 2008. Site du livre : http://www.antoniogoncalves.org/xwiki/bin/view/Book/WebHome |
| [R2] | L'application de référence Java Pet Store 2.0 du programme Java BluePrints de Sun Microsystems : http://java.sun.com/developer/releases/petstore/ |
| [R3] | Site de NetBeans : http://www.netbeans.org |
| [R4] | Description et historique de NetBeans sous Wikipedia : http://fr.wikipedia.org/wiki/NetBeans (Français) http://en.wikipedia.org/wiki/Netbeans (Anglais) |
| [R5] | Cet article décrit les motivations de portage d'une application Web sous NetBeans ainsi que la différence entre la structure de fichier de type Jakarta et Java BluePrints préconisée par SUN : http://www.netbeans.org/kb/articles/porting-webapps.html |

XIII - Glossaire

| | |
|----------------|------------------------------------|
| DB | DataBase |
| EAR | Enterprise Archive |
| EJB | Enterprise JavaBean |
| ESB | Enterprise Service Bus |
| IDE | Integrated Development Environment |
| JAR | Java ARchive |
| Java SE | Java Standard Edition |
| JAX-WS | Java API for XML Web Services |
| JDBC | Java DataBase Connectivity |
| JDK | Java Development Kit |
| JEE | Java Enterprise Edition |
| JEE5 | Java Enterprise Edition version 5 |
| JMS | Java Message Service |
| JPA | Java Persistence API |
| JSF | JavaServer Faces |
| JVM | Java Virtual Machine |
| POC | Proof of Concept |
| SOA | Service Oriented Architecture |
| SQL | Structured Query Language |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| WAR | Web Application aRchive |
| WSDL | Web Service Description Language |
| YapsON | Yaps Pet Store On NetBeans |