 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

Document présentation


Description	Documentation technique de création de web service java avec apache Axis2 et l'IDE Eclipse.
--------------------	---

Document certification

	<i>Name</i>	<i>Fonction</i>
<i>Author</i>	DJEBIEN Tarik RAKOTOBE Eric	Etudiant Miage IPI NT
<i>Decidor</i>	ROOS Jean-François	Enseignant CARA

Document Version history

<i>Version</i>	<i>Date</i>	<i>Change Summary</i>	<i>Change Author</i>
1.0	16 / 02 / 2013	Création du tutoriel	Djebien Tarik Rakotobe Eric

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

[Creation du webService en java en utilisant Apache Axis2 et Eclipse](#)


[1. Installation de l'environnement de développement](#)

[2. Création du projet pour le WebService](#)

[3. Création de la classe de Web Service](#)

[4. Créer un fichier .aar \(Archive Axis\) et déployer le service](#)

[5. Création du WebService client](#)


 Université Lille1 <small>Sciences et Technologies</small>	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

Creation du webService en java en utilisant Apache Axis2 et Eclipse

Les webServices sont des composants d'application qui communiquent en utilisant des protocoles ouverts.

L'utilisation des webServices permettent de publier vos propres fonctions.

Ce document a pour but de vous indiquer pas à pas les démarches que nous avons suivi pour la réalisation du tp1 de CARA, en utilisant Axis2 Web Services / SOAP / WSDL et l'IDE Eclipse.

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

1. Installation de l'environnement de développement


1.1. Avant de commencer, vous devez installer votre environnement de développement.

Les étapes ci-dessous sont nécessaires pour créer vos WebServices avec Axis2 et l'IDE Eclipse.

Ce document a été rédigé avec les pré-requis suivants :

- Apache Axis2 1.6.2
- Eclipse Juno
- Apache Tomcat 6

- 1) Apache Axis2 Binary Distribution - [Download](#)
- 2) Apache Axis2 WAR Distribution - [Download](#)
- 3) Apache Tomcat - [Download](#)
- 4) Eclipse IDE - [Download](#)
- 5) Java installé sur votre machine - [Download](#)


 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

1.2. Ensuite, vous devez configurer vos variables d'environnement pour Java et Tomcat. Ci-dessous les variables d'environnement à ajouter :

JAVA_HOME :- Mettre la valeur au répertoire sur lequel se trouve le JDK value
(exemple : D:\Developpement\Java\jdk1.6.0_39)

TOMCAT_HOME :- Mettre la valeur au répertoire où se trouve votre installation de Tomcat
(exemple D:\Developpement\apache-tomcat-6.0.36-windows-x64)

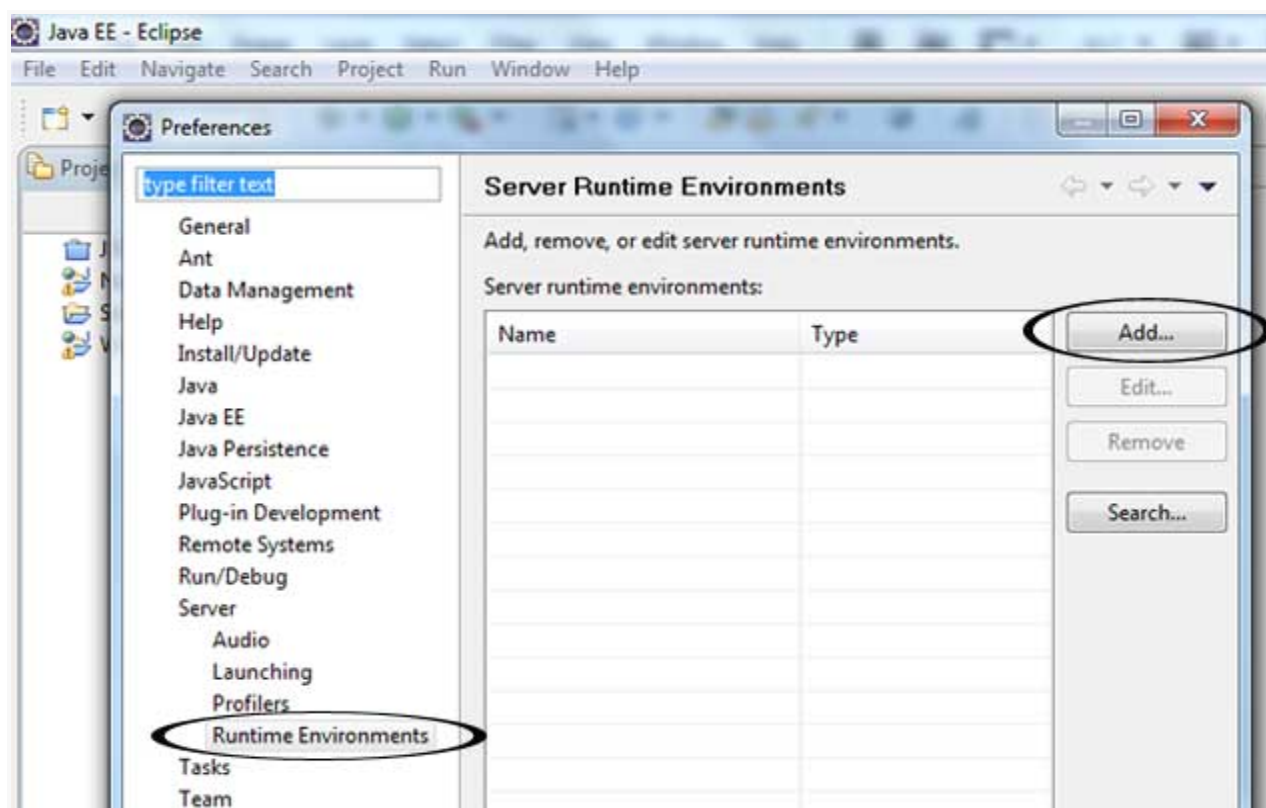
PATH :- Mettre la valeur au répertoire où se trouve le dossier 'bin' de votre JDK
(exemple D:\Developpement\Java\jdk1.6.0_39\bin)


 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

1.3. Maintenant, il faut ajouter le 'runtime environnement' de eclipse.

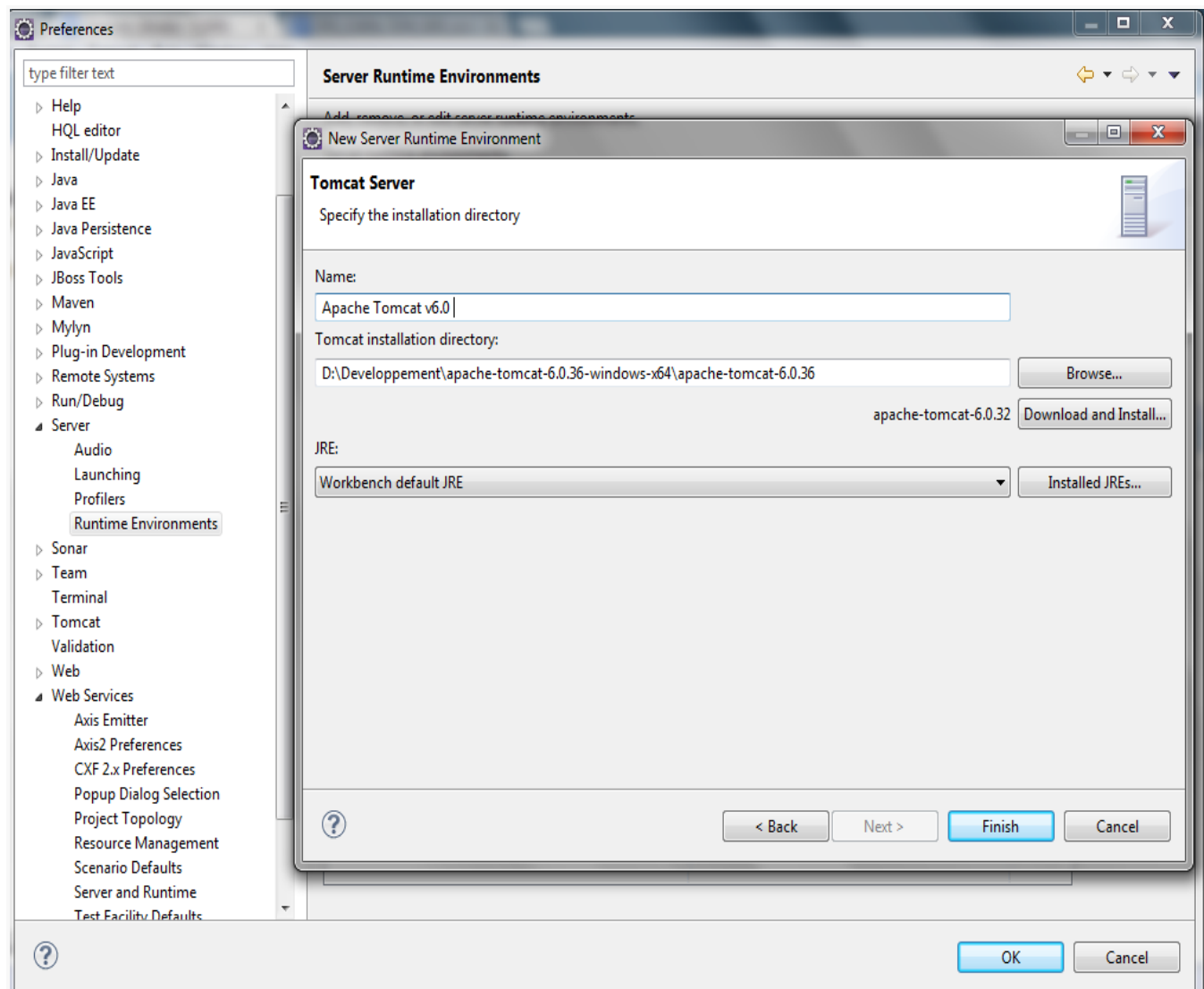
Pour cela, aller dans


Windows --> Preferences --> Server --> Runtime Environments



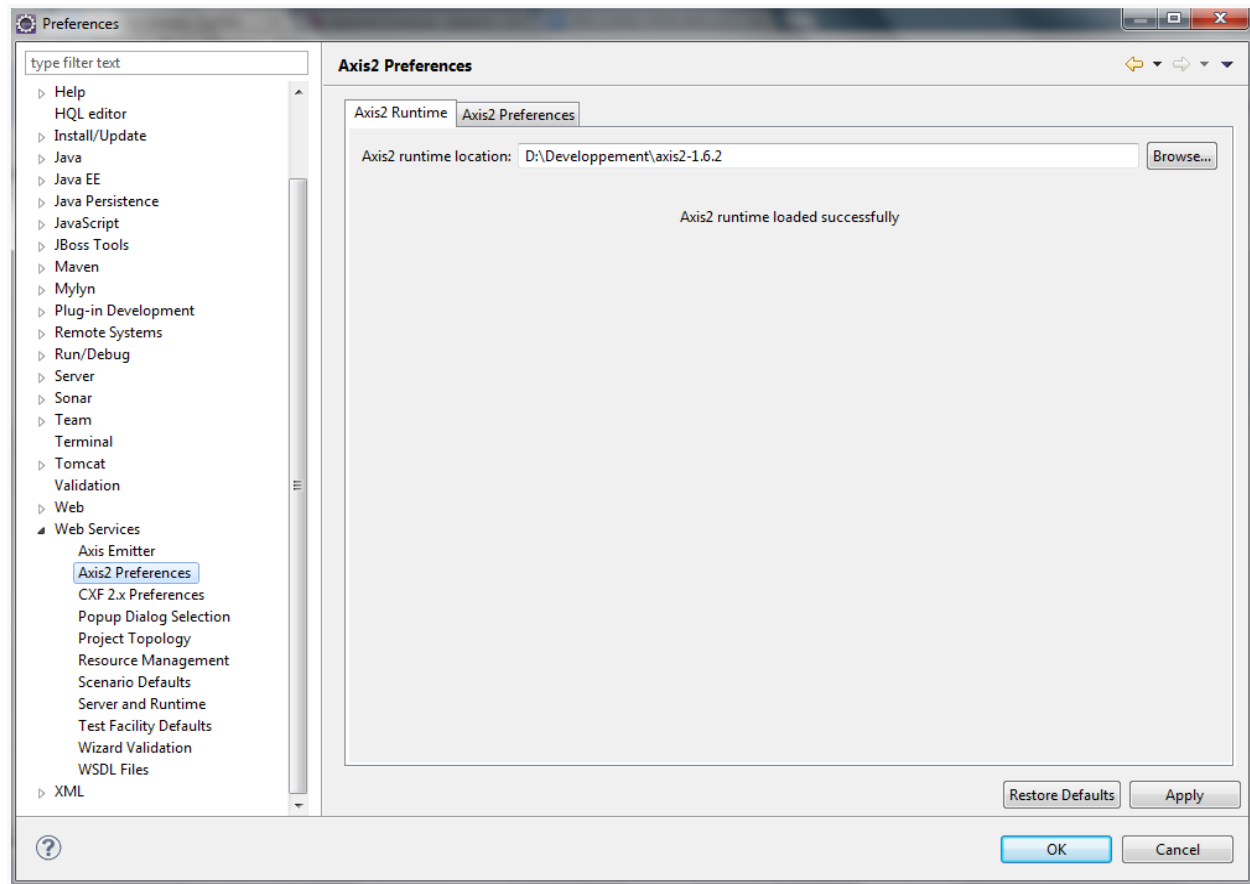
 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0


Ensuite, sélectionner Apache Tomcat v6.0 et dans la prochaine fenêtre, parcourir le dossier d'installation de votre Apache et cliquer sur 'Finish'



 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

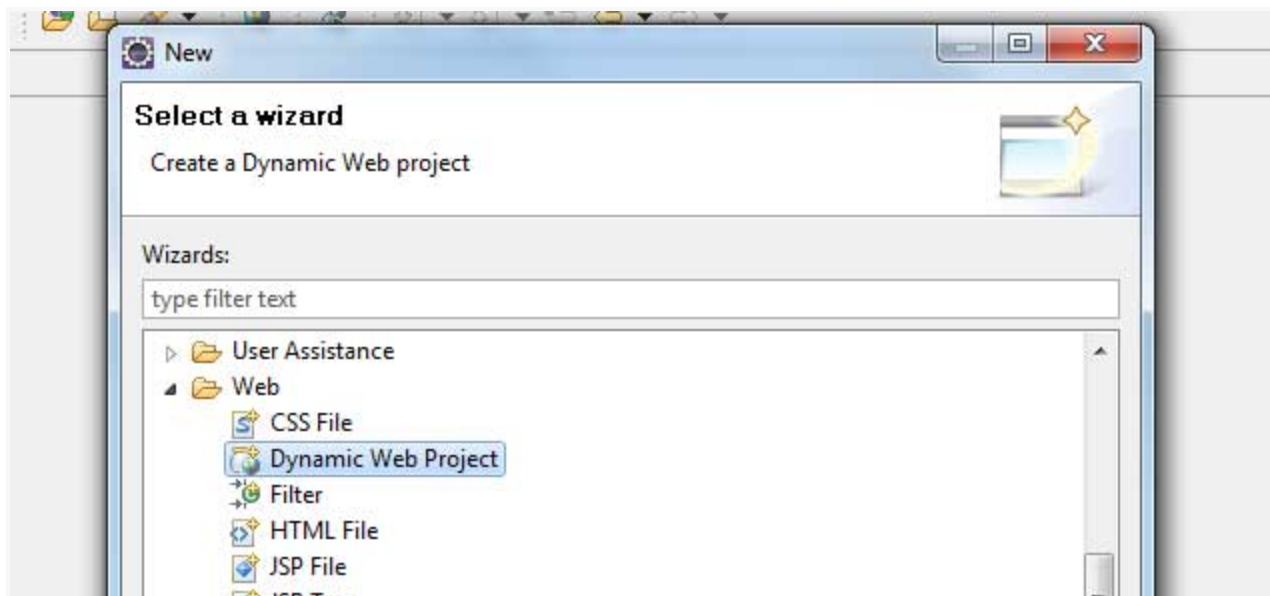
1.4. Cliquer ensuite sur Web Service --> Axis2 Preferences et parcourir vers le dossier d'installation de Apache Axis2




 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

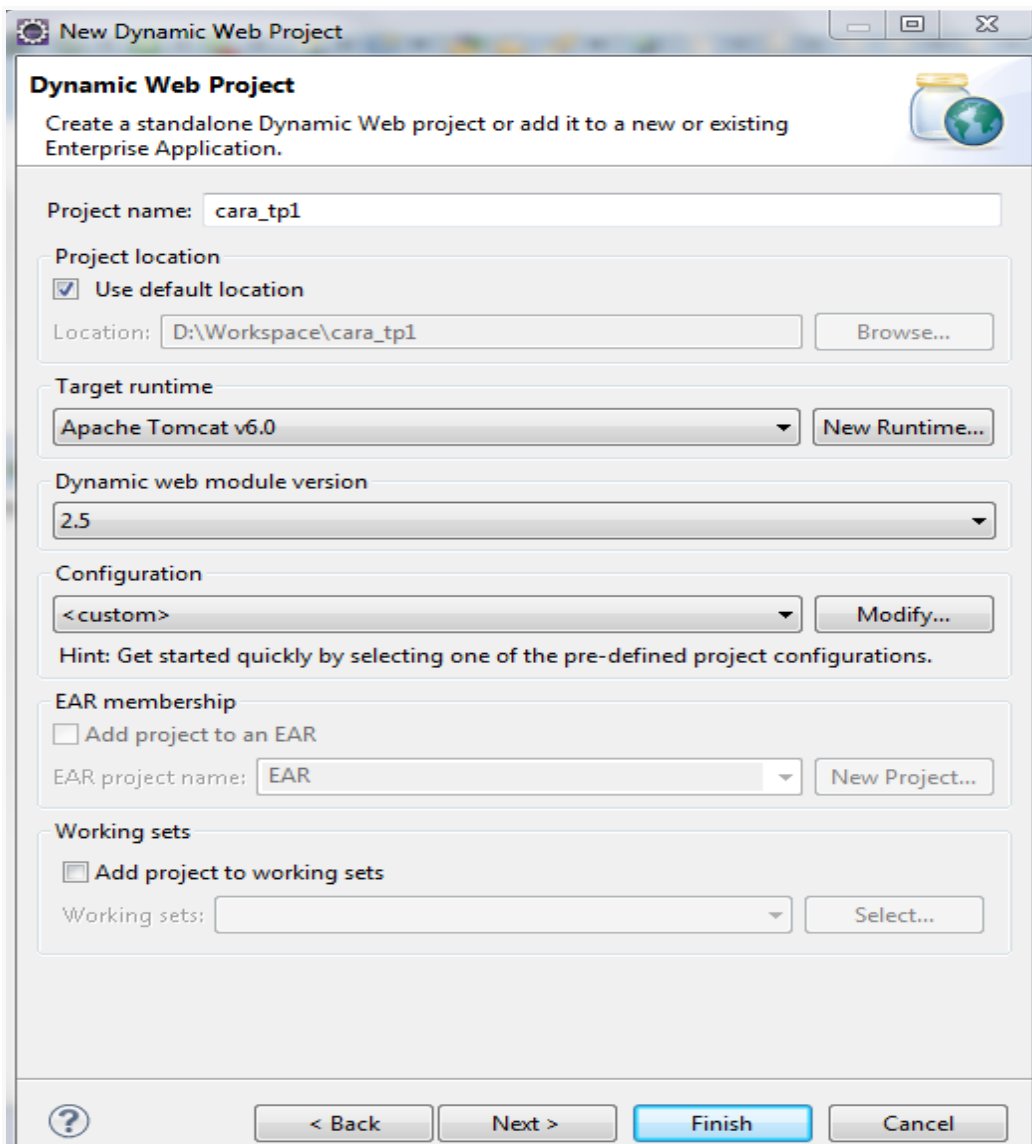
2. Création du projet pour le WebService

2.1 D'abord, créer un nouveau 'Dynamic Web Project (File --> New --> Other...) ' en choisissant Web --> Dynamic Web Project.



 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

2.2 Choisir Apache Tomcat6 comme 'Target Runtime' avec 'Dynamic web module version 2.5 ', et cliquer sur 'Modify' pour installer 'Axis2 Web Services project facet'



New Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR membership


☐ Add project to an EAR

EAR project name:

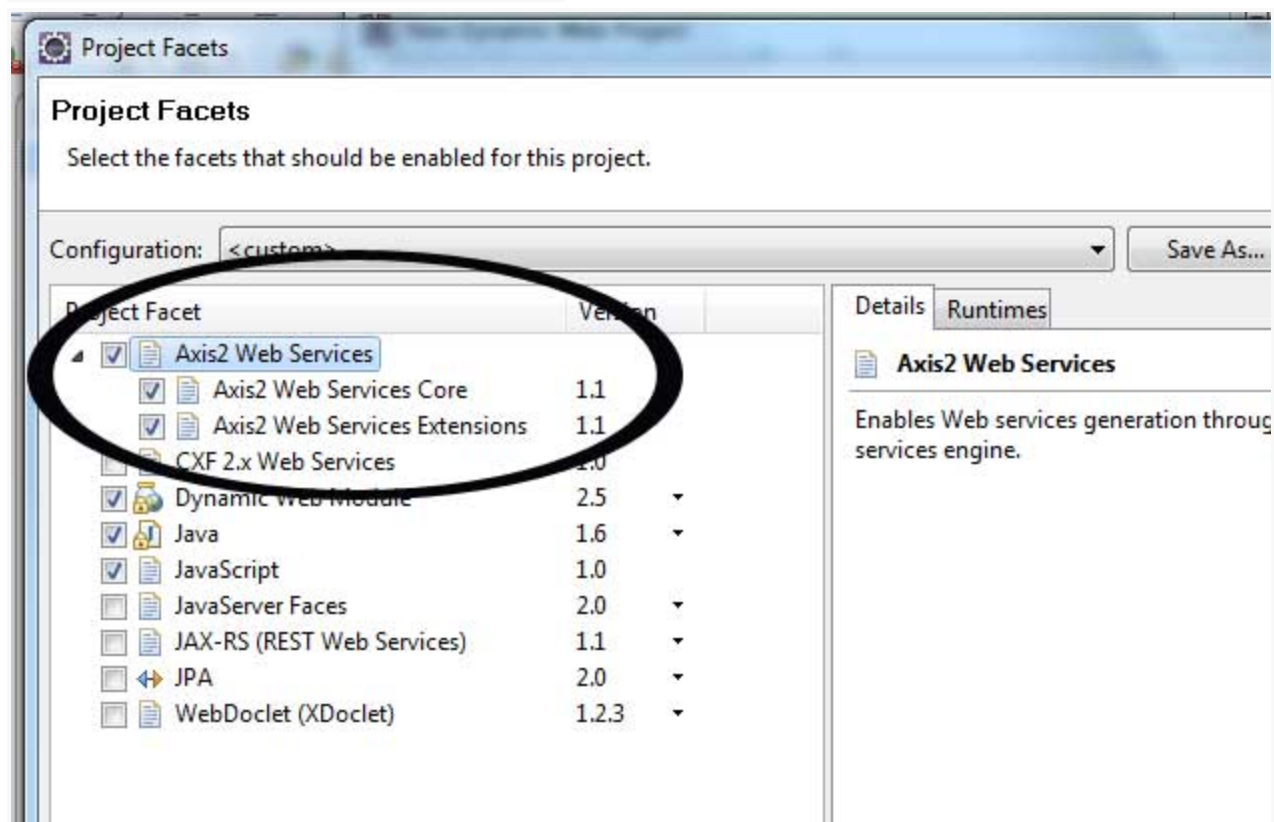
Working sets

☐ Add project to working sets

Working sets:


 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

2.3 Sélectionner 'Axis2 Web Services'



2.4 Cliquer sur OK et ensuite Next.

Maintenant, vous pouvez choisir vos dossiers et cliquer sur Finish.

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

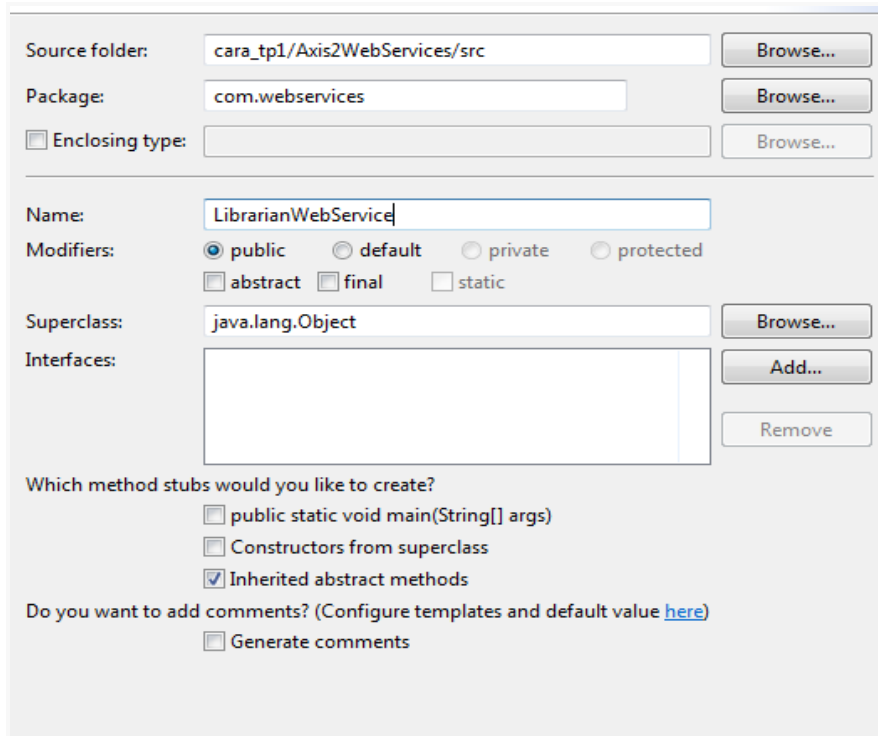
3. Création de la classe de Web Service

Maintenant, vous aller créer la classe Java que vous voudrez exposer en tant que WebService.

Pour cela, créer une classe

3.1 Cliquer droit sur cara_tp1, choisir New --> Class

Créer la classe 'LibrarianWebService' dans le package 'com.webservices'



Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?


☐ public static void main(String[] args)

☐ Constructors from superclass

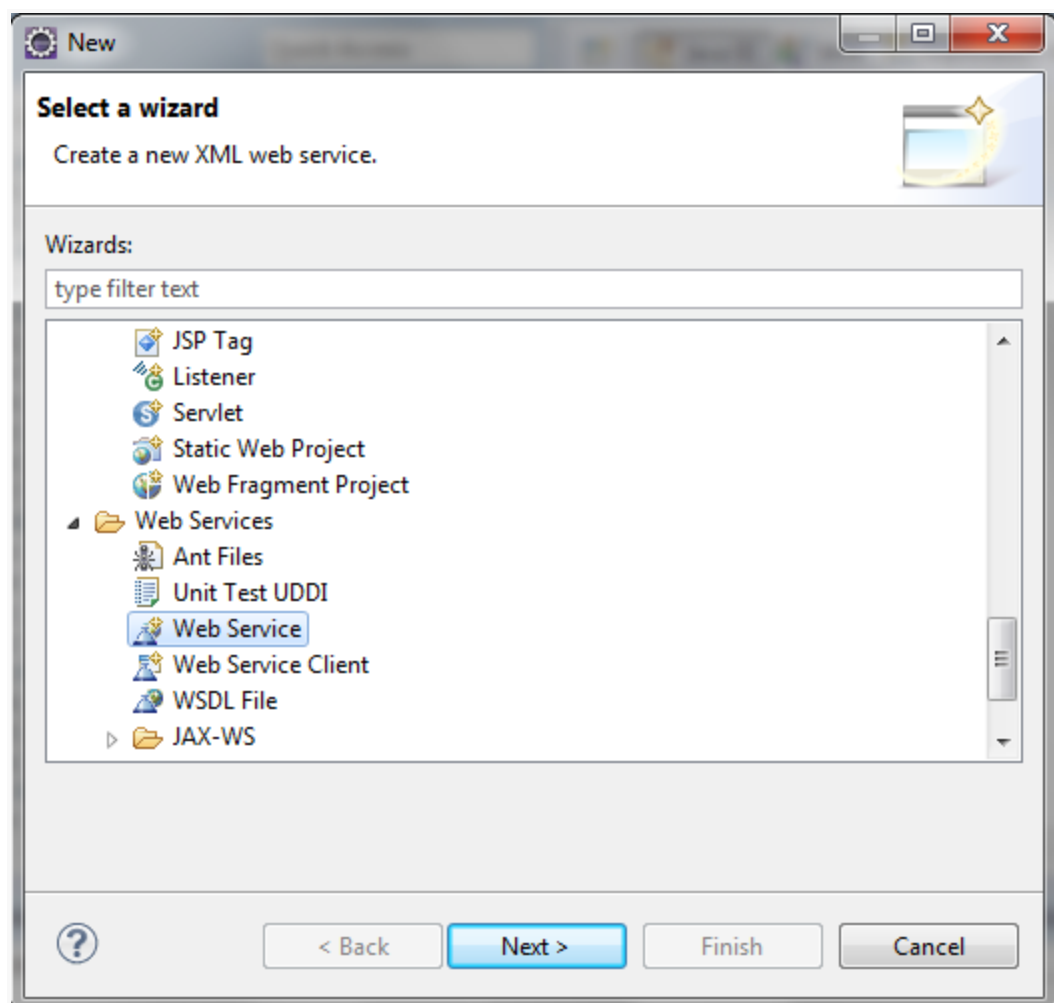
☒ Inherited abstract methods


Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments


 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

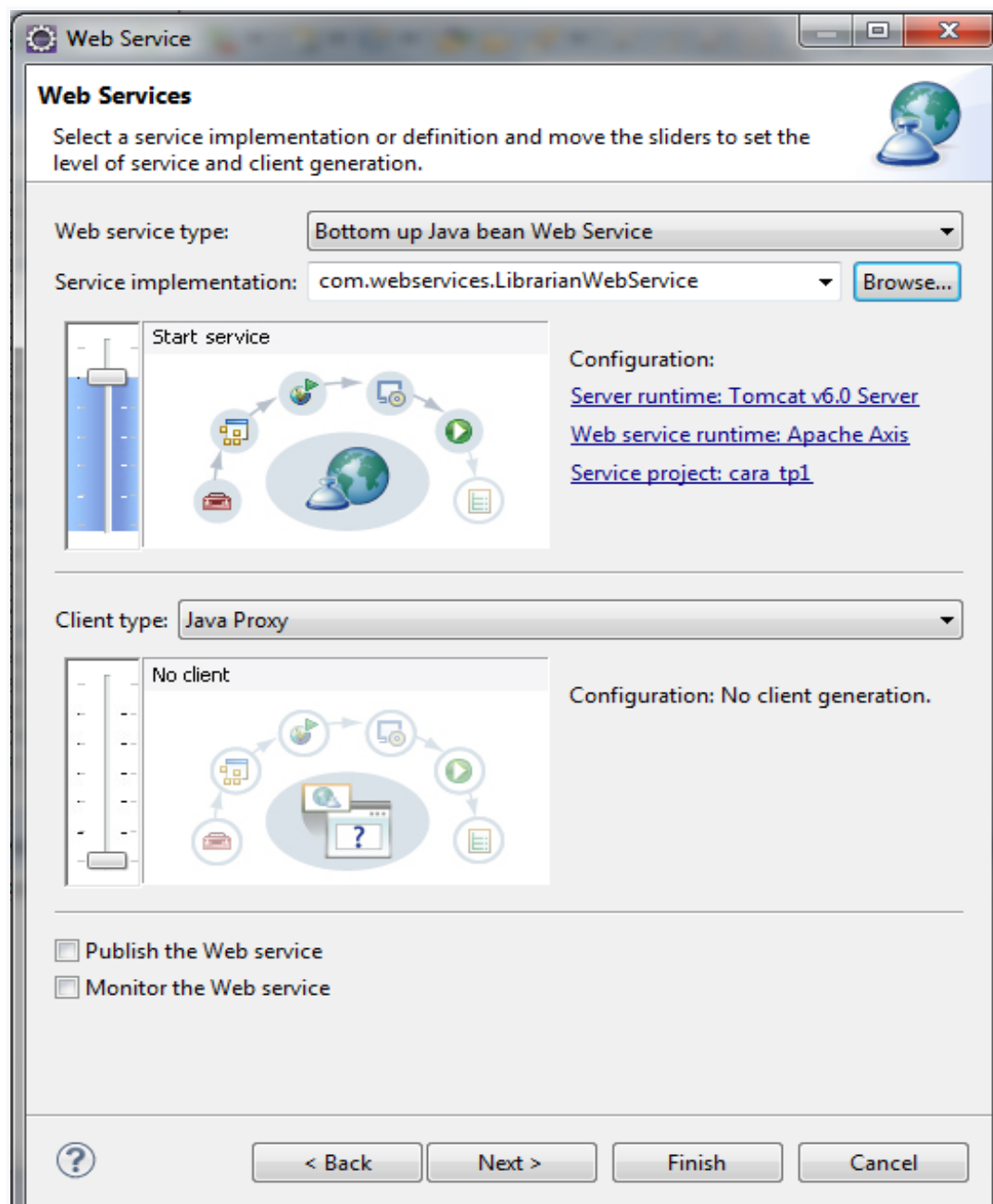
3.2 Ensuite, sélectionner select File --> New --> Other et choisir Web Service




 Université Lille1 <small>Sciences et Technologies</small>	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

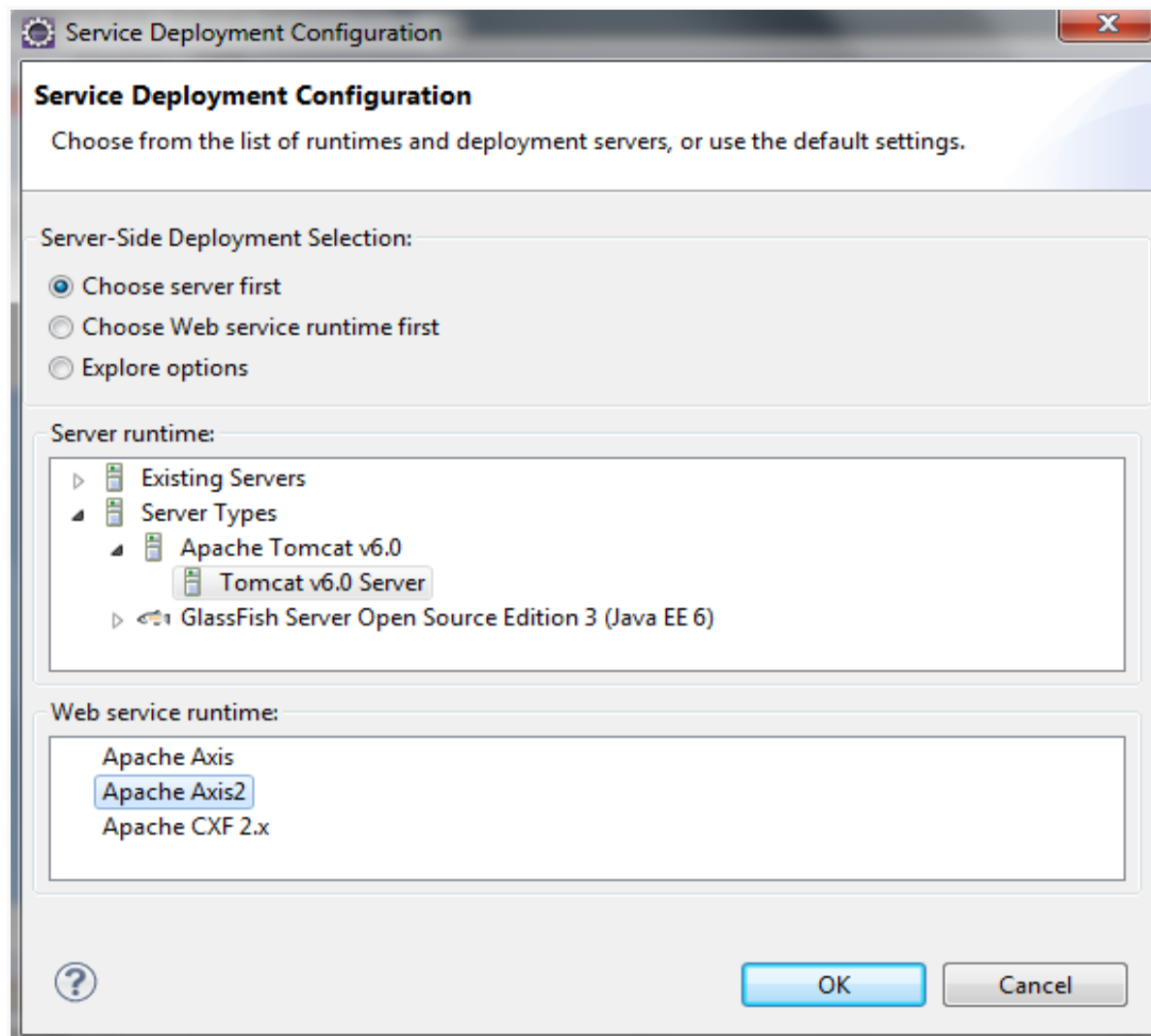
3.3 Sélectionner la classe de WebService 'LibrarianWebService' comme service implementation et pour s'assurer que la configuration s'est bien faite, cliquer sur Server runtime.


 Université Lille 1	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0



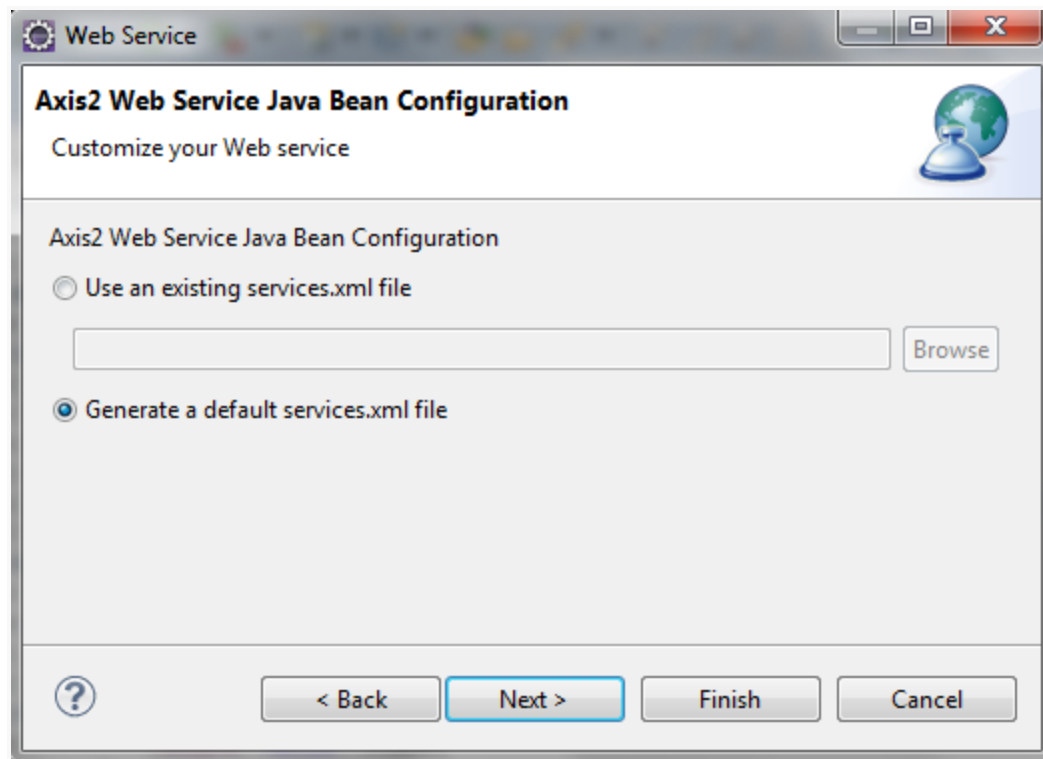
 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0


3.4 Vérifier que le Web Service runtime a été positionné a Axis2 (par défaut, Axis1 est sélectionné) et cliquer sur Ok.



 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

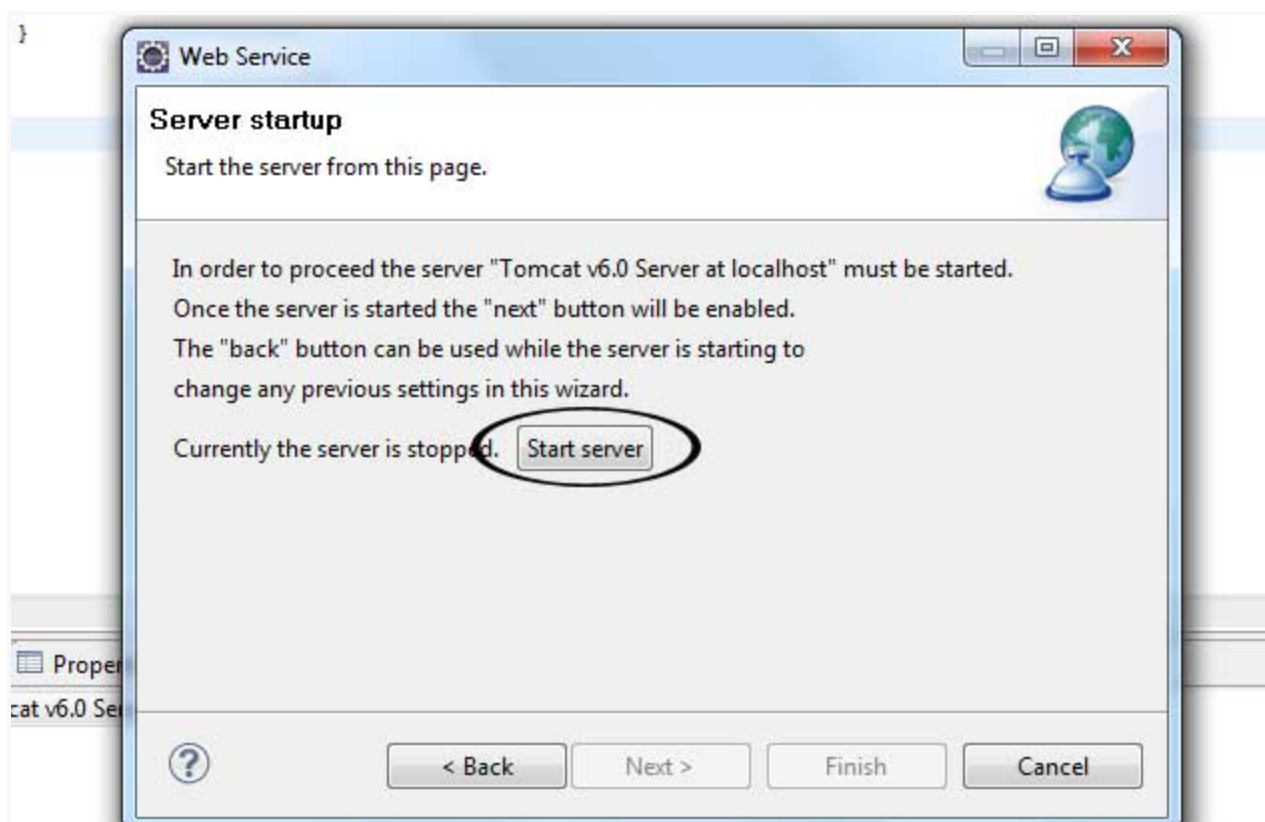
3.5 Cliquer sur Next, et vérifier que 'Generate a default service.xml file' est bien cochée.




 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

3.6 Cliquer sur Next, et 'Start Server'.

Après le démarrage du serveur, vous pouvez cliquer sur 'Finish' au lieu de Next si vous ne voulez pas que votre WebService soit publié dans un annuaire de webServices UDDI.



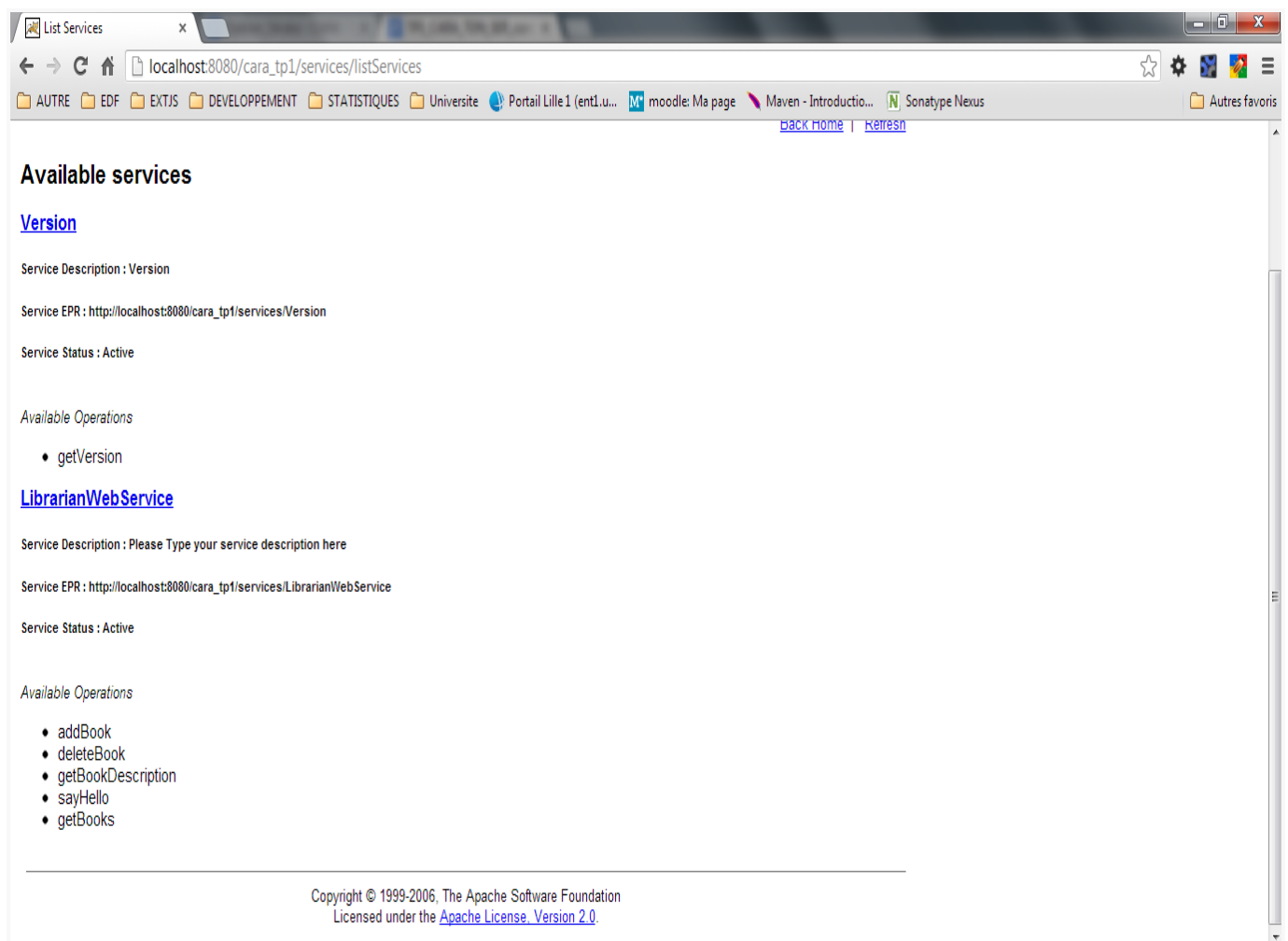
 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

Maintenant, vous pouvez aller vers le lien

http://localhost:8080/cara_tp1/services/listServices

pour voir le service déployé par Axis2.

Vous pouvez voir le WSDL obtenu par approche 'bottom up Java bean webService' en cliquant sur le lien LibrarianWebService.



Available services

[Version](#)

Service Description : Version

Service EPR : http://localhost:8080/cara_tp1/services/Version

Service Status : Active

Available Operations

- getVersion

[LibrarianWebService](#)

Service Description : Please Type your service description here


Service EPR : http://localhost:8080/cara_tp1/services/LibrarianWebService

Service Status : Active

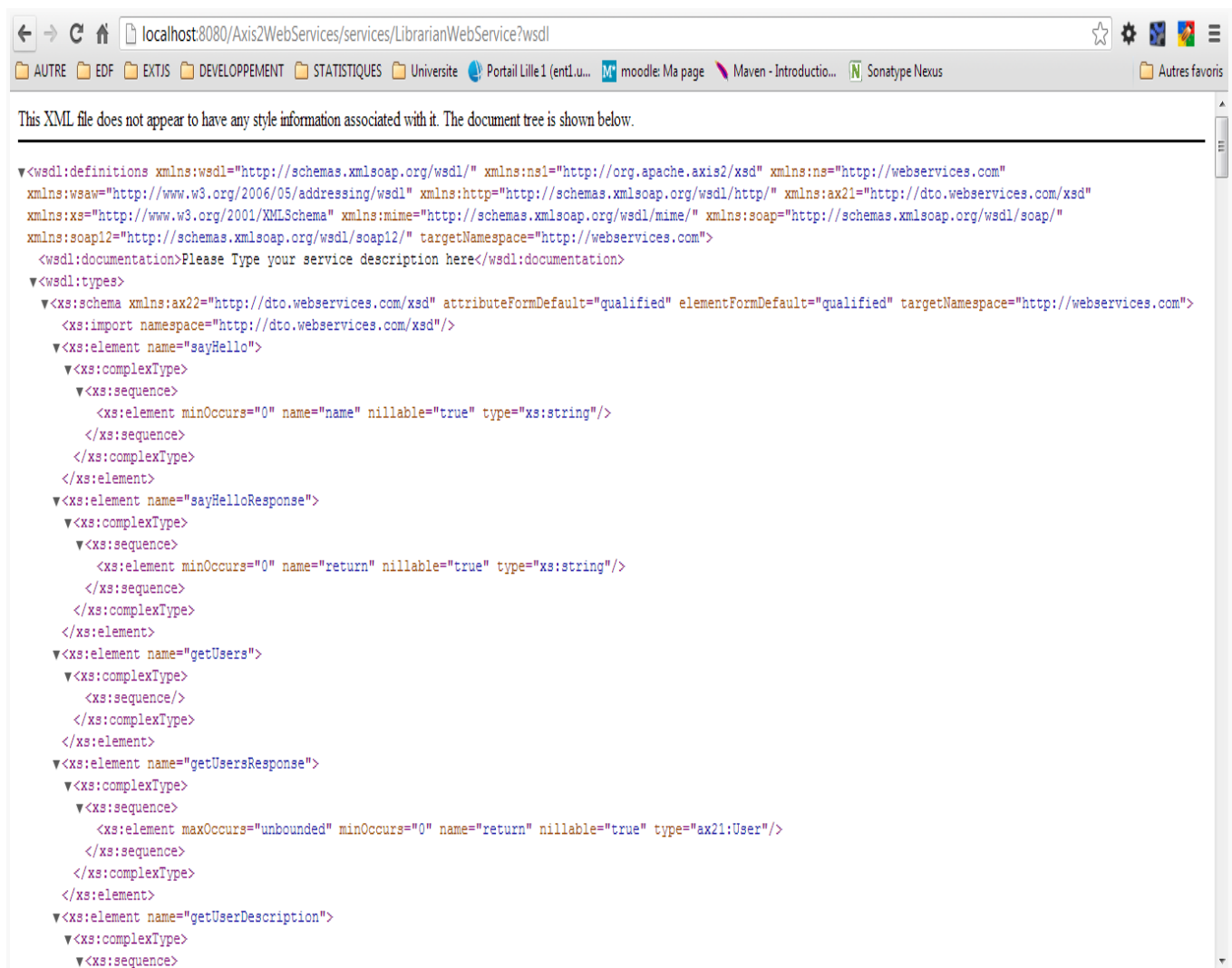
Available Operations

- addBook
- deleteBook
- getBookDescription
- sayHello
- getBooks

Copyright © 1999-2006, The Apache Software Foundation
Licensed under the [Apache License, Version 2.0](#).

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

Voici le WSDL généré par Axis2 :




```

<?xml version='1.0' encoding='UTF-8'?>
<wsc:definitions xmlns:wsc="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://webservicess.com"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:ax21="http://dto.webservicess.com/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://webservicess.com">
  <wsc:documentation>Please Type your service description here</wsc:documentation>
  <wsc:types>
    <xs:schema xmlns:ax22="http://dto.webservicess.com/xsd" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://webservicess.com">
      <xs:import namespace="http://dto.webservicess.com/xsd"/>
      <xs:element name="sayHello">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="name" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="sayHelloResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getUsers">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
      </xs:element>
      <xs:element name="getUsersResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="ax21:User"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="getUserDescription">
        <xs:complexType>
          <xs:sequence>

```

On retrouve les opérations définies dans les méthodes des classes Java développées.

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

```


▼<wsdl:output>
  <mime:content type="application/xml" part="parameters"/>
</wsdl:output>
</wsdl:operation>
▼<wsdl:operation name="getBookDescription">
  <http:operation location="getBookDescription"/>
  ▼<wsdl:input>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:input>
  ▼<wsdl:output>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:output>
</wsdl:operation>
▼<wsdl:operation name="sayHello">
  <http:operation location="sayHello"/>
  ▼<wsdl:input>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:input>
  ▼<wsdl:output>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:output>
</wsdl:operation>
▼<wsdl:operation name="getBooks">
  <http:operation location="getBooks"/>
  ▼<wsdl:input>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:input>
  ▼<wsdl:output>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:output>
</wsdl:operation>
▼<wsdl:operation name="addUser">
  <http:operation location="addUser"/>
  ▼<wsdl:input>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:input>
  ▼<wsdl:output>
    <mime:content type="application/xml" part="parameters"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>

```

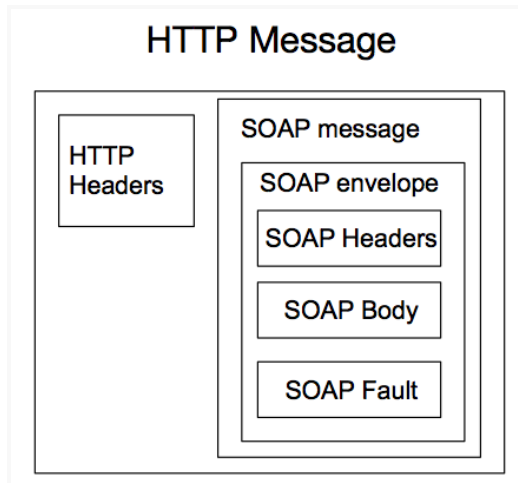
Cette méthode via eclipse est équivalente à la génération en ligne de commande avec axis2-1.6.2/bin/java2wsdl.

Il existe une autre approche 'Top down Java bean webService' permettant de transformer un WSDL en classe java avec la commande axis2-1.6.2/bin/wsdl2java.

On peut appeler notre web service avec un message http via SOAP en accédant au

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

endpoint: <http://localhost:8080/Axis2WebServices/services/LibrarianWebService/>



Une opération du webservice peut alors être consommée de cette façon


<http://localhost:8080/Axis2WebServices/services/LibrarianWebService/sayHello?name=tarik>

où [sayHello?name=tarik](#) correspond à un appel à la méthode suivante (définie dans la classe LibrarianWebService)



Avec cette méthode, nous devons utiliser Eclipse à chaque fois que nous voulons lancer le service, si nous ne créons pas de .aar (Archive Axis).


C'est pourquoi, nous allons voir dans le prochain paragraphe comment créer un .aar et le déployer.

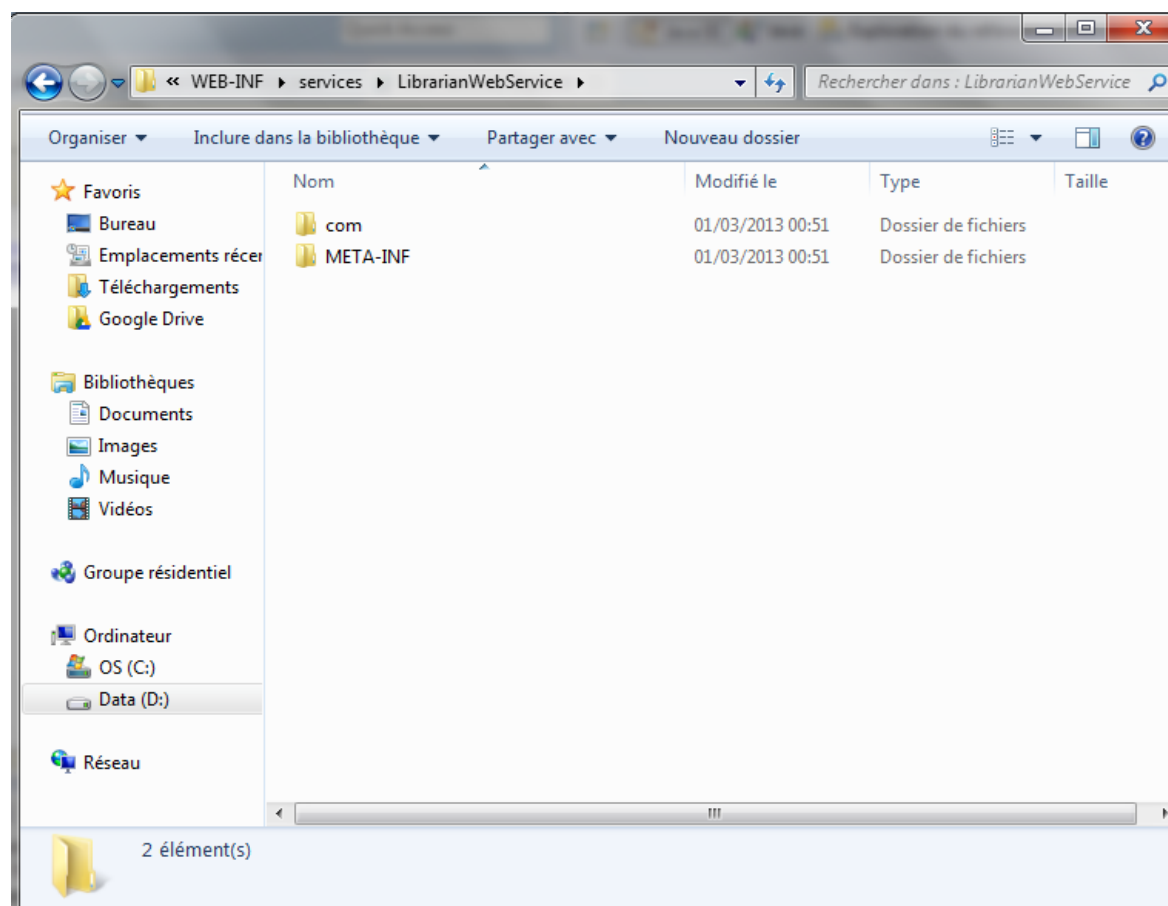
 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0


4. Créer un fichier .aar (Archive Axis) et déployer le service

4.1 Dans votre workspace, aller vers le dossier contenant le LibrarianWebService. Ouvrir une console, et taper la commande suivante pour créer le LibrarianWebService.aar.

```
jar cvf LibrarianWebService.aar com META-INF
```

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0




 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

```

C:\ Administrateur : Invite de commandes

D:\>cd Workspace\cara_tp1\WebContent\WEB-INF\services\LibrarianWebService
D:\Workspace\cara_tp1\WebContent\WEB-INF\services\LibrarianWebService>jar cvf Li
brarianWebService.aar com META-INF
manifeste ajouté
ajout : com/<entrée = 0> <sortie = 0><stockage : 0 %>
ajout : com/client/<entrée = 0> <sortie = 0><stockage : 0 %>
ajout : com/client/test/<entrée = 0> <sortie = 0><stockage : 0 %>
ajout : com/client/test/LibrarianWebServiceTest.class<entrée = 6400> <sortie = 2
622><compression : 59 %>
ajout : com/webservices/<entrée = 0> <sortie = 0><stockage : 0 %>
ajout : com/webservices/dto/<entrée = 0> <sortie = 0><stockage : 0 %>
ajout : com/webservices/dto/Book.class<entrée = 1800> <sortie = 758><compression
: 57 %>
ajout : com/webservices/dto/Library.class<entrée = 3579> <sortie = 1696><compres
sion : 52 %>
ajout : com/webservices/LibrarianWebService.class<entrée = 1823> <sortie = 739><
compression : 59 %>
ajout : com/webservices/LibrarianWebServiceCallbackHandler.class<entrée = 2805>
<sortie = 754><compression : 73 %>
ajout : com/webservices/LibrarianWebServiceStub$1.class<entrée = 5911> <sortie =
2480><compression : 58 %>
ajout : com/webservices/LibrarianWebServiceStub$2.class<entrée = 6021> <sortie =
2495><compression : 58 %>
ajout : com/webservices/LibrarianWebServiceStub$3.class<entrée = 5921> <sortie =
2496><compression : 57 %>
ajout : com/webservices/LibrarianWebServiceStub$4.class<entrée = 5921> <sortie =
2495><compression : 57 %>
ajout : com/webservices/LibrarianWebServiceStub$5.class<entrée = 5941> <sortie =
2489><compression : 58 %>
ajout : com/webservices/LibrarianWebServiceStub$AddBook$Factory.class<entrée = 4
039> <sortie = 1959><compression : 51 %>
ajout : com/webservices/LibrarianWebServiceStub$AddBook.class<entrée = 9294> <so
rtie = 4091><compression : 55 %>
ajout : com/webservices/LibrarianWebServiceStub$AddBookResponse$Factory.class<en
trée = 3583> <sortie = 1712><compression : 52 %>
ajout : com/webservices/LibrarianWebServiceStub$AddBookResponse.class<entrée = 7
937> <sortie = 3583><compression : 54 %>
ajout : com/webservices/LibrarianWebServiceStub$Book$Factory.class<entrée = 4330
> <sortie = 2090><compression : 51 %>

```

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

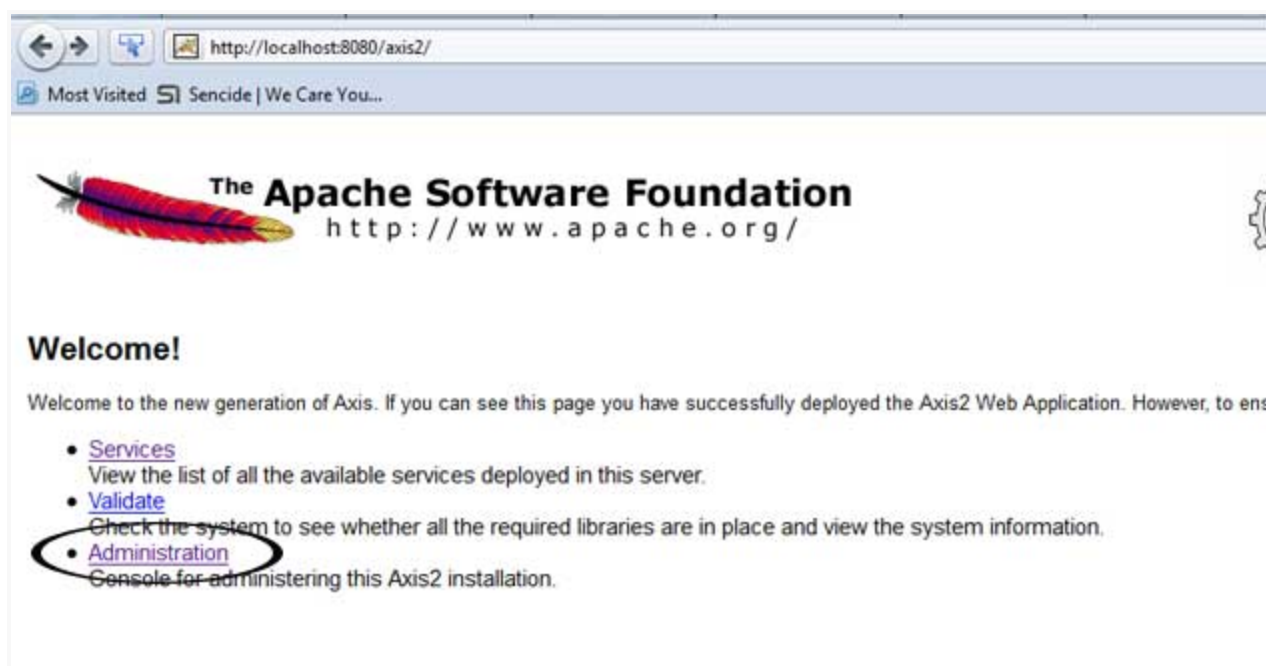
4.2 Ensuite, copier le fichier axis2.war que vous trouverez dans la distribution Apache Axis2 WAR Distribution (téléchargé au tout début), vers le dossier webapps de Apache Tomcat.


Arrêter le serveur Tomcat qui tourne su votre Eclipse IDE.

4.3 Utiliser la ligne de commande pour démarrer tomcat (aller dans le répertoire bin et démarrer via le batch startup.bat).

Maintenant, il y aura un nouveau répertoire nommé axis2 dans le répertoire webapps.

Si vous aller maintenant vers la page <http://localhost:8080/axis2/>, vous trouverez la page d'accueil de l'application Web de Axis2.

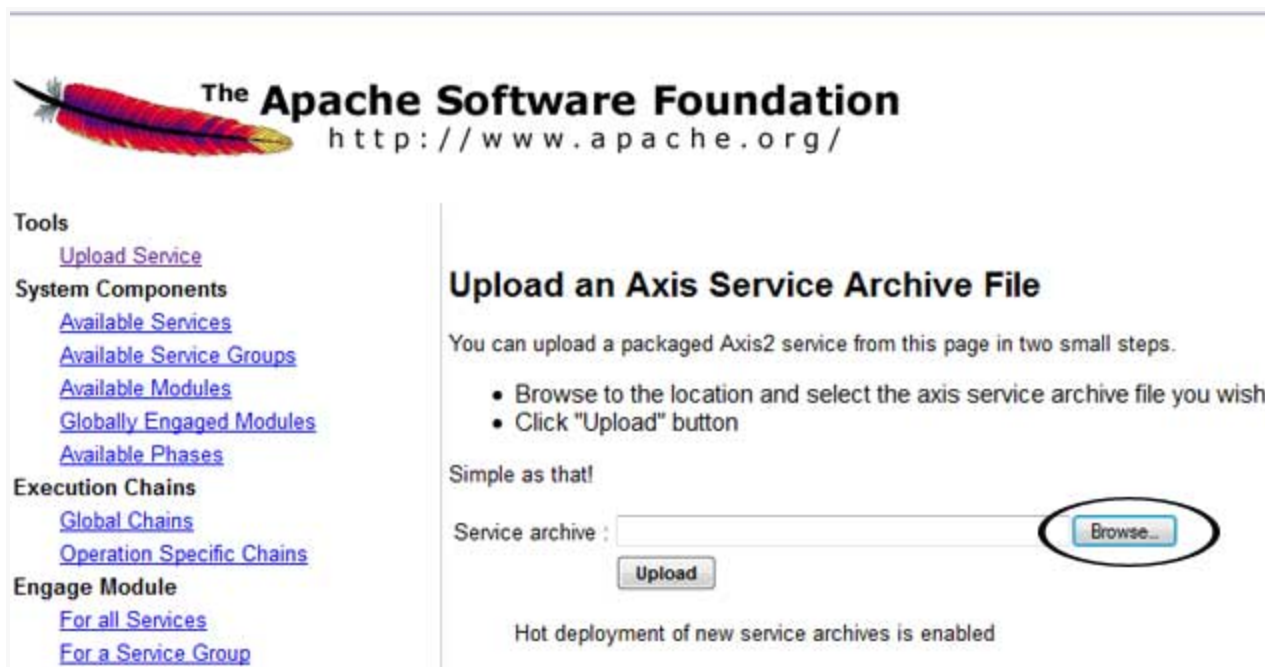


 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

4.4 Cliquer ensuite sur le lien 'Administration', et logger vous avec le login admin et password 'axis2'.

Vous pouvez maintenant uploader votre webService LibrarianWebService en cliquant sur le bouton 'Browse' et ensuite 'Upload'.

(Cela revient à copier manuellement le LibrarianWebService .aar vers le répertoire webapps\axis2\WEB-INF\services)



The Apache Software Foundation
http://www.apache.org/

Tools
[Upload Service](#)

System Components
[Available Services](#)
[Available Service Groups](#)
[Available Modules](#)
[Globally Engaged Modules](#)
[Available Phases](#)

Execution Chains
[Global Chains](#)
[Operation Specific Chains](#)

Engage Module
[For all Services](#)
[For a Service Group](#)

Upload an Axis Service Archive File


You can upload a packaged Axis2 service from this page in two small steps.

- Browse to the location and select the axis service archive file you wish
- Click "Upload" button


Simple as that!


Service archive : **Browse...**

Hot deployment of new service archives is enabled

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

4.5 Vous trouverez alors la liste des services (y compris le service LibrarianWebService) en suivant le lien
<http://localhost:8080/axis2/services/listServices>


The Apache Software Foundation
<http://www.apache.org/>



[Back Home](#) | [Refresh](#)

Available services

[Version](#)

Service Description : Version

Service EPR : http://localhost:8080/cara_tp1/services/Version

Service Status : Active

Available Operations

- getVersion

[LibrarianWebService](#)


Service Description : Please Type your service description here

Service EPR : http://localhost:8080/cara_tp1/services/LibrarianWebService

Service Status : Active

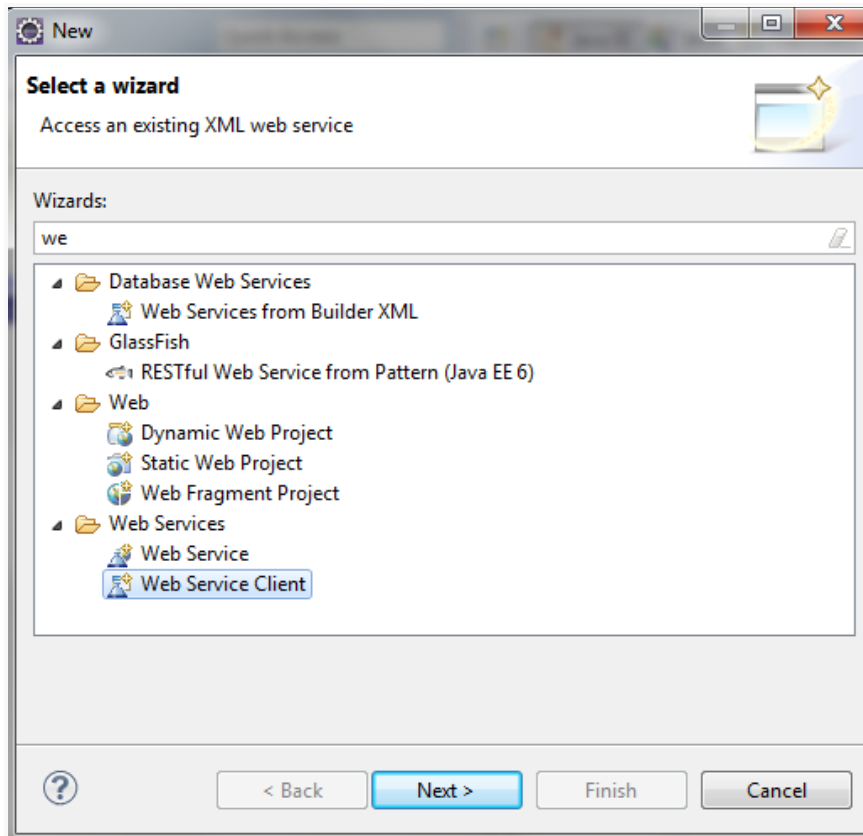
Available Operations


- addBook
- deleteBook
- getBookDescription

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

5. Création d'un client pour le Webservice

5.1 Dans Eclipse, aller dans File --> New --> Other.. --> Web Service Client



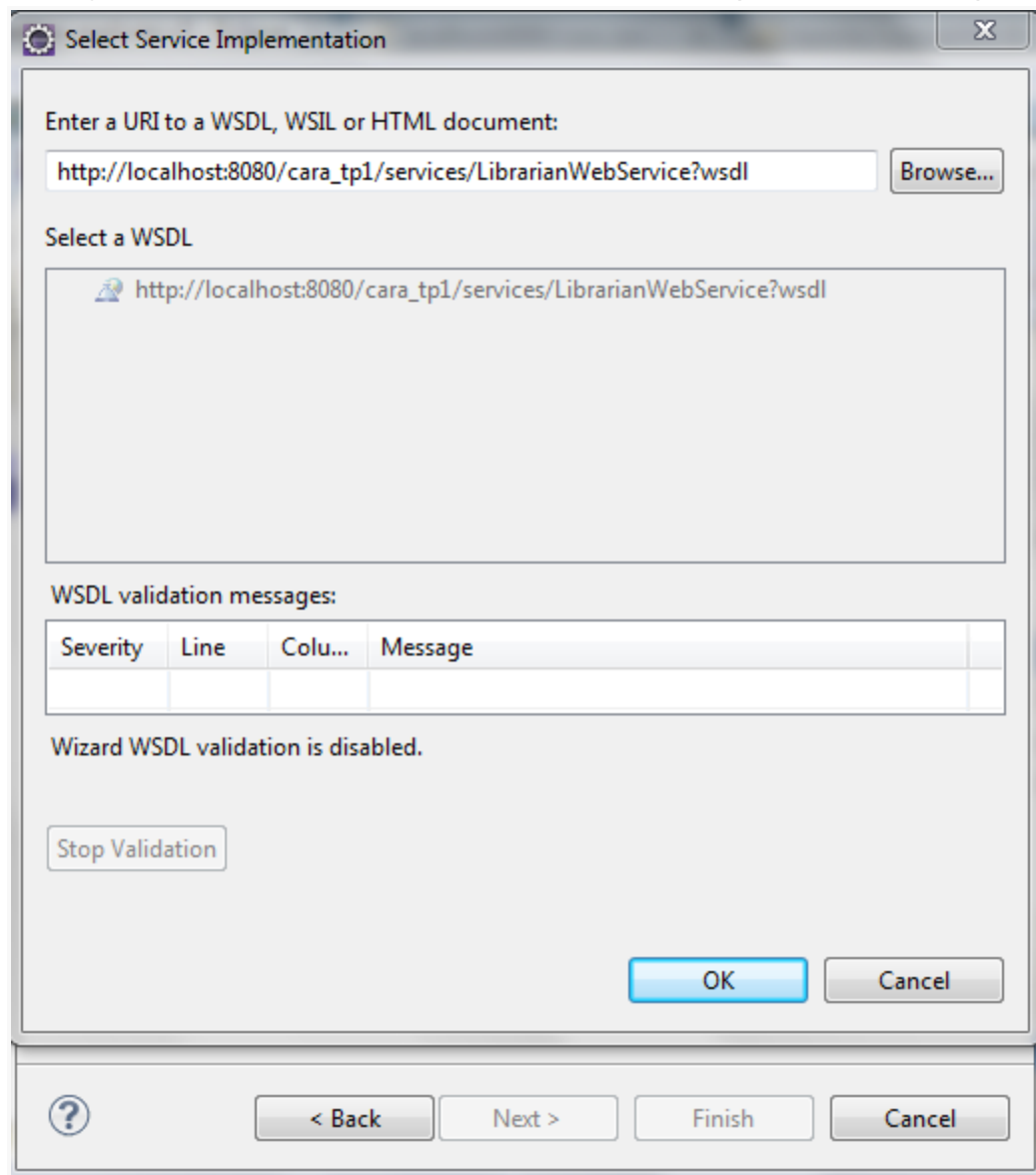
 Université Lille 1	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0


5.2 Rentrer l'URI

http://localhost:8080/cara_tp1/services/LibrarianWebService?wsdl

comme définition du service.

Configurer ensuite le Server Runtime comme auparavant. et cliquer sur Finish.




 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

5.3 Cela va générer deux nouvelles classes appelées :

- LibrarianWebServiceStub.java
- LibrarianWebServiceCallbackHandler.java

Vous pouvez maintenant créer une classe de test pour le client et utiliser votre webservice.

 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

Ci-dessous un exemple de méthode java pour l'ajout d'un livre à la bibliothèque :


```

/**
 * Ajoute un livre dans la bibliotheque
 * @param author : l'auteur du livre
 * @param title : le titre du livre
 * @param editor : l'editeur du livre
 * @param borrower : l'emprunteur du livre
 * @return le livre ajoute
 */
public Book addBook(String author, String title, String editor, String borrower){
    if(author == null) author = "";
    if(title == null) title = "";
    if(editor == null) editor = "";
    if(borrower == null) borrower = "";

    Book newBook = new Book();
    newBook.setAuthor(author);
    newBook.setTitle(title);
    newBook.setEditor(editor);
    newBook.setBorrower(borrower);
    Long isbn = generateIsbn();
    newBook.setIsbn(isbn);

    books.put(isbn, newBook);
    Book bookAdded = books.get(isbn);
    return bookAdded;
}

```


 Université Lille1 Sciences et Technologies	Conception d'applications réparties avancées	Creation d'un Web Service en java en utilisant Apache Axis2 et Eclipse
Latest update : 16/02/2013	Documentation technique	Current doc version : 1.0

Un stub sera généré pour cette méthode, lors de la création d'un nouveau WebService.
Ce stub sera de la forme :

```
public com.webservices.LibrarianWebServiceStub.AddBookResponse addBook
(com.webservices.LibrarianWebServiceStub.AddBook addBook2)
throws java.rmi.RemoteException{

    org.apache.axis2.context.MessageContext _messageContext = null;
    try{
        org.apache.axis2.client.OperationClient _operationClient =
        _serviceClient.createClient(_operations[1].getName());
        _operationClient.getOptions().setAction("\\\\");
        ....
    }
    ...
}
```

Enfin, l'appel à la méthode se fera facilement via la méthode :
librarianWebServiceStub.addBook(...) de la classe **LibrarianWebServiceStub** généré

```
@Test
public void addBookTest() throws RemoteException {
    String bookTitle = "Test Driven Development book";

    AddBook addBook = new AddBook();
    addBook.setTitle(bookTitle);
    AddBookResponse addBookResponse = librarianWebServiceStub.addBook(addBook);
    com.webservices.LibrarianWebServiceStub.Book addedBook = addBookResponse.get_return();

    DeleteBook deleteBook = new DeleteBook();
    deleteBook.setIsbn(addedBook.getIsbn());
    librarianWebServiceStub.deleteBook(deleteBook);

    assertEquals(addedBook.getTitle(),bookTitle);
}
```