

Comment installer et exécuter wsig-exemples en 5 étapes

Auteur : Sebastien Heymann <sebastien.heyman@tremplin-utc.net>

Date : 29 décembre 2008

Pré-requis :

- Savoir utiliser l'IDE Eclipse est fortement conseillé mais pas indispensable
- Savoir installer et configurer Tomcat (non traité ici car hors de propos)
- Savoir utiliser Jade

Notes :

- Ce document traite de la mise en place d'un environnement de développement, il ne doit pas être utilisé en production.
- Ce tutoriel repose sur le guide de WSIG (v2.1), notamment pour l'exemple MathAgent.

Distribution :

- JADE v3.6.1 - <http://jade.tilab.com>
- WSIG v2.1 - <http://jade.tilab.com/community-addons.php>

Logiciels installés dans ce tutoriel :

- Java Run Time Environment version 1.5 nécessaire (testé en 1.6)
- Eclipse 3.4+ - <http://www.eclipse.org>
- Eclipse Web Tools Project (WTP) 3.0.3+
- Java EE Developer Tools minimum installation
- Subclipse 1.4.7+ - <http://subclipse.tigris.org>
- Apache Tomcat 5.5 - <http://tomcat.apache.org>

Etape 1 – Installation de Tomcat et Eclipse

Commencez tout d'abord par installer Tomcat 5.5. Sur Ubuntu, il est disponible dans les dépôts officiels. Pensez à désactiver son démarrage automatique via la commande

```
sudo update-rc.d -f tomcat5.5 remove
```

Installez l'Eclipse que vous voulez (<http://www.eclipse.org/downloads/>). Nous utilisons la distribution Classic dans ce tutoriel.



Eclipse Classic 3.4.1 (151 MB)

The classic Eclipse download: the Eclipse

[Release notes](#) | [Other downloads](#) | [Docum](#)

Downloads: 1,095,515

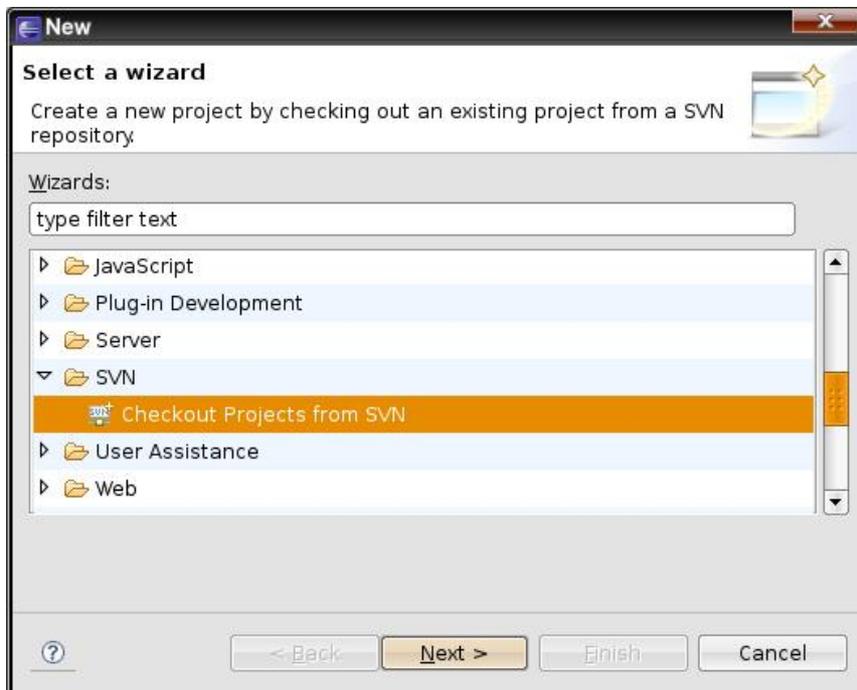
[HOME](#) | [PRIVACY POLICY](#) | [TERMS OF USE](#) **Updated!**

Illustration 1: <http://www.eclipse.org/downloads/>

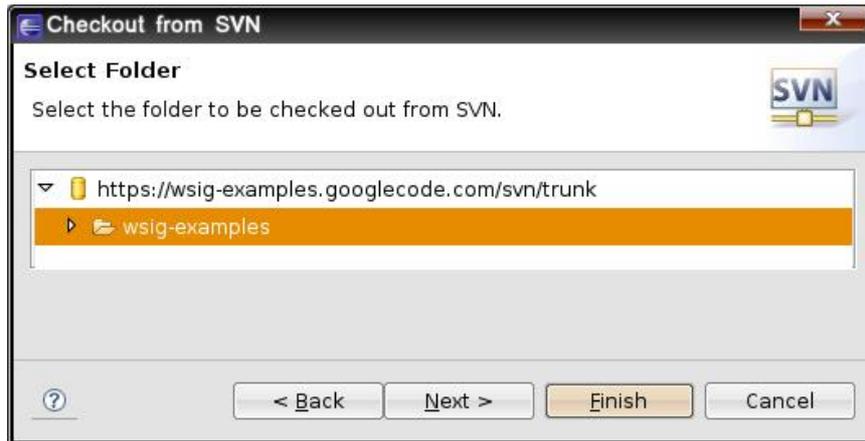
1. Créez ensuite un workspace vide dans Eclipse qui restera dédié au projet, et redémarrez dedans.
2. Allez dans *Help > Software Updates*, puis ajoutez le site <http://download.eclipse.org/webtools/updates>.
3. Installez Java EE Developer Tools (dans Eclipse Web Tools Project (WTP) 3.0.3). Notez qu'Axis n'est pas nécessaire car il est fourni dans la distribution de WSIG.
4. Ajoutez enfin le site http://subclipse.tigris.org/update_1.4.x, et téléchargez Subclipse, qui vous permettra d'avoir la gestion du SVN intégré.

Etape 2 – Récupération du projet

1. Faites *File > New > Other... > SVN > Checkout Projects from SVN*



2. Cliquez sur *create a new repository location*
3. Ajoutez <https://wsig-examples.googlecode.com/svn/trunk/>.
4. Sélectionnez enfin *wsig-examples-read-only* (ou *wsig-examples* si vous avez les droits d'écriture sur le SVN) et terminez.



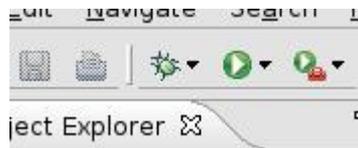
Ce projet est de type Dynamic Web Project. Une fois le contenu du dépôt téléchargé et le workspace mis à jour automatiquement, Eclipse déclare une erreur :

"Target runtime Apache Tomcat v5.5 is not defined. wsig-examples Unknown Faceted Project Problem"

C'est normal à ce stade, nous définirons un serveur à l'étape 4.

Etape 3 – Plateforme Jade et agents

Créez et démarrez ensuite les *Debug Configurations* suivantes en tant que *Java Application* :



runJadePlatform

Project	wsig-examples
Main class	jade.Boot
Program arguments	-gui

Cliquez sur *Debug* pour démarrer la plateforme Jade.

runMathAgent

Project	wsig-examples
Main class	jade.Boot
Program arguments	-gui -container "MathAgent1:com.tilab.wsig.examples.MathAgent(MathFunctions false)" -name "WSIGTestPlatform"

Cliquez sur *Debug* pour démarrer l'agent MathAgent1, instance de MathFunctions.

runMathAgentPrefix

Project	wsig-examples
Main class	jade.Boot
Program arguments	-gui -container "MathAgent2:com.tilab.wsig.examples.MathAgent (MathFunctions false second)" -name "WSIGTestPlatform"

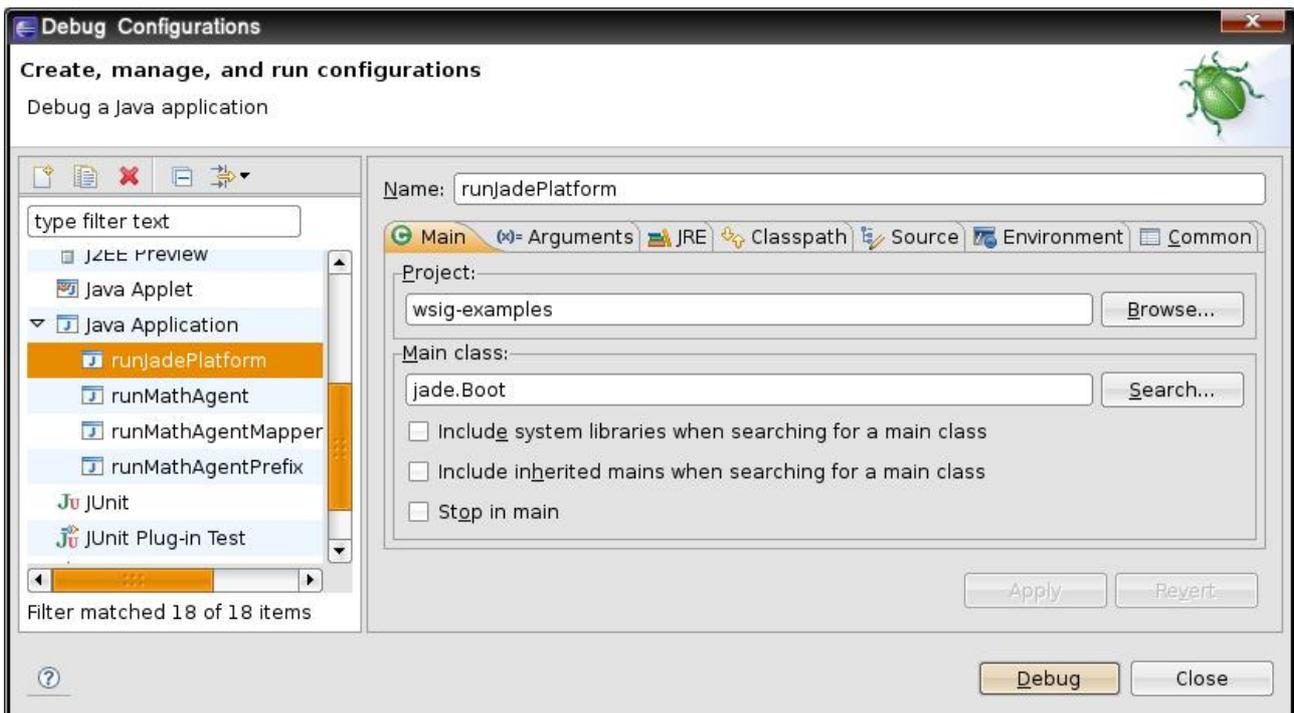
Cliquez sur *Debug* pour démarrer l'agent MathAgent2, instance de MathFunctions.

runMathAgentMapper

Project	wsig-examples
Main class	jade.Boot
Program arguments	-gui -container "MathAgent3:com.tilab.wsig.examples.MathAgent (MathFunctionsM apper true)" -name "WSIGTestPlatform"

Cliquez sur *Debug* pour démarrer l'agent MathAgent3, instance de MathFunctionsMapper.

Pour plus de détails sur ces agents, reportez-vous au guide de WSIG.



Démarrage

Par la suite, les agents peuvent être démarrés directement via la barre d'outils :

Debug > *runJadePlatform*

Debug > *runMathAgent*

Debug > *runMathAgentPrefix*

Debug > *runMathAgentMapper*

L'interface graphique de Jade Platform a dû apparaître, et vous pouvez voir les 3 agents évoluant dans 3 conteneurs différents. Ces agents peuvent être lancés indépendamment les uns des autres.

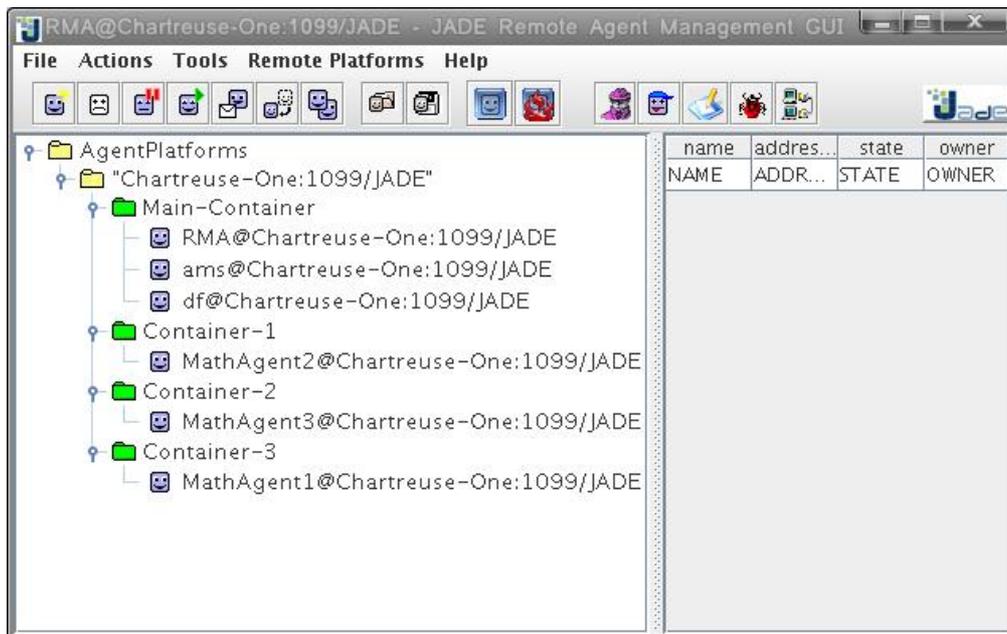


Illustration 2: Interface graphique de Jade

Etape 4 – Ajout du serveur

Procédure

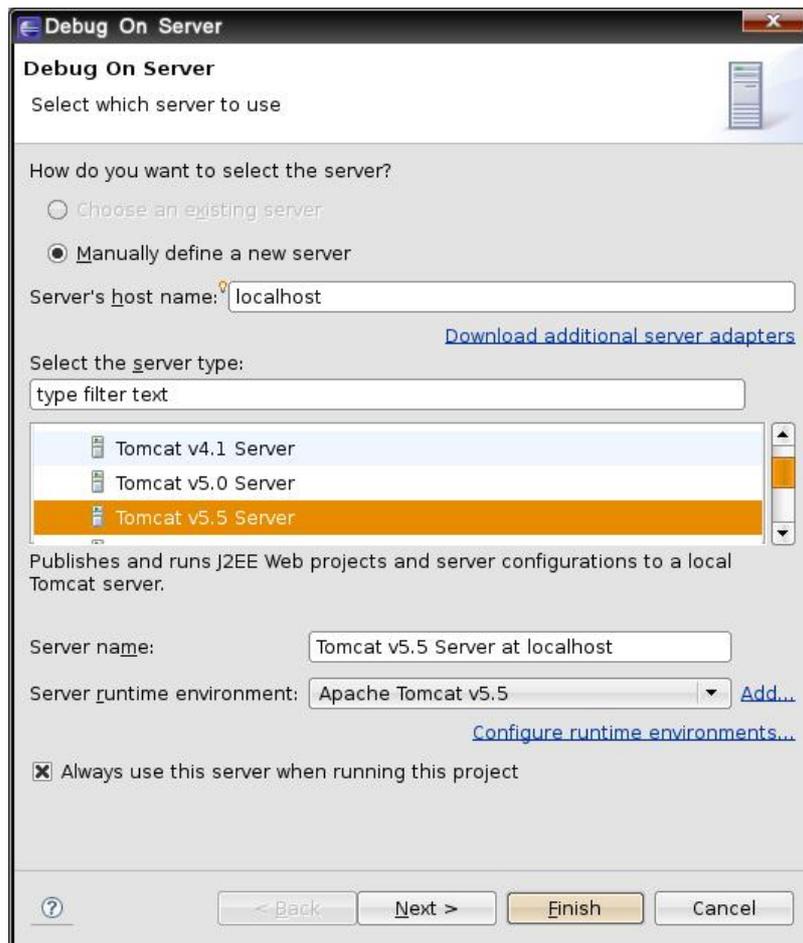
1. Passez en perspective JavaEE.



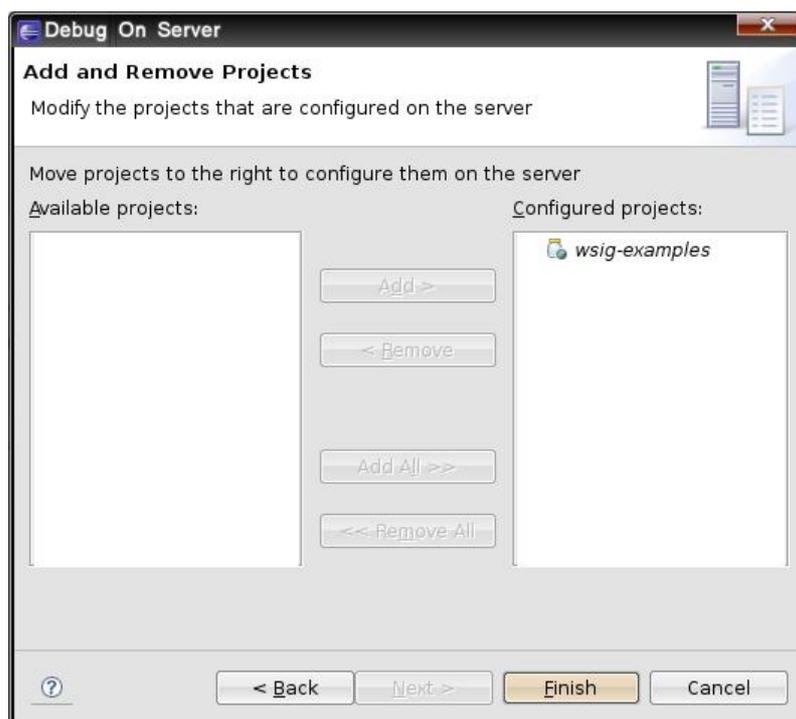
2. Sélectionnez le projet *wsig-examples* dans la liste des projets



3. Créez le *Debug Configuration* de l'application (*Debug > Debug As > Debug on server*) : choisissez *Manually define a new server* de type *Apache > Tomcat v5.5 Server*, cochez la case *Always use this server when running this project*, et faites *Next*.

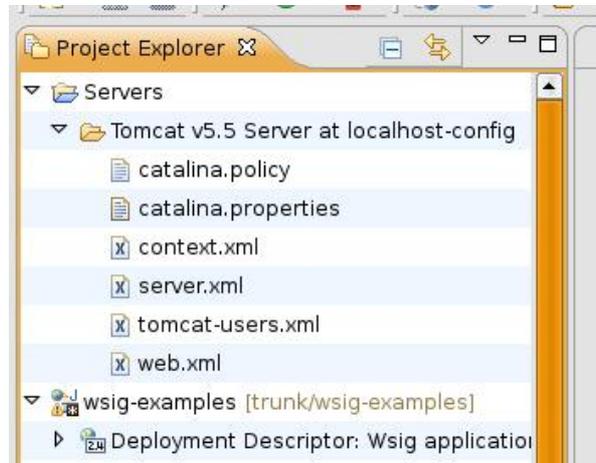


4. Renseignez le chemin d'installation de Tomcat, */usr/share/tomcat5.5* sous Ubuntu 7.10, et faites Next.
5. Ajoutez le projet *wsig-examples* aux projets configurés pour le serveur, et terminez.



Configuration

Dans la vue des projets, un nouveau projet est apparu : *Servers*. Il contient un dossier comprenant les fichiers de configuration de Tomcat (les fichiers XML).



Pour savoir quels ports écoute le serveur, ouvrez *server.xml*. Par exemple, ma configuration par défaut est la suivante :

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8180 -->
<Connector acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
enableLookups="false" maxHttpHeaderSize="8192" maxSpareThreads="75" maxThreads="150"
minSpareThreads="25" port="8180" redirectPort="8443"/>
...
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector enableLookups="false" port="8009" protocol="AJP/1.3" redirectPort="8443"/>
```

Vous pouvez changer les ports, les changements seront pris en compte immédiatement même si le serveur est démarré.

L'application *wsig-examples* est déclarée vers la fin du fichier :

```
<Context docBase="wsig-examples" path="/wsig-examples" reloadable="true"
source="org.eclipse.jst.j2ee.server:wsig-examples"/></Host>
```

La valeur de *reloadable* permet ici de propager les modifications immédiatement sur le serveur, ce qui est très pratique pour éditer et tester du code à la volée. Sans ce mécanisme, il faudrait packager l'application dans une archive war et la déployer sur le serveur à chaque fois.

Etape 5 – Déploiement et utilisation

Le serveur démarre immédiatement en déployant l'application après avoir été ajouté. L'agent

ControlWSIG est alors créé dans *WSIG-Container*. La console de *WSIG* apparaît à l'adresse <http://localhost:8180/wsigs-examples/> (variable selon le port utilisé), que vous pouvez aussi ouvrir dans votre navigateur web habituel.

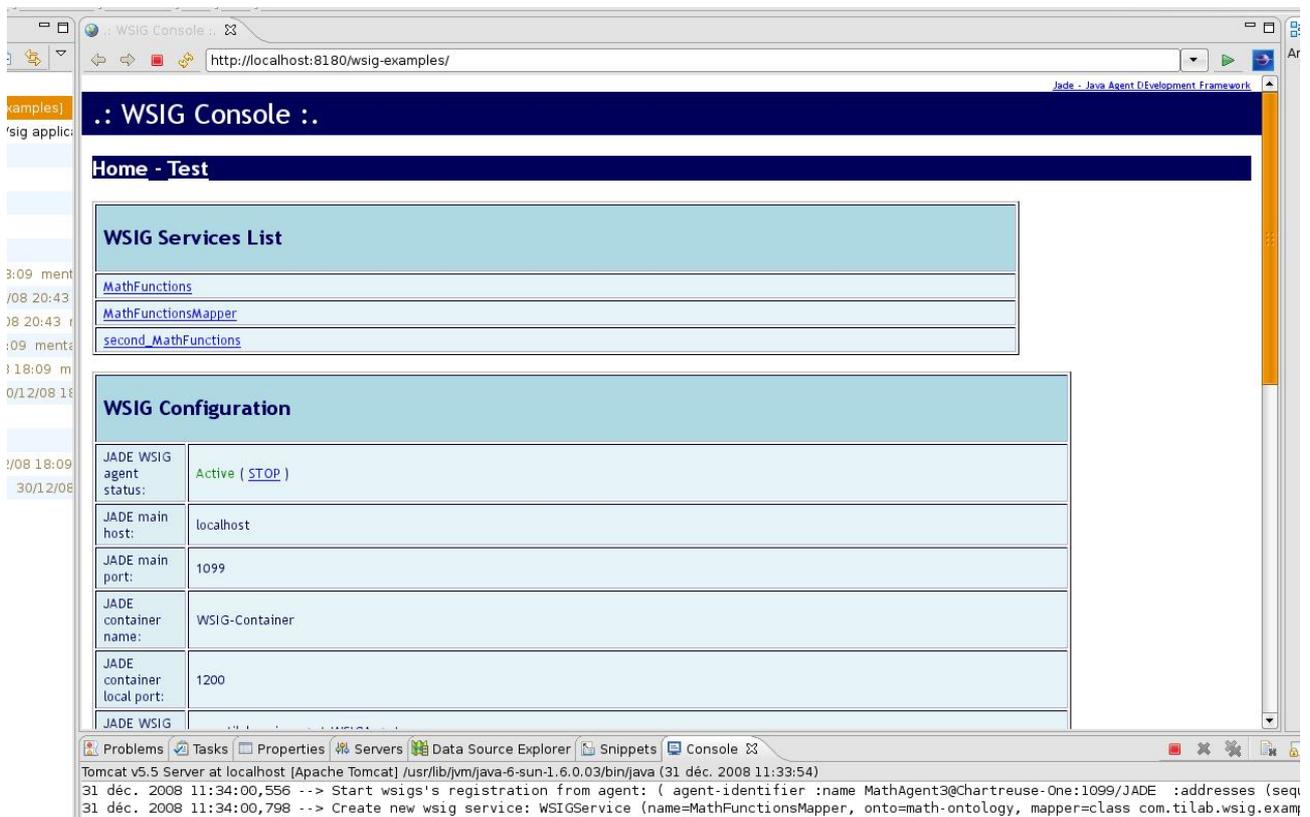
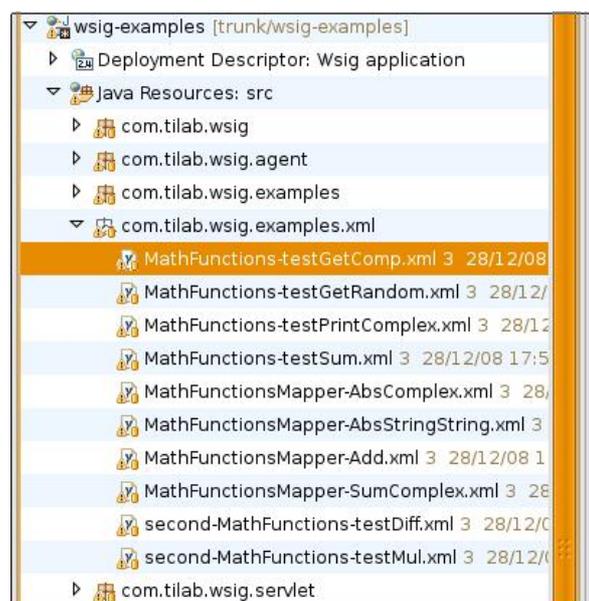


Illustration 3: Console web de *WSIG*

Utilisation

Les web services correspondants à nos agents sont listés sur la page *Home* dans *WSIG Services List*. Les exemples de requête SOAP sont dans le package *src/com.tilab.wsigs-examples.xml* :



Allez sur la page *Test* pour les faire fonctionner : le contenu du fichier XML à tester est à copier dans *SOAP request*. Il ne vous reste plus qu'à envoyer la requête !

Pense-bête :

- Si le serveur répond que le web service n'existe pas ("*Service xxx not present in wsig*"), vérifiez bien que l'agent correspondant est démarré.
- Si vous désactivez l'agent *ControlWSIG* via l'interface web et le réactivez, les autres agents auront été arrêtés. Lancez-les de nouveau (sans arrêter le serveur), et rafraîchissez la page *Home* pour les voir de nouveau dans la liste.

Test page	
WebService url:	http://localhost:8180/wsigs-examples/ws
SOAP request:	<pre><?xml version="1.0" encoding="UTF-8" ?> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:MathFunctions"> <soapenv:Header/> <soapenv:Body> <urn:getComponents soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <complex xsi:type="urn:complex"> <real xsi:type="xsd:float">4</real> </complex> </urn:getComponents> </soapenv:Body> </soapenv:Envelope></pre>
SOAP response:	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><soapenv:Body> <getComponentsResponse xmlns="urn:MathFunctions"><getComponentsReturn xmlns=""><float xmlns="">4.0</float><float xmlns="">5.0</float> </getComponentsReturn></getComponentsResponse></soapenv:Body> </soapenv:Envelope></pre>
<input type="button" value="Send"/> <input type="button" value="Reset"/>	

Illustration 4: Test de requête SOAP

Démarrage manuel

1. Démarrez la plateforme Jade
2. Démarrez les agents
3. Dans l'onglet Servers en bas de l'écran, cliquez droit sur le nom du serveur, puis cliquez sur *Debug*, ou démarrez directement *wsigs-examples* dans le menu *Run > Debug* ou la barre d'outils.

Arrêt

Pour tout arrêter proprement :

1. Cliquez droit sur le nom du serveur, puis cliquez sur *Stop*.
2. Arrêtez Jade via l'interface graphique en faisant *File > Shut down Agent Platform*.



Pour plus de détails sur Jade et WSIG, référez-vous à leurs guides d'utilisation respectifs.