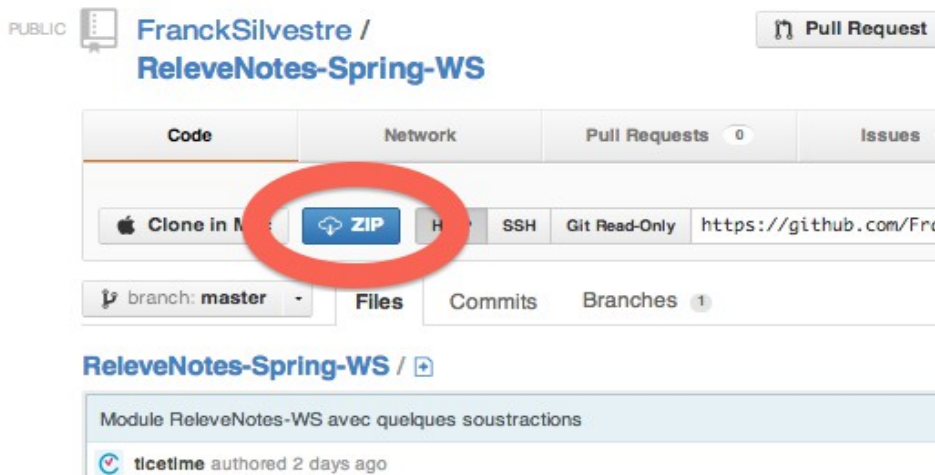


## Feuille de TP n°3 – Web Services SOAP « Contract first » et Web Services REST

### Exercice 1

Objectif : compléter une application existante pour qu'elle publie un Web Services SOAP/WSDL sur le modèle « contract first ».

1. Connectez vous à la machine virtuelle Linux qui vous a été effectuée.
2. Récupérez le projet ReleveNotes-Spring-WS sur Github (<https://github.com/FranckSilvestre/ReleveNotes-Spring-WS>) dans un dossier de votre « home directory ». Sur Github vous pouvez cloner le repository ou simplement télécharger le projet sous forme de zip.



3. Ouvrez Eclipse STS et importer le projet dans votre espace de travail. Le projet est un projet Maven, il faut donc sélectionner un import de type Maven. Les copies d'écran ci-dessous détaillent les étapes de l'import du projet dans Eclipse STS.

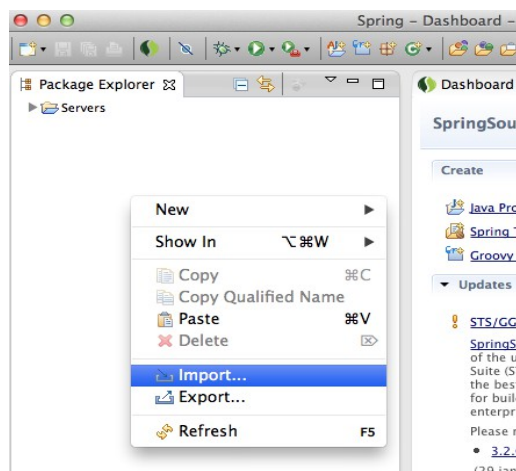


fig1. Sélection de l'action « Import »

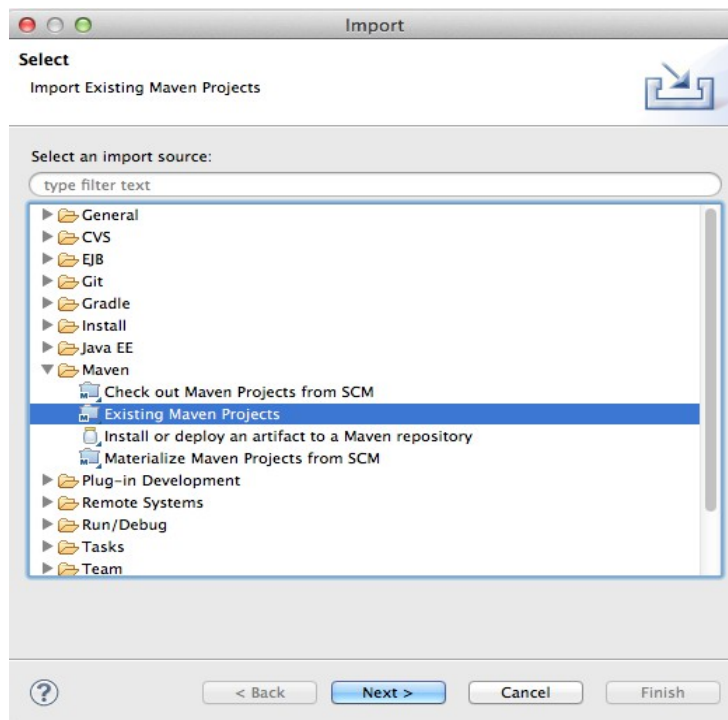


fig2. Sélection de l'import de type « Existing Maven Projects »

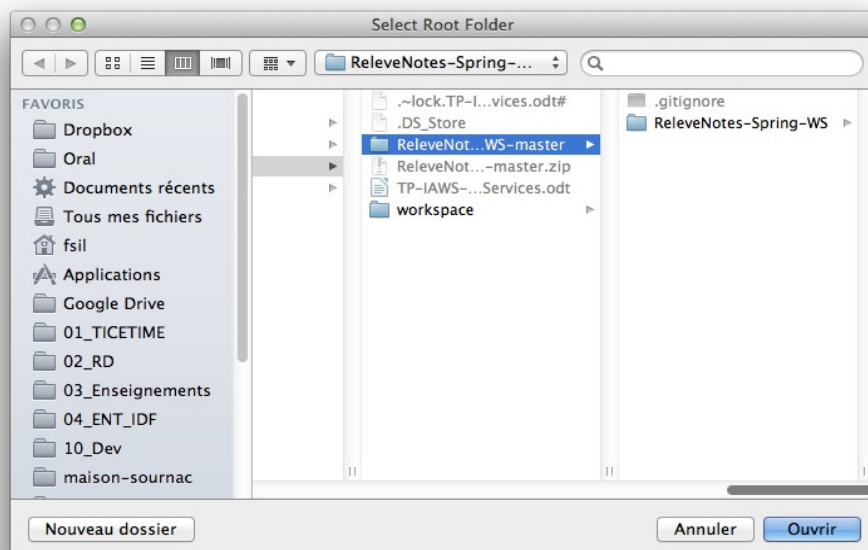


fig3. Sélection de la racine du projet récupéré.

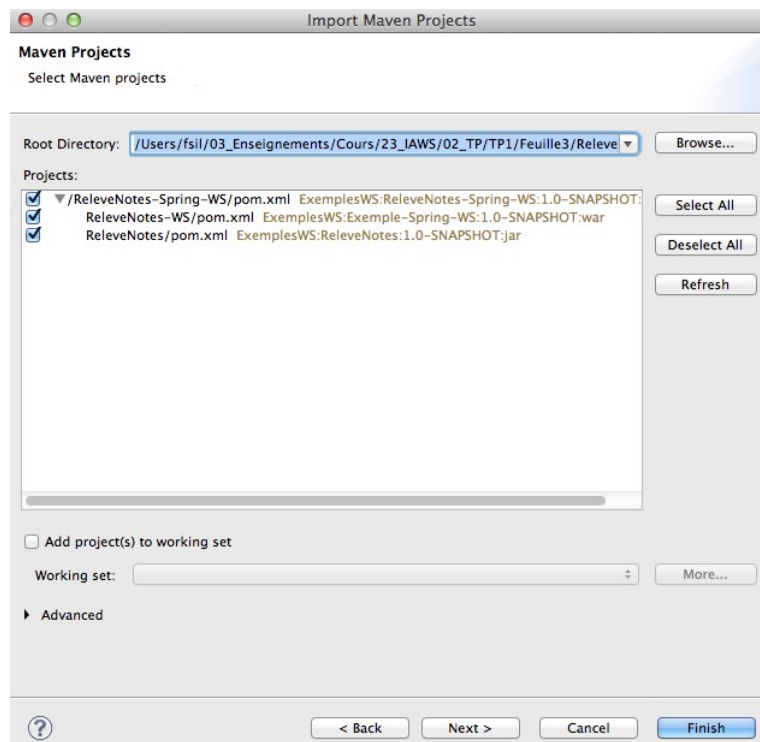


fig4. Sélection des modules à importer

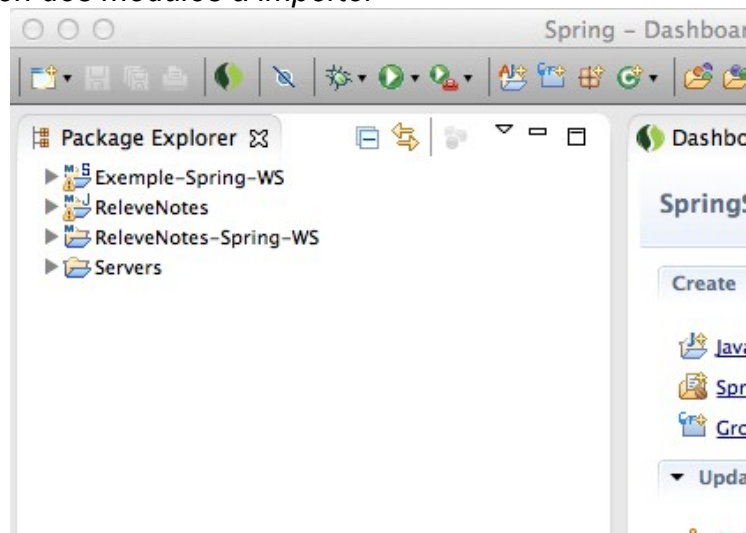
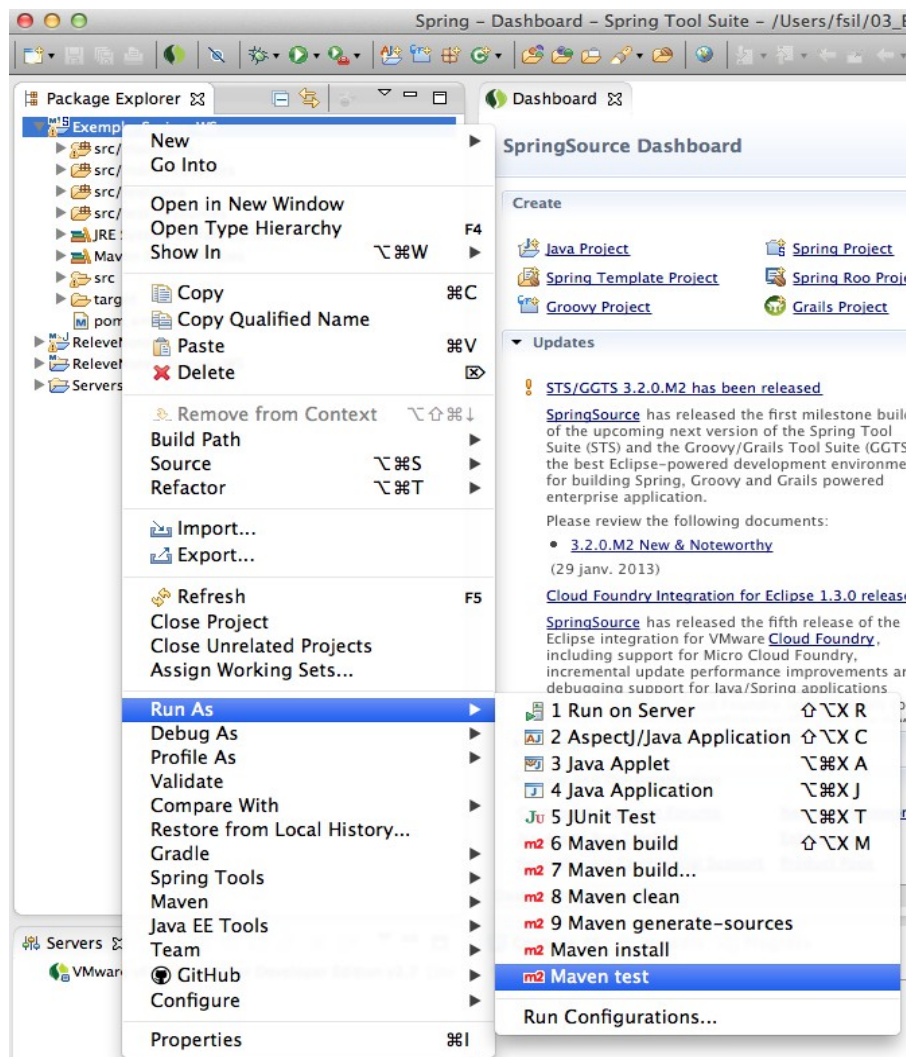


fig5. Résultat de l'import dans la vue Package Explorer

- Le module à modifier est le module ReleveNotes-Spring-WS. En vous appuyant sur la documentation de Spring WS <http://static.springsource.org/spring-ws/sites/2.0/reference/html/tutorial.html> compléter le module pour que le test d'intégration testant la publication du Web Services passe. Pour lancer le test d'intégration, il suffit d'exécuter le projet principal comme « Maven test ».





Université  
Paul Sabatier  
TOULOUSE III

M1 Info DL

Interopérabilité des applications et introduction aux Web Services

## Exercice 2

Objectif : consommer un Web Service de type REST en Groovy et/ou en Java.

1. A l'aide de votre navigateur accédez à l'adresse suivante :  
<http://scolarite.ticetime.cloudbees.net/etudiant/list>  
Explorez l'application.
2. L'application fournit une API REST. Toujours avec votre navigateur, accédez aux URLs suivantes :  
<http://scolarite.ticetime.cloudbees.net/api/v1/etudiants>  
<http://scolarite.ticetime.cloudbees.net/api/v1/etudiants/1>  
<http://scolarite.ticetime.cloudbees.net/api/v1/evaluations>  
<http://scolarite.ticetime.cloudbees.net/api/v1/evaluations/3>  
L'API REST proposée souffre d'un défaut : le quel ?
3. Récupérer sur Github le projet Maven ScolariteClient en suivant le même procédé que pour l'exercice 1. Le projet se trouve à l'adresse suivante :  
<https://github.com/FranckSilvestre/ScolariteClient/>  
Il se peut qu'Eclipse STS relève une erreur au niveau du POM, ignorez là.
4. Explorez le projet et exécutez le projet en Maven Test. En vous inspirant de l'exemple donné dans le projet (code principal et test), écrivez un programme en Java ou en Groovy qui permette d'afficher pour un étudiant donné le détail de ses évaluations. L'affichage pourra s'effectuer dans la console (le résultat de l'affichage de la console lorsqu'on exécute un Maven Test se trouve dans le fichier target/surefire-reports/scolarite.client....-output.txt).



Ce(tte) oeuvre est mise à disposition selon les termes de la [Licence Creative Commons Paternité - Pas d'Utilisation Commerciale 3.0 France](#).