

# STI 3<sup>e</sup> année – Outils de développement

## TD : Ant+Ivy

J.-F. Lalande

Dans ce TD, nous proposons de développer une application en s'appuyant sur Ant+Ivy. Ivy permet de gérer les dépendances de projet et s'interface avec ant. Il utilise les dépôts Maven.

**Pensez à créer un compte rendu PDF pour ce TD ! Il est attendu, dans ce compte rendu, de mentionner le dépôt github associé.**

### 1 Mon premier projet Ant+Ivy

**Exercice 1** Si Ant n'est pas présent (testez en ligne de commande la commande "ant"), récupérez les binaires de ant et décompressez les dans un coin. Changez votre PATH afin de pouvoir exécuter la commande "ant".

**Exercice 2** Créer un répertoire "ProjectIvy". Placez-y le fichier build.xml que vous pouvez télécharger à l'étape 2 du "First Tutorial" situé à <http://ant.apache.org/ivy/history/latest-milestone/samples/build.xml>

**Exercice 3** Transformez ce répertoire en dépôt git. Faites un premier push de votre projet sur github. Précisez l'url de votre dépôt sur votre compte rendu.

**Exercice 4** Tapez "ant". Regardez les fichiers générés, notamment dans src. Que fait le programme principal ?

**Exercice 5** Ajoutez le répertoire des sources à votre projet git.

**Exercice 6** Analysez la tâche "go". Que est le but des tags "javac" et "java" dans cette tâche ?

**Exercice 7** Délocalisez le tag "java" (exécution) dans une tâche ant nommée "run" (on pourra ainsi faire "ant run"). Donnez le contenu de votre tag "target".

**Exercice 8** Dans votre tâche "go", avant la compilation, nous allons afficher le contenu de la variable "lib.path.id" avec le code ci-dessous. Que contient votre variable ?

```
<pathconvert property="tmp" refid="lib.path.id"/>
<echo>Dans le Classpath : ${tmp}</echo>
```

1  
2

**Exercice 9** Mettez en commentaire la ligne avec la tâche "ivy :cachepath" : vous obtenez une erreur sur la variable "lib.path.id" sur la la ligne de compilation javac. Supprimer l'attribut classpathref. Reexécutez la tâche "go" : vous obtenez une erreur dans vos sources. Pourquoi ? A quoi sert le tag "ivy :cachepath" ?

**Exercice 10** Dans cet exercice nous allons remplacer la dépendance "ivy :cachepath" par un fichier Ivy externe. Restaurez l'attribut "classpathref" précédemment supprimé. Créez un fichier ivy.xml à la racine de votre projet, qui déclare une dépendance à opencsv, et contient :

```
<ivy-module version="2.0">
  <info organisation="apache" module="hello-ivy"/>
  <dependencies>

    <dependency org="com.opencsv" name="opencsv" rev="3.9">
```

1  
2  
3  
4  
5

```

    <artifact name="opencsv" type="jar" />
  </dependency>

</dependencies>
</ivy-module>

```

6  
7  
8  
9  
10

Pour faire le lien entre build.xml et ivy.xml, créez la tâche suivante dont vous ferez dépendre la tâche "go" :

```

<target name="resolve" description="retrieve dependencies with ivy">
  <ivy :resolve /> <!-- résoudre les dépendances déclarées dans ivy.xml -->
  <ivy :cachepath pathid="lib.path.id" /> <!-- définir la variable lib.path.id -->
</target>

```

1  
2  
3  
4

**Exercice 11** Ajoutez à ce fichier ivy.xml la dépendance "commons-lang" que l'on a mis en commentaire dans build.xml. Recompilez votre projet : vous devez réussir à compiler et exécuter à nouveau votre projet. Combien Ant a-t-il téléchargé d'artefacts ? Que contient votre variable lib.path.id désormais ?

**Exercice 12** Ajoutez à votre dépôt git ivy.xml et les fichiers modifiés. Faites un push.

## 2 OpenCSV

Nous proposons de lire un fichier csv contenant des nombres. Nous allons chercher le max des nombres de ce fichier. Pour ne pas faire du parsing inutile nous allons utiliser une bibliothèque qui gère les csv et c'est Ivy qui va se charger de l'intégration.

**Exercice 13** Créez un fichier data.csv et mettez le à la racine de votre projet.

**Exercice 14** Désactivez la tâche "generate-src". En effet, il nous faut modifier les sources de votre programme Java. Ajoutez le code suivant à votre main qui va permettre de lire les données depuis votre fichier data.csv. Parcourez les valeurs de myEntries pour les afficher.

```

int monmax = 0;
CSVReader reader = new CSVReader(new FileReader("data.csv"));
List<String[]> myEntries = reader.readAll();
// parcourt
// System.out.println("Nombre lu : " + nb);

```

1  
2  
3  
4  
5

**Exercice 15** Affichez le max en fin d'algorithme.

**Exercice 16** Faites un push de votre code.

## 3 Documentation

**Exercice 17** Créez une tâche "report", qui dépend de la tâche "go". Dans cette tâche, créez une sous tâche qui crée le répertoire "report", puis ajoutez la sous tâche permettant de créer un rapport :

```

<ivy :report todir="report" />

```

1

Qu'obtenez-vous dans le répertoire report ?

**Exercice 18** Customisez le tag info de ivy.xml en utilisant les tags suivants :

```

<info organisation="apache"
  module="info-all" branch="trunk"
  revision="1.0" status="release" />

```

1  
2  
3

**Exercice 19** Regénérez la doc. Réalisez un screenshot du début de la doc que vous ajouterez à votre compte rendu.