
LOG240 - Rapport de Lab 1

Asma Alibert
Quentin Marques

Table des matières

| | |
|--|----|
| 1. Connexion à la machine virtuelle VMware | 2 |
| 2. Gestionnaire de sources | 3 |
| 2.1. Configuration de SVN | 3 |
| 2.2. Import/Export du code source vers SVN | 4 |
| 3. Système de suivi de projet | 7 |
| 3.1. Configuration de Trac | 7 |
| 3.2. Gestion des permissions de Trac | 7 |
| 3.3. Fonctionnalités de Trac | 9 |
| 3.3.1. Tickets | 9 |
| 3.3.2. Composants | 11 |
| 3.3.3. Milestones | 11 |
| 3.3.4. Versions | 11 |
| 4. Gestionnaire de projet | 11 |
| 4.1. Configuration Maven | 11 |

1. Connexion à la machine virtuelle VMware

On se connecte à la machine virtuelle VMware pour s'assurer de son bon fonctionnement. Le laboratoire est réalisé sous environnement Windows et nous utilisons le terminal PuTTY pour nous connecter avec les paramètres suivants:

adresse IP: Log240-20133-14.logti.etsmtl.ca

port: 22

protocole: SSH

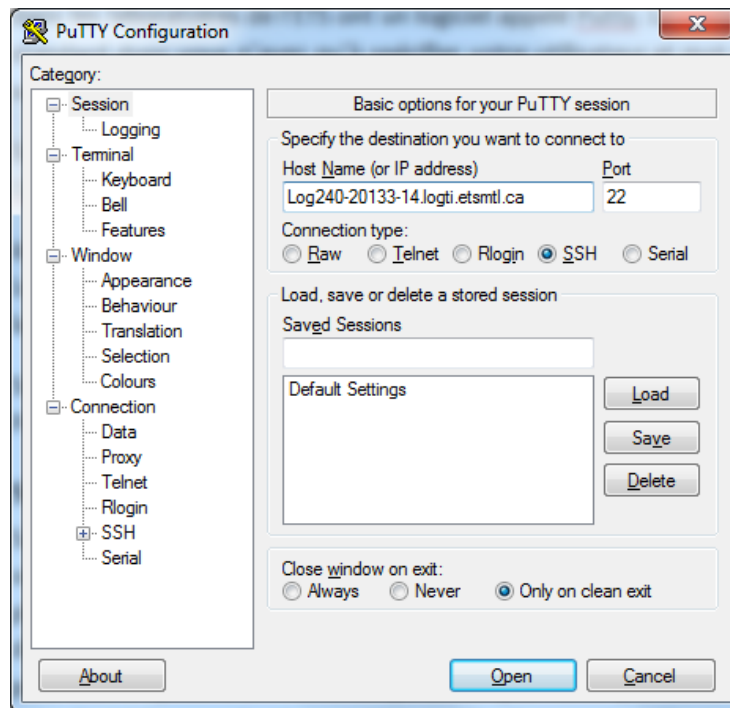


Figure 1. Connexion SSH au serveur Ubuntu en utilisant PuTTY

Un terminal s'ouvre alors et nous propose de nous connecter. Nous utilisons les identifiants suivants:

login: system

password: SystemPass14



Note

Sur un terminal UNIX, les mots de passe ne s'affichent pas lors de la saisie par soucis de sécurité, c'est pourquoi la sortie console suivante en est dépourvue.

```
login as: system
system@Log240-20133-14.logti.etsmtl.ca's password:
Linux log240-20133-14 2.6.32-51-generic-pae
#113-Ubuntu SMP Wed Aug 21 20:02:16 UTC 2013 i686 GNU/Linux
Ubuntu 10.04.4 LTS
```

```
Welcome to Ubuntu!
* Documentation:  https://help.ubuntu.com/

System information as of Mon Sep 16 15:48:01 EDT 2013

System load: 0.53           Memory usage: 18%    Processes:          75
Usage of /:  76.1% of 4.92GB Swap usage:   0%      Users logged in:  0

Graph this data and manage this system at https://landscape.canonical.com/

6 packages can be updated.
3 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Sep 16 15:45:09 2013 from logti-a3324-19.ens.ad.etsmtl.ca
system@log240-20133-14:~$
```



Note

Par soucis de clarté, l'invite de commande du shell (prompt) sera abrégée \$ par la suite.

On constate que la machine virtuelle est correctement lancée et en état de fonctionner.

2. Gestionnaire de sources

2.1. Configuration de SVN

Pour assurer le suivi du code source, nous utilisons le gestionnaire de sources Subversion (SVN). Nous commençons par changer les droits sur le dossier /opt qui sera notre dossier de travail pour lancer SVN en tant que daemon.

```
$ cd /opt
$ sudo /bin/chown -R www-data:svnusers svn
[sudo] password for system:
$ sudo /bin/chmod -R u+wx,g+wx,o-wrx svn
$ sudo /usr/bin/svnserve -d
```

On ajoute ensuite les différents utilisateurs:

client: L'utilisateur que notre client utilisera pour accéder au code

```
$ sudo /usr/sbin/useradd client -G svnusers
-c "Utilisateur client (BudgetPersonnel2000)" -m
$ sudo /usr/bin/passwd client
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

asma: L'utilisateur du développeur 1

```
$ sudo /usr/sbin/useradd asma -G svnusers  
-c "Utilisateur developpeur (MaintenancePlus)" -m  
$ sudo /usr/bin/passwd asma  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```

quentin: L'utilisateur du développeur 2

```
$ sudo /usr/sbin/useradd quentin -G svnusers  
-c "Utilisateur developpeur (MaintenancePlus)" -m  
$ sudo /usr/bin/passwd quentin  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```



Important

Tout les mots de passes utilisés sont identiques aux noms des utilisateurs. Il est évident que ceci ne constitue pas une bonne pratique de sécurité mais, dans un soucis de simplification du rapport, nous avons décidé de faire ainsi.



Astuce

Nous pouvons vérifier que ces utilisateurs ont bien été créés sur la VMware de la manière suivante:

```
$ cat /etc/passwd | grep "Utilisateur" | cut -d: -f1  
client  
asma  
quentin
```

2.2. Import/Export du code source vers SVN

Nous envoyons le code source de FinanceJ sur le dépôt SVN nouvellement configuré. Nous utilisons le logiciel TortoiseSVN pour réaliser cette opération. Dans un premier temps, nous créons un dossier Code contenant les dossiers branches, tags et trunk. Nous avons ajouté à ce dernier dossier les codes sources du projet FinanceJ.



Attention

Le vocabulaire de TortoiseSVN diffère de celui couramment utilisé: L'importation SVN est un envoi depuis un poste local vers le serveur et inversement concernant l'exportation. Afin de lever l'ambiguïté, nous utiliserons les termes classiques *import/export*.

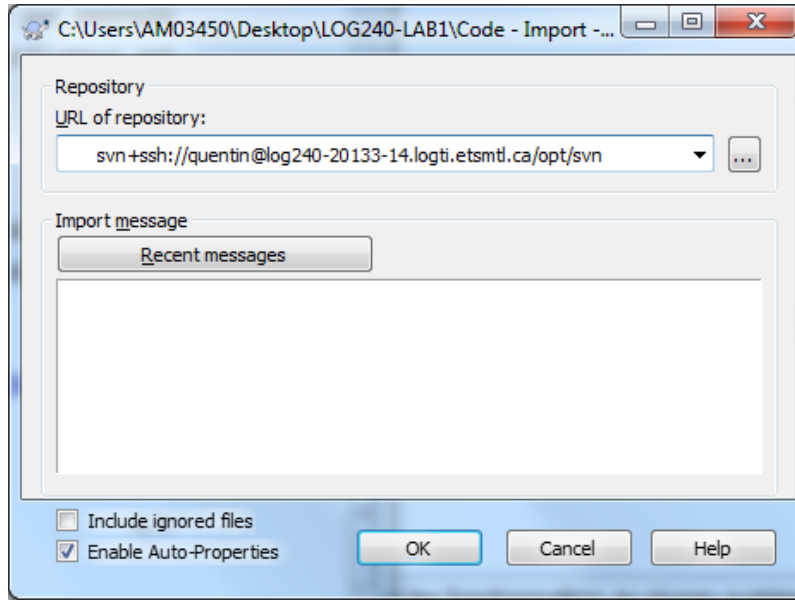


Figure 2. Exportation du dépôt en utilisant TortoiseSVN

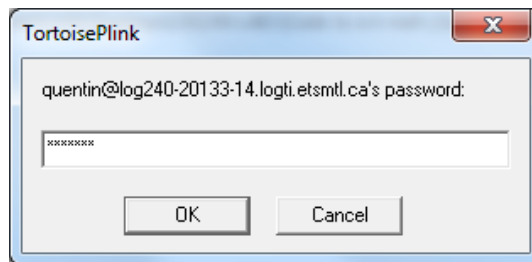


Figure 3. Confirmation du mot de passe SVN

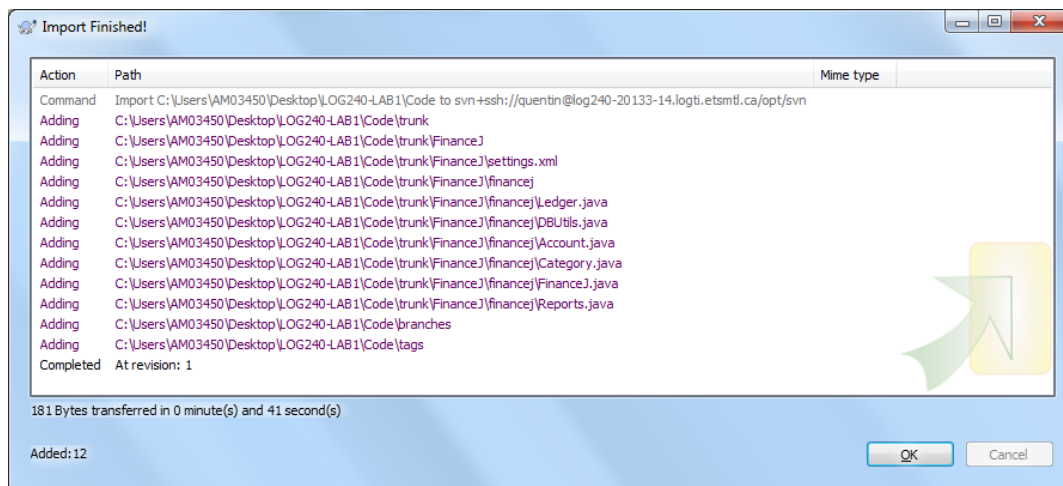


Figure 4. Résultat de l'exportation

Nous vérifions ensuite que l'envoi a correctement fonctionné en essayant d'importer le code envoyé dans un second dossier en local.



Note

Nous aurions également pu effectuer un **ls** directement sur le serveur via PuTTY mais des restrictions sur les VMwares nous en empêchent.

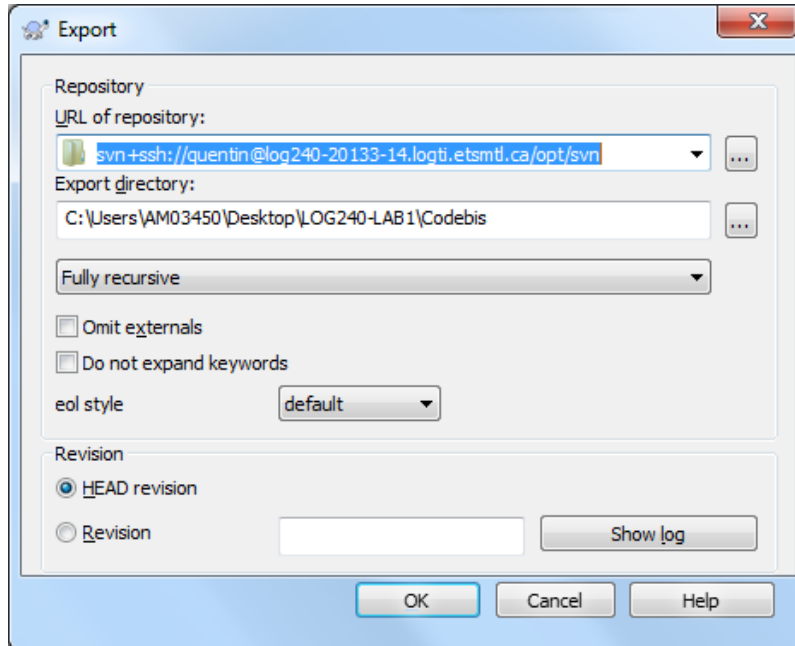


Figure 5. Vérification de l'importation dans le dossier Codebis

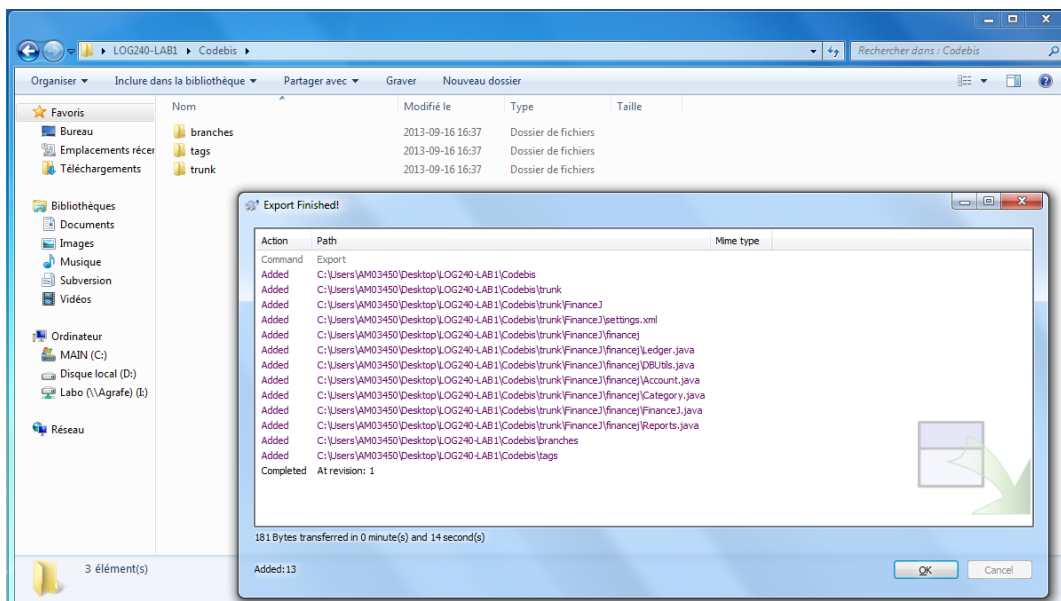


Figure 6. Succès du test d'importation dans le dossier Codebis

3. Système de suivi de projet

3.1. Configuration de Trac

Nous avons utilisé Trac, un système minimaliste de bug tracking et de wiki permettant la suivi de projet de développement. Tout d'abord, nous avons vérifié que Trac était correctement installé avec la bonne version:

```
$ tracd --version
tracd 0.11.7
```

Nous créons ensuite les 3 utilisateurs Trac de la manière suivante:

```
$ cd trac/conf/
$ sudo htpasswd trac.htpasswd client
New password:
Re-type new password:
Adding password for user client
```

```
$ sudo htpasswd trac.htpasswd asma
New password:
Re-type new password:
Adding password for user asma
```

```
$ sudo htpasswd trac.htpasswd quentin
New password:
Re-type new password:
Adding password for user quentin
```



Astuce

Nous pouvons vérifier que les utilisateurs ont correctement été rajoutés de la manière suivante:

```
$ cat trac.htpasswd | cut -d: -f1
admin
asma
quentin
client
```

3.2. Gestion des permissions de Trac

Nous nous sommes ensuite connectés à l'interface web de Trac. Nous nous sommes connectés en tant qu'administrateur avec les identifiants par défaut suivants:

login: admin

password: admin

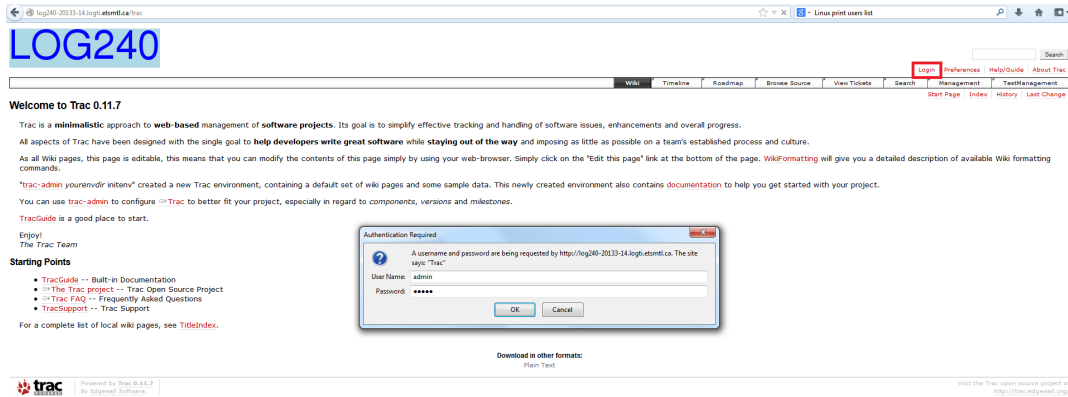


Figure 7. Connexion à l'interface web de Trac en admin

Nous avons ensuite défini les permissions de chaque utilisateur de Trac en suivant les contraintes suivantes:

- Il n'y a qu'un seul développeur ayant les droits d'administrateur (asma)
- Les développeurs doivent avoir accès à la plupart des fonctionnalités exceptés les tâches d'administration
- Le client peut uniquement consulter Trac et créer des tickets pour remonter des anomalies aux développeurs



Note

Afin d'éviter de redéfinir chaque permission pour chaque nouvel utilisateur éventuel, nous avons décidé de créer deux groupes de permissions:

- developpers
- clients

Ceci nous permet d'ajouter simplement des développeurs ou des comptes clients tout en conservant des permissions cohérentes.

Manage Permissions

| Subject | Action | | | | | |
|---------------|--|--|---------------------------------------|--|---|--|
| admin | <input checked="" type="checkbox"/> TRAC_ADMIN | | | | | |
| anonymous | <input type="checkbox"/> BROWSER_VIEW | <input type="checkbox"/> CHANGESSET_VIEW | <input type="checkbox"/> FILE_VIEW | <input type="checkbox"/> LOG_VIEW | <input type="checkbox"/> MILESTONE_VIEW | |
| | <input type="checkbox"/> REPORT_SQL_VIEW | <input type="checkbox"/> REPORT_VIEW | <input type="checkbox"/> ROADMAP_VIEW | <input type="checkbox"/> SEARCH_VIEW | <input type="checkbox"/> TICKET_VIEW | |
| | <input type="checkbox"/> TIMELINE_VIEW | <input type="checkbox"/> WIKI_VIEW | | | | |
| asma | <input checked="" type="checkbox"/> admin | | | | | |
| authenticated | <input type="checkbox"/> TICKET_CREATE | <input type="checkbox"/> TICKET_MODIFY | <input type="checkbox"/> WIKI_CREATE | <input type="checkbox"/> WIKI_MODIFY | | |
| client | <input type="checkbox"/> clients | | | | | |
| clients | <input type="checkbox"/> CHANGESSET_VIEW | <input type="checkbox"/> MILESTONE_VIEW | <input type="checkbox"/> REPORT_VIEW | <input type="checkbox"/> ROADMAP_VIEW | <input type="checkbox"/> SEARCH_VIEW | |
| | <input type="checkbox"/> TICKET_CREATE | <input type="checkbox"/> TICKET_VIEW | | | | |
| developers | <input type="checkbox"/> BROWSER_VIEW | <input type="checkbox"/> CHANGESSET_VIEW | <input type="checkbox"/> FILE_VIEW | <input type="checkbox"/> LOG_VIEW | <input type="checkbox"/> MILESTONE_VIEW | |
| | <input type="checkbox"/> REPORT_VIEW | <input type="checkbox"/> ROADMAP_VIEW | <input type="checkbox"/> SEARCH_VIEW | <input type="checkbox"/> TICKET_CREATE | <input type="checkbox"/> TICKET_MODIFY | |
| | <input type="checkbox"/> TICKET_VIEW | <input type="checkbox"/> TIMELINE_VIEW | <input type="checkbox"/> WIKI_ADMIN | | | |
| quentin | <input type="checkbox"/> developers | | | | | |

Remove selected items

Figure 8. Modifications des permissions clients et développeurs

3.3. Fonctionnalités de Trac

3.3.1. Tickets

Les tickets dans Trac peuvent être utilisés pour assigner des tâches de projet, pour faire des requêtes, des reports de bugs et du support logiciel. Ici nous avons créé 2 requêtes / tickets / tâches. À l'aide de Trac, nous avons la possibilité de:

- Définir la priorité de la tâche
- Préciser le composant affecté
- Estimer le nombre d'heures pour la tâche
- Préciser la version
- Ajouter des commentaires
- Joindre des fichiers
- etc.

Ticket #1 (new task)

| | | | |
|----------------------------|-------|-----------------------------|---------|
| Vérifier le code | | Opened 0 seconds ago | |
| Reported by: | asma | Owned by: | quentin |
| Priority: | major | Milestone: | |
| Component: | Code | Version: | Lab1 |
| Keywords: | | Cc: | |
| Estimated Number of Hours: | 0.0 | | |
| Billable?: | yes | Total Hours: | 0 |
| Description | | | |

Attachments

Attach file

Add/Change #1 (Vérifier le code)Comment (you may use [WikiFormatting](#) here):

B *I* A      

Add Hours to Ticket:

Show stopwatch

Change Properties

Summary:

Ticket #2 (new task)

| | | | |
|---|----------|-----------------------------|---------|
| S'assurer que le site web se lance | | Opened 0 seconds ago | |
| Reported by: | asma | Owned by: | quentin |
| Priority: | major | Milestone: | |
| Component: | Site Web | Version: | Lab1 |
| Keywords: | | Cc: | |
| Estimated Number of Hours: | 0.0 | | |
| Billable?: | yes | Total Hours: | 0 |
| Description | | | |

Attachments

Attach file

Add/Change #2 (S'assurer que le site web se lance)Comment (you may use [WikiFormatting](#) here):

B *I* A      

Figure 9. Création de tickets Trac

3.3.2. Composants

Les composants sont les systèmes/modules qui peuvent être affectés par des problèmes et donc être concernés par les tickets. Nous avons créé 2 composants:

- code
- site web

Figure 10. Création de composants Trac

3.3.3. Milestones

Les milestones sont les étapes importantes de la maintenance du projet.

Figure 11. Création de milestones Trac

3.3.4. Versions

Nous considérons que les versions représenteront les 5 laboratoires

Figure 12. Création de versions Trac

4. Gestionnaire de projet

4.1. Configuration Maven

Nous avons déplacé le fichier `settings.xml` fourni dans le répertoire `C:\Documents and Settings\am03450\.m2\`. Nous avons configuré le fichier `pom.xml` puis nous l'avons exécuté. Maven a alors compilé et exécuté le projet. Maven pouvant générer un site web du projet à partir des sources, nous avons configuré le fichier `pom.xml` pour ajouter un plugin permettant la génération de rapports. Maven permet également de générer des rapports d'analyse de la qualité du code source:

- PMD qui analyse la qualité du code source Java
- CheckStyle permet de vérifier le style du code source
- QALab permet de récupérer les rapports de PMD et CheckStyle au cours du temps

Nous avons ensuite configuré Maven afin qu'il puisse ajouter au site web du projet les informations provenant de Trac et SVN. Pour faire cela, nous avons ajouté des plugins au fichier `settings.xml` ainsi qu'au `pom.xml`. Les rapports produits par QALab sont générés sur le site web du projet par Maven dans la section "Projects Reports".

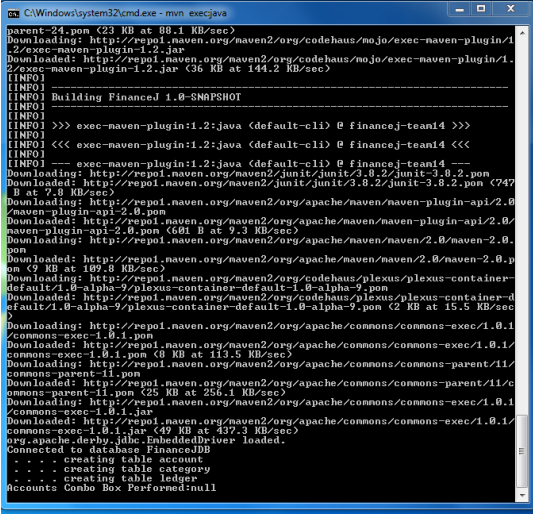


Figure 13. Compilation du projet FinanceJ par Maven

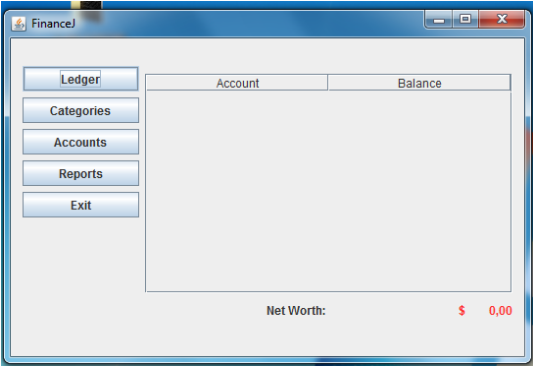


Figure 14. Code compilé et exécuté par Maven

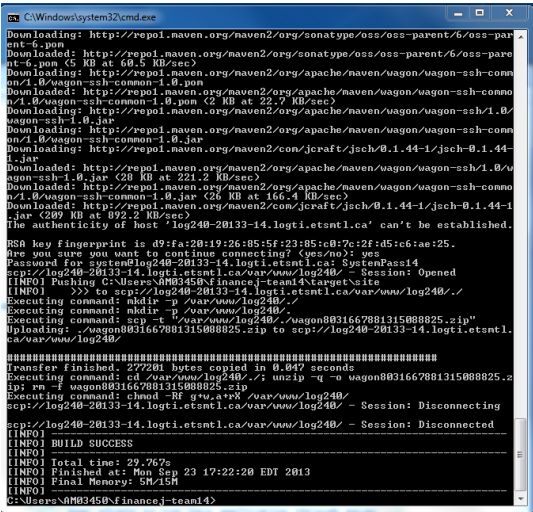


Figure 15. Compilation du site web par Maven

The screenshot shows a web browser displaying the 'Project Information' page for 'FinanceJ'. The page has a sidebar on the left with a 'Project Documentation' menu containing links like 'About', 'Dependencies', 'Plugin Management', 'Project Team', 'Project Summary', 'Issue Tracking', and 'Source Repository'. The main content area is titled 'Project Information' and includes an 'Overview' section with a table of documents and their descriptions.

| Document | Description |
|-------------------|--|
| About | FinanceJ is a personal finance Java application. A simple interface allows the user to track income/expenses vs budget. No accounting experience required. |
| Dependencies | This document lists the project's dependencies and provides information on each dependency. |
| Plugin Management | This document lists the plugins that are defined through pluginManagement. |
| Project Team | This document provides information on the members of this project. These are the individuals who have contributed to the project in one form or another. |
| Project Plugins | This document lists the build plugins and the report plugins used by this project. |
| Project Summary | This document lists other related information of this project. |
| Issue Tracking | This is a link to the issue management system for this project. Issues (bugs, features, change requests) can be created and queried using this link. |
| Source Repository | This is a link to the online source repository that can be viewed via a web browser. |

Copyright © 2013 #78 - LOG240 - Team 14. All Rights Reserved.

Figure 16. Site web générés par Maven

The screenshot shows a web browser displaying the 'Generated Reports' page for 'FinanceJ'. The sidebar on the left has a 'Project Reports' menu with links like 'Test JavaDocs', 'JavaDocs', 'Checkstyle', 'Trac Report', 'PMD Report', 'QALab Main Report', and 'QALab Movers Report'. The main content area is titled 'Generated Reports' and includes an 'Overview' section with a table of documents and their descriptions.

| Document | Description |
|---------------------|-------------------------------------|
| Test JavaDocs | Test JavaDoc API documentation. |
| JavaDocs | JavaDoc API documentation. |
| Checkstyle | Report on coding style conventions. |
| Trac Report | Report on Ticket from the Trac. |
| PMD Report | Verification of coding rules. |
| QALab Main Report | QALab Report. |
| QALab Movers Report | The QALab Movers Report. |

Copyright © 2013 #78 - LOG240 - Team 14. All Rights Reserved.

Figure 17. Rapports générés disponibles sur le site web