Projet OTI

CHENG Xiaojun
JIN Benli

*** 1/ Introduction

L'objective de ce projet est d'utiliser QUnit, Selenium et Sonar pour tester et de surveiller la qualité de code.

*** 2/ Architecture de projet

|____assets : les souce code de projet webAvance
|____index.html : le point entée de notre application de webAvance
|____jsTestDriver.conf : configuration pour jsTestDriver

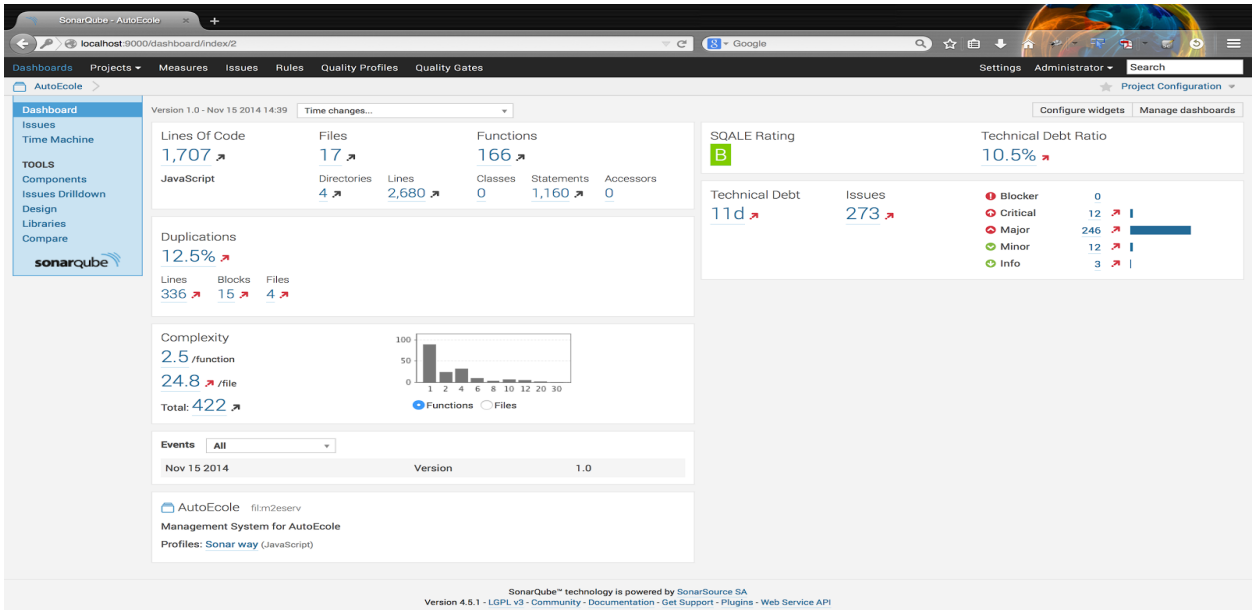
|____qunit
| |____model : les tests en QUnit
| | |____database : les tests pour les obejct DAO
| | |____object: les test pour les objects d'entity

|____seleinum-webdriver-java-projet : le test sur seleinum en java
| |____javadoc_seleinum : le javadoc y compris toutes les document pour seleinum
| |____selenium
| | |____src : le code source de selenium webdriver en java

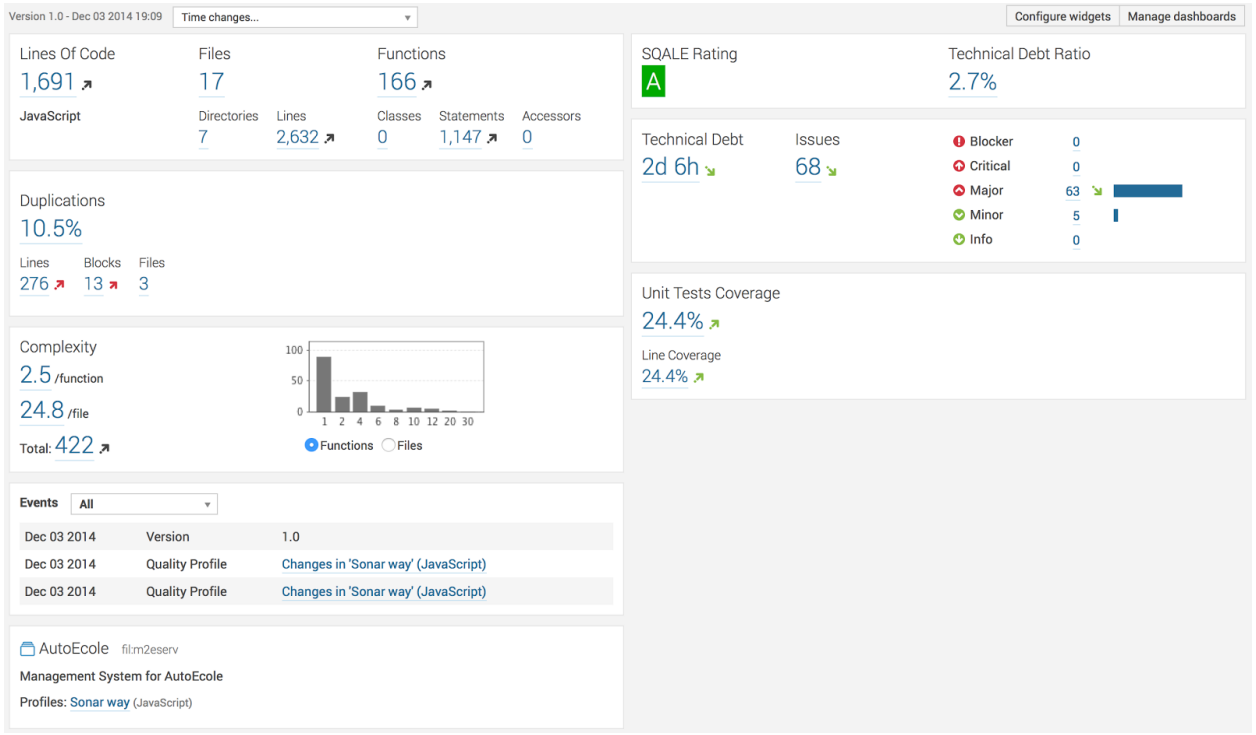
|____seleinumIDE : les scripts pour seleiumIDE
| |____student_login_test
| |____teacher_login_add_delete_class
| |____teacher_login_otherfunctions

*** 3/ Sonar qualité de code

Voici la capture d'écran comme qu'on a lancé l'analyse de projet sur Sonar la première fois:



Finalement, voici l'état de notre projet après on ajoute les tests et optimise:



*** 4/ Coverage et QUnit

Comme nous avons crée les tests que pour le couche modèle, donc on s'intéresse aux le coverage pour le répertoire model

Par rapport que les entities comme nous avons créé pour l'application, les test de QUnit concernent sur les aspects suivants:

1. La construction d'entity
2. Le rechargement sur LocalStorage

Pour les objects DAO nous donnent l'accès à manipuler avec les entites, donc les tests concernent sur les aspects suivants:

1. La construction d'objet
2. Le rechargement sur LocalStroage
3. Les opérations CRUD

*** 5/ Seleinum-webdriver

Nous avons utilisé Seleinum-webdriver pour tester en java. vous trouvez toutes les documentations dans répartoire :

/seleinum-webdriver-java-projet/javadoc_seleinum/index.html

pour tester, il faut ajouter les bibliothèques nécessites (comme **selenium-java-2.43.1.jar** et les **libs**), et également changer les path pour index.html en changant le ligne :

driver.get(getURLString()) dans chaque .java;

Liste tests réalisé :

- Login (seleniumTest.java)
    - test login as client
    - test login as secret
    - test login error label display
- Secretary (packageSecretary)
    - TestSecretaryAddTeacher.java
        - test add a teacher
    - TestSecretaryStudentInformationPage.java
        - test display student information page
        - test display student information : search and display a student lesson list
    - TestSecretaryHomePage.java

            - test home page add a class
            - test home page add a lecture class : with choose in radiobox
            - test home page delete a class
            - test home page display data picker
            - test home page open and close popover : for add/delete/edit a class to a student
    - Student(package Student)
        - TestStudentClassTale.java
            - test student information table
            - test student table display color : we use different to know whether a class to already passed or not

*** 6/ Seleinum-IDE

Nous avons mis toutes les scrpits pour seleinumIDE dans pépartoire : /seleinumIDE, pour tester il fault change le configuration de PATH de index.html

- student_login_test : student login and student fonction
- teacher_login_add_delete_class : moniter login and add and detele a class
- teacher_login_otherfunctions : moniter login and other functions