



Définition: Procédure permettant de **vérifier** que chaque fonction est correctement implémentée. On vérifie chaque fonction **indépendamment** les unes des autres.

Annotation @Test

Méthode de Test de  
la Story Homepage

```
// Test de la Story #1-homepage (https://trello.com/c/WKTneu9o/1-homepage)
@Test
public void testHomepage() throws Exception {
    driver.get("https://www.meetup.com/fr-FR/");
    // Vérification du critère 1: le titre doit être Partagez vos passions | Meetup
    assertEquals(driver.getTitle(), "Partagez vos passions | Meetup");
    // TODO
    // To Be Completed By Coders From Coding Factory
}
```

Assertion du titre de la  
page

Test fonctionnel du 1<sup>er</sup>  
Critère d'Acceptation

# Test Fonctionnel

Ecrit par le Développeur/Testeur

Moyen de vérifier la conformité aux  
« *Spécifications* »

En Agile, ces tests visent plutôt à valider les  
critères d'acceptation d'une User Story

Tests orientés couche métier Backend

# Exemple: Story Fiche Offre d'emploi

Un test fonctionnel testerait que la méthode qui gère la fiche d'une offre renvoi toutes les données demandées en backend.

Ne teste donc pas le frontend.

# Test Fonctionnel d'Interface

Même objectif que le test fonctionnel côté backend mais teste également le frontend.

Consiste à piloter les interfaces pour vérifier la présence des bonnes données (fond) et leur rendu (forme).

# Exemple: Story Fiche Offre d'emploi

On testerait que la méthode qui gère la fiche d'une offre renvoi toutes les données demandées en backend.

Puis on testerait le rendu des données sur la page de la fiche de l'offre d'emploi.





# Selenium, quel intérêt ?

Outil n°1 de pilotage de navigateurs web

Existe pour pratiquement tous les langages (java, php, ruby, js, ...)

Méthodes transposables dans d'autres langages



# Pilotage de site web

Utilisation du site Meetup.com

Base pour commencer à tester  
fonctionnellement des pages web

✦ Vous épargnez des heures de configs

Code sur mon Github (adesousa)

# Liens pratiques

Doc Selenium

[http://www.seleniumhq.org/docs/03\\_webdriver.jsp#selenium-webdriver-api-commands-and-operations](http://www.seleniumhq.org/docs/03_webdriver.jsp#selenium-webdriver-api-commands-and-operations)

Doc Sélecteurs CSS

[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

Mon GitHub

<https://github.com/adesousa>



*## Download Selenium webdriver & Selenium ChromeDriver*

```
cd /Library/Java/JUNIT/
```

```
mkdir selenium
```

```
sudo wget http://selenium-release.storage.googleapis.  
com/3.8/selenium-java-3.8.1.zip
```

```
sudo unzip selenium-java-3.8.1.zip -d selenium
```

```
sudo chmod 755 selenium/libs
```

```
sudo wget https://chromedriver.storage.googleapis.  
com/2.34/chromedriver_mac64.zip
```

```
sudo unzip chromedriver_mac64.zip
```

```
sudo rm chromedriver_mac64.zip
```

```
mv selenium/client-combined*.jar .
```

```
mv selenium/libs/* .
```

```
sudo rm -r selenium-java-3.8.1.zip selenium/
```

```
sudo chown $USER:$USER /Library/Java/JUNIT/*
```

```
sudo chmod 755 /Library/Java/JUNIT/*
```

*## Selenium configuration for Junit*

vim ~/.bash\_profile

*# delete this line at the end of file*

export CLASSPATH=/Library/Java/JUNIT/junit-4.12.jar:/  
Library/Java/JUNIT/hamcrest-all-1.3.jar:.

*# Put this line at the end of file*

export CLASSPATH=/Library/Java/JUNIT/junit-4.12.jar:/  
Library/Java/JUNIT/hamcrest-all-1.3.jar:/Library/  
Java/JUNIT/chromedriver:/Library/Java/JUNIT/client-  
combined-3.8.1.jar:/Library/Java/JUNIT/client-  
combined-3.8.1-sources.jar:/Library/Java/JUNIT/  
selenium-server-standalone-3.8.1.jar:.

*# Save and Quit VIM*

*# Refresh Bash Conf*

source ~/.bash\_profile

```
## First of all, you need to compile your test class (  
from src folder)
```

```
# Important, to avoid compile errors  
cd src/
```

```
### Classes
```

```
javac -d ../bin/ test/functional/FunctionalTest.java
```



```
## Then run your test cases. (from bin tests folder)
cd bin/
java org.junit.runner.JUnitCore test.functional.
    FunctionalTest
```





# Testez <<Meetup.com>>

Consultez le Product Backlog :

<https://trello.com/b/CNZPMXNS/functional-testing-agile-testing-course>

Appropriiez-vous les critères à tester.

Code de test sur mon github

# Objectif

Couverture fonctionnelle à 100%

Chaque critère d'acceptation doit être testée par un test fonctionnel



