



**eBook Gratuit**

# APPRENEZ selenium

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

**#selenium**

# Table des matières

À propos.....	1
<b>Chapitre 1: Démarrer avec le sélénium.....</b>	<b>2</b>
Remarques.....	2
Versions.....	2
Exemples.....	3
Test simple de sélénium en Java.....	3
Test de sélénium simple en python.....	4
Configuration de python Selenium via le terminal (BASH).....	4
Exemple simple de sélénium en C #.....	5
<b>Chapitre 2: Accepter les alertes popup avec Selenium.....</b>	<b>7</b>
Exemples.....	7
Exemple Python d'alerte d'acceptation.....	7
Extensions C # vers WebDriver.....	7
Java.....	7
<b>Chapitre 3: Automatisation des applications mobiles.....</b>	<b>9</b>
Exemples.....	9
Android + Chrome + Python.....	9
Python + Chrome + Android.....	9
<b>Chapitre 4: Démarrer avec Selenium en python.....</b>	<b>10</b>
Remarques.....	10
Exemples.....	11
Selenium de base en python.....	11
Test de base du sélénium.....	11
<b>Chapitre 5: En attente dans Selenium.....</b>	<b>12</b>
Introduction.....	12
Exemples.....	12
Attente explicite en Python.....	12
Attendez à Java avec du sélénium.....	13
<b>Chapitre 6: Premier projet en sélénium avec Java.....</b>	<b>14</b>
Introduction.....	14

Exemples.....	14
Configurer IntelliJ Idea pour Selenium.....	14
Configurer ChromeDriver.....	16
Ouvrir un site Web en utilisant Selenium.....	17
Obtenir des éléments dans Selenium.....	17
Exemple de travail dans le sélénium.....	17
Obtenir les attributs des WebElements dans Selenium.....	18
<b>Chapitre 7: Prenez une capture d'écran d'une page Web.....</b>	<b>19</b>
Exemples.....	19
Capture d'écran Python Selenium Take / Save de la page Web.....	19
Extension C # TakeScreenshot.....	19
Capture d'écran de sélection / enregistrement Java de la page Web et ajout d'un rapport.....	19
<b>Chapitre 8: Selenium IDE.....</b>	<b>21</b>
Exemples.....	21
Essayez un script Selenium simple: Rechercher Wikipedia sur Google.....	21
<b>Chapitre 9: Selenium simple exemple C # et Nunit.....</b>	<b>23</b>
Remarques.....	23
Exemples.....	23
Simple Selenium-NUnit.....	23
<b>Chapitre 10: Selenium simple exemple C # et Nunit.....</b>	<b>25</b>
Exemples.....	25
Test de chargement de page simple et assurez-vous que le titre de la page est correct.....	25
<b>Chapitre 11: Usine WebDriver.....</b>	<b>26</b>
Exemples.....	26
WebDriver Factory C #.....	26
<b>Chapitre 12: WebDriverManager pour Selenium - un outil très soigné de Boni Garcia.....</b>	<b>28</b>
Introduction.....	28
Exemples.....	28
Les exemples ci-dessous montrent à quel point il est facile à utiliser.....	28
<b>Crédits.....</b>	<b>29</b>

---

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [selenium](#)

It is an unofficial and free selenium ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official selenium.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Démarrer avec le sélénium

## Remarques

Selenium est une puissante bibliothèque de commandes en plusieurs langues (C #, Haskell, Java, JavaScript, Objective-C, Perl, PHP, Python, R et Ruby) qui permettent à un programmeur d'automatiser l'interaction du navigateur. C'est incroyablement utile pour les développeurs qui testent des applications.

Selenium propose des méthodes pour:

- Rechercher un élément dans une page Web
- Cliquez sur un élément
- Envoyer une chaîne à un élément
- Naviguer vers une page Web
- Passer à un autre onglet dans la même fenêtre du navigateur
- Prenez une capture d'écran d'une page Web

En utilisant ces méthodes, un développeur peut vérifier automatiquement les tests:

- Si un élément est dans une page et visible pour un utilisateur
- Un formulaire de recherche ou de connexion
- Boutons ou éléments interactifs
- Vérifier les valeurs ou les attributs d'un élément

Selenium s'exécute dans les Webdrivers, qui sont similaires à un navigateur Web normal mais permettent à Selenium d'interagir avec eux. Un test Selenium ouvre généralement une nouvelle instance de pilote, quel que soit le navigateur testé par le développeur, ce qui est toujours une ardoise propre. De cette façon, lors de l'exécution d'un test Selenium, le développeur n'a pas à s'inquiéter des cookies précédents, ni du cache du navigateur affectant les résultats de son application.

Selenium fonctionne également lors de l'exécution d'un pilote Web en mode sans tête.

## Versions

Version	Date de sortie
3.4.0	2017-04-11
3.3	2017-04-07
3.2	2017-02-27
3.1	2017-02-13
3.0.1	2016-11-19

Version	Date de sortie
3.0	2016-10-11

## Exemples

### Test simple de sélénium en Java

Le code ci-dessous est un programme Java simple utilisant du sélénium. Le voyage du code ci-dessous est

1. Ouvrez le navigateur Firefox
2. Ouvrir la page google
3. Imprimer le titre de la page Google
4. Trouver l'emplacement de la zone de recherche
5. Passez la valeur en tant que Selenium dans la zone de recherche
6. Soumettre le formulaire
7. Arrêter le navigateur

```
package org.openqa.selenium.example;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import java.util.concurrent.TimeUnit;

public class Selenium2Example {
    public static void main(String[] args) {
        // Create a new instance of the Firefox driver
        WebDriver driver = new FirefoxDriver();

        // An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time
        // when trying to find an element or elements if they are not immediately available.
        // The default setting is 0. Once set, the implicit wait is set for the life of the
        // WebDriver object instance.
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        // Maximize the browser window to fit into screen
        driver.manage().window().maximize();

        // Visit Google
        driver.get("http://www.google.com");

        // Check the title of the page
        System.out.println("Page title is: " + driver.getTitle());

        // Find the text input element by its name
        WebElement element = driver.findElement(By.name("q"));

        // Enter something to search for
        element.sendKeys("Selenium!");

        // Now submit the form. WebDriver will find the form for us from the element
        element.submit();
    }
}
```

```
        //Close the browser
        driver.quit();
    }
}
```

## Test de sélénium simple en python

```
from selenium import webdriver

# Create a new chromedriver
driver = webdriver.Chrome()

# Go to www.google.com
driver.get("https://www.google.com")

# Get the webelement of the text input box
search_box = driver.find_element_by_name("q")

# Send the string "Selenium!" to the input box
search_box.send_keys("Selenium!")

# Submit the input, which starts a search
search_box.submit()

# Wait to see the results of the search
time.sleep(5)

# Close the driver
driver.quit()
```

## Configuration de python Selenium via le terminal (BASH)

Le plus simple est d'utiliser [pip](#) et [VirtualEnv](#) . Le sélénium nécessite également [python 3.\\*](#) .

Installez virtualenv en utilisant:

```
$: pip install virtualenv
```

Créez / entrez un répertoire pour vos fichiers Selenium:

```
$: cd my_selenium_project
```

Créez un nouveau VirtualEnv dans le répertoire de vos fichiers Selenium:

```
$: virtualenv -p /usr/bin/python3.0 venv
```

Activer le VirtualEnv:

```
$: source venv/bin/active
```

Vous devriez voir maintenant voir (venv) au début de chaque ligne de bash. Installez Selenium en

utilisant pip:

```
$: pip install selenium
```

Selenium est livré avec le pilote FireFox par défaut.

Si vous souhaitez exécuter Selenium dans Google Chrome, procédez comme suit:

```
$: pip install chromedriver
```

Vous avez maintenant un VirtualEnv contrôlé par la version. Pour vous assurer que tout est correctement configuré:

Démarrer python:

```
$: python
```

Imprime:

```
Python 2.7.10 (default, Jul 14 2015, 19:46:27)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
```

Créez un nouveau webdriver (dans ce cas, un chromedriver) et accédez à [www.google.com](http://www.google.com):

```
>>> from selenium import webdriver
>>> driver = webdriver.Chrome()
>>> driver.get("https://www.google.com")
```

Fermez le pilote et l'interpréteur python:

```
>>> driver.quit()
>>> quit()
```

Désactiver le VirtualEnv:

```
$: deactivate
```

Si la ligne `driver = webdriver.Chrome()` erreurs:

- Assurez-vous que le navigateur chrome est également installé. Si vous ne le faites pas, le chromedriver Selenium ne peut pas accéder au fichier binaire Chrome.
- `webdriver.Chrome()` peut également prendre un paramètre pour votre emplacement chromedriver. Si vous l'avez installé avec pip, essayez (sur mac) `driver = webdriver.Chrome("./venv/selenium/webdriver/chromedriver")`.

## Exemple simple de sélénium en C #

```
//Create a new ChromeDriver
```



```
IWebDriver driver = new ChromeDriver();

//Navigate to www.google.com
driver.Navigate().GoToUrl("https://www.google.com");

//Find the WebElement of the search bar
WebElement element = driver.FindElement(By.Name("q"));

//Type Hello World into search bar
element.SendKeys("Hello World");

//Submit the input
element.Submit();

//Close the browser
driver.Quit();
```

Lire Démarrer avec le sélénium en ligne: <https://riptutorial.com/fr/selenium/topic/1840/demarrer-avec-le-selenium>

---

# Chapitre 2: Accepter les alertes popup avec Selenium

## Exemples

### Exemple Python d'alerte d'acceptation

```
from selenium import webdriver

# Create a new webdriver
driver = webdriver.Chrome()
# Get a page that has a popup window (Use mouse to click "try it" button)
driver.get("http://www.w3schools.com/js/tryit.asp?filename=tryjs_alert")
# Accept the opened alert
driver.switch_to.alert.accept()
```

### Extensions C # vers WebDriver

```
public static IWebDriver dismissAlert(this IWebDriver driver)
{
    try
    {
        IAlert alert = driver.SwitchTo().Alert();
        alert.Dismiss();
    }
    catch {}
    return driver;
}
public static IWebDriver acceptAlert(this IWebDriver driver)
{
    try
    {
        IAlert alert = driver.SwitchTo().Alert();
        alert.Accept();
    }
    catch { }
    return driver;
}
```

#### Comment utiliser:

```
driver.acceptAlert();
driver.dismissAlert();
```

## Java

#### Pour une alerte simple:

```
Alert simpleAlert = driver.switchTo().alert();
```

```
String alertText = simpleAlert.getText();
System.out.println("Alert text is " + alertText);
simpleAlert.accept();
```

Pour une alerte rapide:

```
Alert promptAlert = driver.switchTo().alert();
String alertText = promptAlert.getText();
System.out.println("Alert text is " + alertText);
//Send some text to the alert
promptAlert.sendKeys("Accepting the alert");
Thread.sleep(4000); //This sleep is not necessary, just for demonstration
promptAlert.accept();
```

Pour une alerte de confirmation:

```
Alert confirmationAlert = driver.switchTo().alert();
String alertText = confirmationAlert.getText();
System.out.println("Alert text is " + alertText);
confirmationAlert.accept();
```

Une autre façon de procéder est d'emballer votre code dans un try-catch:

```
try{
    // Your logic here.
} catch (UnhandledAlertException e){
    Alert alert = driver.switchTo().alert();
    alert.accept();
}

// Continue.
```

Lire [Accepter les alertes popup avec Selenium en ligne](https://riptutorial.com/fr/selenium/topic/6787/accepter-les-alertes-popup-avec-selenium):

<https://riptutorial.com/fr/selenium/topic/6787/accepter-les-alertes-popup-avec-selenium>

---

# Chapitre 3: Automatisation des applications mobiles

## Exemples

### Android + Chrome + Python

Pour pouvoir exécuter des tests, Chrome browser doit être préinstallé sur Android appareil Android ,

### Python + Chrome + Android

Pour être en mesure de travailler avec une application Web sur Android appareil Android utilisant Selenium ci-dessous, les conditions préalables doivent être remplies:

- [Android SDK](#) installé sur l'ordinateur
- [Navigateur Chrome](#) installé sur Android appareil Android
- [Mode de débogage](#) activé sur l'appareil Android

Démarrez le serveur adb et chromedriver avec les commandes ci-dessous depuis cmd / Terminal :

```
adb start-server
chromedriver
```

Notez le chromedriver port du serveur chromedriver du journal qui ressemble à

```
Démarrage de ChromeDriver 2.15.322448
(52179c1b310fec1797c81ea9a20326839860b7d3) sur le port 9515
```

Connecter Android appareil Android à un ordinateur avec USB câble USB

Vous trouverez ci-dessous un code Python simple pour obtenir la page Google :

```
from selenium import webdriver

capabilities = {
    'chromeOptions': {
        'androidPackage': 'com.android.chrome',
    }
}

driver = webdriver.Remote('http://localhost:9515', capabilities) # Specify your port number
value
driver.get('http://google.com')
driver.quit()
```

Lire Automatisation des applications mobiles en ligne:

<https://riptutorial.com/fr/selenium/topic/9449/automatisation-des-applications-mobiles>

---

# Chapitre 4: Démarrer avec Selenium en python

## Remarques

Qu'est-ce que le sélénium?

Selenium est une bibliothèque de commandes pour aider une interface de programmation avec un navigateur comme un véritable utilisateur.

Choses que Selenium fait:

Recherche d'éléments dans le fichier HTML d'une page Web

- Trouve un seul élément:

- `driver.find_element_by_css_selector("css.selector.of.element")` [Aide du sélecteur CSS](#)
- `driver.find_element_by_xpath("//xpath//of//element")` [Aide XPATH](#)
- `driver.find_element_by_name("name_of_element")`
- `driver.find_element_by_id("id_of_element")`
- `driver.find_element_by_partial_link_text("element_link_text")`
- `driver.find_element_by_class_name("class_name_of_element")`
- `driver.find_element_by_tag_name("tag_name_of_element")`

- Trouve une liste d'éléments:

- `driver.find_elements_by_css_selector("css.selector.of.elements")`
- `driver.find_elements_by_xpath("//xpath//of//elements")`
- `driver.find_elements_by_name("name_of_elements")`
- `driver.find_elements_by_partial_link_text("elements_link_text")`
- `driver.find_elements_by_class_name("class_name_of_elements")`
- `driver.find_elements_by_tag_name("tag_name_of_elements")`

- Documentation officielle: [sélénium-python lire les docs](#)

Interagir avec des éléments:

"method" représente l'une des méthodes ci-dessus pour trouver un élément ou une liste d'éléments.

- fonction de clic:

- `driver.find_element_by_method.click()`

- fonction `send_keys`:

- `driver.find_element_by_method.send_keys("text")` envoie la chaîne "text" à l'élément trouvé.
- `driver.find_element_by_method.send_keys(KeyCode.UP)` envoie le KeyCode de la flèche

vers le haut à l'élément trouvé.

## Exemples

### Selenium de base en python

```
from selenium import webdriver

driver = webdriver.Chrome() # Creates a new chromedriver instance
driver.get("https://www.python.org/") # Go to https://www.python.org/
# Sends the text "python" to the text search box
driver.find_element_by_id("id-search-field").send_keys("python")
# Click on the search button
driver.find_element_by_css_selector("button[type=\"submit\"]").click()
```

### Test de base du sélénium

Ceci est un exemple de base d'un test Selenium utilisant la bibliothèque python Unittest

```
from selenium import webdriver
import unittest

class SeleniumTest(unittest.TestCase):

    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(30)

    def test(self):
        self.driver.get("https://www.google.com")
        self.driver.find_element_by_id("lst-ib").send_keys("python")
        self.driver.find_element_by_css_selector("span[class=\"sbico\"]").click()

    def tearDown(self):
        self.driver.quit()
```

Lire Démarrer avec Selenium en python en ligne:

<https://riptutorial.com/fr/selenium/topic/6786/demarrer-avec-selenium-en-python>

---

# Chapitre 5: En attente dans Selenium

## Introduction

L'un des obstacles les plus courants pour les nouveaux utilisateurs de Selenium est d'attendre qu'une page soit complètement chargée. Les utilisateurs humains peuvent facilement savoir si une page est complètement chargée ou si elle est toujours en cours de chargement. Le sélénium, cependant, attend juste un certain temps. Par conséquent, il est souvent pratique d'avoir un bon moyen d'attendre des éléments dans une page. Bien qu'il soit *possible* de le faire avec une boucle et des fonctions `sleep ()`, il existe déjà des moyens beaucoup plus propres à Selenium.

## Exemples

### Attente explicite en Python

Lorsque le navigateur navigue sur une page dynamique (le plus souvent une application Web basée sur AJAX), le chargement des éléments sur la page peut prendre un temps différent, et de plus, certains éléments ne se chargent que pour répondre à certaines actions de l'utilisateur. Dans de tels cas, la récupération directe d'un élément peut échouer:

```
# Don't do this: may fail on dynamic page with ElementNotVisibleException
element = driver.find_element_by_name('q')
```

La solution la plus évidente semble introduire l'attente avant de récupérer des éléments:

```
# Don't do this: inefficient solution for ElementNotVisibleException
time.sleep(5) # delays for 5 seconds
element = driver.find_element_by_name('q')
```

Mais une telle solution est inefficace, car elle fait que le test attend toujours 5 secondes, même si dans la plupart des cas, l'élément apparaît après 1 seconde (et ne nécessite parfois que 5 secondes). Cela ne semble pas grand-chose si c'est un seul endroit, mais généralement chaque test traite de plusieurs éléments, et il existe plusieurs tests, ce qui correspond à la durée totale du test.

Une meilleure solution consiste à attendre que l'élément apparaisse pendant 5 secondes au maximum, mais revenez de l'attente dès que l'élément est trouvé. `WebDriverWait` vous permet de faire exactement cela.

L'exemple suivant navigue sur `www.google.com`, attend (pendant 5 secondes) le chargement de la barre de recherche, puis recherche "sélénium".

```
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
```

```

from selenium.webdriver.common.by import By

# Create a new chromedriver instance
driver = webdriver.Chrome()

# Go to www.google.com
driver.get("https://www.google.com")

try:
    # Wait as long as required, or maximum of 5 sec for element to appear
    # If successful, retrieves the element
    element = WebDriverWait(driver,5).until(
        EC.presence_of_element_located((By.NAME, "q")))

    # Type "selenium"
    element.send_keys("selenium")

    #Type Enter
    element.send_keys(Keys.ENTER)

except TimeoutException:
    print("Failed to load search bar at www.google.com")
finally:
    driver.quit()

```

## Attendez à Java avec du sélénium

Attente **explicite** : attendez qu'une certaine condition se produise avant de continuer dans le code.

```

WebDriver driver = new FirefoxDriver();
driver.get("http://google.com");
WebElement myElement = (new WebDriverWait(driver, 10))
    .until(ExpectedConditions.presenceOfElementLocated(By.id("myElement")));

```

Attente **implicite**: attendez un certain temps lorsque vous tentez de trouver un élément ou des éléments s'ils ne sont pas immédiatement disponibles.

```

WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.get("http://google.com");
WebElement myElement = driver.findElement(By.id("myElement"));

```

Lire En attente dans Selenium en ligne: <https://riptutorial.com/fr/selenium/topic/6901/en-attente-dans-selenium>



---

# Chapitre 6: Premier projet en sélénium avec Java

## Introduction

Ceci est une introduction à Selenium, en utilisant Java. Bien que nous ne nous attendions pas à ce que vous sachiez quoi que ce soit concernant Selenium, vous devez avoir des connaissances préalables sur Java pour suivre ce cours.

### Liens de téléchargement :

[Sélénium](#)

[IntelliJ IDEA](#)

[ChromeDriver](#)

[JDK 8](#)

## Exemples

### Configurer IntelliJ Idea pour Selenium

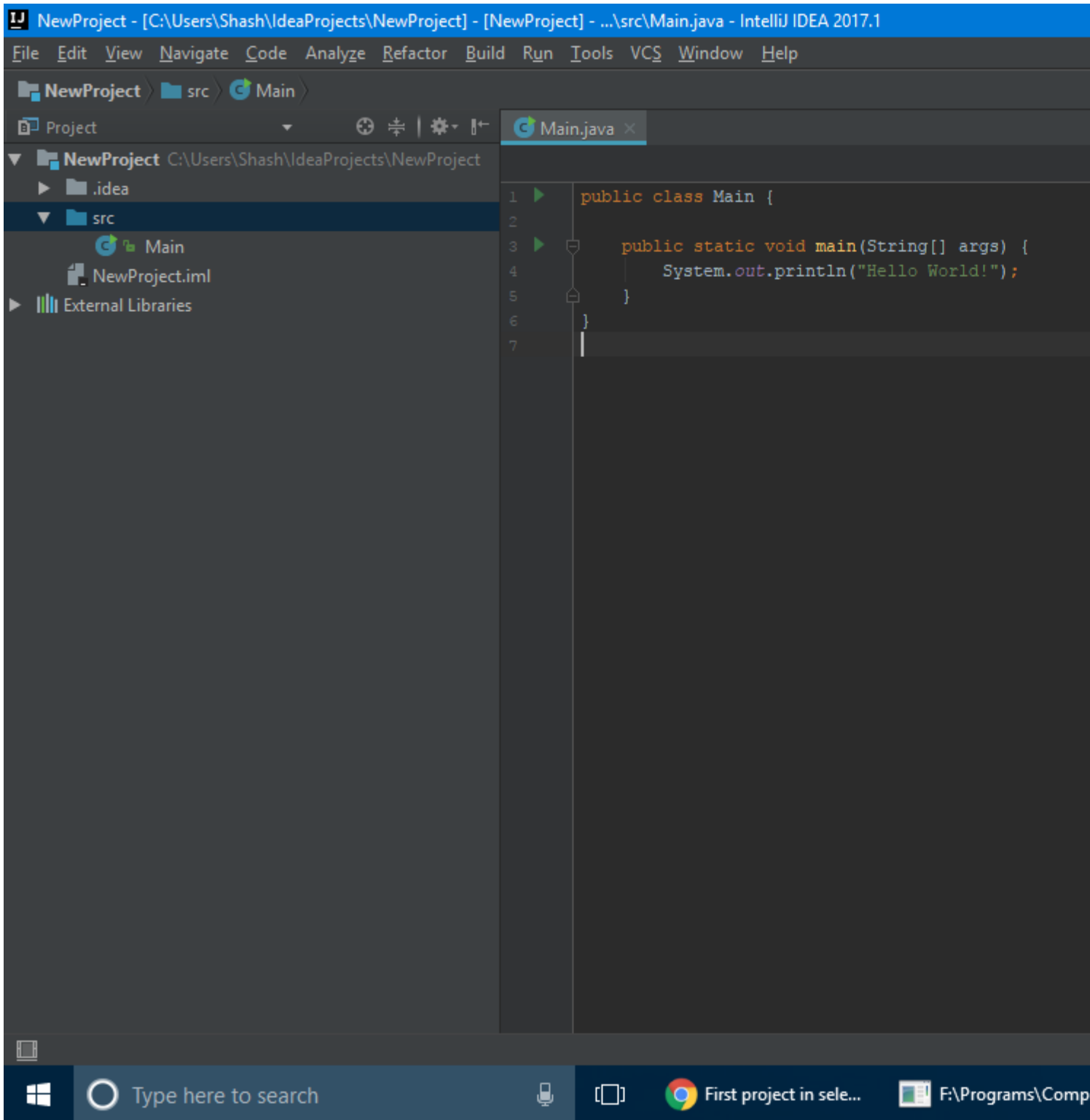
Conditions préalables:

1. Java est installé
2. Le sélénium est extrait dans un dossier (contient 2 fichiers et 1 dossier)

Suivez ces étapes pour configurer IntelliJ Idea for Selenium.

1. Cliquez sur "**Nouveau projet**".
2. Choisissez Java <Application "Hello World"
3. Tapez le nom du projet et créez-le.

Votre écran devrait ressembler à ceci

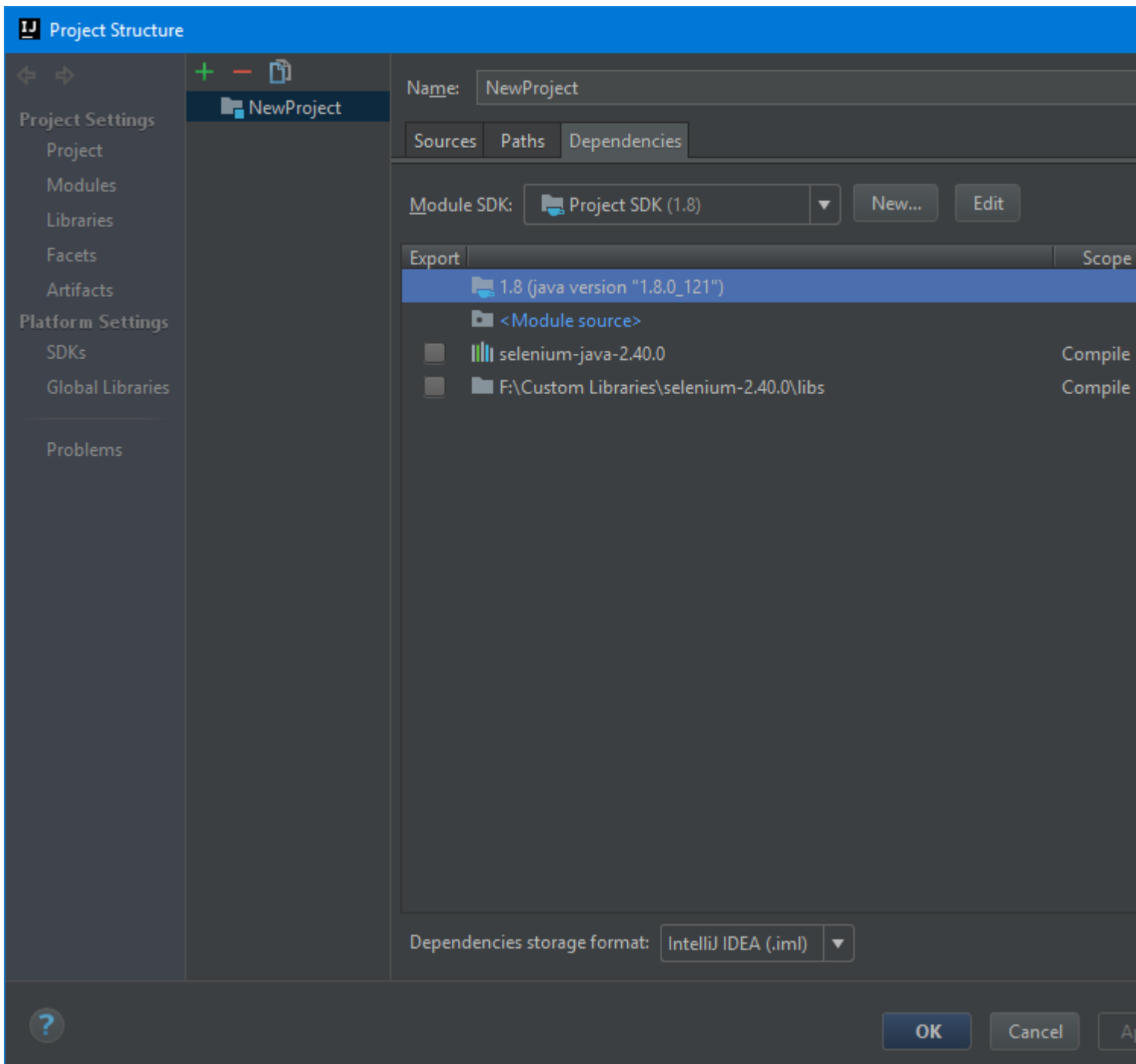


Maintenant, allez à

File < Project Structure < Modules < Dependencies

Cliquez sur l'icône verte plus (+) et choisissez Bibliothèque. Ensuite, accédez au dossier Selenium extrait et ajoutez " *selenium-java 2.4.0.jar*". Après l'ajout de ceci, cliquez à nouveau sur l'icône verte plus (+) et choisissez maintenant "*Répertoire*". Cette fois, localisez le dossier *libs* de Selenium et cliquez sur ok tout en le sélectionnant.

À la fin, votre structure de projet devrait ressembler à ceci



Maintenant, cliquez sur OK et vous êtes tous prêts.

## Configurer ChromeDriver

Conditions préalables: ChromeDriver est téléchargé

Copiez le code suivant dans votre classe.

```
public static void main(String[] args) {  
    System.setProperty("webdriver.chrome.driver", "path of the exe file\\chromedriver.exe");  
}
```

Si vous utilisez Linux, indiquez le chemin vers le binaire ChromeDriver.

## Ouvrir un site Web en utilisant Selenium

Nous utilisons la méthode `get` pour accéder à un site Web. Par exemple, cela ouvrirait google

```
public static void main(String[] args) throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "path of the exe file\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.google.com");
    Thread.sleep(3000); //wait for 3 seconds
    driver.quit();      //close Chrome
}
```

`driver.quit()` ferme le navigateur. Pour créer un délai, nous utilisons `Thread.sleep(3000)`.

## Obtenir des éléments dans Selenium

Chaque élément HTML dans Selenium est appelé `WebElement`. Par exemple, une `p` étiquette serait un `WebElement`, une `a` étiquette serait un `WebElement`, etc. Considérez la structure HTML suivante:

```
<a id="link1" href="https://www.google.com">google</a>
<p class="p1">
This is a paragraph
</p>
```

Maintenant, si nous voulions obtenir le `a` étiquette, nous pourrions faire

```
WebElement link = driver.findElement(By.id("link1"));
```

Maintenant, nous pouvons cliquer sur ceci, en

```
link.click();
```

Prenons un autre exemple. Si nous voulions le texte de la balise `p`, c'est-à-dire " *Ceci est un paragraphe* ", nous pouvons le faire

```
WebElement p = driver.findElement(By.className("p1"));
System.out.println(p.getText());
```

Nous pouvons également obtenir des éléments par balises, comme

```
WebElement tag = driver.findElement(By.tagName("a"));
```

## Exemple de travail dans le sélénium

Maintenant que nous connaissons les bases du sélénium, nous pouvons réaliser notre propre projet. Pour cet exemple, nous allons créer un programme qui détecte les dernières questions sur le dépassement de pile.

Nous commençons facilement, permet d'ouvrir le débordement de la pile.

```

public static void main(String[] args) throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "path of the exe file\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://stackoverflow.com");
    Thread.sleep(3000);
    driver.quit();
}

```

Maintenant, si vous regardez la source de la page, vous trouvez que toutes les questions, sont a balises, avec une `className` `question-hyperlink` . Cependant, comme il y a plusieurs questions, nous utilisons une `List` de `WebElement` au lieu de `WebElement` . Ainsi, nous pouvons faire

```

public static void main(String[] args) throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "path to chromedriver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://stackoverflow.com");
    List<WebElement> list = driver.findElements(By.className("question-hyperlink"));
}

```

Maintenant, nous avons besoin d'obtenir le `href` attribut de l' a balise, qui a le lien de la question. Pour ce faire, nous pouvons utiliser `getAttribute("href")` sur chaque `WebElement` , comme

```

public static void main(String[] args) throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "path to chromedriver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://stackoverflow.com");
    List<WebElement> list = driver.findElements(By.className("question-hyperlink"));
    System.out.println(list.size());
    list.forEach(e->System.out.println(e.getAttribute("href")));
    driver.quit();
}

```

Cela affiche les liens des questions principales sur le débordement de pile.

## Obtenir les attributs des WebElements dans Selenium

Pour obtenir l'attribut d'un `WebElement` , nous utilisons `getAttribute` sur cet objet `WebElement` . Par exemple, considérez la balise HTML suivante

```
<a id="click" href="https://www.google.com">
```

Nous pouvons trouver l'attribut `href` l'élément par

```

WebElement e = driver.findElement(By.id("click"));
System.out.println(e.getAttribute("href")); //prints https://www.google.com

```

Lire Premier projet en sélénium avec Java en ligne:

<https://riptutorial.com/fr/selenium/topic/10012/premier-projet-en-selenium-avec-java>

# Chapitre 7: Prenez une capture d'écran d'une page Web

## Exemples

### Capture d'écran Python Selenium Take / Save de la page Web

```
from selenium import webdriver

# Create a new chromedriver
driver = webdriver.Chrome()
# Go to www.google.com
driver.get("https://www.google.com")
# Saves a .png file with name my_screenshot_name to the directory that
# you are running the program from.
screenshot_name = "my_screenshot_name.png"
driver.save_screenshot(screenshot_name)
```

`driver.save_screenshot` renvoie "true" si la capture d'écran a été prise et "false" si ce n'est pas le cas. L'enregistrement des captures d'écran fonctionne également avec les navigateurs sans tête. Si vous souhaitez enregistrer une capture d'écran dans un répertoire différent, ajoutez simplement le chemin du fichier (relatif à l'endroit où vous exécutez le code). Par exemple:

```
screenshot_name = "screenshots/my_screenshot_name.png"
```

Sauvegarde la capture d'écran dans le répertoire "screenshots" dans le répertoire à partir duquel python est exécuté.

### Extension C # TakeScreenshot

```
public static Screenshot TakeScreenshot(this IWebDriver _driver)
{
    return ((ITakesScreenshot)_driver).GetScreenshot();
}
```

Exemple d'utilisation:

```
driver.TakeScreenshot().SaveAsFile(@"Test/Test.png", ImageFormat.Png);
```

### Capture d'écran de sélection / enregistrement Java de la page Web et ajout d'un rapport

```
public void Screenshot() throws Throwable{
    final byte[] screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.BYTES);
    scenario.embed(screenshot, "image/png"); // ... and embed it in the report.
    Thread.sleep(1000);
}
```

```
}
```

## Alternativement

```
public static void captureScreenShot(WebDriver ldriver){  
  
    // Take screenshot and store as a file format  
    File src= ((TakesScreenshot)ldriver).getScreenshotAs(OutputType.FILE);  
    try {  
        // now copy the screenshot to desired location using copyFile method  
  
        FileUtils.copyFile(src, new File("C:/selenium/"+System.currentTimeMillis()+".png"));  
    }  
  
    catch (IOException e)  
  
    {  
        System.out.println(e.getMessage());  
    }  
}
```

Lire Prenez une capture d'écran d'une page Web en ligne:

<https://riptutorial.com/fr/selenium/topic/6789/prenez-une-capture-d-ecran-d-une-page-web>

# Chapitre 8: Selenium IDE

## Exemples

### Essayez un script Selenium simple: Rechercher Wikipedia sur Google

Conditions préalables:

- Installer Firefox
- Installez l'addon Selenium IDE ( <https://addons.mozilla.org/fr/firefox/addon/selenium-ide/>)

Ouvrez le plugin. Un bouton affichant un cercle rouge doit être affiché. Si elle est pressée, cela signifie que vous pouvez commencer votre scénario. Le plugin enregistre tout ce que vous faites **dans cette instance de Firefox** .

Faites ce que vous voulez être enregistré.

En fin de compte, sauvegardez votre scénario; vous remarquerez que les scénarios de Selenium IDE sont des fichiers HTML.

Vous pouvez également ouvrir des fichiers d'autres utilisateurs. Par exemple, copiez et collez le code ci-dessous dans un nouveau fichier HTML et importez-le via votre plug-in. Vous pourrez exécuter le scénario très simple.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="https://www.google.com/" />
<title>sample-test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">sample-test</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>id=lst-ib</td>
<td>Wikipedia</td>
</tr>
</tbody></table>
</body>
</html>
```

Ce DSL (langage spécifique au domaine) est communément appelé "sélénese". Ses fonctions les plus courantes sont listées [ici](#) .



Lire Selenium IDE en ligne: <https://riptutorial.com/fr/selenium/topic/7694/selenium-ide>

---

# Chapitre 9: Selenium simple exemple C # et NUnit

## Remarques

Ceci est un exemple très simple ou le démarrage de Selenium, l'accès à une page et son utilisation, puis la fermeture de Selenium dans NUnit.

## Exemples

### Simple Selenium-NUnit

Prérequis:

- Le sélénium et les pilotes de navigateur requis sont installés (disponible dans Nuget)
- NUnit a été installé dans VS et ajouté au projet

```
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;
using OpenQA.Selenium.IE;
using System;
[TestFixture]
public class GoToGoogle
{
    //The WebDriver object
    IWebDriver driver;
    //Ran before test cases
    [TestFixtureSetUp]
    public void setup()
    {
        //Initialize the webdriver
        //An example of IE
        driver = new InternetExplorerDriver();
        //Firefox Example
        //driver = new FirefoxDriver();
        //An example of Chrome
        //driver = new ChromeDriver();

        //Wait x seconds to find the element and then fail, x = 5 here
        driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(5));
    }
    //Ran after the test case has completed
    [TestFixtureTearDown]
    public void tearDown()
    {
        driver.Quit();
    }
    [Test]
    public void gotoGoogle()
    {
```

```
//going to google.com
driver.Navigate().GoToUrl("www.google.com");
//Assert we are on google.com
Assert.AreEqual(driver.Title, "Google");
//Getting the search field
IWebElement searchField = driver.FindElement(By.Name("q"));
//Typing in the search field
searchField.SendKeys("Selenium Tutorial");
//Submitting
searchField.Submit();

}

}
```

Lire Selenium simple exemple C # et Nunit en ligne:

<https://riptutorial.com/fr/selenium/topic/6537/selenium-simple-exemple-c-sharp-et-nunit>

---

# Chapitre 10: Selenium simple exemple C # et Nunit

## Exemples

Test de chargement de page simple et assurez-vous que le titre de la page est correct

```
public class HomepageTests
{
    private IWebDriver _driver;

    [SetUp]
    public void LoadDriver()
    {
        _driver = new ChromeDriver();
    }

    [Test]
    public void Valid_Home_Page_Title()
    {
        _driver.Navigate().GoToUrl("Your homeoage url / local or remote");
        StringAssert.Contains("Expected title of your page", _driver.Title);
    }

    [TearDown]
    public void UnloadDriver()
    {
        _driver.Quit();
    }
}
```

Lire Selenium simple exemple C # et Nunit en ligne:

<https://riptutorial.com/fr/selenium/topic/6554/selenium-simple-exemple-c-sharp-et-nunit>

# Chapitre 11: Usine WebDriver

## Examples

### WebDriver Factory C #

```
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;
using OpenQA.Selenium.IE;

/// <summary>
/// Factory for creating WebDriver for various browsers.
/// </summary>
public static class WebDriverFactory
{
    /// <summary>
    /// Initilizes IWebDriver base on the given WebBrowser name.
    /// </summary>
    /// <param name="name"></param>
    /// <returns></returns>
    public static IWebDriver CreateWebDriver(WebBrowser name)
    {
        switch (name)
        {
            case WebBrowser.Firefox:
                return new FirefoxDriver();
            case WebBrowser.IE:
            case WebBrowser.InternetExplorer:
                InternetExplorerOptions ieOption = new InternetExplorerOptions();
                ieOption.IntroduceInstabilityByIgnoringProtectedModeSettings = true;
                ieOption.EnsureCleanSession = true;
                ieOption.RequireWindowFocus = true;
                return new InternetExplorerDriver(@"./", ieOption);
            case "safari":
                return new RemoteWebDriver(new Uri("http://mac-ip-address:the-opened-port"),
                DesiredCapabilities.Safari());
            case WebBrowser.Chrome:
            default:
                ChromeOptions chromeOption = new ChromeOptions();
                string location = @"./";
                chromeOption.AddArguments("--disable-extensions");
                return new ChromeDriver(location, chromeOption);
        }
    }
}

public enum WebBrowser
{
    IE,
    InternetExplorer,
    Firefox,
    Chrome
}
```

```
// Usage  
var driver = WebDriverFactory.CreateWebDriver(WebBrowser.Chrome);
```

Lire Usine WebDriver en ligne: <https://riptutorial.com/fr/selenium/topic/7727/usine-webdriver>

---

# Chapitre 12: WebDriverManager pour Selenium - un outil très soigné de Boni Garcia

## Introduction

Je suis passé à Selenium 3 et j'ai commencé à utiliser Chrome au lieu de Firefox. J'ai découvert que pour Chrome, je devais télécharger un fichier binaire pour WebDriver pour gérer le navigateur. Pour que cela fonctionne, je dois définir le chemin absolu de cette binaire en tant que variable Java. Si binaire est mis à jour, je dois mettre à jour ce binaire manuellement dans mon environnement de test - ce qui prend du temps et est vraiment agaçant. J'ai découvert une bibliothèque Java très soignée qui le fait pour moi:

<https://github.com/bonigarcia/webdrivermanager>

## Exemples

Les exemples ci-dessous montrent à quel point il est facile à utiliser

```
ChromeDriverManager.getInstance().setup();
FirefoxDriverManager.getInstance().setup();
OperaDriverManager.getInstance().setup();
PhantomJsDriverManager.getInstance().setup();
EdgeDriverManager.getInstance().setup();
InternetExplorerDriverManager.getInstance().setup();
```

Lire [WebDriverManager pour Selenium - un outil très soigné de Boni Garcia en ligne](#):

<https://riptutorial.com/fr/selenium/topic/10582/webdrivermanager-pour-selenium---un-outil-tres-soigne-de-boni-garcia>

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec le sélénium	<a href="#">akhilsk</a> , <a href="#">Alex.K.</a> , <a href="#">Brydenr</a> , <a href="#">Community</a> , <a href="#">Eugene S</a> , <a href="#">Priya</a> , <a href="#">Pseudo Sudo</a> , <a href="#">Ruslan López Carro</a> , <a href="#">Thomas</a> , <a href="#">Venkatesh Achanta</a>
2	Accepter les alertes popup avec Selenium	<a href="#">Brydenr</a> , <a href="#">Paul Muir</a> , <a href="#">Priya</a> , <a href="#">slackmart</a>
3	Automatisation des applications mobiles	<a href="#">Andersson</a>
4	Démarrer avec Selenium en python	<a href="#">Brydenr</a>
5	En attente dans Selenium	<a href="#">Brydenr</a> , <a href="#">John Fisher</a> , <a href="#">Kiril S.</a> , <a href="#">Priya</a> , <a href="#">Pseudo Sudo</a>
6	Premier projet en sélénium avec Java	<a href="#">Frank Underwood</a>
7	Prenez une capture d'écran d'une page Web	<a href="#">Brydenr</a> , <a href="#">DuarteNGF</a> , <a href="#">Paul Muir</a> , <a href="#">Priya</a>
8	Selenium IDE	<a href="#">Y-B Cause</a>
9	Selenium simple exemple C # et Nunit	<a href="#">Paul Muir</a>
10	Usine WebDriver	<a href="#">Buaban</a>
11	WebDriverManager pour Selenium - un outil très soigné de Boni Garcia	<a href="#">sen4ik</a>