
TP 5

Selenium web driver

1- Qu'est-ce que Selenium ?

Selenium désigne une suite d'outils largement utilisée dans la communauté des tests, notamment pour les tests multi-navigateurs. Selenium ne peut pas automatiser les applications de bureau ; il est uniquement destiné à l'automatisation des navigateurs. Il prend en charge plusieurs navigateurs tels que **Chrome, Firefox, Edge, Safari, Opera**, ainsi que divers systèmes d'exploitation comme **Windows, Mac et Linux/Unix**.

Selenium est également compatible avec différents langages de programmation – **C#, Java, JavaScript, Ruby, Python, PHP**. Les testeurs peuvent choisir le langage dans lequel concevoir leurs cas de test, ce qui rend Selenium très apprécié pour sa flexibilité.

2- Composants de Selenium:

La suite de tests Selenium comprend trois composants principaux :

- **Selenium Integrated Development Environment (IDE)** est une extension de navigateur pour Firefox et Chrome qui permet de générer rapidement des tests grâce à sa fonctionnalité d'enregistrement et de relecture. L'IDE enregistre les actions de l'utilisateur sur le navigateur et les exporte sous forme de scripts réutilisables dans différents langages tels que Java, C# et JavaScript.
- **Selenium WebDriver** est le composant principal de Selenium qui fournit une interface de programmation permettant de contrôler les navigateurs web. Il permet d'écrire des tests dans différents langages de programmation pour interagir avec les éléments d'une page, simuler des interactions utilisateur et effectuer des vérifications (assertions).
- **Selenium Grid** est un outil utilisé pour l'exécution simultanée de cas de test sur différents navigateurs, machines et systèmes d'exploitation. Il permet l'exécution parallèle des tests, ce qui accélère l'exécution de grandes suites de tests.

Dans la suite on s'intéresse au **Selenium WebDriver**.

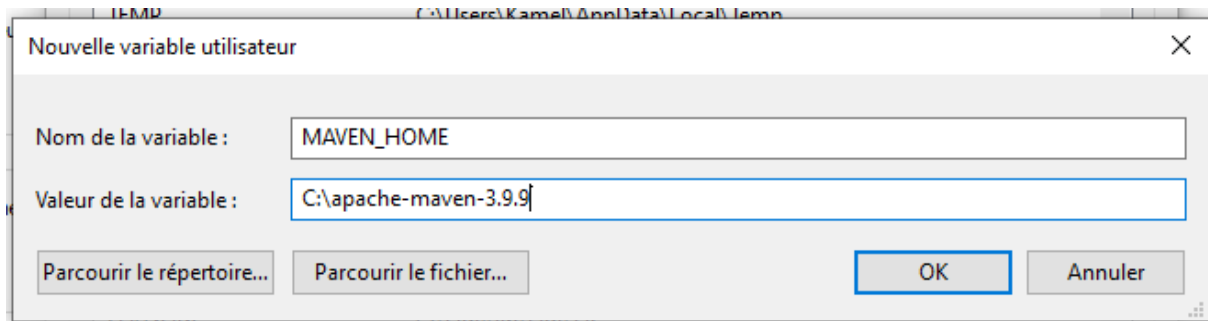
3- Installation et premier test :

3-1: Installation et configuration de Maven :

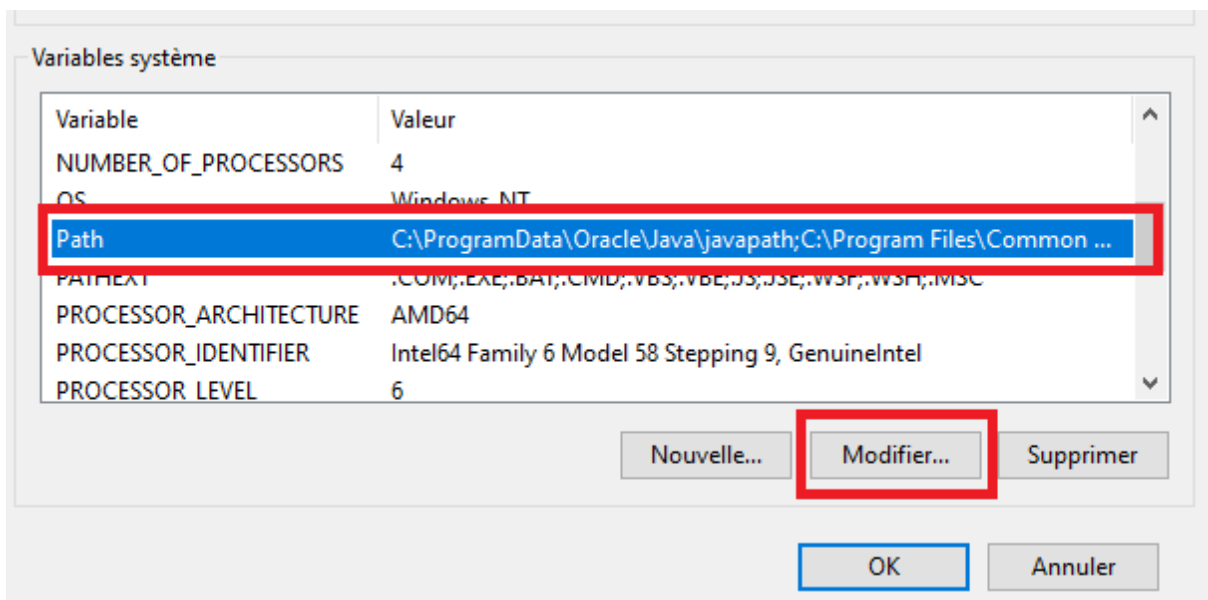
Etape1: téléchargez **Apache Maven 3.9.9** de <https://maven.apache.org/download.cgi>

Etape 2: décompressez le fichier **apache-maven-3.9.9-bin.zip** dans le dossier où nous souhaitons que Maven soit installé par exemple **C:/** pour obtenir le répertoire Maven comme suit **C:/apache-maven-3.9.9/**

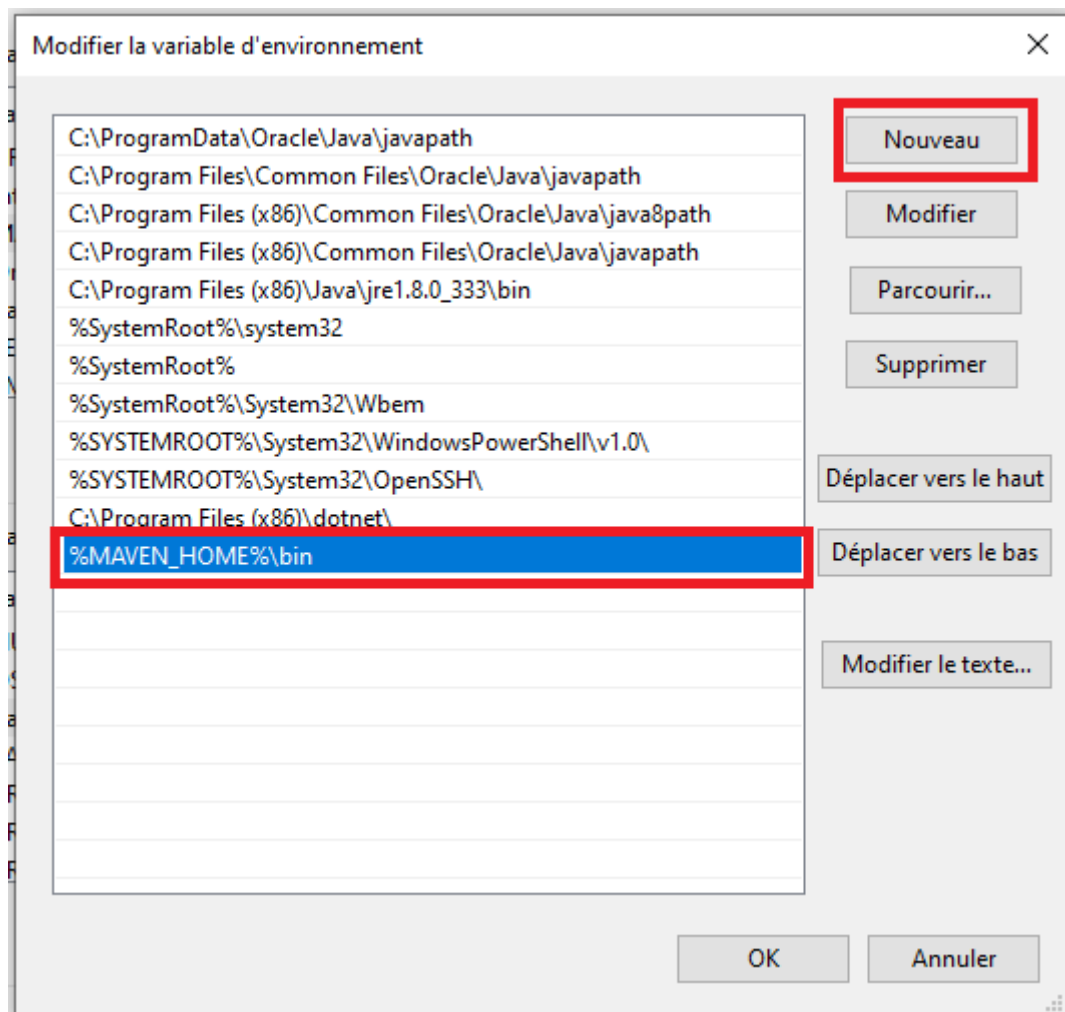
Etape 3: Configurer le *path* : *posteDeTravail->Button Droit->Propriétés->Paramètres avancées du système-> variable d'environnement-> Nouvelle Variable utilisateur*, puis ajouter dans *Nouvelle variable utilisateur* : Nom de la variable **MAVEN_HOME**. Valeur de la variable : **C:\apache-maven-3.9.9** comme illustré dans la figure suivante :



Ensuite : dans Variable système on sélectionne **path** et on clique sur le bouton **modifier** comme illustré dans la figure suivante :



On obtient une fenêtre, on clique sur le bouton **Nouveau** et taper **%MAVEN_HOME%\bin** comme illustré dans la figure suivante :



Etape 4: Tester MVN, Ouvrez l'invite de commandes et tapez : *mvn -version*.

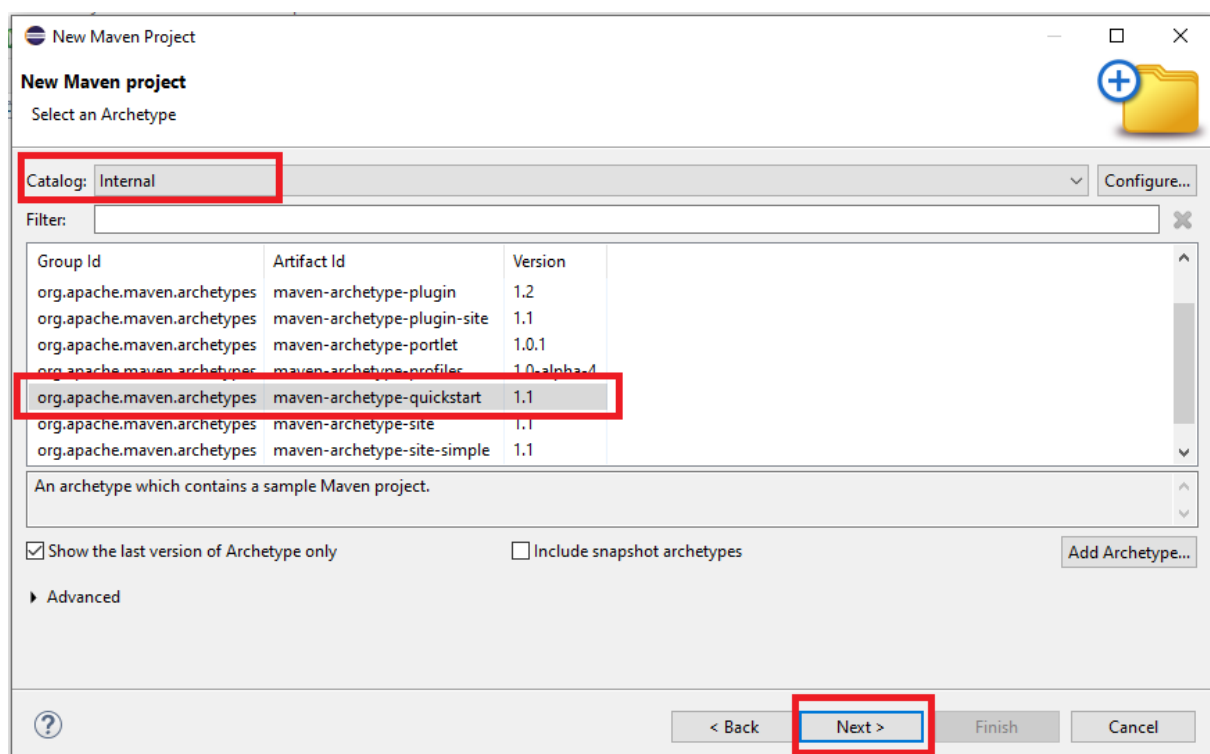
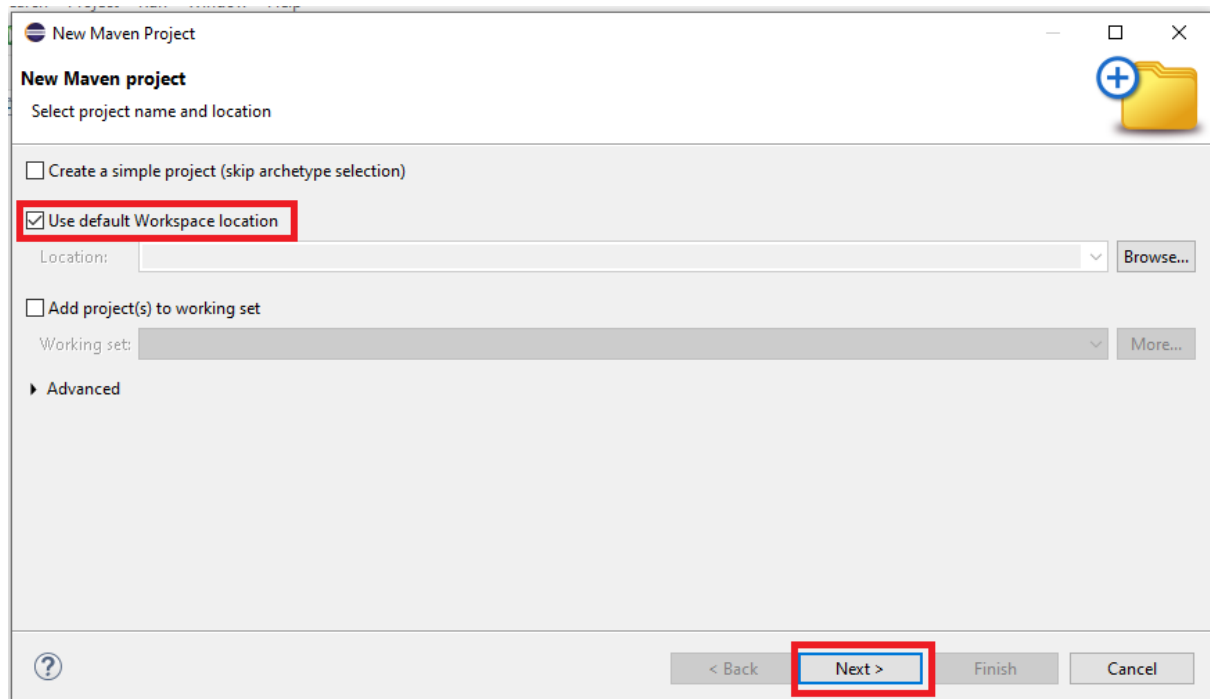
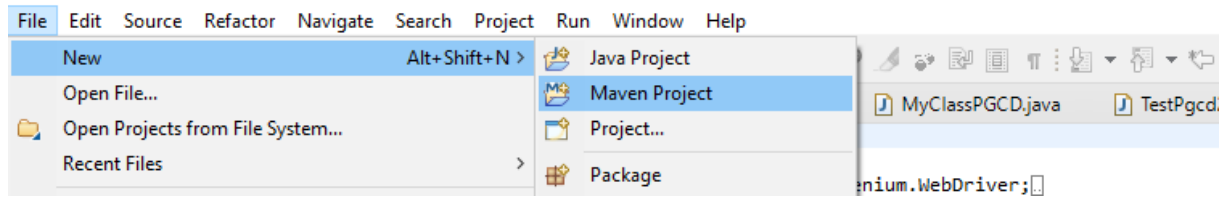
```

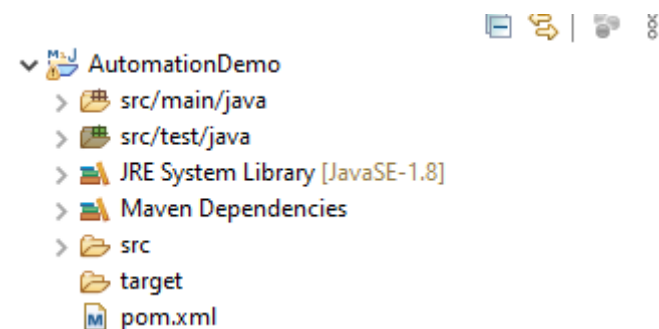
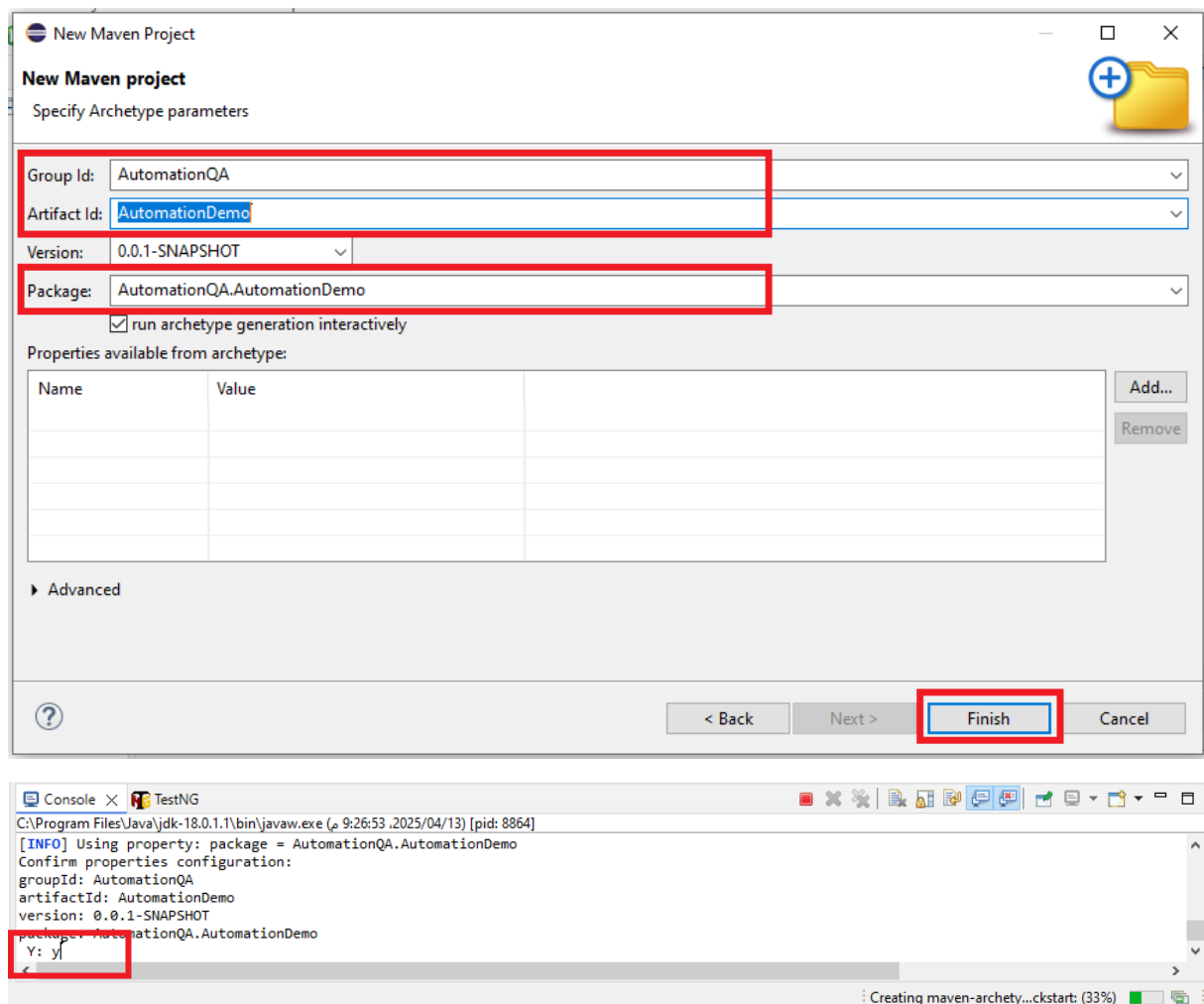
C:\> Invite de commandes
Microsoft Windows [version 10.0.19045.5737]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kamel>mvn -version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfc9c97d260186937)
Maven home: C:\apache-maven-3.9.9
Java version: 18.0.1.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-18.0.1.1
Default locale: fr_FR, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
  
```

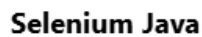
3-2 Premier test selenium:

Etape 1: création d'un projet MAVEN dans l'onglet New de Eclipse choisi Maven Project comme illustré dans les figures suivantes :





Etape 2 : configuration de pom.xml : allez dans le site : <https://mvnrepository.com/> et dans la barre de recherche tapez **selenium java** :



License	Apache 2.0
Categories	Web Testing
Tags	quality selenium testing web
HomePage	https://selenium.dev/
Ranking	#291 in MvnRepository (See Top Artifacts) #1 in Web Testing
Used By	1,865 artifacts

6

HomePage	https://selenium.dev/
Date	Apr 05, 2025
Files	pom (4 KB) jar (545 bytes) View All
Repositories	Central
Ranking	#291 in MvnRepository (See Top Artifacts) #1 in Web Testing
Used By	1,865 artifacts

Maven
Gradle
SBT
Mill
Ivy
Grape
Leiningen
Buildr

Scope: **Compile**

```

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>4.31.0</version>
</dependency>

```

Copier-coller dans pom.xml

```

10 <name>AutomationDemo</name>
11 <url>http://maven.apache.org</url>
12
13 <properties>
14   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15 </properties>
16
17 <dependencies>
18   <dependency>
19     <groupId>junit</groupId>
20     <artifactId>junit</artifactId>
21     <version>3.8.1</version>
22     <scope>test</scope>
23   </dependency>
24   <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
25   <dependency>
26     <groupId>org.seleniumhq.selenium</groupId>
27     <artifactId>selenium-java</artifactId>
28     <version>4.31.0</version>
29   </dependency>
30
31 </dependencies>

```

On ajoute à pom.xml la dependency suivantes :

```

<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>6.0.1</version> <!-- Check for latest version if needed -->
</dependency>

```

3-3 Créer votre programme selenium :

Dans la classe **App.java** (ou créer une nouvelle classe) tapez le code suivant :

```
public static void main( String[] args )
{
    WebDriverManager.firefoxdriver().setup();
    FirefoxDriver driver = new FirefoxDriver();
    driver.get("https://www.google.com/");
    driver.manage().window().maximize();
}
```

- on change le code précédent pour faire une recherche sur le mot “test logiciel”:

```
public static void main( String[] args )
{
    /*
        Le code précédent
    */

    WebElement searchBar = driver.findElement(By.cssSelector("#APjFqb"));
    searchBar.sendKeys("test logiciel");

    WebElement bouton =
driver.findElement(By.xpath("/html/body/div[1]/div[3]/form/div[1]/div[1]/div[3]/center/input
[1]"));

    bouton.click();
}
```

- on change le code précédent pour fermer la page web après 10 seconds:

```
public static void main( String[] args )
{
    /*
        Le code précédent
    */

    try {

        Thread.sleep(10000);
```



```

    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    driver.close();
}

```

Un autre exemple :

```

public static void main( String[] args )

```

```

{
    WebDriverManager.firefoxdriver().setup();
    FirefoxDriver driver = new FirefoxDriver();
    driver.get("https://www.bing.com/");
    driver.manage().window().maximize();

    WebElement searchBar = driver.findElement(By.name("q")); // Trouver le champ de
recherche avec name="q"
    searchBar.sendKeys("test logiciel"); // Taper une requête
    searchBar.submit(); // Soumet le formulaire contenant le champ de recherche
    (équivalent à appuyer sur "Entrée")

    try {
        Thread.sleep(10000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    // driver.close();
}

```