



Le mode « *headless* » est permet d'utiliser Firefox normalement sans toutefois afficher aucun élément d'interface graphique. Cela n'est sans doute pas très utile pour surfer sur le Web, mais peut être extrêmement utile pour les tests industrialisés. Cet article fournit tout ce dont vous pouvez avoir besoin pour utiliser le mode *headless* de Firefox.

## Utiliser le mode *headless*

Vous pouvez démarrer Firefox dans son mode *headless* grâce à une ligne de commande incluant le drapeau (*flag*) `-headless`. Par exemple :

```
1 | /chemin/vers/firefox -headless
```

Pour le moment, nous n'avons pas inclus davantage d'options, mais plus seront ajoutées plus tard.

Par exemple, nous travaillons à implémenter une option `--screenshot`, qui permettra de faire des captures d'écran depuis le mode *headless* de Firefox. Voir [bug 1378010](#) pour suivre l'avancée.

## Prise en charge

Le mode *headless* de Firefox fonctionne à partir de la version 55 sur Linux et à partir de la version 56+ sur Windows et Mac.

## Tests industrialisés à l'aide du mode *headless*

La façon la plus utile d'utiliser ce mode *headless* est de faire tourner des tests industrialisés dans Firefox. Cela signifie que vous pouvez rendre votre processus de test bien plus efficace grâce à ce mode.

## Selenium

Pour fournir un exemple d'utilisation du mode headless pour test industrialisés, nous allons créer un test recourant à [Selenium](#) via [Node.js](#) et [selenium-webdriver](#). Pour cela, nous supposons que vous êtes déjà à l'aise avec les bases de Selenium, WebDriver et Node, puis que vous avez préparé un environnement de test. Si vous ne l'avez pas fait, rendez-vous sur le guide développant la [mise en place d'un environnement de test](#), puis revenez lire cette documentation.

Tout d'abord, soyez sûr d'avoir installé Node ainsi que `selenium-webdriver` sur votre machine. Ensuite, créez un nouveau fichier nommé `selenium-test.js`.

**Note** : Vous pouvez également cloner ce dépôt : [headless-examples](#) qui contient un fichier de *package*. Il suffit donc de lancer `npm install` afin d'installer les dépendances nécessaires.

1. Ajouter quelques lignes de code. À l'intérieur de ce fichier, commencez en important le module principal selenium-webdriver, ainsi que le sous-module firefox :

```
1 | var webdriver = require('selenium-webdriver'),
2 |   By = webdriver.By,
3 |   until = webdriver.until;
4 |
5 | var firefox = require('selenium-webdriver/firefox');
```

2. Puis, créez un objet `binary` qui représente Firefox Nightly et ajouter l'argument `-headless` afin qu'il puisse être lancé avec ce mode :

```
1 | var binary = new firefox.Binary(firefox.Channel.NIGHTLY);
2 | binary.addArguments("-headless");
```

3. Maintenant, créez une nouvelle instance de driver utilisant Firefox et utilisez `setFirefoxOptions()` afin d'inclure une option qui spécifiera que le test devra tourner sur le Nightly channel de Firefox (cette étape n'est pas nécessaire sur Linux, mais reste utile pour utiliser les fonctions avancées de la version Nightly de Firefox sur Windows/Mac tant que celle-ci n'est pas disponible en *release*) :

```
1 | var driver = new webdriver.Builder()
2 |   .forBrowser('firefox')
3 |   .setFirefoxOptions(new firefox.Options().setBinary(binary))
4 |   .build();
```

4. Il faut maintenant ajouter la ligne de code qui initiera la navigation sur la page de recherche Google :

```
1 | driver.get('https://www.google.com');
2 | driver.findElement(By.name('q')).sendKeys('webdriver');
3 |
4 | driver.sleep(1000).then(function() {
5 |   driver.findElement(By.name('q')).sendKeys(webdriver.Key.TAB);
6 | });
7 |
8 | driver.findElement(By.name('btnK')).click();
9 |
10 | driver.sleep(2000).then(function() {
11 |   driver.getTitle().then(function(title) {
12 |     if(title === 'webdriver - Google Search') {
13 |       console.log('Test passed');
14 |     } else {
15 |       console.log('Test failed');
16 |     }
17 |   });
18 | });
19 |
20 | driver.quit();
```

5. Enfin, démarrez le test en utilisant la commande suivante :

```
1 | node selenium-test
```

Et c'est tout ! Après quelques secondes, vous devriez voir apparaître le message "Test passed" sur la console

L'article [Headless Firefox in Node.js with selenium-webdriver](#) de Myk Melez contient d'autres conseils utiles pour créer un test industrialisé via Node.js et Selenium dans le mode *headless*.

## D'autres solutions de test

Slimerjs supporte Firefox sur Linux et bientôt sur Mac et Windows. Voir l'article [Headless SlimerJS with Firefox](#) de Brendan Dahl pour plus détails.

De plus, vous pourrez utiliser le mode headless de Firefox pour faire tourner des tests industrialisés développés dans la plupart des autres applications de tests, pour autant qu'elles permettent de définir une variable d'environnement.

## Dépannage et aide supplémentaire

Si vous avez le moindre problème en utilisant le mode *headless*, ne vous inquiétez pas, nous sommes ici pour vous aider. Cette section a pour but de référencer toutes vos questions et les réponses que nous leur apportons.

- Sur Linux, certaines bibliothèques sont nécessaires pour faire tourner le mode *headless* de Firefox, même si ce dernier ne les utilise pas directement. Voir le [bug 1372998](#) pour suivre la résolution du problème.

Si vous souhaitez poser une question à nos ingénieurs, le meilleur moyen est de se rendre sur le [canal IRC #headless](#) de Mozilla. Si vous êtes certain d'avoir trouvé un bug, documentez le sur la plateforme [Mozilla Bugzilla](#).

## Voir aussi

- [Using Selenium with Headless Firefox \(on Windows\)](#) (en anglais) par Andre Perunicic (utilise Python)
- [Headless Firefox in Node.js with selenium-webdriver](#) (en anglais) par Myk Melez
- [Headless SlimerJS with Firefox](#) (en anglais) par Brendan Dahl