

Wayne Out There (.com)  
wayne-out-there  
Stuff that matters to Wayne

---

## Dismissing Firefox popups using Selenium and Python

Posted on November 19,2019 by admin

In my new journey to figure out how to automate stuff in my life, one of the time-sucking adventures was to try to stop browser popups such as geolocation and notifications. I just wanted to click 'ok' or 'dismiss' and move on since this was my 'bot' instance of the browser.

The issue with these browser-based pop ups is that they are handled by the browser, not by selenium so it seems selenium can't dismiss them or dismiss them easily.

The solution workflow that I found to work well for me is as follows:

1. Switch to your Selenium Firefox profile
2. Do your browser setting that you need
3. Save the change
4. Start your Selenium script with new changes

Here are the details on how to do this:

### Create a new Firefox Profile

1. in browser, type about:profiles
2. Create a new profile
3. Take a note of the location of the root directory of that profile (ie paste to your Selenium script for now)
4. 'Launch profile in new browser' (this will launch your new profile in a new instance nicely)
5. Make your browser changes and save

### Do your browser setting changes

The following two changes were the main two that bothered me so perhaps I'll highlight those here:

1. Don't allow websites to send you notifications
  - Preferences/privacy and security / Notifications
1. Don't let websites ask for geolocation stuff
  - Preferences/privacy and security / Permissions - Locations (Settings button)

TIP! It's very useful to actually do a dry run as a human before you let your bot run free so that you can deal with these popups logged in, one time, as this profile. So, workflow is to switch to your Selenium profile, launch it, do a dry run on with real human clicks, deal with any popups or browser setting stuff, save changes, go run the bot script.

### Setting up your Python Script to use your Selenium Profile

Now that you've got your new profile, let's actually use it in your script, instead of what will always be a fresh browser instance

Here is my code block which you can add in your project:

```
## SETUP SELENIUM TO USE CUSTOM FIREFOX PROFILE
#Pretty sure you need to import this to use 'FirefoxProfile' but I'm too lazy to confirm - feel free
from selenium.webdriver.firefox.options import Options
#Root directory copied from Step 3 above
profilePath = '/home/user/.mozilla/firefox/cjda7321.Selena'
#Directs profile selector thing to the right path created above
profile = webdriver.FirefoxProfile(profilePath)
#Tells Selenium to use the custom profiledriver = webdriver.Firefox(firefox_profile=profile)
```

Hope that helps!

Posted in: [Technology](#), [Tutorial](#) | Tagged: [Python](#), [Selenium](#), [Tutorial](#) | With 0 comments