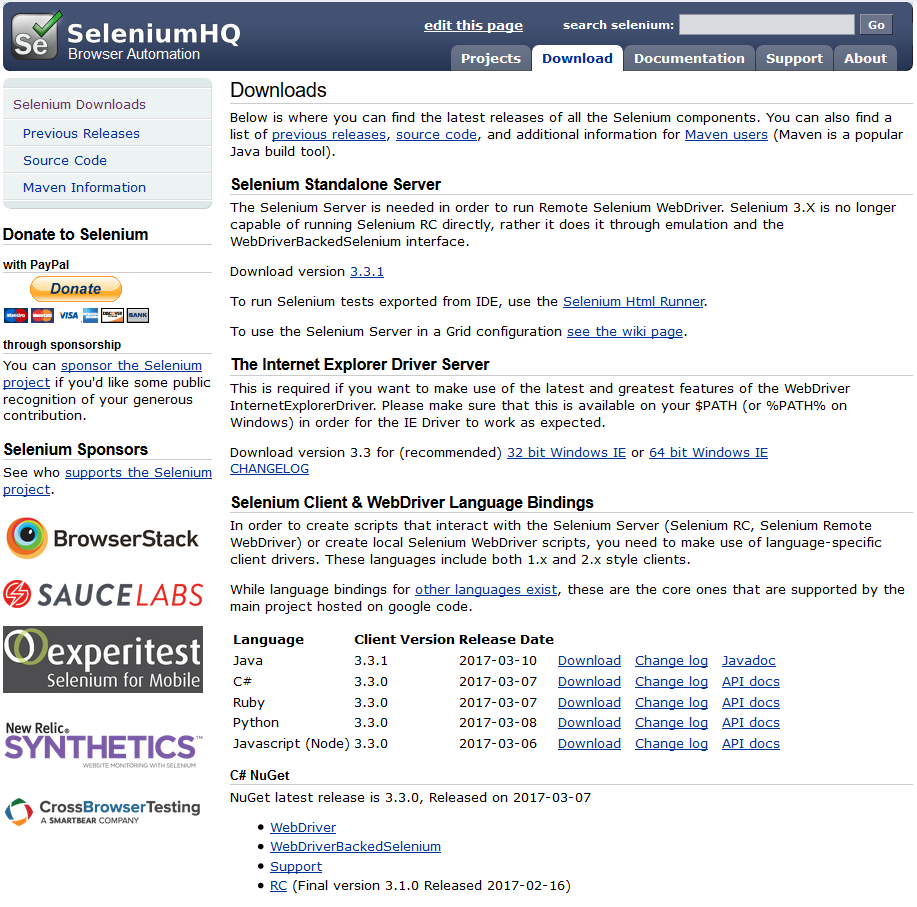
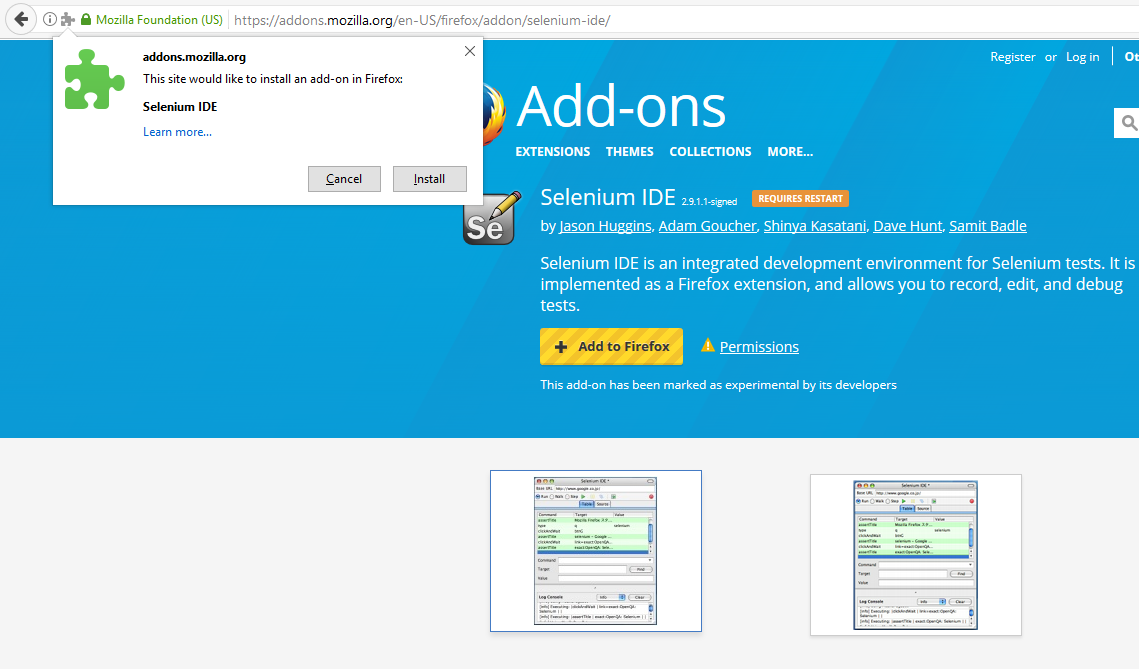
**Steps to Install Selenium on a Conestoga Lab Machine**

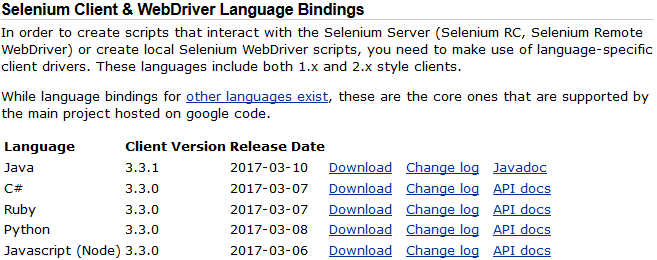
1. Download and install Firefox from <https://www.mozilla.org/en-US/firefox/new/>
2. Using Firefox browser, go to <http://www.seleniumhq.org/>
3. Click on the "Download Selenium" link on the right-hand side of the Selenium homepage. This gets you to the download page:



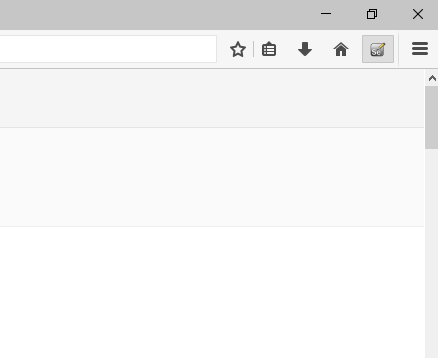
1. The documentation for Selenium can be found at <http://docs.seleniumhq.org/docs/>
2. Download the Selenium IDE (further down the download page), which includes several Firefox plugins. The current version is Selenium IDE 2.9.1.1. Click the link "Download latest released version from addons.mozilla.org" which will take you to <https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>
3. Click the yellow "Add to Firefox" button. This will present you with a popup asking you to authorize the installation. Click "Install." You will then be prompted to restart Firefox. Click the "Restart" button.



1. After Firefox has restarted, return to <http://www.seleniumhq.org/download/> and download the C# version of the "Selenium Client and WebDriver Language Bindings", version 3.3.0. You will want to bookmark the "API docs" link in your browser so that you can find the online Selenium API documentation easily.



1. Save the selenium-dotnet-3.3.0.zip file somewhere on your machine, and unzip the contents to a separate directory, say C:\selenium-dotnet. The .zip file contains the .dll files that will permit you to use Selenium with NUnit. You will be using the .dll files in the net40 subdirectory when you add references for your C# NUnit test project in Visual Studio.
2. After downloading the Selenium NUnit .dll files, you are ready to start to use Selenium by using the IDE from within Firefox. There will be a Selenium icon added to Firefox in the top right corner. Alternatively, you can access Selenium by clicking the Firefox menu button, choosing "Developer", then choosing "Selenium IDE."



1. Once you start the IDE, then you can record web interactions and save them in a script for replay, or you can export the script as a C# NUnit test case. Once saved, you can then execute the test case from within the NUnit console.

