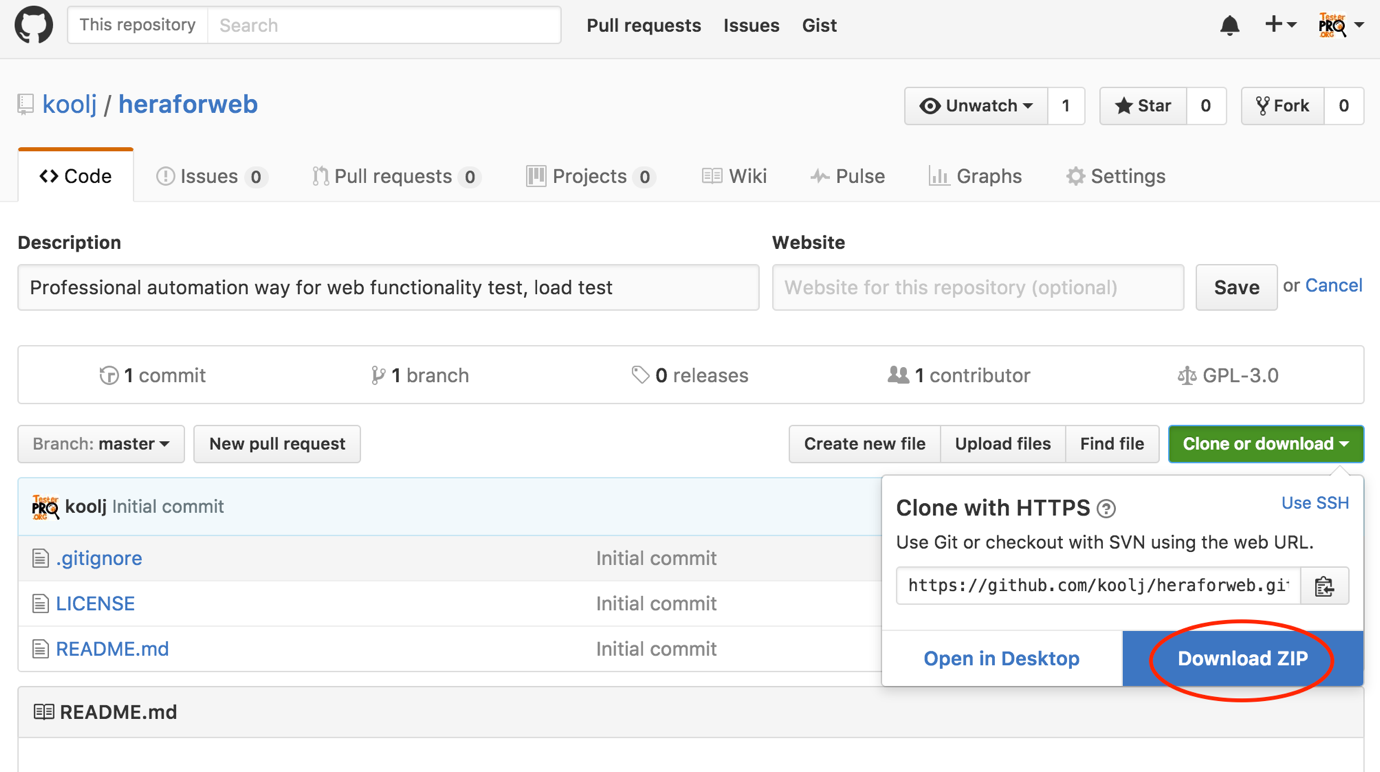
By KoolJ

User guide for HERA Web

1. [SetUp](#_1._1)
2. [Config](#_2._1)
3. [Setup TestCase](#_3._1)
4. [View the report](#_4._1)
5. [Keywords reference](#_5._1)
6. [How to finding id or xpath node on HTML page](#_6._1)

# 1.

Get .ZIP from GitHub: <https://github.com/koolj/heraforweb>



1. Setup Oracle JDK

* Check Java should be installed, by typing in cmd commandline/terminal (Win/Mac):

Java –version

<should return>

java version "1.8.xxx"

Java(TM) SE Runtime Environment (build 1.8.0\_92-b14)

If not, then setup JDK.

Follow this to setup (use this version of jdk or newer): <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

* Check after installation, in commandline terminal by typing

Java –version

<should return>

java version "1.8.xx"

Java(TM) SE Runtime Environment (build 1.8.0\_92-b14)

* System setting:
* On Windows OS:
  + Open system variable:
    - MyComputer> Property > SystemProperty> Tab Advanced > Environment Variables > System variable
    - Add variables: On system variables
  + tap New...
    - Variable Name: JAVA\_HOME
    - Variable Value: <path to java folder, ex: C:\Program Files (x86)\Java\jdk1.8.xxx>
  + Edit Path on System Variables
    - Choose: Path > edit
    - add the string of “;%JAVA\_HOME%/bin"
  + Save
* On Mac OS:
  + Make sure you have Java SDK 1.8.x for Mac (download from <https://support.apple.com/kb/DL1572?locale=en_US>) default installed on your mac. Check it by command lines:
  + In Launchpad>Terminal> type “java -version” -> it returns something like: java version “1.8.xxxx" -> it is ok
  + In Launchpad>Terminal> type “javac” ->should return Usage: javac <options> <source files>
  + -> it is ok
  + If not, go to this LINK to download suitable a 1.8.x version for mac: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
  + Then set JAVA\_HOME, in Launchpad>Terminal> type” export JAVA\_HOME=`/usr/libexec/java\_home` “

# 2.

Setting config:

1. Note:

* Firefox, Safari, Chrome should be installed first before run correlatively.
* On Windows will have IE, but for Mac, Linux, Unix OS.

In .ZIP, it includes **drivers** folder. Setting driver path in (b).

1. Setting file **config\_web.xls** :

|  |  |
| --- | --- |
| firefox\_link | “./drivers/geckodriver” <default to “./drivers/geckodriver.exe” for Windows, or download from: <https://github.com/mozilla/geckodriver/releases>, with Mac/Linux, get the correlative version |
| chrome\_link | with Window/Mac/Linux, get the correct driver version, and set the path to right folder, from: https://sites.google.com/a/chromium.org/chromedriver |
| ie\_link | Only on Window, and set the path to right folder. Download the driver from: http://selenium-release.storage.googleapis.com/2.53/IEDriverServer\_x64\_2.53.1.zip |
| safari\_link | with Window/Mac/Linux, get the correct driver version, and set the path to right folder, from: http://selenium-release.storage.googleapis.com/2.48/SafariDriver.safariextz |
| webhost\_link | <website URL root for auto testing> |
| dev | <tester name> |

1. Setting **run.xml**:

-For one browser:

<test name="firefoxTest">

<parameter name="browser" value="firefox" />

<classes>

<class name="hera.web.control" />

</classes>

</test>

-For multiple browsers, multi threads:

<test name="firefoxTest">

<parameter name="browser" value="firefox" />

<classes>

<class name="hera.web.control" />

</classes>

</test>

<test name="chromeTest">

<parameter name="browser" value="chrome" />

<classes>

<class name="hera.web.control" />

</classes>

</test>

<test name="ieTest">

<parameter name="browser" value="ie" />

<classes>

<class name="hera.web.control" />

</classes>

</test>

<test name="safariTest">

<parameter name="browser" value="safari" />

<classes>

<class name="hera.web.control" />

</classes>

</test>

-For multiple threads in one browser:

<test name="test1">

<parameter name="browser" value=" chrome " />

<classes>

<class name="hera.web.control" />

</classes>

</test>

<test name="test2">

<parameter name="browser" value="chrome" />

<classes>

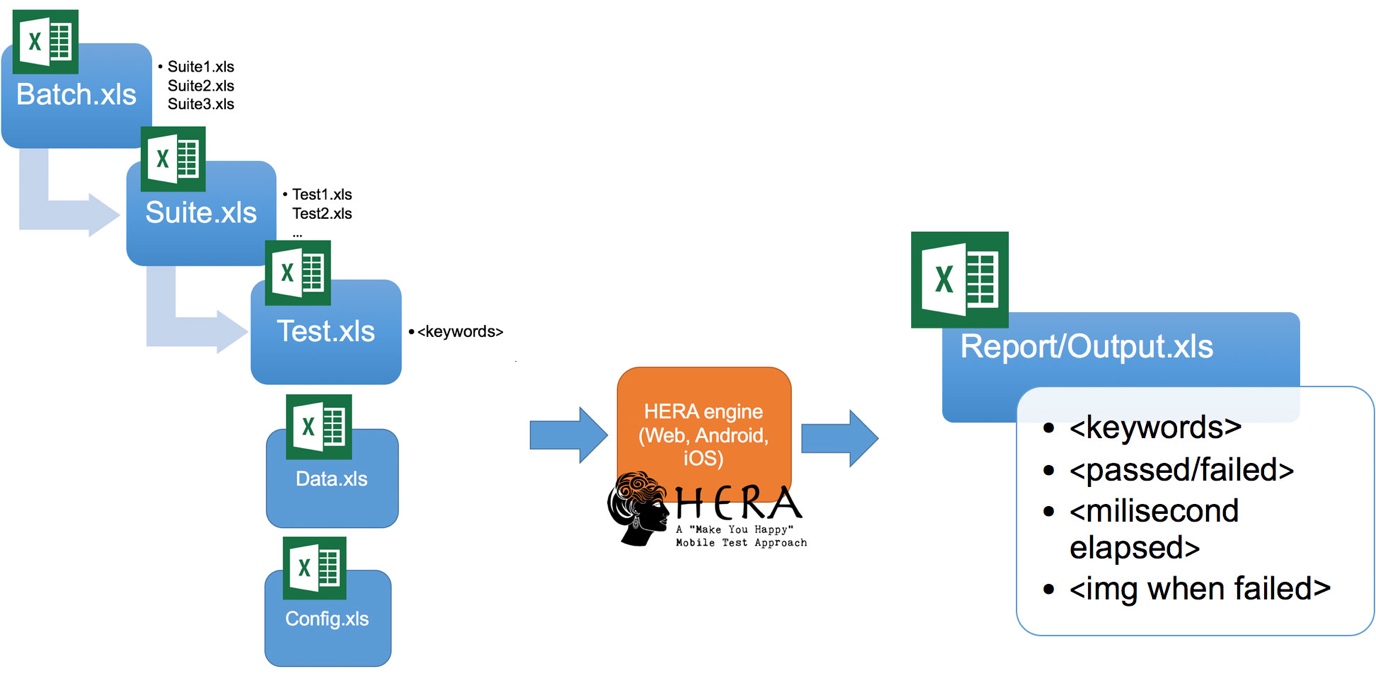
<class name="hera.web.control" />

</classes>

</test>

# 3.

1. The Architecture:



1. Example for login test steps & keywords

|  |  |  |  |
| --- | --- | --- | --- |
| R | sel\_XPATHclick | .//\*[@id='account']/a |  |
| R | sel\_eXwaitforID | 60000 | log |
| R | sel\_CSVsetText | log | Anhpham |
| R | sel\_CSVsetText | pwd | Sundayall97 |
| R | sel\_sleep | 4000 |  |

1. Run cmd/terminal (win/mac), after “cd” to .ZIP folder:

* In Windows:

java -cp lib\\*;bin org.testng.TestNG run.xml

* In Mac:

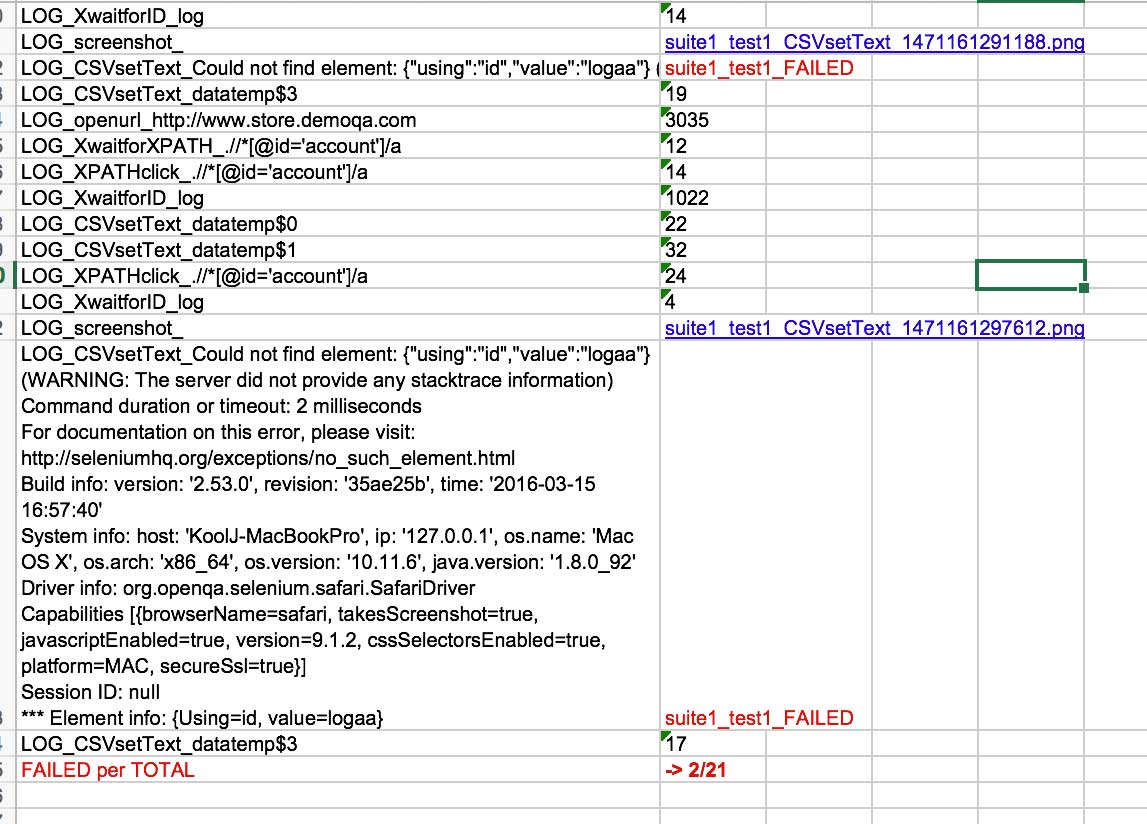
java -cp lib/\*:bin org.testng.TestNG run.xml

1. View report from “report” folder**: output\_1466135420329.xls**, auto returns an Excel report each time run. Refer how to read a report on (4)

# 4.

Understand a report:

1. Eachtime HERA runs, it returns an output\_xxx.xls, that “xxx” is the time it was born. Ex: **output\_1466135420329.xls**

****

1. If the step failed, it notify: **suite\_test\_<tên step>\_FAILED**. And then capture an image of .png format, stored on same folder.
2. The last line is the total FAILED per TOTAL run.

# 5.

Keyword reference:

**Just sleep**

sel\_sleep | <milisecond> |

**Store the variable to apply the loop for…endfor**

store | <variable\_name> | <value\_to\_store> |

for | <var\_first\_step> | <var\_step> | <var\_total\_step> |

endfor

**To click:**sel\_XPATHclick | <html\_xpath> |

sel\_IDclick | <html\_id> |sel\_MENUclick | <submenu\_name> |<menu\_item\_name> |

sel\_LINKclick | <link\_name> |

(bổ sung)

sel\_CSSclick

sel\_NAMEclick

**Get data from EXCEL, need “CSV read” first:**

sel\_CSVread | <fileExcel\_name> |

sel\_CSVsetText | < html\_id> | <data\_EXCELfile\_name>$<column\_number>, |

(lưu ý cần có sel\_CSVread, trước khi áp dụng sel\_CSVsetText, trước khi vào vòng for..endfor. Xem mẫu “test\_loadExcel\_data\_sample.xls” để tham khảo)

(bổ sung)

sel\_CSVXPATHsetText

sel\_CSVCSSsetText

sel\_CSVNAMEsetText

**Add text to textbox**

sel\_XPATHsendkey | <html\_xpath> |<value\_to\_send>|sel\_IDsendkey | <html\_id> |<value\_to\_send>|

(bổ sung)

sel\_NAMEsendkey

sel\_CSSsendkey

Ram dom:

sel\_XPATHsendkey\_ran | <html\_xpath> |<value\_to\_send>|

* (random value)\_ <value\_to\_send>

sel\_NAMEsendkey\_ran

sel\_IDsendkey\_ran

sel\_CSSsendkey\_ran

set keep same random, or randomly

sel\_randomkeep (same random value)

sel\_randomend (randomly)

**Get value from combobox, checkbox, option**

(bổ sung)

sel\_CSVcomboboxbynamevalue | < html\_name> | <data\_EXCELfile\_name>$<column\_number>|

sel\_comboboxbycsstext | < html\_css> | <value>|

sel\_comboboxbyxpathtext | < html\_xpath> | <value>|

sel\_comboboxbynamevalue | < html\_name> | <value>|

**Compare text**

sel\_XPATHtextvalidate |<html\_xpath>| <value\_to\_validate>|

sel\_IDtextvalidate | <html\_id> |<value\_to\_validate>|

sel\_CSVtextvalidateCSS | < html\_css> | <data\_EXCELfile\_name>$<column\_number>, |

(bổ sung)

sel\_NAMEtextvalidate

sel\_CSStextvalidate

**Take photo and compare img:**

sel\_screenshotC | <img\_name> |

sel\_screenshotX | <img\_name> in looping |

sel\_compareimg | <img1\_path>.png|<img2\_path>.png|

**Get and compare property CSS from a node:**

sel\_getprob |<id\_name>|<id\_CSSproperty>|<id\_CSSexpectation\_value>|

**Keys for wait**

sel\_eXwaitforXPATH | <time\_for\_waiting> | <xpath\_to\_wait> |

sel\_eXwaitforID | <time\_for\_waiting> | <id\_to\_wait> |

sel\_iMwaitforXPATH | <time\_for\_waiting> | <site\_address\_to\_load> | <xpath\_to\_wait> |

sel\_iMwaitforID | <time\_for\_waiting> | <site\_address \_to\_load> | <id\_to\_wait> |

(bổ sung)

sel\_eXwaitforCSSSelector

sel\_eXwaitforLinkText

sel\_eXwaitforNAME

**Hit ENTER**

(bổ sung)

sel\_keyENTERid | <id\_name> |

sel\_keyENTERname | <name> |

sel\_keyENTERcss | <css\_link> |

**Open URL**

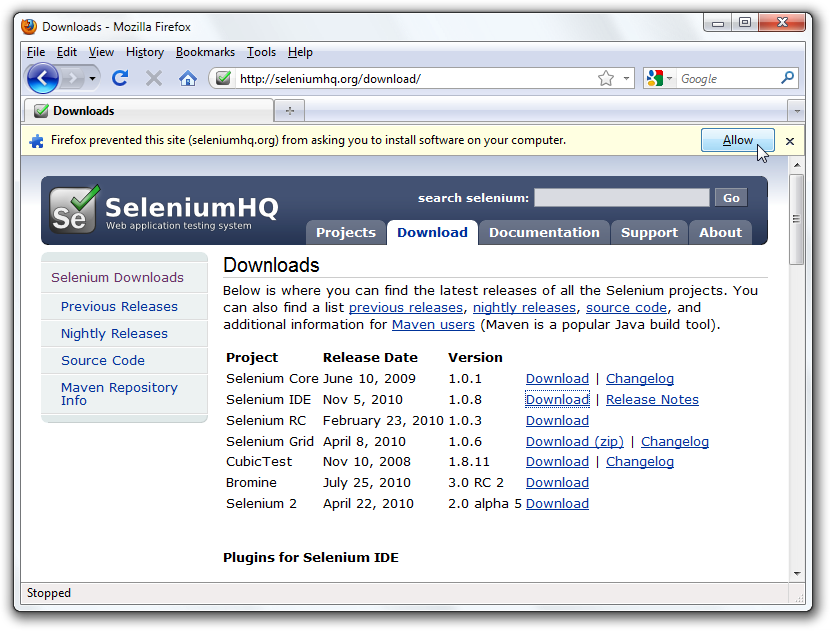
sel\_openurl | <url\_name> |

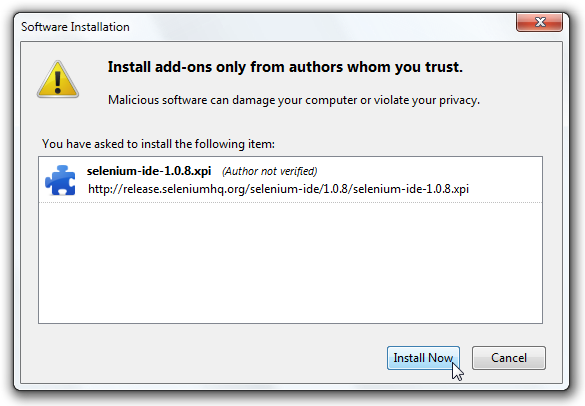
sel\_gotourl | <part\_of\_url\_name> |

# 6.

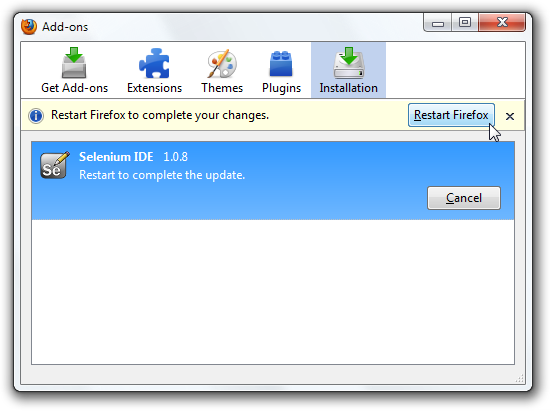
Cài Selenium IDE để tìm xpath và id của HTML node

Lấy từ SeleniumHQ hoặc từ Firefox addin.

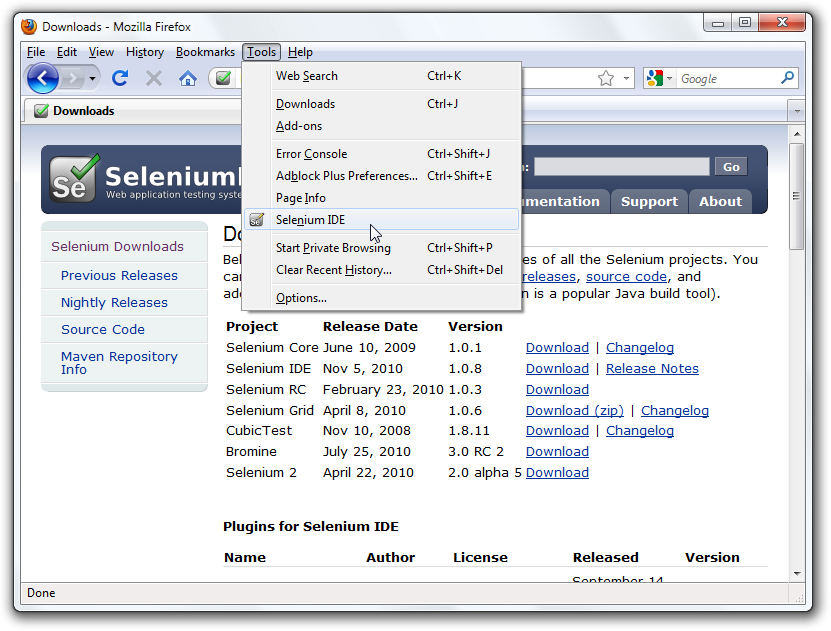


Down và cài cắp lên firefox

Accept



Cài và khơi động lại firefox



Sau đó record và tìm đc xpath/id/css của node HTML chúng ta cần

