

CÔNG CỤ KIỂM CHỨNG PHẦN MỀM

.....

CÔNG CỤ KIỂM THỬ TỰ ĐỘNG (SELENIUM)



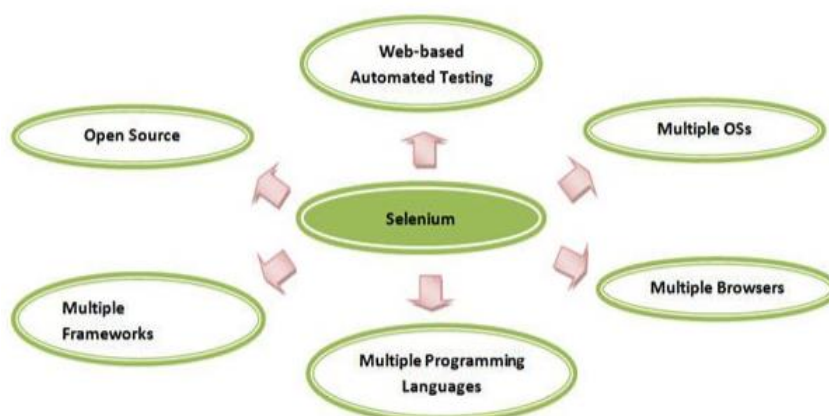
MỤC LỤC

| | |
|---|----|
| 1. Selenium | 2 |
| 1.1 Selenium là gì? | 2 |
| 1.2 Mục đích của kiểm thử tự động | 2 |
| 1.3 Ưu và nhược điểm của Selenium | 2 |
| 1.3.1 Ưu điểm | 2 |
| 1.3.2 Nhược điểm | 2 |
| 1.4 Cấu trúc của Selenium | 3 |
| 1.4.1 Selenium IDE | 3 |
| 1.4.2 Selenium RC | 3 |
| 1.4.3 WebDriver | 4 |
| 1.4.4 Selenium Grid | 4 |
| 1.5 So sánh Selenium và QTP | 4 |
| 2. Selenium IDE | 6 |
| 2.1 Hướng dẫn cài đặt Selenium IDE | 6 |
| 2.2 Các loại command của Selenium IDE | 8 |
| 2.2.1 Actions | 8 |
| 2.2.2 Accessors | 8 |
| 2.2.3 Assertions | 8 |
| 2.3 Các câu lệnh kiểm tra của Selenium IDE | 8 |
| 2.3.1 Các câu lệnh về đối tượng UI | 8 |
| 2.3.2 Các câu lệnh về Text | 9 |
| 2.3.3 Các câu lệnh về tổng thể trang web | 9 |
| 2.4 Xác định đối tượng UI | 9 |
| 2.4.1 Đối tượng UI - Locators | 9 |
| 2.4.2 Xác định đối tượng UI với Selenium IDE | 10 |
| 2.5 Các câu lệnh trong Selenium IDE Flow Control | 11 |
| 2.6 Các tính năng của Selenium IDE | 11 |
| 2.7 Kiểm thử theo hướng Data Driven | 12 |
| 2.7.1 Data Driven Testing là gì? | 12 |
| 2.7.2 Hướng dẫn cài đặt SelBlocks | 12 |
| 2.7.3 Câu lệnh trong SelBlocks | 13 |
| 2.8 Quy trình thực hiện kiểm thử tự động với Selenium IDE | 13 |
| 2.9 Hướng dẫn sử dụng Selenium IDE | 13 |
| 2.9.1 Demo 1 | 13 |
| 2.9.2 Demo 2 | 16 |
| 2.9.3 Demo 3 | 18 |

1. Selenium

1.1 Selenium là gì?

- Selenium là công cụ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau
- Selenium chạy trên nhiều trình duyệt và các hệ điều hành có thể kiểm soát bởi nhiều ngôn ngữ lập trình.



Selenium supports a broad range of browsers, technologies and platforms

1.2 Mục đích của kiểm thử tự động

Kiểm thử tự động giúp giảm bớt công sức thực hiện, tăng độ tin cậy, giảm sự nhầm lẫn, giảm chi phí cho tổng quá trình kiểm thử và rèn luyện kỹ năng lập trình cho kiểm thử viên

1.3 Ưu và nhược điểm của Selenium

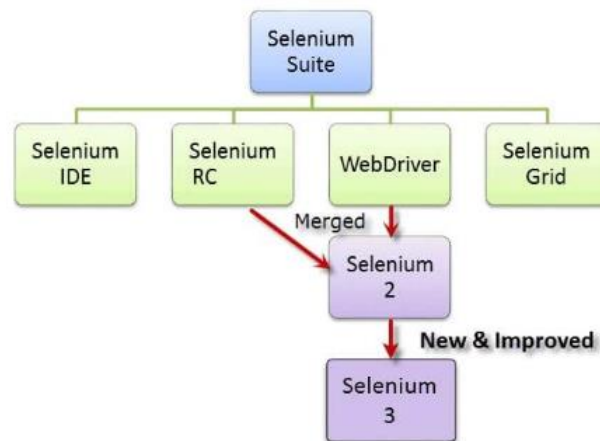
1.3.1 Ưu điểm

- Là công cụ mã nguồn mở
- Đơn giản, dễ cài đặt, dễ làm việc
- Sử dụng miễn phí
- Dễ dàng ghi và phát lại các thao tác của người dùng với trình duyệt
- Các lựa chọn trường thông minh sẽ sử dụng ID, Name, XPath khi cần thiết
- Tự động hoàn chỉnh cho tất cả các lệnh Selenium thường gặp
- Gỡ lỗi và thiết lập các điểm dừng
- Lưu các kiểm thử như là với các định dạng khác nhau, ví dụ: HTML, PHP ...
- Hỗ trợ các tệp tin selenium user-extensions.js
- Dễ dàng điều chỉnh thông qua các plugin

1.3.2 Nhược điểm

- Không thể đảm bảo bất kỳ add-in nào được yêu cầu nếu selenium không thể nhận ra các đối tượng
- Hỗ trợ được cung cấp cho selenium là rất ít

1.4 Cấu trúc của Selenium



1.4.1 Selenium IDE

Selenium Integrated Development Environment (Selenium IDE) là một công cụ cho phép chúng ta Record / Playback một test script. Đây là một add-on hỗ trợ cho Firefox, chúng ta cũng có thể Record trên trình duyệt Firefox.

| Ưu điểm | Khuyết điểm |
|---|---|
| Mã nguồn mở | Không hỗ trợ biểu thức điều kiện/ vòng lặp |
| Cộng đồng hỗ trợ | Không hỗ trợ test report |
| Hỗ trợ nhiều ngôn ngữ lập trình | Không hỗ trợ win app kể cả win dialog như download/upload – ngoại trừ browser alarm |
| Hỗ trợ chạy trên nhiều OS khác nhau với mức độ chỉnh sửa script hầu như là không có | Thực hiện kiểm thử chậm so với Selenium RC và WebDriver |
| Chạy testcase ở background | Không hỗ trợ database/ mobile testing |
| | Khó khăn khi xử lý những case/workflow phức tạp |

1.4.2 Selenium RC

Selenium Remote Control (Selenium RC) là công cụ kiểm tra web tự động đầu tiên cho phép người dùng sử dụng ngôn ngữ lập trình. Test script có thể hỗ trợ các ngôn ngữ lập trình khác nhau như: java, C#, PHP, Python, Perl, Ruby

| Ưu điểm | Khuyết điểm |
|---|--|
| Nhiều trình duyệt và đa nền tảng | Cài đặt phức tạp hơn IDE |
| Có thể hỗ trợ việc lặp lại hoặc câu lệnh có điều kiện cho tập lệnh kiểm tra | Yêu cầu phải có kiến thức lập trình |
| Có thể hỗ trợ kiểm tra theo dữ liệu | Cần phải có Selenium RC Server để chạy |

| | |
|--|---|
| Đã hoàn chỉnh API | API có các lệnh thừa và khó hiểu |
| Có thể sẵn sàng hỗ trợ các trình duyệt mới | Tương tác với trình duyệt ít hơn thực tế |
| Thực hiện nhanh hơn IDE | Kết quả không phù hợp và sử dụng javascript |
| | Thời gian thực thi chậm hơn WebDriver |

1.4.3 WebDriver

WebDriver tốt hơn Selenium IDE và Selenium RC về nhiều mặt. WebDriver thực hiện một cách hiện đại và ổn định hơn trong tự động hóa các hành động của trình duyệt. WebDriver không giống Selenium RC không dựa vào javascript dành cho tự động hóa. WebDriver điều khiển trình duyệt bằng cách trực tiếp liên lạc với WebDriver. Các ngôn ngữ được hỗ trợ giống với những ngôn ngữ trong Selenium RC như Java, C#, PHP, Python, Perl, Ruby

| Ưu điểm | Khuyết điểm |
|---|--|
| Cài đặt đơn giản hơn Selenium RC | Cài đặt phức tạp hơn Selenium IDE |
| Giao tiếp trực tiếp với trình duyệt | Yêu cầu kiến thức lập trình |
| Sự tương tác của trình duyệt thực tế hơn | Không thể sẵn sàng hỗ trợ các trình duyệt mới |
| Không cần một thành phần riêng biệt như RC Server | Không có cơ chế tích hợp để ghi lại các thông điệp thời gian chạy và tạo ra các kết quả kiểm tra |
| Thời gian thực hiện nhanh hơn IDE và RC | |

1.4.4 Selenium Grid

Selenium Grid là một công cụ được sử dụng cùng với Selenium RC để chạy thử nghiệm song song trên các máy khác nhau và các trình duyệt khác nhau cùng một lúc (thực hiện song song có nghĩa là chạy nhiều testcase cùng một lúc). Đặc biệt hơn nữa là Selenium Grid còn cho phép chúng ta thực thi kiểm thử với chế độ phân tán

Đặc điểm:

- Cho phép chạy đồng thời các testcase trong nhiều trình duyệt và môi trường
- Tiết kiệm nhiều thời gian
- Sử dụng khái niệm hub-and-nodes. Hub hoạt động như một nguồn chính của lệnh Selenium cho mỗi kết nối với nó.

1.5 So sánh Selenium và QTP

| STT | Tiêu chí | Selenium | QTP |
|-----|--------------------------|---|--------------------------------------|
| 1 | Chi phí bản quyền | Đây là công cụ mã nguồn mở cho nên không có chi phí bản quyền, hoàn toàn miễn phí | Rất mắc...khoảng 5000 USD cho 1 máy |
| 2 | Thực sự mô | Selenium thực hiện hành động trong nền của | QTP thực hiện kiểm thử tự động tương |

| | | | |
|----|--|---|---|
| | phỏng hành động của người dùng cuối | trình duyệt tức là người sử dụng, có thể thực hiện các kiểm thử tự động với trình duyệt được thu nhỏ tối thiểu (minimized). | đương với một người thực hiện các thao tác vào ứng dụng. Vì vậy, các trình duyệt không có thể được giảm thiểu (người sử dụng không thể thực hiện bất kỳ hành động nào khác trên máy tính trong khi kiểm thử đang chạy). |
| 3 | Plug-ins | Có và hoàn toàn miễn phí | Có. Người sử dụng phải trả tiền bản quyền |
| 4 | Nỗ lực và kỹ năng để sử dụng công cụ | Người dùng cần phải có kỹ năng tốt về lập trình và coding vì đây là điều kiện cần. | Đòi hỏi nỗ lực ít hơn để tạo ra một kịch bản kiểm thử, vì nó có một môi trường phát triển rất tốt, thân thiện với người sử dụng. |
| 5 | Dễ dàng tạo ra các kịch bản kiểm thử? | Tốt. Selenium IDE Recorder không mạnh mẽ như QTP nhưng là một công cụ miễn phí, nhiều hành động không được ghi lại bởi các IDE và có thể tự nhập vào. | Hoàn hảo. |
| 6 | Hỗ trợ trên các HĐH | Windows, Linux, Unix và Macintosh. | Windows. |
| 7 | Loại ứng dụng | Web based applications. | Web-based và desktop applications. |
| 8 | Trình duyệt | IE, Firefox, Safari, Opera và một vài trình duyệt khác | IE, Chrome và Firefox. |
| 9 | Ngôn ngữ phát triển kịch bản kiểm thử | Java, Ruby, Python, C#, Perl, Groovy và nhiều ngôn ngữ khác | VBScript và/hoặc JavaScript. |
| 10 | Tích hợp với các công cụ quản lý thử nghiệm | Khi phát triển kiểm thử (test case) sử dụng C# trong VS thì có thể sử dụng MTM để quản lý test cases | HP Quality Center và TestDirector |
| 11 | Môi trường | Kịch bản thử nghiệm có thể được phát triển | Kịch bản thử nghiệm chỉ có thể được |

| | | | |
|----|---|---|---|
| | phát triển kiểm thử | trong nhiều trình soạn thảo như Eclipse, Visual Studio, Netbeans... | phát triển trong QTP. |
| 12 | Hỗ trợ kỹ thuật | Vì đây là công cụ mã nguồn mở nên không có hỗ trợ kỹ thuật chính thức. | QTP cung cấp hỗ trợ kỹ thuật rất tốt qua điện thoại, mail, web, forum. |
| 13 | Quản lý và lưu trữ đối tượng giao diện (UI-Object) | Không. Đối tượng có thể được quản lý bằng cách sử dụng giao diện người dùng mở rộng (UI-Element user extension) hoặc properties files. Tất cả các phương pháp trên đều phải sử dụng code cứng (hard-code) | Có. QTP được xây dựng trong với Object Repository. Quản lý Object repository là khá dễ dàng trong QTP |
| 14 | Hỗ trợ cho các hộp thoại | Chỉ hỗ trợ 1 phần. Không thể thực hiện một số hành động như lấy tiêu đề của hộp thoại. | Có. QTP hỗ trợ tất cả các loại hộp thoại. |
| 15 | Hỗ trợ cho các kịch bản tải lên tập tin | Không. Selenium không hỗ trợ kịch bản này vì hạn chế JavaScript. | QTP không có vấn đề gì trong |

Selenium là lựa chọn hoàn hảo nếu...

- Developer chịu trách nhiệm phát triển và bảo trì các kiểm thử
- Những kiểm thử bạn đang phát triển sẽ là 1 phần của một cơ sở hạ tầng lớn hơn
- Chắc chắn là bạn sẽ thử nghiệm những kiểm thử trên trình duyệt mà thôi
- Bạn nghĩ là bạn sẽ thực hiện những kiểm thử trên những môi trường khác ngoài Windows
- Bạn muốn sử dụng những công cụ miễn phí

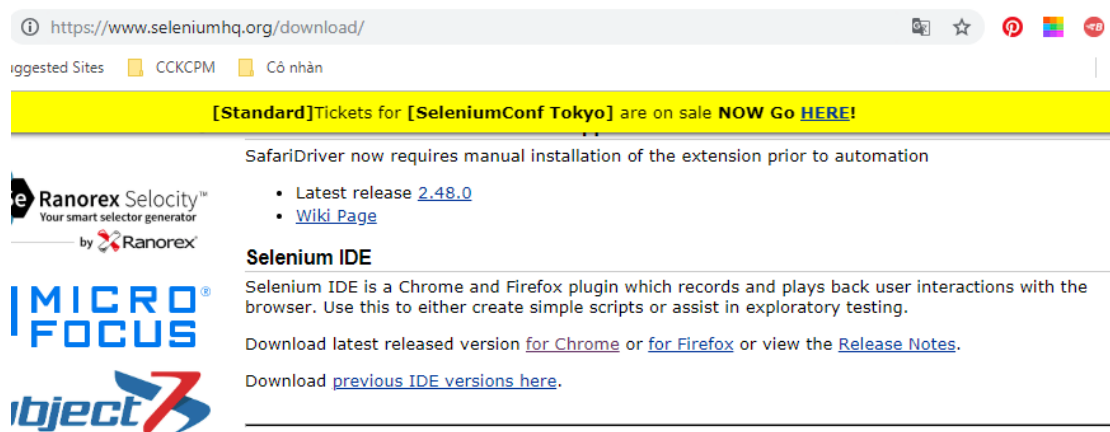
QTP có lẽ là lựa chọn tốt nếu...

- Bạn muốn phát triển và duy trì các kiểm thử một cách nhanh chóng và dễ dàng.
- Nếu bạn cần kiểm thử đáng kể bên ngoài trình duyệt

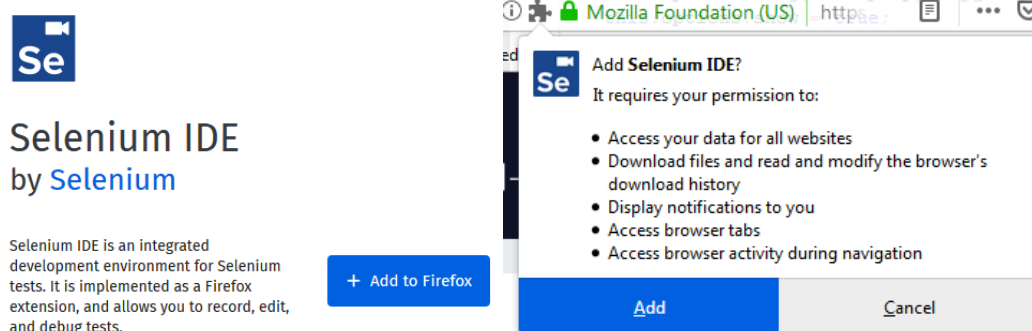
2. Selenium IDE

2.1 Hướng dẫn cài đặt Selenium IDE

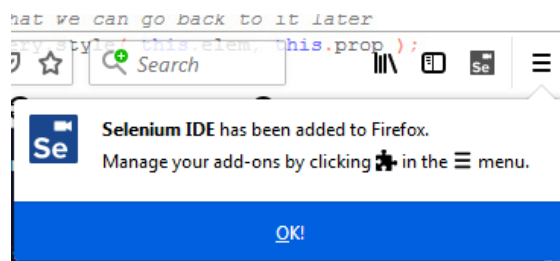
Bước 1: Mở trang chủ của Selenium <https://www.seleniumhq.org/download/> → chọn trình duyệt mà bạn muốn để cài tiện ích



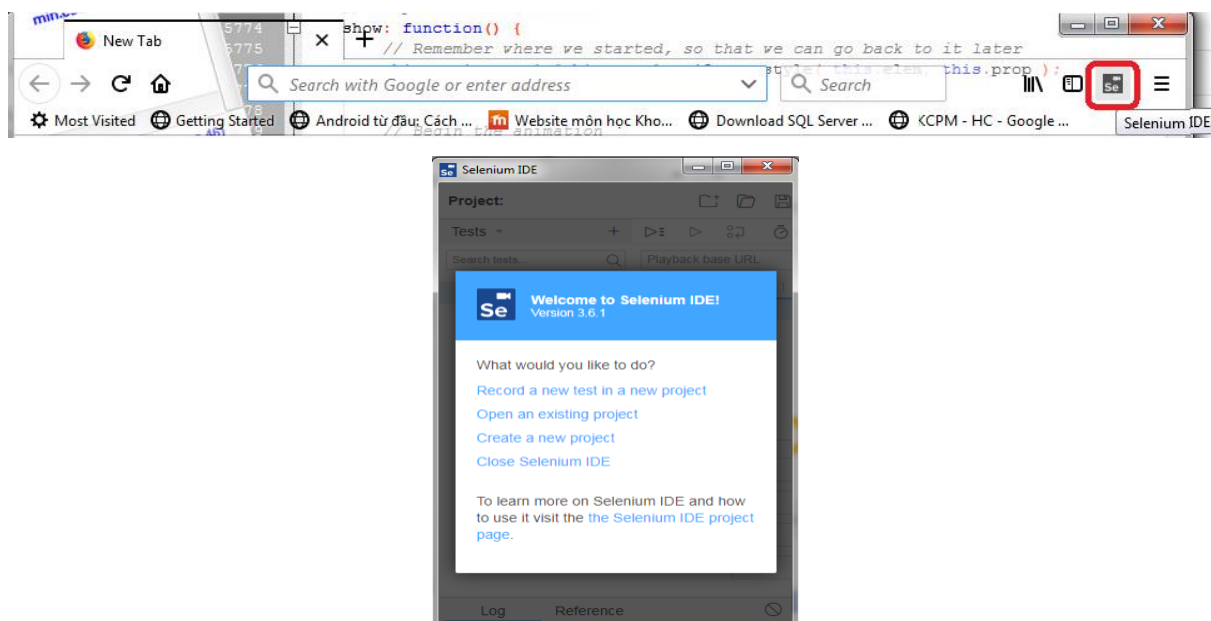
Bước 2: Cài đặt Selenium IDE giống như cài đặt một plugin trên trình duyệt Firefox



Kết quả: Selenium IDE đã được thêm vào Firefox



Bước 3: Khởi động Selenium IDE bằng cách click vào icon Selenium trên menu bar của Firefox



2.2 Các loại command của Selenium IDE

Các câu lệnh Selenium có 3 loại: Actions, Accessors và Assertions

2.2.1 Actions

Actions (hành động): là những command tương tác trực tiếp với các phần tử trên trang web (Page elements)

- Click: click vào một phần tử bất kỳ
- Type: nhập dữ liệu vào một phần tử trên trang

2.2.2 Accessors

Accessors (truy cập): là những lệnh cho phép lưu trữ giá trị cho biến

Ví dụ: storeTitle command là một accessor đọc giá trị page title và lưu giá trị vào một biến. Nó không có khả năng tương tác với bất kỳ một element nào trên trang.

2.2.3 Assertions

Assertions (xác nhận): là những lệnh xác minh một điều kiện nào đó được thỏa mãn

Tất cả các Assertions Selenium có thể được sử dụng trong 3 cách thức:

1. **assert:** khi một lệnh assert fail → quá trình test dừng ngay lập tức
2. **verify:** khi lệnh verify fail → selenium IDE ghi nhận kết quả failure và tiếp tục quá trình test
3. **waitFor:** tạo ra một lệnh chờ một điều kiện nào đó xảy ra rồi mới tiếp tục lệnh tiếp theo. Theo mặc định mỗi lệnh chờ sẽ có thời gian là 30 giây (thời gian này có thể thay đổi trong option của Selenium IDE). Nếu condition xảy ra thì sẽ tiếp tục các lệnh khác, trường hợp condition không xảy ra thì sẽ ghi nhận kết quả Failure và tiếp tục thực hiện lệnh tiếp theo.

2.3 Các câu lệnh kiểm tra của Selenium IDE

2.3.1 Các câu lệnh về đối tượng UI

Đối với các đối tượng UI, thông thường chúng ta chỉ có hai loại kiểm tra: tồn tại hay không tồn tại

- verifyElementPresent/ assertElementPresent: hai câu lệnh này dùng để kiểm tra một đối tượng UI tồn tại trên trang web
- verifyElementNotPresent/ assertElementNotPresent: hai câu lệnh này dùng để kiểm tra một đối tượng UI không tồn tại trên trang web

Bên cạnh hai loại kiểm tra chính, Selenium IDE còn hỗ trợ chúng ta kiểm tra đối tượng UI ở mức chi tiết hơn như:

- Cách hiển thị của đối tượng UI trên trang web
 - **Chiều cao:** verifyElementHeight/assertElementHeight – verifyNotElementHeight/assertNotElementHeight
 - **Chiều rộng:** verifyElementWidth/assertElementWidth – verifyNotElementWidth/assertNotElementWidth

- **Toạ độ trái so với trang web:** `verifyElementPositionLeft/assertElementPositionLeft` – `verifyNotElementPositionLeft/assertNotElementPositionLeft`
- **Toạ độ trên so với trang web:** `verifyElementPositionTop/assertElementPositionTop` – `verifyNotElementPositionTop/assertNotElementPositionTop`
- Các đối tượng UI được thiết kế với **thuộc tính đặc biệt** và chúng ta cần kiểm tra thuộc tính đó: `verifyAttribute/assertAttribute` – `verifyNotAttribute/assertNotAttribute`
- Đối tượng UI đặc biệt như **Checkbox** hay **RadioBox**: `verifyChecked/assertChecked` – `verifyNotChecked/assertNotChecked`

2.3.2 Các câu lệnh về Text

Chúng ta không cần kiểm tra một đối tượng UI, chúng ta muốn biết một dòng text có hay không có xuất hiện trên trang web. Lúc này, chúng ta có các câu lệnh kiểm tra về Text:

- **`verifyTextPresent/assertTextPresent`:** hai câu lệnh này dùng để kiểm tra một dòng text tồn tại trên trang web.
- **`verifyTextNotPresent/assertTextNotPresent`:** hai câu lệnh này dùng để kiểm tra một dòng text không tồn tại trên trang web.

Vì bốn câu lệnh kiểm tra text này không tương tác với đối tượng UI nào, nên phần Target của câu lệnh chính là dòng text mà chúng ta muốn kiểm tra.

Trong trường hợp đặc biệt, chúng ta muốn kiểm tra text của một đối tượng UI xác định, chúng ta có hai câu lệnh **`verifyText/assertText`**. Lúc này, target chính là đối tượng UI của chúng ta, và Value chính là dòng text cần kiểm tra.

2.3.3 Các câu lệnh về tổng thể trang web

Selenium IDE hỗ trợ chúng ta các câu lệnh có thể làm việc với toàn bộ trang web như:

- **`verifyBodyText/assertBodyText`:** Kiểm tra toàn bộ text trên trang web.
- **`verifyCssCount/assertCssCount`:** Kiểm tra số lượng đối tượng UI sử dụng chung một thiết kế CSS
- **`verifyXPathCount/assertXPathCount`:** Kiểm tra số lượng đối tượng UI có cùng XPath

2.4 Xác định đối tượng UI

2.4.1 Đối tượng UI - Locators

Selenium hỗ trợ chúng ta xác định UI dựa trên các thuộc tính và chúng ta gọi một cách xác định đối tượng UI thông qua thuộc tính của nó là locator, như sau:

- ID
- Name
- Link Text
- CSS Selector
 - Tag và ID

- Tag và Class
- Tag và Attribute
- Inner Text

○ XPath

Đối với ID: thuộc tính đơn nhất và ổn định nhất của hệ thống phần mềm, nếu sử dụng ID cho việc xác định UI thì tốt. Nhưng tất cả chúng ta đều biết, hiếm có một nhà phát triển phần mềm nào lại gán ID cho mọi đối tượng UI, vì ID không phải là một thuộc tính bắt buộc và nó không hiển thị lên giao diện

Đối với Name: thuộc tính Name đôi khi không đơn nhất, cũng không có một ràng buộc nào bắt Name đơn nhất.

Đối với Link Text: thuộc tính Link Text là không đơn nhất và cũng không ổn định. Một trang web có thể có nhiều liên kết đến một trang khác hay thay đổi từ ngữ nhưng không đổi ý nghĩa.

Đối với CSS Selector: sử dụng CSS là chúng ta phụ thuộc vào cách thiết kế web của nhà phát triển. Và hiển nhiên, CSS Selector không đơn nhất, không một nhà phát triển nào tạo ra một CSS để dùng cho một đối tượng cả. Cho nên, chúng ta phải kết hợp CSS với Tag/ID, Tag/class ...

Đối với XPath: đây được xem là thuộc tính hay dùng nhất của Selenium. Tuy nhiên, cách này lại thiếu chính xác, thiếu đơn nhất và thiếu ổn định nhất trong tất cả các cách xác định đối tượng UI

2.4.2 Xác định đối tượng UI với Selenium IDE

Trong Selenium IDE, khi chúng ta record một testcase, IDE đã tự động xác định đối tượng UI cho chúng ta. Nhưng đôi khi thuộc tính IDE lấy ra không thể sử dụng được. ví dụ: Google có 2 đối tượng UI có cùng ID. Để xác định một đối tượng UI với Selenium IDE, chúng ta làm như sau:

- *Bước 1:* mở trang web mà chúng ta đang muốn kiểm tra
- *Bước 2:* mở selenium IDE
- *Bước 3:* gõ locator của đối tượng mà chúng ta muốn xác định vào Target textbox
- *Bước 4:* click Find button

Lưu ý: Nếu các locator của đối tượng mà chúng ta điền vào Target là đúng, UI sẽ sáng nhấp nháy trên trang web

Đây là cách chúng ta tạo ra các locator với từng phương thức khác nhau

| Phương thức | Cấu trúc | Ví dụ |
|---------------|-------------------------------------|--------------------------|
| ID | id= <i>id_của_đối_tượng</i> | id=email |
| Name | name= <i>name_của_đối_tượng</i> | name=username |
| Name | name= <i>name_của_đối_tượng</i> | name=tripType |
| Link Text | link= <i>link_text</i> | link=REGISTER |
| Tag/ID | css=tag# <i>id</i> | css=input#email |
| Tag/Class | css=tag. <i>class</i> | css=input.inputtext |
| Tag/Attribute | css=tag[attribute= <i>giá trị</i>] | css=input[name=lastName] |

| | | |
|---------------------|----------------------------------|-----------------------------------|
| Tag/Class/Attribute | css=tag.class[attribute=giá trị] | css=input.inputtext[tabindex=1] |
| XPath | XPath | //html/body//bookstore/book/title |

Ngoài ra, để có thể xác định một locator một cách nhanh chóng và chính xác hơn, các bạn có thể sử dụng hai công cụ khá mạnh của **Firefox** là **FireBug** và **XPath Finder**.

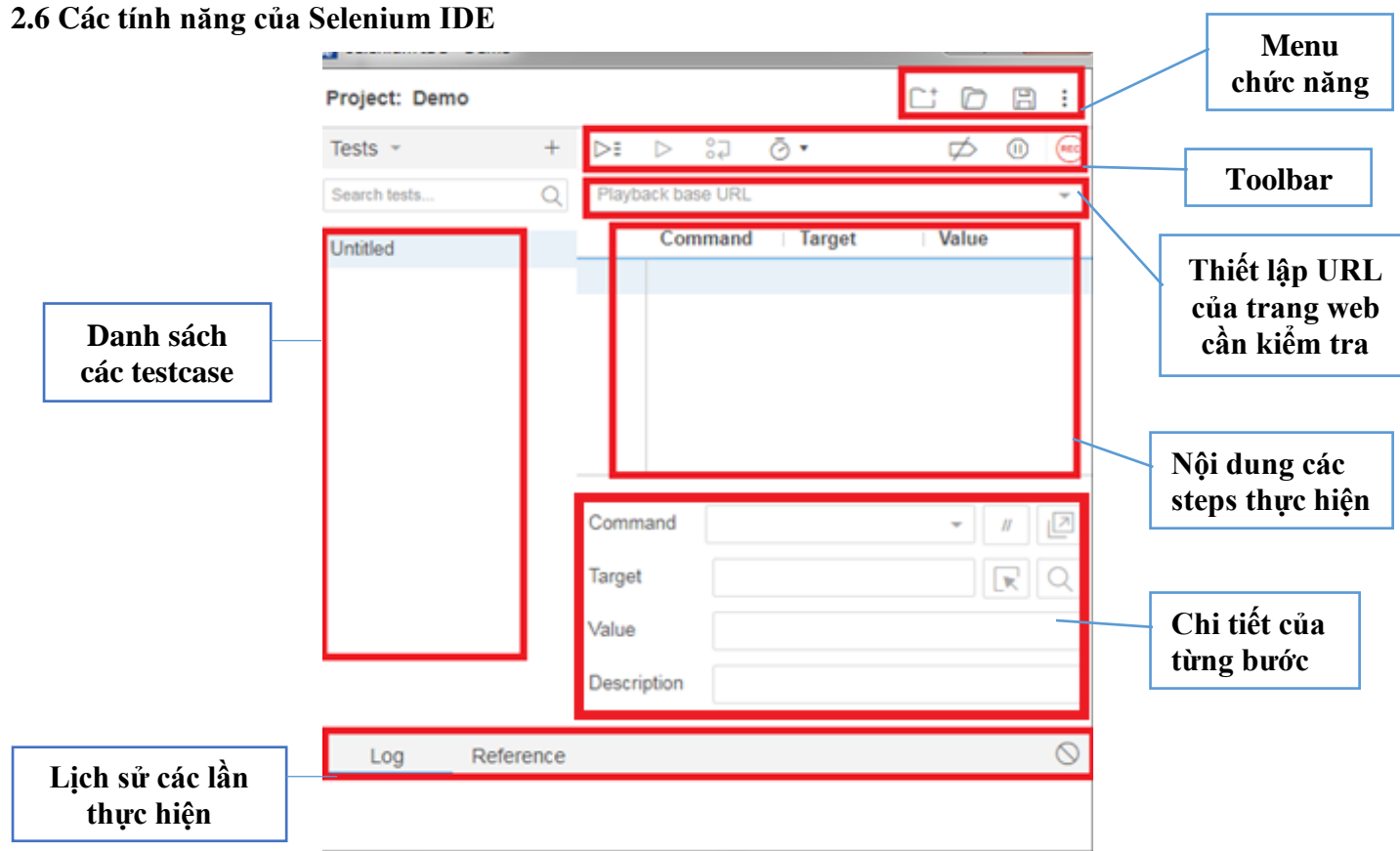
- **FireBug** giúp chúng ta đọc mã nguồn trang web một cách rõ ràng
- **XPath Finder** giúp chúng ta lấy XPath của đối tượng UI một cách chính xác và đơn giản

2.5 Các câu lệnh trong Selenium IDE Flow Control

Trong Command Dropdown List của Selenium IDE, chúng ta sẽ có 7 câu lệnh mới:

- label | mylabel – Tạo ra một nhãn
- goto | mylabel – Di chuyển đến một nhãn
- gotoLabel | mylabel – Tương tự như câu lệnh “goto”
- gotoIf | expression – Đi đến một nhãn nếu thỏa điều kiện
- while | expression – Lặp một đoạn mã với điều kiện
- endWhile – Kết thúc một vòng lặp
- push | value | arrayName – Điền một giá trị vào bên trong một mảng

2.6 Các tính năng của Selenium IDE



2.7 Kiểm thử theo hướng Data Driven

2.7.1 Data Driven Testing là gì?

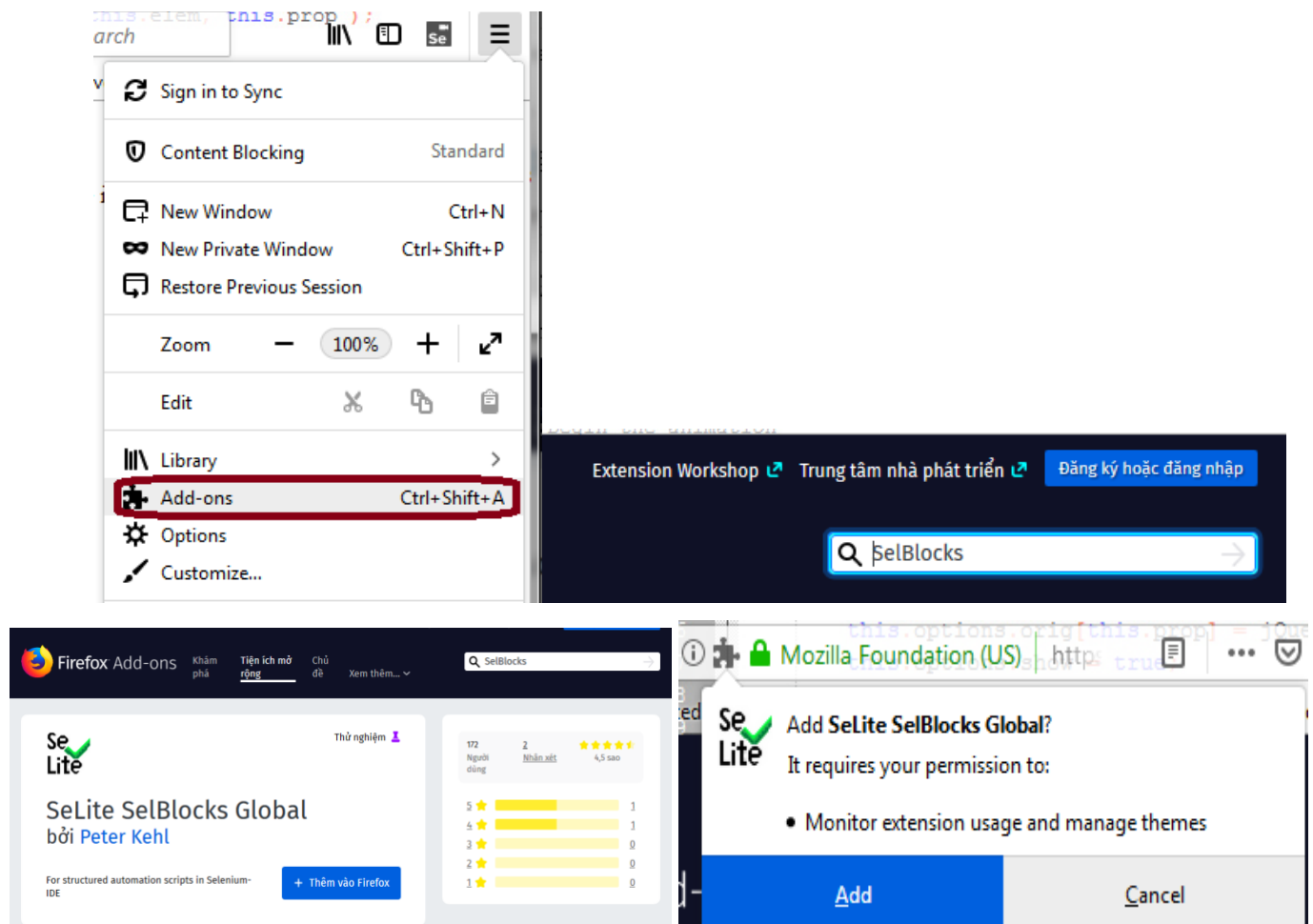
- Data Driven Testing nghĩa là chỉ dùng một kịch bản test nhưng có thể verify được nhiều trường hợp khác nhau với đầu vào/ đầu ra từ file data bên ngoài (.csv, .txt, .xls ...) thay vì sử dụng data cứng trong mỗi lần chạy

Ví dụ: 100 test scripts, mỗi scripts đều phải yêu cầu login (email/password) vào hệ thống để sử dụng, nếu bạn set cứng 2 biến email/password trong script thì mỗi lần thay đổi email/password phải sửa 100 lần cho 100 scripts. Thay vì đó chỉ cần tạo 1 file lưu trữ data của 2 biến đó và đưa tên biến vào 100 scripts, chỉ cần sửa 1 lần và sẽ apply cho toàn bộ

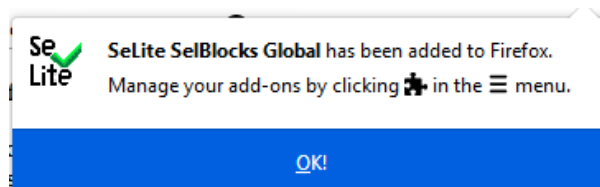
- Chúng ta sẽ sử dụng một add-on hỗ trợ cho Selenium IDE tên là SelBlocks, nó cho phép thêm các command về vòng lặp, điều kiện và data driven testing trong kịch bản test.

2.7.2 Hướng dẫn cài đặt SelBlocks

SelBlocks cài đặt tương tự như Selenium IDE



Kết quả



2.7.3 Câu lệnh trong SelBlocks

SelBlocks, bên cạnh việc hỗ trợ chúng ta thực thi kiểm thử theo hướng data-driven, SelBlocks cung cấp một số câu lệnh đặc biệt như:

- *if/elseIf/else*
- *try/catch/finally/throw*
- *for/foreach/while*
- *continue/break*
- *call/function/return*
- *loadXmlVars/loadJsonVars*
- *forXml/forJson*
- *exitTest*

Nhưng trong bài hướng dẫn sử dụng kiểm thử tự động, sẽ chỉ tập trung vào phần Data-Driven.

2.8 Quy trình thực hiện kiểm thử tự động với Selenium IDE

Để thực hiện kiểm thử tự động với Selenium IDE ta thực hiện theo các bước sau đây:

- *Bước 1:* Thiết lập base URL về trang web muốn kiểm tra
- *Bước 2:* Thực hiện viết các kịch bản kiểm thử (sử dụng các command kết hợp với các Element trên Webpage để tạo ra các hành động tương tác theo các testcase đã định nghĩa)
- *Bước 3:* Thực hiện kiểm thử, ghi nhận kết quả → Post Bug

Chú ý: việc viết kịch bản kiểm thử có thể làm bằng tay hoặc sử dụng chức năng record của Selenium để tự động ghi lại các thao tác kiểm thử

2.9 Hướng dẫn sử dụng Selenium IDE

2.9.1 Demo 1

Sử dụng chuột/ keyboard tương tác với trình duyệt tiến hành record và playback

❖ Kịch bản

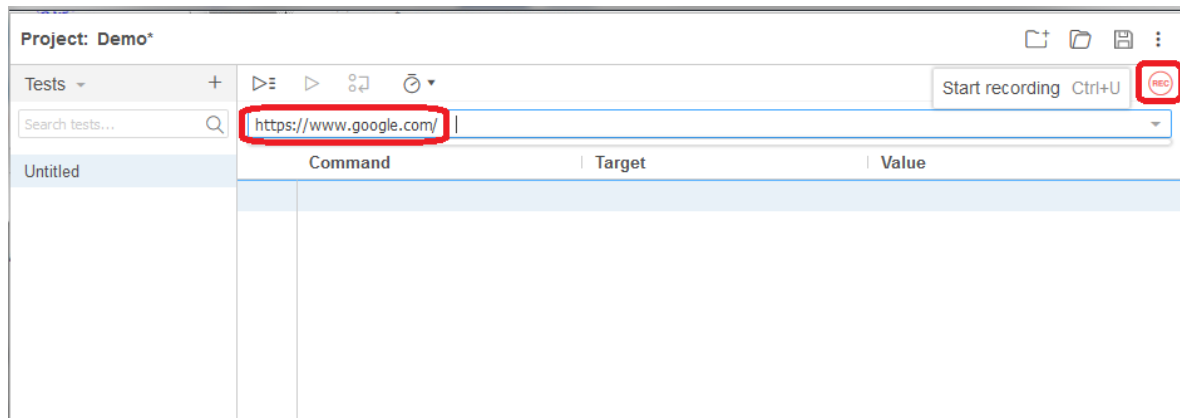
1. Truy cập trang web **Google**
2. Tìm kiếm từ khóa **Selenium**
3. Nhấp vào đường dẫn đến trang chủ của Selenium
4. Nếu tiêu đề của trình duyệt hiện tại không phải là **Downloads** thì nhấn nút **Download Selenium**

❖ Thực hiện trên Selenium IDE

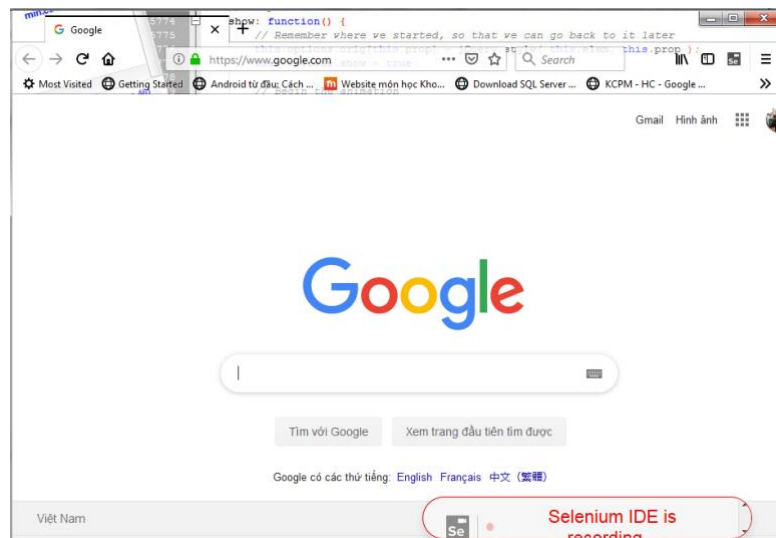
RECORD

Bước 1

- Nhập URL: <https://www.google.com/> và nhấn **Start Recording**



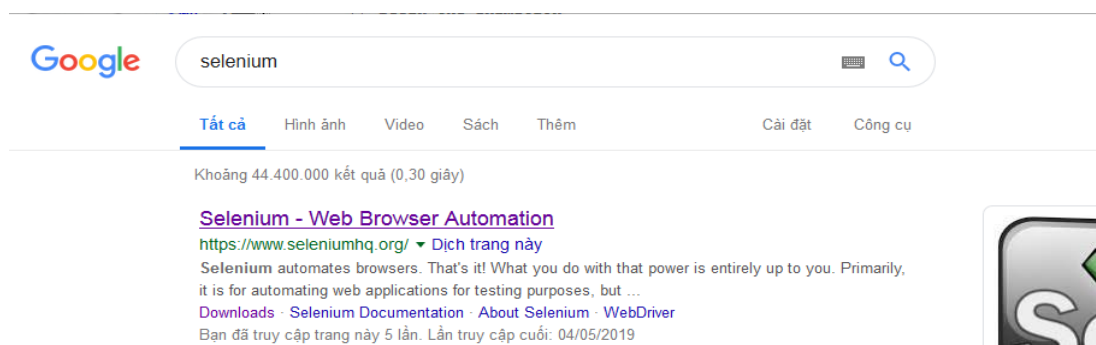
- Sau khi nhấn **Start Recording**, Firefox sẽ hiển thị trang Google và ở chế độ đang recording

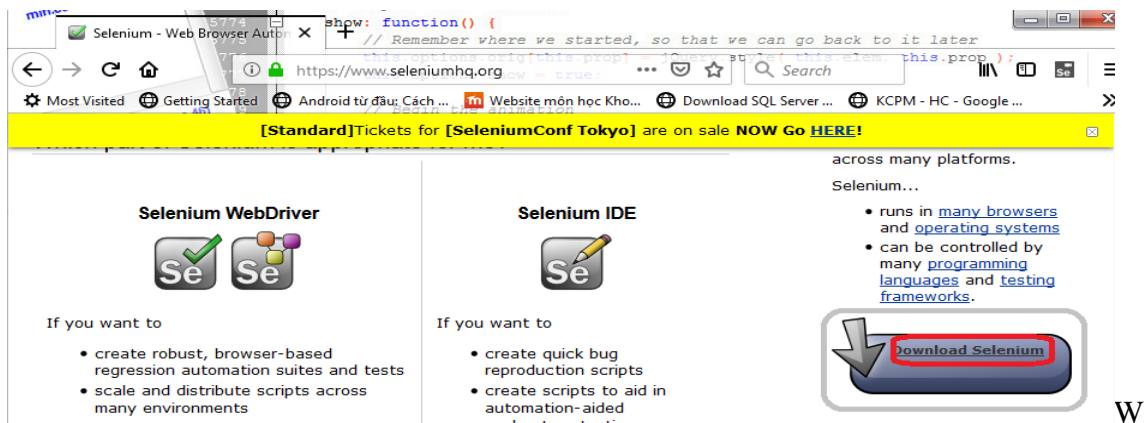


Bước 2

- Chúng ta sử dụng chuột/ keyboard tương tác với trình duyệt

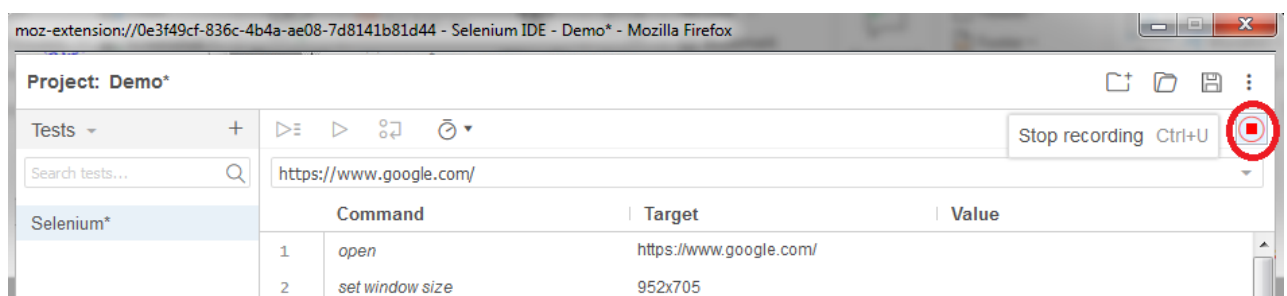
Nhập tìm kiếm “Selenium”, nhấn Enter → chọn “Selenium – Web Browser Automation” → nhấn “Download Selenium”





Bước 3

- Nhấn “Stop Record” để kết thúc việc ghi



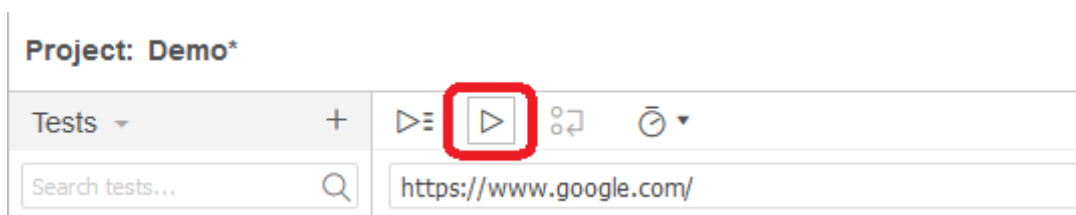
- Kết quả

| | Command | Target | Value |
|---|-----------------|---------------------------|---------------|
| 1 | open | https://www.google.com/ | |
| 2 | set window size | 952x705 | |
| 3 | type | name=q | selenium |
| 4 | send keys | name=q | \${KEY_ENTER} |
| 5 | click | css=g:nth-child(1).LC20Ib | |
| 6 | mouse over | css=g:nth-child(1).LC20Ib | |
| 7 | click | id=mainContent | |

Bước 4

PLAY BACK

Chọn “Run Current Test”



→ Chạy lại đúng như những gì chúng ta tương tác với trình duyệt

2.9.2 Demo 2

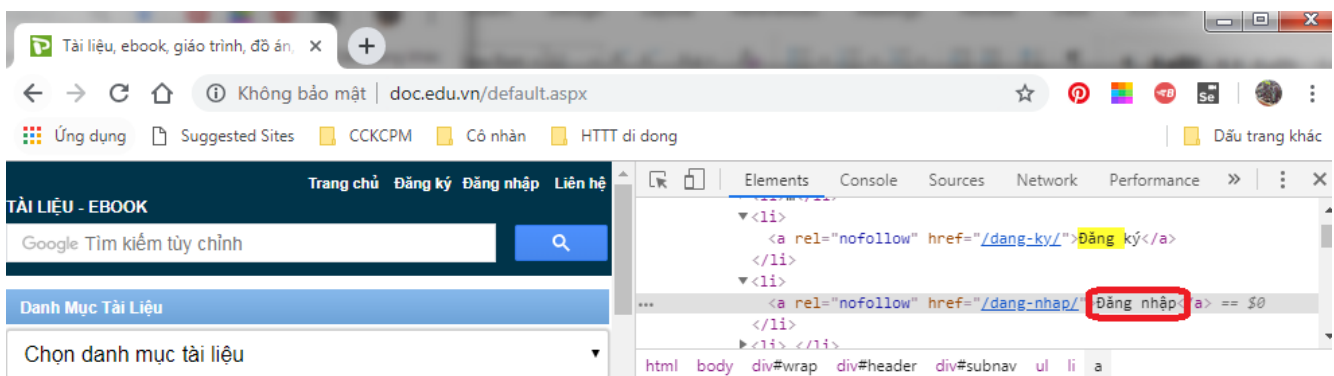
Sử dụng dòng lệnh nhập vào để tiến hành record và playback

Kịch bản: đăng nhập vào trang chủ ebook

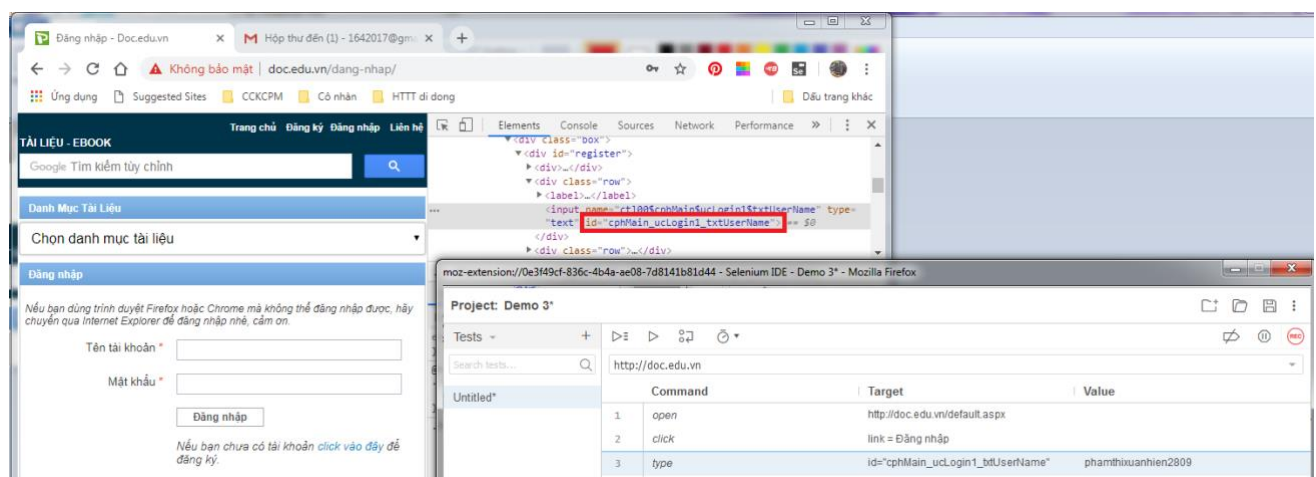
Bước 1: Nhập URL <http://doc.edu.vn>

Bước 2: Nhập các dòng lệnh

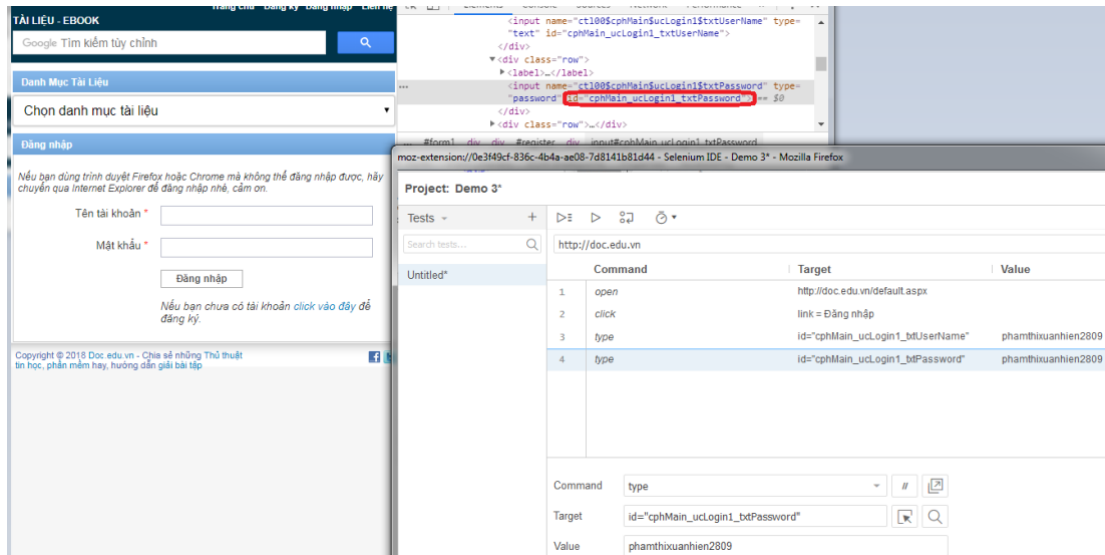
- Dòng lệnh 1: tương ứng với việc vào trang chủ của ebook
 - Open: <http://doc.edu.vn/default.aspx>
- Dòng lệnh 2: set window size: **982x788**
- Dòng lệnh 3: tương ứng với di chuyển chuột đến link đăng nhập trên menu bar
 - Command: **mouse over**
 - Target: **linkText=Đăng nhập**
- Dòng lệnh 4: tương ứng với click vào “**Đăng nhập**” trên menu bar trang chủ
 - Command: **click**
 - Target: **linkText=Đăng nhập**



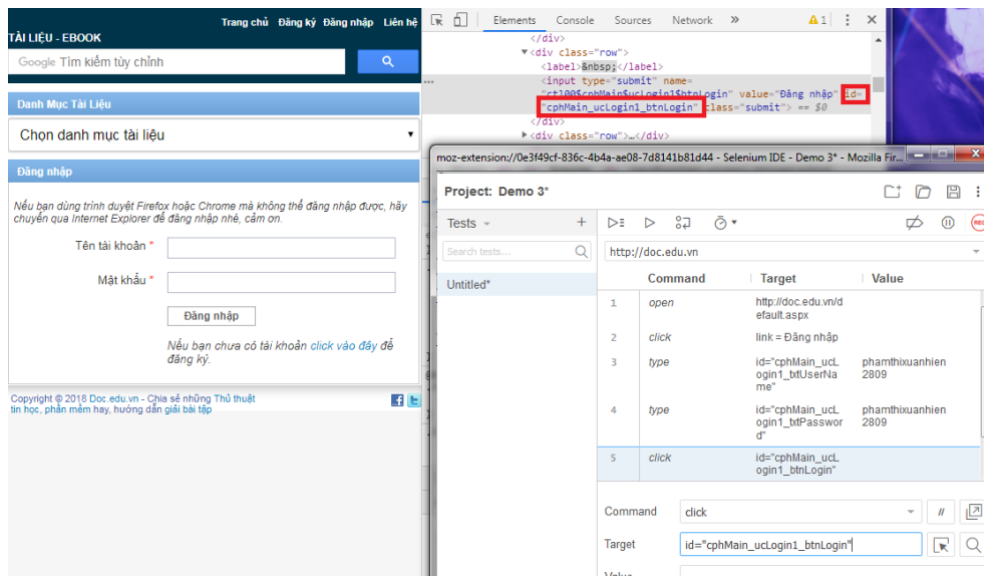
- Dòng lệnh 5: tương ứng với textbox **tên tài khoản**
 - Command: **type**
 - Target: **id=cphMain_ucLogin1_txtUserName**
 - Value: **phamthixuanhien2809**



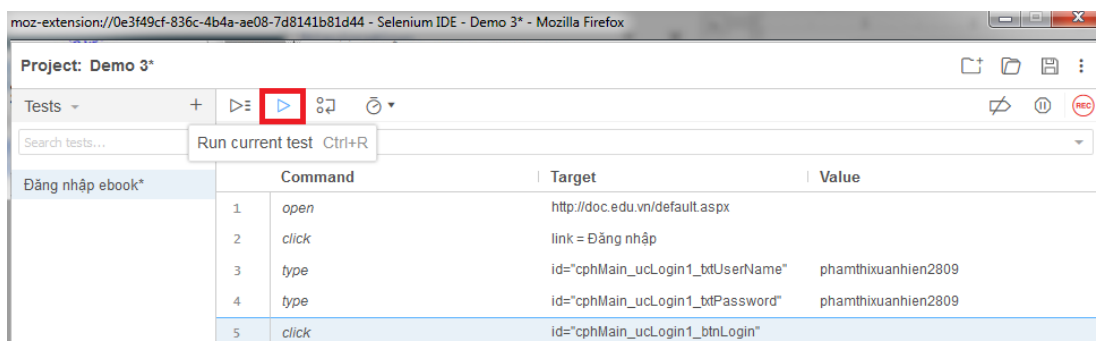
- Dòng lệnh 4: tương ứng với textbox **mật khẩu**
 - Command: **type**
 - Target: **id=cphMain_ucLogin1_txtPassword**
 - Value: **phamthixuanhien2809**



- Dòng lệnh 5: tương ứng với click vào button “Đăng nhập”
 - Click: **id=cphMain_ucLogin1_btnLogin**



Bước 3: Nhấn “**Run current test**” để chạy



Kết quả sau khi run

Project: Demo 3*

Tests ▾ +

Search tests... 🔍

http://doc.edu.vn

| | Command | Target | Value |
|-----------|-----------------|---------------------------------|---------------------|
| Login* | | | |
| Đăng nhập | | | |
| 1 | open | http://doc.edu.vn/default.aspx | |
| 2 | set window size | 982x788 | |
| 3 | mouse over | linkText=Đăng nhập | |
| 4 | click | linkText=Đăng nhập | |
| 5 | type | id=cphMain_ucLogin1_txtUserName | phamthixuanhien2809 |
| 6 | type | id=cphMain_ucLogin1_txtPassword | phamthixuanhien2809 |
| 7 | click | id=cphMain_ucLogin1_btnLogin | |
| 8 | // | | |

Command: click

Target: id=cphMain_ucLogin1_btnLogin

Value:

Description:

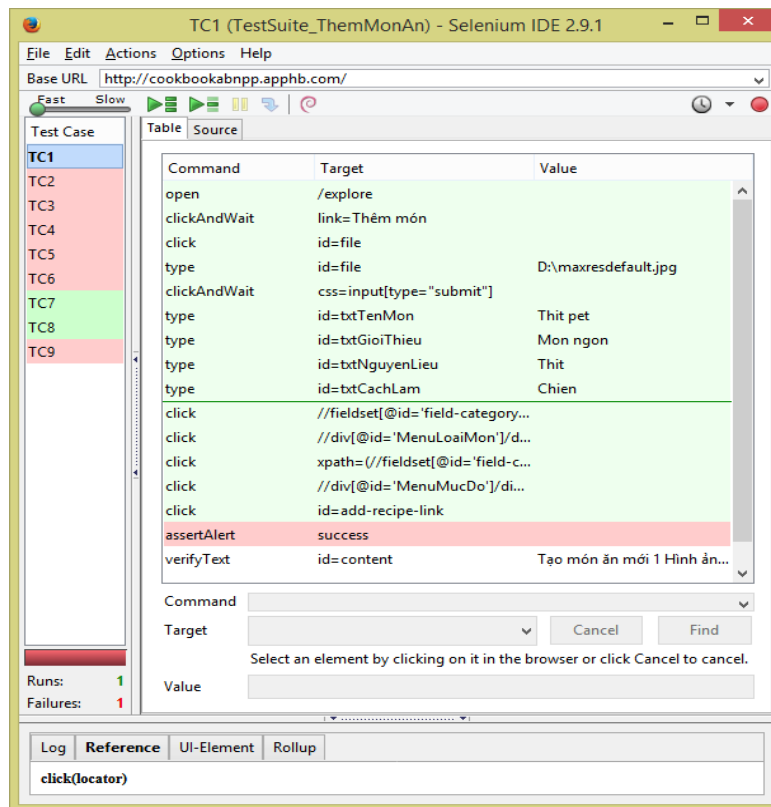
Log Reference

- 4. click on linkText=Đăng nhập OK
- 5. type on id=cphMain_ucLogin1_txtUserName with value phamthixuanhien2809 OK
- 6. type on id=cphMain_ucLogin1_txtPassword with value phamthixuanhien2809 OK
- 7. click on id=cphMain_ucLogin1_btnLogin OK

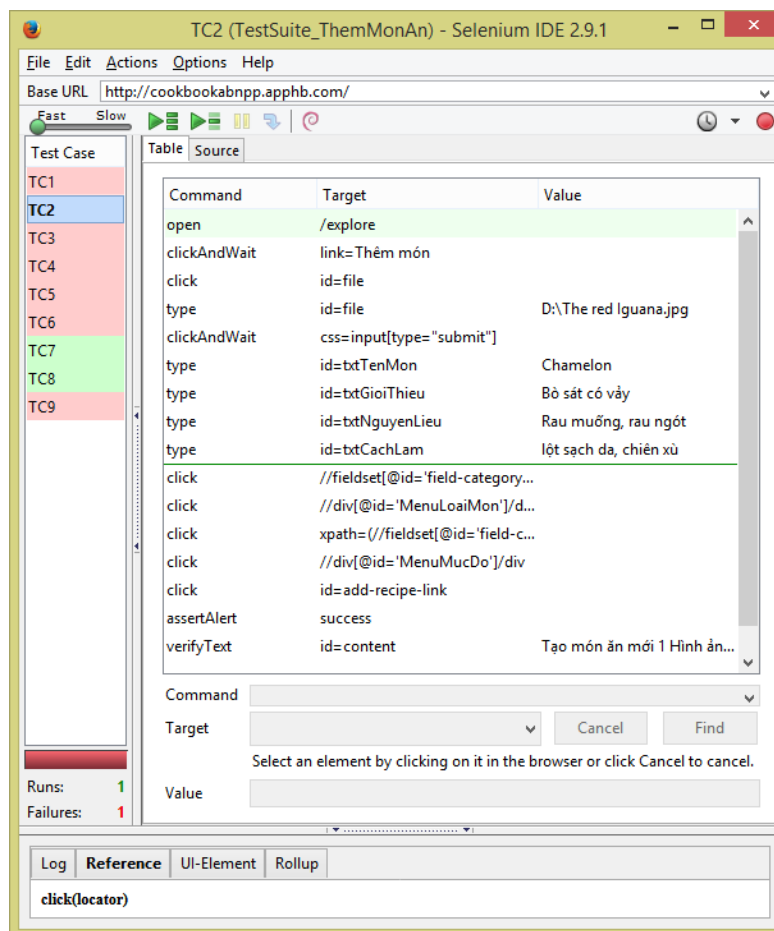
'Login' completed successfully

2.9.3 Demo 3

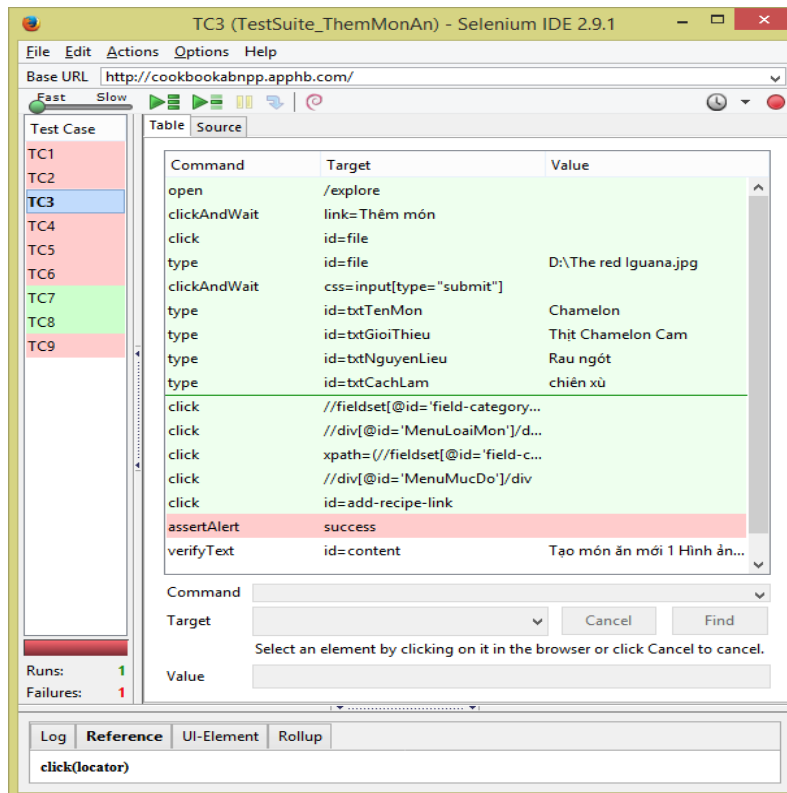
Dựa vào bộ TestCase đã đề ra, tạo bộ TestSuite để test tự động
TC1:



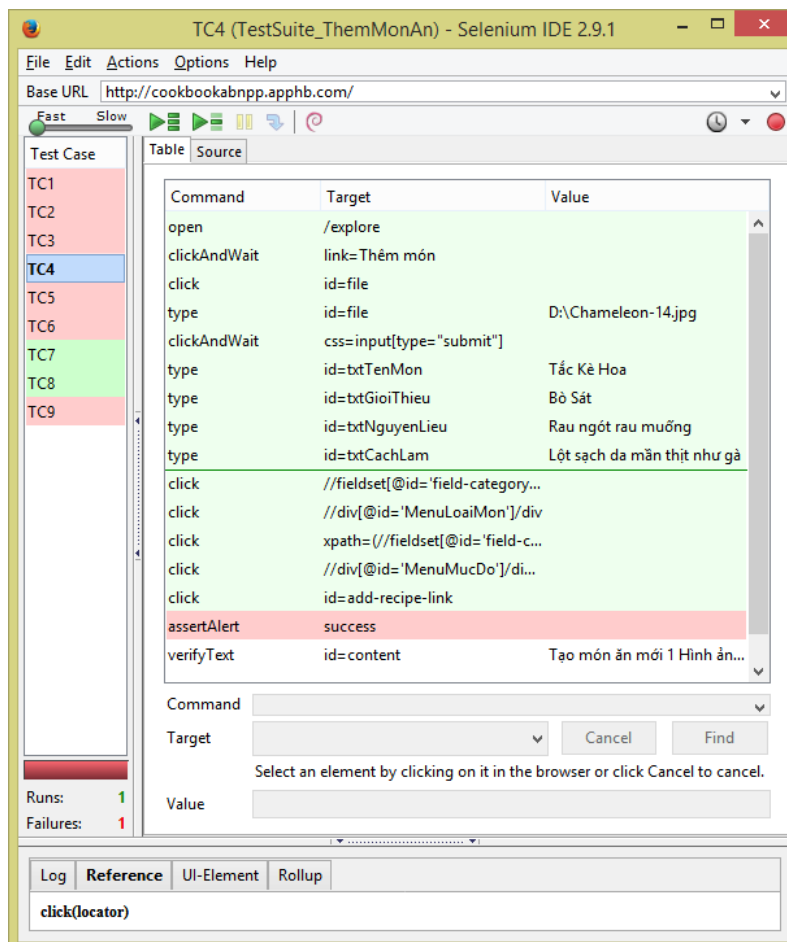
TC2:



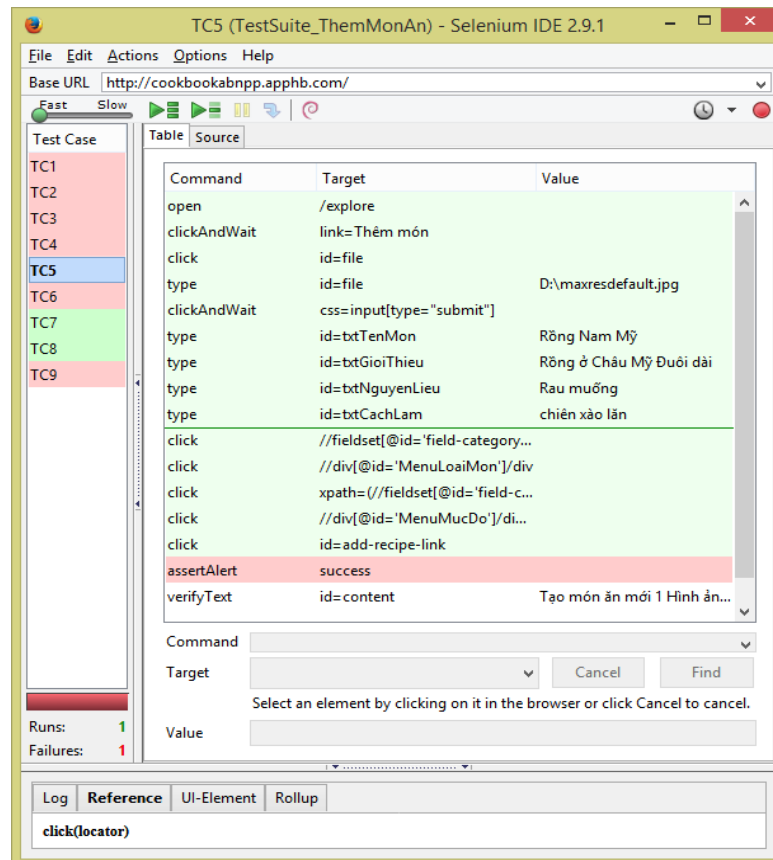
TC3:



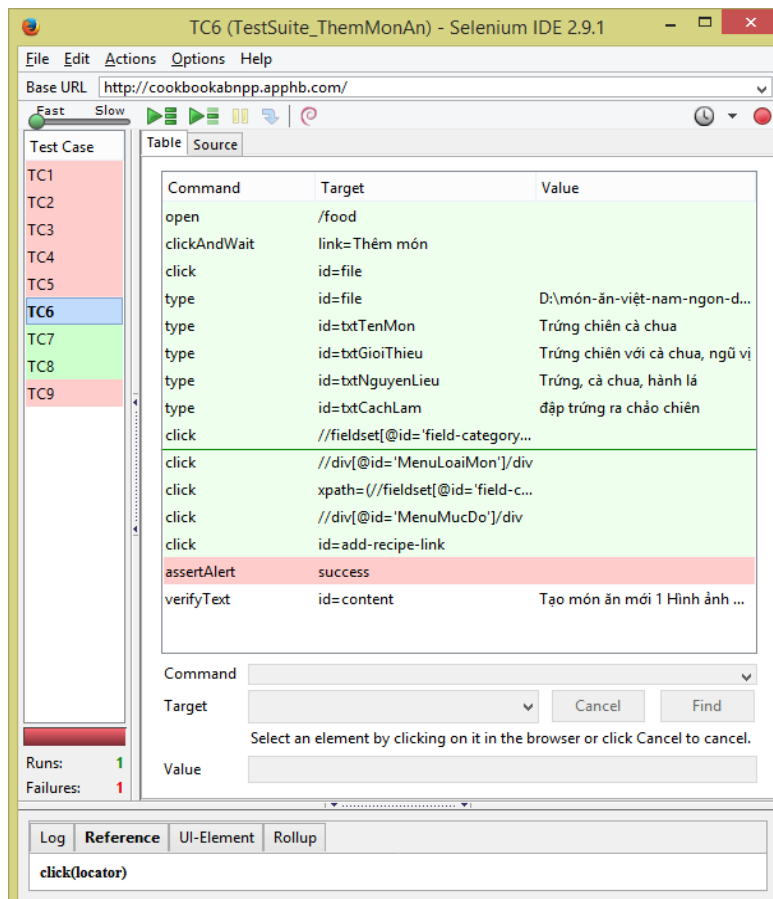
TC4:



TC5:



TC6:



TC7:

TC7 (TestSuite_ThemMonAn) - Selenium IDE 2.9.1

Base URL: <http://cookbookabnpp.apphb.com/>

Test Case: TC1, TC2, TC3, TC4, TC5, TC6, **TC7**, TC8, TC9

Runs: 1, Failures: 1

| Command | Target | Value |
|--------------|--------------------------|------------------------------|
| open | /explore | |
| clickAndWait | link=Thêm món | |
| click | id=file | |
| type | id=file | D:\1_205700.jpg |
| clickAndWait | css=input[type="submit"] | |
| type | id=txtTenMon | 2132321312312325435435523423 |
| verifyValue | id=txtTenMon | 2132321312312325435435523423 |

Command:
 Target:
 Select an element by clicking on it in the browser or click Cancel to cancel.
 Value:

Log Reference UI-Element Rollup

TC8:

TC8 (TestSuite_ThemMonAn) - Selenium IDE 2.9.1

Base URL: <http://cookbookabnpp.apphb.com/>

Test Case: TC1, TC2, TC3, TC4, TC5, TC6, TC7, **TC8**, TC9

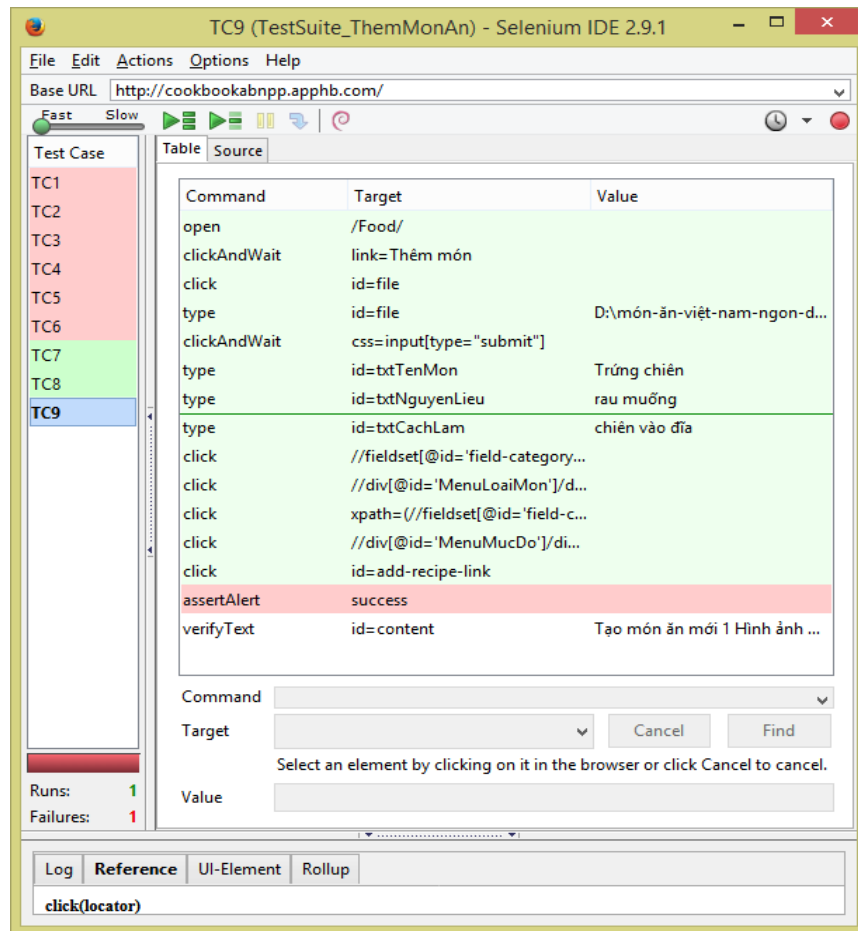
Runs: 1, Failures: 1

| Command | Target | Value |
|--------------|--|------------------------------|
| open | /food | |
| clickAndWait | link=Thêm món | |
| click | id=file | |
| type | id=file | D:\món-ăn-việt-nam-ngon-d... |
| clickAndWait | css=input[type="submit"] | |
| type | id=txtTenMon | 2222233333222223333322223... |
| verifyText | css=#field-title-container > di... Tên món ăn 60 | |

Command:
 Target:
 Select an element by clicking on it in the browser or click Cancel to cancel.
 Value:

Log Reference UI-Element Rollup

TC9:



⇒ Kết quả: 2 testcase Pass, 7 testcase Fail