### **Selenium là gì?**

### **Sẵn sàng để đáp ứng ngôi sao tự động hóa?**

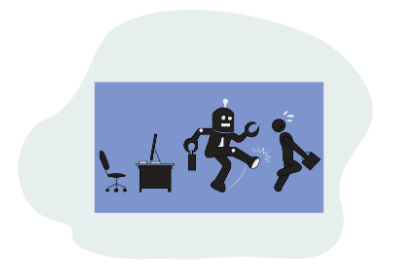
  
SELEN! Nhớ tên!

#### Nó loại bỏ kiểm tra thủ công lặp đi lặp lại.

#### Đó là lý do tại sao chúng tôi gọi nó là ngôi sao Tự động hóa!

#### Vì vậy, chúng ta hãy nhảy ngay vào nó.

### **Ông Selenium thực sự là ai?**



#### Selenium được sử dụng để tự động hóa các thử nghiệm được thực hiện trên các trình duyệt web.

#### Hoặc chúng ta có thể nói nó là một công cụ mã nguồn mở để xác thực các ứng dụng web trên các trình duyệt khác nhau.

### **Tự động hóa & xác nhận là gì?**

#### 

#### 

#### Làm cho một hệ thống hoặc quy trình tự động là tự động hóa.

#### Loại bỏ tất cả các kiểm tra và xác nhận thủ công, và làm cho chúng tự động là những gì ông Selenium làm.

### **Tại sao bạn cần tự động hóa & xác nhận?**

### 

Bạn không thể chỉ khởi chạy một ứng dụng web mà không kiểm tra nó và loại bỏ lỗi hoặc lỗi.

Tất cả các thử nghiệm nghiêm ngặt và lặp đi lặp lại phải được thực hiện cho một ứng dụng hoạt động đúng

Các thử nghiệm dài thường tránh trong quá trình kiểm tra thủ công có thể được chạy mà không cần giám sát.

### **Lịch sử của Selen**

### **Gặp gỡ ông Jason Huggins**

#### Selenium được tạo ra bởi Jason Huggins vào năm 2004.

#### Một kỹ sư tại ThoughtWorks

#### Nhưng tại sao anh ta lại tạo ra nó?

### **Lịch sử**

#### Một ngày nọ tại Thoughtworks, Jason Huggins đang làm việc trên một ứng dụng web đòi hỏi phải kiểm tra thường xuyên.

#### Nhận ra rằng việc kiểm tra thủ công lặp đi lặp lại của ứng dụng của họ ngày càng trở nên kém hiệu quả, ông đã tạo ra một chương trình JavaScript sẽ tự động kiểm soát các hành động của trình duyệt.

#### Ông đặt tên cho chương trình này là "JavaScript Test Runner".



#### Ông nhìn thấy tiềm năng trong ý tưởng này để giúp tự động hóa các ứng dụng web khác.

#### Ông đã tạo ra JavaScript Runner mã nguồn mở mà sau này được đổi tên thành Selenium Core.

#### Đây là cách ngôi sao Tự động hóa, ông Selenium được tạo ra.

### **Tóm tắt**

### 

### Selenium được sử dụng để tự động hóa các thử nghiệm được thực hiện trên các trình duyệt web.

### Loại bỏ tất cả các kiểm tra và xác nhận thủ công, và làm cho chúng tự động được thực hiện Selenium.

### **Các thành phần khác nhau**

#### Selenium Software không chỉ là một công cụ duy nhất mà là một bộ phần mềm.

#### Mỗi phần của nó phục vụ cho các nhu cầu thử nghiệm khác nhau của một tổ chức.

#### Có bốn thành phần của ông Selenium:

1. Selenium IDE (Môi trường phát triển tích hợp).
2. Selenium RC (Điều khiển từ xa)
3. Selenium WebDriver
4. Lưới Selen

### **Selenium IDE**

#### Selenium IDE (Môi trường phát triển tích hợp) là công cụ đơn giản nhất trong Selenium Suite.

#### Đây là một tiện ích bổ sung đơn giản của Firefox tạo ra các bài kiểm tra rất nhanh thông qua chức năng ghi và phát lại của nó.

#### Nó là dễ dàng để cài đặt và dễ học.

#### Nói tóm lại, các tiện ích bổ sung này giúp thử nghiệm thuận tiện và tiện dụng hơn.

### **Ưu điểm của IDE**

Selenium IDE rất dễ sử dụng.

Nó có khả năng chuyển đổi bài kiểm tra sang các ngôn ngữ lập trình khác nhau như html, java, v.v

Kinh nghiệm ngôn ngữ lập trình là không cần thiết để sử dụng Selenium IDE

Selenium IDE cung cấp khả năng Ghi nhật ký bằng cách sử dụng trình cắm đăng nhập tệp.

Trong Selenium IDE, người dùng có thể gỡ lỗi và thiết lập các điểm ngắt có nghĩa là bạn có thể kiểm tra cho đến một điểm và sau đó đặt điểm ngắt và bắt đầu sau từ điểm đó.

Selenium IDE rất linh hoạt cho người dùng.

### **Các tính năng của Selenium IDE**

#### Có một số tính năng được cung cấp trong IDE dưới thanh công cụ, sử dụng mà bạn có thể kiểm soát việc thực hiện các trường hợp kiểm thử trên đầu ngón tay của mình:

1. Kiểm soát tốc độ – Giúp kiểm soát tốc độ của các trường hợp kiểm thử
2. Run All – Cho phép thực thi toàn bộ Test Suite
3. Chạy - Chạy thử nghiệm hiện được chọn
4. Tạm dừng / Tiếp tục - cho phép người dùng tạm dừng và tiếp tục một trường hợp kiểm thử cụ thể
5. Bước - Giúp bước vào từng lệnh cụ thể trong kịch bản kiểm tra.

#### IDE thực sự khiến bạn trở thành Thanos của thế giới thử nghiệm này, mọi thứ trong tích tắc.

### **Selenium RC**

#### Selenium RC (Điều khiển từ xa) là một thành phần quan trọng trong bộ thử nghiệm Selenium.

#### Nó là một khung thử nghiệm.

#### Đây là công cụ kiểm tra web tự động đầu tiên cho phép người dùng sử dụng ngôn ngữ lập trình mà họ thích.

#### Nói tóm lại, Selenium RC cung cấp điều khiển từ xa cho tay bạn.

### **RC có một lịch sử thú vị.**

#### Thật không may, cùng một vấn đề chính sách ban đầu.

#### Trong những ngày đầu, một hạn chế đã được áp đặt bởi cùng một chính sách ban đầu.

#### Đó là những người thử nghiệm sử dụng Selenium Core phải cài đặt toàn bộ ứng dụng đang được thử nghiệm và máy chủ web trên máy tính cục bộ của riêng họ.

#### Đó chỉ là một sự lãng phí thời gian.

### **Một suy nghĩ khác Kỹ sư công trình đã đến giải cứu**

#### Đây là vị cứu tinh.

#### Vì vậy, một kỹ sư khác của ThoughtWorks, Paul Hammant, đã quyết định tạo ra một máy chủ sẽ hoạt động như một proxy để "lừa" trình duyệt tin rằng Selenium Core và ứng dụng web đang được thử nghiệm đến từ cùng một miền.

#### Thủ thuật hay! Và đây là cách Selenium RC được phát minh.

### **Tính năng**

Nó dễ dàng hỗ trợ các trình duyệt mới.

Hỗ trợ kiểm tra theo hướng dữ liệu, tức là tất cả dữ liệu được lưu trữ trong một bảng và có thể được kiểm tra bằng một tập lệnh kiểm tra duy nhất.

Thực thi nhanh hơn IDE

### **Selenium WebDriver**

#### Selenium WebDriver là một tập hợp các API mã nguồn mở được sử dụng để tự động hóa việc kiểm tra một ứng dụng web.

#### WebDriver chứng tỏ mình tốt hơn cả Selenium IDE và Selenium RC về nhiều mặt.

### 

#### Nó thực hiện một cách tiếp cận hiện đại và ổn định hơn trong việc tự động hóa các hành động của trình duyệt. WebDriver, không giống như Selenium RC, không dựa vào JavaScript để tự động hóa.

#### Nó kiểm soát trình duyệt bằng cách giao tiếp trực tiếp với nó.

### **Pros**

Cài đặt đơn giản hơn Selenium RC

Giao tiếp trực tiếp với trình duyệt

Không cần một thành phần riêng biệt như máy chủ RC

Thời gian thực thi nhanh hơn IDE và RC

### **Selenium Grid**

### **What is the grid?**

#### Selenium Grid là một công cụ được sử dụng cùng với Selenium RC để chạy các bài kiểm tra 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 thử nghiệm cùng một lúc.

#### Bộ phần mềm chuyên chạy song song nhiều thử nghiệm trên các trình duyệt, hệ điều hành và máy khác nhau.

#### Nó chạy thử nghiệm trên nhiều nền tảng cùng một lúc.

#### Nó rất giống với việc bạn thực hiện đa nhiệm trên điện thoại di động của mình.

### **Khái niệm Hub & node**

#### Selenium Grid sử dụng khái niệm nút trung tâm, nơi bạn chỉ chạy thử nghiệm trên một máy duy nhất được gọi là **hub**, nhưng việc thực thi sẽ được thực hiện bởi các máy khác nhau được gọi là **node**.

#### Điều này được thực hiện bằng cách định tuyến các lệnh đến một trình duyệt web từ xa, nơi một máy chủ hoạt động như trung tâm.

#### Hub định tuyến các lệnh kiểm tra đến nhiều nút Grid đã đăng ký.

#### Hub tiến hành thực hiện đồng thời các thử nghiệm trên nhiều máy, quản lý các trình duyệt khác nhau một cách tập trung, thay vì tiến hành các thử nghiệm khác nhau cho từng máy

### **Khi nào nên sử dụng Selenium Grid?**

#### Người kiểm tra nên sử dụng Selenium Grid trong các trường hợp sau:

Để chạy thử nghiệm trên nhiều trình duyệt và phiên bản của chúng, các thiết bị và hệ điều hành khác nhau.

Để giảm thời gian mà một bộ kiểm thử cần để hoàn thành một bài kiểm tra.

### **Tóm tắt**

Selenium IDE, một tiện ích bổ sung của Firefox mà bạn chỉ có thể sử dụng trong việc tạo các trường hợp kiểm thử và bộ thử nghiệm tương đối đơn giản.

Selenium Remote Control, còn được gọi là Selenium 1, là công cụ Selenium đầu tiên cho phép người dùng sử dụng ngôn ngữ lập trình trong việc tạo các bài kiểm tra phức tạp.

WebDriver, bước đột phá mới hơn cho phép các tập lệnh thử nghiệm của bạn giao tiếp trực tiếp với trình duyệt, do đó kiểm soát nó từ cấp hệ điều hành.

Selenium Grid cũng là một công cụ được sử dụng với Selenium RC để thực hiện các bài kiểm tra song song trên các trình duyệt và hệ điều hành khác nhau.

### **Selenium trên Python**

#### **Tìm hiểu nhanh về Python là gì**.

#### Python là một ngôn ngữ kịch bản hướng đối tượng cấp cao.

#### Nó được thiết kế theo cách thân thiện với người dùng.

#### Python sử dụng các từ khóa tiếng Anh đơn giản, dễ hiểu.

#### Nó có ít biến chứng cú pháp hơn bất kỳ ngôn ngữ lập trình nào khác.

### **Tại sao Mr Python tốt nhất cho Selenium?**

Mr Python rất dễ so với các ngôn ngữ lập trình khác, ít dài dòng hơn nhiều.

Các API Python cho phép bạn kết nối với trình duyệt thông qua Selenium.

Selenium gửi các lệnh Python tiêu chuẩn đến các trình duyệt khác nhau, bất chấp sự thay đổi trong thiết kế trình duyệt của chúng.

### **Python trên Java cho Selenium**

#### Một vài điểm ủng hộ Python hơn Java để sử dụng với Selenium là

Các chương trình Java có xu hướng chạy chậm hơn so với các chương trình Python.

Java sử dụng dấu ngoặc nhọn truyền thống để bắt đầu và kết thúc các khối, trong khi Python sử dụng thụt lề.

Java sử dụng gõ tĩnh, trong khi Mr Python được gõ động.

Ông Python đơn giản và nhỏ gọn hơn so với Java.

### **Cài đặt Python**

#### **Bước 1**: Tải xuống Trình cài đặt Python 3

Mở cửa sổ trình duyệt và điều hướng đến trang Tải xuống cho Windows tại python.org.

Nhấp vào liên kết cho Bản phát hành Python 3 mới nhất - Python 3.x.x.

Cuộn xuống dưới cùng và chọn trình cài đặt thực thi Windows x86-64 cho trình cài đặt thực thi 64 bit hoặc Windows x86 cho 32 bit.

  
**Bước 2**: Chạy trình cài đặt

#### Khi bạn đã chọn và tải xuống trình cài đặt, chỉ cần chạy nó bằng cách nhấp đúp vào tệp đã tải xuống.

#### Sau đó, chỉ cần nhấp vào Cài đặt ngay.

#### Đó nên là tất cả những gì cần có.

#### Một vài phút sau, bạn sẽ có một cài đặt Python 3 hoạt động trên hệ thống của bạn.

### **Thiết lập Selenium trên Python**

### **Sử dụng pip**

#### Một cách tiếp cận tốt hơn sẽ là sử dụng **pip** để cài đặt gói selenium.

#### Python 3.6 có pip có sẵn trong thư viện tiêu chuẩn.

#### Sử dụng pip, bạn có thể cài đặt selenium như thế này:

pip install selenium

### **Hướng dẫn chi tiết cho người dùng Windows**

#### 1. Cài đặt Python như đã nói trong phiên trước.

#### 2. Bắt đầu dấu nhắc lệnh bằng **cmd**. **exe** lập trình và chạy lệnh **pip** như được đưa ra dưới đây để cài đặt selenium:

C:\Python35\Scripts\pip.exe install selenium

#### Selen được cài đặt | Bây giờ bạn có thể chạy các kịch bản kiểm tra selenium

### **Các trình điều khiển**

#### Như bạn đã biết, Selenium yêu cầu trình điều khiển để giao tiếp với trình duyệt đã chọn.

#### Firefox, ví dụ, yêu cầu geckodriver.

#### Đảm bảo rằng nó nằm trong PATH của bạn, ví dụ: đặt nó vào / usr / bin hoặc / usr / local / bin.

#### Không tuân thủ bước này sẽ cung cấp cho bạn một lỗi selenium.common.exceptions.

WebDriverException: Message: ‘geckodriver’ executable needs to be in PATH.

### **Pip là gì?**

#### **pip** là trình quản lý gói tiêu chuẩn cho Mr Python.

#### Nó cho phép bạn cài đặt và quản lý các gói bổ sung không phải là một phần của thư viện chuẩn Python.

#### Điều đó có nghĩa là nó là một công cụ cho phép bạn cài đặt và quản lý các thư viện và phụ thuộc bổ sung không được phân phối như một phần của thư viện tiêu chuẩn.

### **Framework là gì?**

#### Bạn đã bao giờ nghĩ nếu bạn không có xương sườn hoặc hốc mắt thì sao?

#### Bộ xương người là khuôn khổ bên trong và bạn có thể tưởng tượng cơ thể của bạn mà không có khung.

#### Tương tự, ông Selenium cũng yêu cầu một khuôn khổ!

#### Nếu không có framework, chúng ta sẽ đặt "mã" cũng như "dữ liệu" ở cùng một nơi không thể tái sử dụng cũng như không thể đọc được.

#### Selenium Framework là một cấu trúc mã giúp bảo trì mã dễ dàng.

#### Sử dụng Framework, tạo ra các kết quả có lợi như tăng sử dụng lại mã, tính di động cao hơn, giảm chi phí bảo trì tập lệnh, khả năng đọc mã cao hơn, v.v

### 

### 

### 

### **Các loại framework**

#### Chủ yếu có ba loại framework được tạo bởi Selenium WebDriver để tự động hóa các trường hợp kiểm thử thủ công:

Khung kiểm tra theo hướng dữ liệu

Khung kiểm tra theo hướng từ khóa

Khung kiểm thử kết hợp

### 

### 

### 

### 

### 

### **Khung kiểm thử theo hướng dữ liệu là gì?**

#### Kiểm tra tự động theo hướng dữ liệu là một phương pháp trong đó tập dữ liệu thử nghiệm được tạo trong trang tính excel,

#### và sau đó được nhập vào các công cụ kiểm tra tự động hóa để cung cấp cho phần mềm đang thử nghiệm.

#### Trang tính excel lưu trữ tất cả dữ liệu

#### Data Driven Framework là một trong những Automation Testing Framework phổ biến trên thị trường hiện nay.

### 

### 

### **Khung kiểm tra theo hướng từ khóa**

### **Thử nghiệm theo từ khóa là gì?**

#### KEYWORD-DRIVEN TESTING là một kỹ thuật kịch bản sử dụng các tệp dữ liệu để chứa các từ khóa liên quan đến ứng dụng đang được kiểm tra.

#### Những từ khóa này mô tả tập hợp các hành động cần thiết để thực hiện một bước cụ thể.

#### Trong Thử nghiệm theo hướng từ khóa, trước tiên bạn xác định một tập hợp các từ khóa và sau đó liên kết một hành động liên quan đến các từ khóa này.

#### Ở đây, mọi hành động thử nghiệm như mở hoặc đóng trình duyệt, nhấp chuột, v.v. được mô tả bằng một từ khóa như openbrowser, click, Type Text, v.v.

#### Đừng lo lắng, tôi có một ví dụ để giải thích cho bạn.

### **Ví dụ**

#### 1. **đăng nhập** vào trang web Netflix:

#### Từ khóa 'đăng nhập' sẽ được sử dụng trong khung tự động hóa của chúng tôi, để kiểm tra chức năng đăng nhập hoặc hành động được liên kết với nó.

#### 2. **đăng xuất** khỏi trang web Netflix:

#### Từ khóa 'logout' sẽ được sử dụng trong khung tự động hóa của chúng tôi, để kiểm tra chức năng đăng xuất hoặc hành động được liên kết với nó.

### **Lợi thế**

Nó cho phép người kiểm tra chức năng lập kế hoạch tự động hóa kiểm thử trước khi ứng dụng sẵn sàng

Các bài kiểm tra có thể được phát triển mà không cần kiến thức lập trình

Nó không phụ thuộc vào một ngôn ngữ lập trình hoặc công cụ cụ thể

Tương thích với bất kỳ công cụ tự động hóa nào có sẵn trên thị trường

### **Mục tiêu đằng sau việc sử dụng thử nghiệm theo từ khóa**

Nó giúp giảm chi phí bảo trì

Tránh các thông số kỹ thuật trùng lặp

Tái sử dụng nhiều hơn các tập lệnh chức năng

Hỗ trợ kiểm tra tốt hơn và tính di động

Đạt được nhiều thử nghiệm hơn với ít nỗ lực hơn hoặc tương tự

### **Khung kiểm thử kết hợp**

#### Như tên cho thấy,

#### Hybrid Driven Framework là sự kết hợp của cả Data-Driven framework và Keyword-Driven.

#### Kỹ thuật trong đó chúng ta có thể sử dụng tốt nhất cả khung Selenium theo hướng dữ liệu &; từ khóa

#### Ở đây, các từ khóa, cũng như dữ liệu thử nghiệm, được đưa ra bên ngoài.

#### Hybrid Driven framework chủ yếu được sử dụng bởi những người kiểm thử thủ công, những người không có nhiều kiến thức về ngôn ngữ lập trình.

#### Những người như vậy chỉ cần xem Keywords, Test data, Object repository và bắt đầu tạo test case ngay lập tức mà không cần phải code bất cứ thứ gì trong framework.

### **Tóm tắt**

* Kiểm thử theo hướng từ khóa là một kỹ thuật kịch bản sử dụng các tệp dữ liệu để chứa các từ khóa liên quan đến ứng dụng đang được kiểm tra.
* Một thử nghiệm theo hướng từ khóa thường được thực hiện bằng thử nghiệm tự động.
* Các bài kiểm tra có thể được phát triển mà không cần kiến thức lập trình.
* Các thử nghiệm tương thích với bất kỳ công cụ tự động hóa nào có sẵn trên thị trường.

### **Cơ sở dữ liệu và thử nghiệm**

### **Cơ sở dữ liệu là gì?**

#### Bạn phải nhận thức được cơ sở dữ liệu là gì

#### Nhưng chỉ để nhắc nhở bạn, **cơ sở dữ liệu** là một tập hợp thông tin được tổ chức để nó có thể dễ dàng truy cập, quản lý và cập nhật.

#### Cơ sở dữ liệu máy tính thường chứa tập hợp các bản ghi hoặc tệp dữ liệu, chứa thông tin về các giao dịch bán hàng hoặc tương tác với các khách hàng cụ thể.

### **Kiểm thử cơ sở dữ liệu là gì?**

#### Kiểm thử cơ sở dữ liệu là một loại kiểm thử phần mềm kiểm tra lược đồ, bảng, trình kích hoạt, v.v. của **cơ sở dữ liệu** đang được kiểm thử.

#### Nó cũng kiểm tra tính toàn vẹn và nhất quán của dữ liệu.

#### Nó có thể liên quan đến việc tạo các truy vấn phức tạp để tải / kiểm tra căng thẳng cơ sở dữ liệu và kiểm tra khả năng đáp ứng của nó.

#### Vì vậy, bây giờ hãy cho biết cách kiểm tra cơ sở dữ liệu được thực hiện bằng Selenium.

### **Kiểm tra cơ sở dữ liệu bằng Selenium**

#### Selenium Webdriver bị giới hạn trong việc kiểm tra các ứng dụng của bạn bằng Trình duyệt.

#### Để sử dụng Selenium Webdriver để xác minh cơ sở dữ liệu, bạn cần sử dụng JDBC.

#### Bây giờ chúng ta sẽ xem JDBC là gì.

### **Cơ bản của JDBC**

#### JDBC (Java Database Connectivity) là một API cấp SQL cho phép bạn thực thi các câu lệnh SQL.

#### Nó chịu trách nhiệm kết nối giữa ngôn ngữ lập trình Java và một loạt các cơ sở dữ liệu.



#### Về cơ bản những gì JDBC làm là Với sự trợ giúp của JDBC, Selenium có thể truy cập và kiểm tra Cơ sở dữ liệu.

#### Kết nối cơ sở dữ liệu Java (JDBC) là một API Java được sử dụng để kết nối và tương tác với Cơ **sở dữ liệu**.

### **JDBC và selen**

### **JDBC hoạt động như thế nào**

#### **Bước đầu tiên**

#### JDBC là một trong những API Java tiêu chuẩn để kết nối độc lập với cơ sở dữ liệu giữa ngôn ngữ lập trình Java và một loạt các cơ sở dữ liệu.

#### Giao diện chương trình ứng dụng (API) này, cho phép bạn chuyển đổi các câu lệnh yêu cầu truy cập, bằng Ngôn ngữ truy vấn có cấu trúc (SQL).

#### **Bước 2**

#### Sau đó, chúng được chuyển đến chương trình quản lý cơ sở dữ liệu.

#### Nó chủ yếu liên quan đến việc mở kết nối, tạo Cơ sở dữ liệu SQL, thực hiện các truy vấn SQL và sau đó đến đầu ra.

### **Các thành phần JDBC phổ biến**

#### JDBC API cung cấp các giao diện và lớp sau:



Trình điều khiển: Nó là một giao diện xử lý các thông tin liên lạc với máy chủ cơ sở dữ liệu.

Kết nối: Đây là một giao diện bao gồm tất cả các phương thức cần thiết để kết nối với cơ sở dữ liệu.

Đối tượng kết nối đại diện cho bối cảnh giao tiếp trong đó toàn bộ giao tiếp với cơ sở dữ liệu chỉ thông qua đối tượng kết nối.

### **Apache POI**

#### Selenium Webdriver là một công cụ tuyệt vời để tự động hóa các ứng dụng dựa trên web.

#### Nhưng nó không hỗ trợ các thao tác đọc và ghi trên các tệp excel.

#### Do đó, chúng tôi sử dụng Apache POI

### **Apache POI là gì**

#### POI là viết tắt của - Thực hiện xáo trộn kém

#### Apache POI là một API được viết bằng Java để hỗ trợ các hoạt động đọc và ghi – sửa đổi các tệp văn phòng.

#### Đây là API phổ biến nhất được sử dụng cho các bài kiểm tra theo hướng dữ liệu Selenium.

#### Phát triển khung Data Driven trong Selenium bằng POI giúp giảm bảo trì, cải thiện phạm vi thử nghiệm do đó mang lại lợi tức đầu tư tốt.

### **Tại sao kiểm tra theo hướng dữ liệu**

#### Chúng ta sẽ hiểu điều này bằng một ví dụ đơn giản.

#### Giả sử chúng ta cần kiểm tra đăng nhập / Đăng ký với nhiều trường nhập liệu với 100 tập dữ liệu khác nhau.

#### Cũng giống như một biểu mẫu trực tuyến với nhiều trường nhập liệu



#### Để kiểm tra điều này, bạn có ba cách tiếp cận khác nhau:

1. Tạo 100 tập lệnh cho mỗi tập dữ liệu và thực hiện từng thử nghiệm một.
2. Thay đổi dữ liệu trong tập lệnh và thực thi nó nhiều lần.
3. Nhập dữ liệu từ trang tính excel và thực thi tập lệnh nhiều lần với dữ liệu khác nhau.

#### Hai kịch bản đầu tiên tốn nhiều công sức, tốn thời gian - ngụ ý ROI thấp. Do đó, chúng ta phải tuân theo cách tiếp cận thứ ba.

#### Trong cách tiếp cận thứ ba, chúng tôi đang triển khai khung Data Driven, nơi tất cả dữ liệu của chúng tôi nằm trong một bảng excel, nơi nó được nhập và sử dụng để kiểm tra các tính năng của ứng dụng.

### **Tóm tắt**

Chúng ta có thể tạo ba loại khung kiểm thử bằng Selenium WebDriver.

Đây là khung thử nghiệm Data Driven, Keyword Driven và Hybrid.

Chúng tôi có thể đạt được khung theo hướng dữ liệu bằng cách sử dụng nhà cung cấp dữ liệu của TestNG.

Trong framework hướng từ khóa, từ khóa được viết bằng một số tệp bên ngoài như tệp excel và mã java sẽ gọi tệp này và thực thi các trường hợp kiểm thử.

### **Tự động hóa quy trình đăng nhập**

#### Selenium được sử dụng để tự động hóa các ứng dụng web cho mục đích thử nghiệm, nhưng chắc chắn không chỉ giới hạn ở đó.

#### Các tác vụ quản trị dựa trên web nhàm chán cũng có thể được tự động hóa.

#### Khi bạn tìm hiểu thêm, thật thú vị khi thấy mọi thứ xảy ra tự động và tiết kiệm thời gian trong việc thực hiện các nhiệm vụ vô ích hết lần này đến lần khác.

### **Tự động hóa đăng nhập là gì?**

#### Khi chúng tôi học cách tự động hóa quá trình đăng nhập

#### Chúng tôi không phải lãng phí thời gian nhập thông tin đăng nhập nhiều lần.

#### Bây giờ chúng ta sẽ tìm hiểu làm thế nào để tự động hóa quá trình đăng nhập facebook.

#### Selenium có thể tự động hóa và kiểm soát trình duyệt và nhấp, điền văn bản, gửi các nút xuất hiện trên các trang web khác nhau.

### **Thế nào**

#### Để đăng nhập vào Facebook, chúng tôi sẽ sử dụng Python Script điều khiển Selenium.

#### Selenium Python Script sẽ:

* Bước 1) Mở Firefox
* Bước 2) Điều hướng đến Facebook
* Bước 3) Tìm kiếm &; Nhập trường Email hoặc Điện thoại &; Nhập mật khẩu
* Bước 4) Nhấp vào Đăng nhập

#### Đây là những gì ông Selenium sẽ làm **AUTOMATION!**

#### Bạn có thể định cấu hình Selenium để sử dụng bất kỳ trình duyệt nào như Chrome, Safari, IE, v.v. Trong hướng dẫn này, chúng ta sẽ sử dụng FireFox

### **Mã đơn giản**

### **Điều đầu tiên Python**

#### Bây giờ chúng ta đã biết tại sao Python được chọn cho selenium, vì vậy câu hỏi lớn là:

#### Bạn có biết về mã hóa Python không?

#### Nếu bạn biết thì hãy bắt đầu tự động hóa Facebook.

#### **Nếu không thì chúng tôi có Khóa học Python tương tác và tốt nhất trong menu chính của mình**.

#### Bạn có thể kiểm tra điều đó.

### 

### 

### 

### 

### **Mã hóa**

#### Mã để đăng nhập vào Facebook bằng Python

#### Hai bước đầu tiên:

from selenium import webdriver

from selenium.webdriver.support.ui import WebDriverWait

#### **Bước 1**. **Mở Firefox**

browser = webdriver.Firefox()

#### **Bước 2**. **Điều hướng đến Facebook**

browser.get("http://www.facebook.com")

### **Hai bước cuối cùng**

#### **Bước 3**. Tìm kiếm &; Nhập **trường Email hoặc Điện thoại &; Nhập mật khẩu**

username = browser.find\_element\_by\_id("email")

password = browser.find\_element\_by\_id("pass")

submit = browser.find\_element\_by\_id("loginbutton")

username.send\_keys("you@email.com")

password.send\_keys("yourpassword")

#### **Bước 4**. **Nhấp vào Đăng nhập**

submit.click()

#### Đừng lo lắng, chúng tôi có lời giải thích cho từng dòng trong phiên tiếp theo.

### **Lời giải thích**

from selenium import webdriver

#### Dòng mã 1: Từ mô-đun selenium nhập webdriver

from selenium.webdriver.support.ui import WebDriverWait

#### Từ khóa nhập mô-đun selen

browser = webdriver.Firefox()

#### Dòng mã 4: Trong dòng này, chúng ta đang khởi tạo "FireFox" bằng cách tạo một đối tượng của nó.

browser.get("http://www.facebook.com")

#### Dòng mã 6: "Phương thức browser.get" sẽ điều hướng đến một trang được cung cấp bởi URL.

#### WebDriver đợi cho đến khi trang đã được tải hoàn toàn (nghĩa là dịp "onload" đã buông bỏ), trước khi trả lại quyền kiểm soát cho thử nghiệm hoặc tập lệnh của bạn.



username = browser.find\_element\_by\_id("email")

#### Dòng mã 8: Trong dòng này, chúng tôi đang tìm phần tử của hộp văn bản nơi "email" phải được viết.

password = browser.find\_element\_by\_id("pass")

#### Dòng mã 9: Trong dòng này, chúng tôi đang tìm phần tử của hộp văn bản nơi "mật khẩu" phải được viết.

submit = browser.find\_element\_by\_id("loginbutton")

#### Dòng mã 10: Trong dòng này, chúng tôi đang tìm phần tử nút gửi mà chúng tôi cần nhấp vào

username.send\_keys("you@email.com")

#### Dòng mã 11: Bây giờ chúng ta đang gửi các giá trị đến phần email

password.send\_keys("yourpassword")

#### Dòng mã 12: Gửi giá trị đến phần mật khẩu

submit.click()

#### Dòng mã 14: Nhấp vào nút "Gửi"

### **Giới thiệu về TestNG**

### **Thế hệ tiếp theo**

#### Thuật ngữ thế hệ tiếp theo nghe có vẻ tiên tiến và tương lai phải không?

#### TestNG là một khung kiểm thử tự động hóa, trong đó NG là viết tắt của "Thế hệ tiếp theo", được thiết kế để giúp kiểm tra end-to-end dễ dàng.

#### Nó được thiết kế để đơn giản hóa tất cả các yêu cầu kiểm thử của bạn như kiểm tra chức năng, hồi quy, kiểm tra end-to-end, v.v.



#### Sử dụng TestNG, bạn có thể tạo một báo cáo thích hợp và bạn có thể dễ dàng biết có bao nhiêu trường hợp kiểm thử được thông qua, thất bại và bỏ qua.

#### Bạn có thể thực hiện các trường hợp kiểm thử thất bại một cách riêng biệt.

#### Tôi sẽ giải thích nó bằng một ví dụ

### **Ví dụ**

#### Chỉ kiểm thử trường hợp kiểm thử thất bại một cách riêng biệt.

#### Giả sử, bạn có năm trường hợp kiểm thử, một phương thức được viết cho mỗi trường hợp kiểm thử.

#### Giả sử rằng chương trình được viết bằng phương pháp chính mà không sử dụng testNG.

#### Khi bạn chạy chương trình này đầu tiên, ba phương pháp được thực hiện thành công và phương pháp thứ tư không thành công.

#### Sau đó sửa các lỗi có trong phương thức thứ tư, bây giờ bạn chỉ muốn chạy phương thức thứ tư vì ba phương thức đầu tiên dù sao cũng được thực thi thành công.

#### Điều này là không thể nếu không sử dụng TestNG.

### **TestNG với Selen**

### **Tại sao nên sử dụng TestNG với Selenium?**

#### Bạn biết về ông Selenium bây giờ và tôi đoán ông Selenium là một chuyên gia về kiểm thử tự động hóa.

#### Vậy thì tại sao phải sử dụng TestNG với nó.

#### Các bài kiểm tra Selen mặc định không tạo ra một định dạng thích hợp cho kết quả kiểm tra. Sử dụng TestNG, chúng tôi có thể tạo ra kết quả kiểm tra.

#### TestNG có rất nhiều tính năng sẽ giúp việc kiểm tra dễ dàng và nhanh chóng hơn.

#### Có rất nhiều tính năng của TestNG, nhưng chúng tôi sẽ chỉ tập trung vào những tính năng quan trọng nhất mà chúng tôi có thể sử dụng trong Selenium:

Tạo báo cáo ở định dạng thích hợp bao gồm một số trường hợp kiểm thử chạy, số trường hợp kiểm thử được thông qua, số trường hợp kiểm thử không thành công và số trường hợp kiểm thử bị bỏ qua.

Nhiều trường hợp kiểm thử có thể được nhóm lại dễ dàng hơn và bạn có thể ưu tiên trường hợp kiểm thử nào nên được thực hiện trước.

Cùng một trường hợp kiểm thử có thể được thực hiện nhiều lần mà không có vòng lặp chỉ bằng cách sử dụng một từ khóa được gọi là 'số lượng gọi'.

Sử dụng testng, bạn có thể thực hiện nhiều trường hợp kiểm thử trên nhiều trình duyệt, tức là kiểm tra trình duyệt chéo.

### **Tóm tắt**

1. TestNG là một khung kiểm thử có khả năng làm cho các bài kiểm tra Selenium dễ hiểu hơn và tạo ra các báo cáo dễ hiểu.
2. Những ưu điểm chính của TestNG là như sau.

* Chú thích dễ sử dụng và dễ hiểu hơn.
* Các trường hợp kiểm thử có thể được nhóm lại dễ dàng hơn.
* TestNG cho phép chúng tôi tạo các bài kiểm tra song song.

### **Tìm liên kết bị hỏng bằng Selenium Liên kết bị hỏng là gì?**

#### Liên kết bị hỏng là một trang web mà người dùng không thể tìm thấy hoặc truy cập được, vì nhiều lý do.

#### Máy chủ web thường sẽ trả về thông báo lỗi khi người dùng cố gắng truy cập vào một liên kết bị hỏng.

#### Liên kết bị hỏng còn được gọi là 'liên kết chết' hoặc 'thối liên kết'.

#### Tất cả các bạn sẽ thấy tình huống đó.

### **Tại sao bạn nên kiểm tra các liên kết bị hỏng**

#### Bạn nên luôn đảm bảo rằng không có liên kết bị hỏng trên trang web vì người dùng không nên truy cập vào trang lỗi.

#### Lỗi xảy ra nếu các quy tắc không được cập nhật chính xác hoặc các tài nguyên được yêu cầu không tồn tại tại máy chủ.

### **Tại sao sử dụng trình điều khiển web selenium cho việc này**

#### Kiểm tra thủ công các liên kết là một công việc tẻ nhạt, bởi vì mỗi trang web có thể có một số lượng lớn các liên kết và quy trình thủ công phải được lặp lại cho tất cả các trang.

#### Bây giờ trình điều khiển web selen đi vào hình ảnh.

#### Một tập lệnh Tự động hóa sử dụng Selenium sẽ tự động hóa quy trình là một giải pháp phù hợp hơn.

### **Hãy lấy một ví dụ**

#### Ví dụ 1- Giả sử chúng ta có một ứng dụng chứa 400 liên kết và chúng ta cần xác minh liên kết có bị hỏng hay không.

#### **Cách tiếp cận 1**.

#### Quy trình thủ công: Truy cập từng liên kết và xác minh liên kết có hoạt động hay không.

#### Bạn có nghĩ rằng đó là cách tiếp cận hợp lệ?

#### Không, sẽ mất cả ngày để xác minh và bạn cũng sẽ không nhận được hiệu quả và lãi suất tương tự.

### **Sử dụng lưới selen**

#### **Cách tiếp cận 2**.

#### Công việc thông minh: Viết mã sẽ kiểm tra tất cả các liên kết và cũng sẽ xác minh trạng thái.

#### Vì tất cả chúng ta đều thông minh nên chúng ta sẽ thực hiện công việc Thông minh và sẽ xem cách tìm các liên kết bị hỏng bằng cách sử dụng selenium.

#### Vì vậy, chúng tôi sử dụng một mã trình điều khiển web đơn giản có sẵn trên internet sẽ cho biết liên kết nào bị hỏng và liên kết nào không.

#### Điều này làm cho toàn bộ điều rất dễ dàng.

#### Do đó, chúng tôi có thể lấy tất cả các liên kết từ một trang web và in xem các liên kết hợp lệ hay bị hỏng.

### **Tại sao nó là công cụ tốt nhất? Công cụ mã nguồn mở**

#### Công cụ mã nguồn mở là một cụm từ được sử dụng để chỉ một chương trình hoặc công cụ thực hiện một nhiệm vụ rất cụ thể,

#### Trong đó mã nguồn được công bố công khai để sử dụng và / hoặc sửa đổi từ thiết kế ban đầu của nó, miễn phí.

#### Vì nó là một công cụ mã nguồn mở, nó không yêu cầu bất kỳ chi phí cấp phép nào giúp nó chiếm ưu thế hơn các công cụ kiểm tra tự động hóa khác.

### **Công cụ cho mọi nhu cầu**

#### Như đã đề cập trước đó, Selenium có một bộ công cụ, vì vậy nó phù hợp với mọi nhu cầu của người dùng.

#### Bạn có thể sử dụng các công cụ khác nhau như WebDriver, Grid, IDE để đáp ứng các nhu cầu khác nhau của bạn.

### **Hỗ trợ tất cả các ngôn ngữ chính**

#### Bạn có biết thách thức lớn mà tester hoặc developer phải đối mặt với một công cụ kiểm thử tự động hóa là sự hỗ trợ cho các ngôn ngữ.

#### Kể từ khi chúng tôi có selenium Nó hỗ trợ tất cả các ngôn ngữ chính như Java, JavaScript, Python, Ruby, C sharp, Perl, .Net và PHP.

#### Nó dễ dàng hơn cho người thử nghiệm sử dụng.

### **Hỗ trợ trình duyệt và Hệ điều hành**

#### Selenium hỗ trợ các trình duyệt khác nhau như Chrome, Firefox, Opera, Internet Explorer, Edge và Safari.

#### Nó hỗ trợ các hệ điều hành khác nhau như Windows, Linux, Mac.

#### Điều này làm cho nó linh hoạt để sử dụng.

### **Các loại kiểm thử tự động hóa với Selenium**

### **Kiểm tra khả năng tương thích**

#### Được thực hiện bởi Người thử nghiệm để đảm bảo rằng ứng dụng web đáp ứng các tiêu chuẩn hiệu suất trên các kết hợp trình duyệt-hệ điều hành khác nhau.

#### Chẳng hạn:

* Thử nghiệm trên các thiết bị khác nhau (thiết bị di động và máy tính để bàn) để đảm bảo rằng giao diện người dùng phù hợp với quy mô (đáp ứng);
* Thử nghiệm trên các trình duyệt khác nhau để xem liệu quảng cáo video có hiển thị trên các trang như bình thường hay không.

### **Kiểm tra hiệu suất**

#### Một loạt các thử nghiệm được thực hiện bởi Testers để đảm bảo rằng dự án đáp ứng các tiêu chuẩn hiệu suất do các bên liên quan đặt ra.

#### Tester viết một tập lệnh kiểm tra xem tất cả các yếu tố trên trang chủ có tải trong vòng 2 giây trên các phiên bản trình duyệt khác nhau hay không.

### **Kiểm thử hồi quy**

#### Một loạt các thử nghiệm được thực hiện để đảm bảo rằng các tính năng mới được xây dựng hoạt động với hệ thống hiện có.

#### Trong ví dụ tương tự, giả sử trang web sản phẩm ra mắt một tính năng mới (mã khuyến mại) tự động áp dụng cho các mặt hàng đủ điều kiện trước khi thanh toán.

#### Người thử nghiệm có thể viết các trường hợp để xác minh rằng nó không phá vỡ phần còn lại của tính năng thanh toán.

### **Kiểm tra end-to-end**

#### Người kiểm tra / chuyên gia QA, thường là từ quan điểm của người dùng.

#### Mục đích là để xác minh rằng tất cả các điểm tiếp xúc trên ứng dụng web đều hoạt động.

#### Từ ví dụ trước,

#### Người thử nghiệm có thể viết một loạt các trường hợp kiểm thử để kiểm tra xem đăng ký, tìm kiếm sản phẩm, thanh toán, đánh giá, đánh dấu trang và tất cả các tính năng khác có hoạt động hay không.

### **Chúc mừng!**

#### Với điều này, chúng tôi đã hoàn thành khóa học của chúng tôi về Selenium.

#### Đó là một hành trình tuyệt vời với bạn!

#### Chào tạm biệt bạn! Tada!!