

# Bài: Thao tác cơ bản với Selenium trong WPF

Xem bài học trên website để ủng hộ Kteam: [Thao tác cơ bản với Selenium trong WPF](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Trong bài trước, Kteam đã giới thiệu đến bạn [TỔNG QUAN VỀ SELENIUM](#) một thư viện tuyệt vời cho việc Auto Web Browser.

Ở bài này, Kteam sẽ giới thiệu tới bạn các **Thao tác cơ bản với Selenium trong WPF**.

## Nội dung

Để đọc hiểu bài này tốt nhất bạn cần:

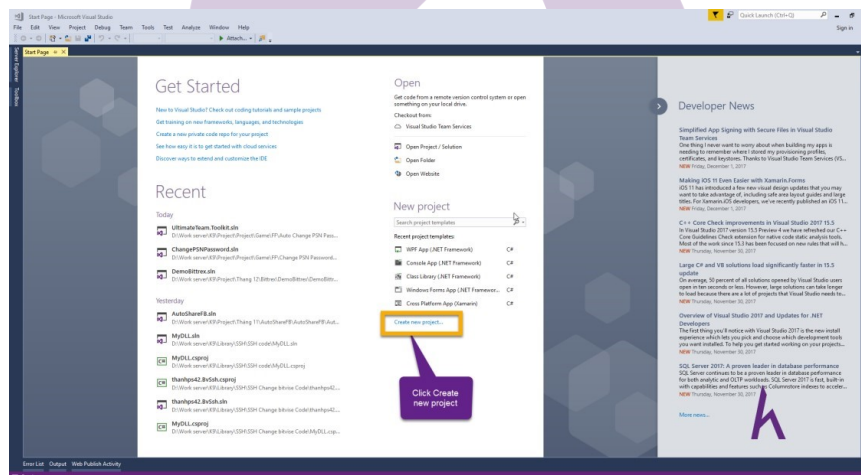
- Cài đặt sẵn [VISUAL STUDIO 2015 hoặc 2017 community](#).
- Xem qua bài [TỔNG QUAN VỀ SELENIUM](#).
- Có kiến thức về [C# CƠ BẢN](#), [LẬP TRÌNH OOP trong C#](#), và [LẬP TRÌNH WPF](#).
- Có kiến thức HTML cơ bản.

Trong bài này, chúng ta sẽ cùng tìm hiểu những nội dung sau:

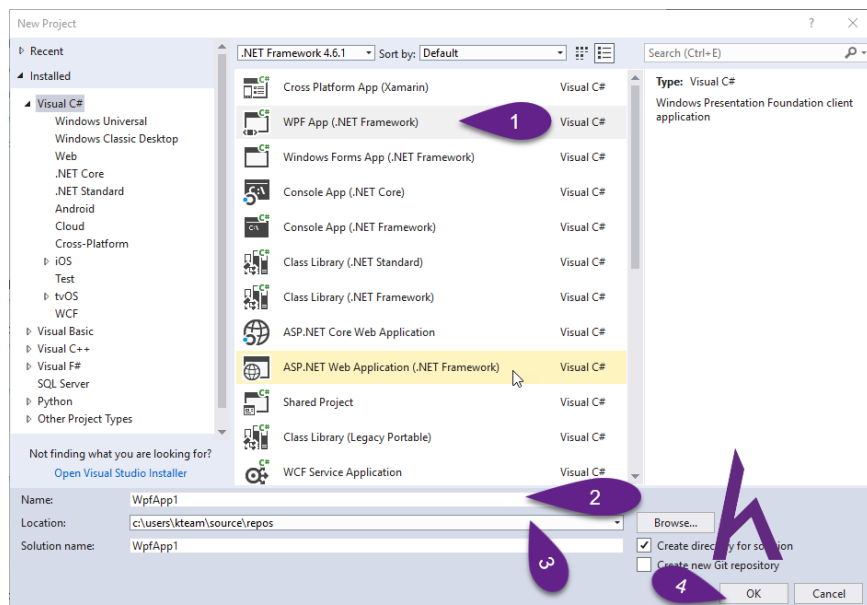
- Tạo một Project import thư viện Selenium Firefox và Chrome Driver.
- Navigate URL với Selenium.
- Một số thuộc tính và phương thức cơ bản.
- Get Element với Selenium.

## Tạo một project import thư viện Selenium Firefox và Chrome Driver

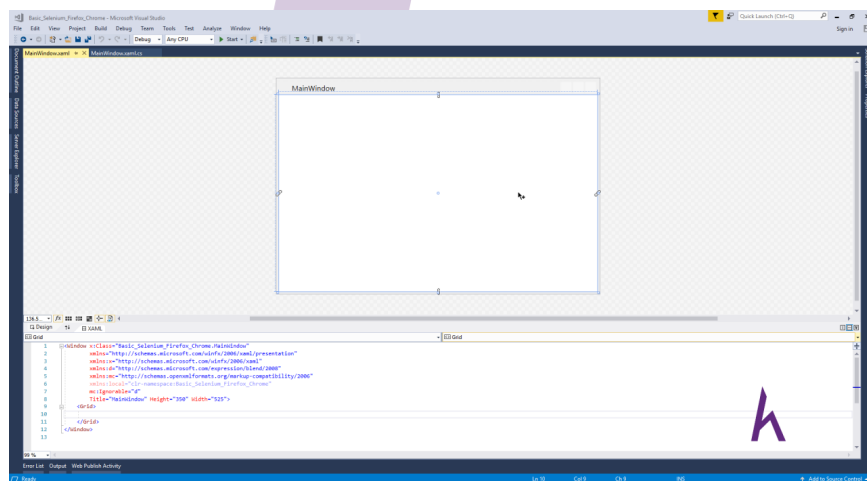
Đầu tiên chúng ta mở **Visual Studio** và tạo một project mới



Chọn **WPF App(.NET Framework)** hoặc **Windows Forms App** nếu bạn không rành WPF. Đặt tên project và chọn đường dẫn lưu solution của bạn.



Tại màn hình ban đầu của project.



Bạn thêm đoạn code:

```
<Button Content="Start" Click="Button_Click"></Button>
```

Tạo Button ngay bên trong cặp thẻ bên dưới, để khi chúng ta nhấn vào Button đó sẽ cho chạy Selenium.

```
<Grid>

</Grid>
```

Code hoàn chỉnh của **MainWindow.xaml**

```
<?xml version="1.0" encoding="utf-8" ?>
<Window x:Class="SeleniumFramework.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Button Content="Start" Click="Button_Click"></Button>
    </Grid>
</Window>
```

```
<Window x:Class="Basic_Selenium_Firefox_Chrome.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Basic_Selenium_Firefox_Chrome"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Button Content="Start" Click="Button_Click"></Button>
    </Grid>
</Window>
```

Một lưu ý nhỏ, trong sự kiện Click mình đã tạo sẵn một **Event Click** theo cách nhắc lệnh mà Visual Studio hỗ trợ, đó là:

**Gõ Click=**

**Một New Event Handler sẽ hiện lên bạn chỉ cần nhấn Enter**

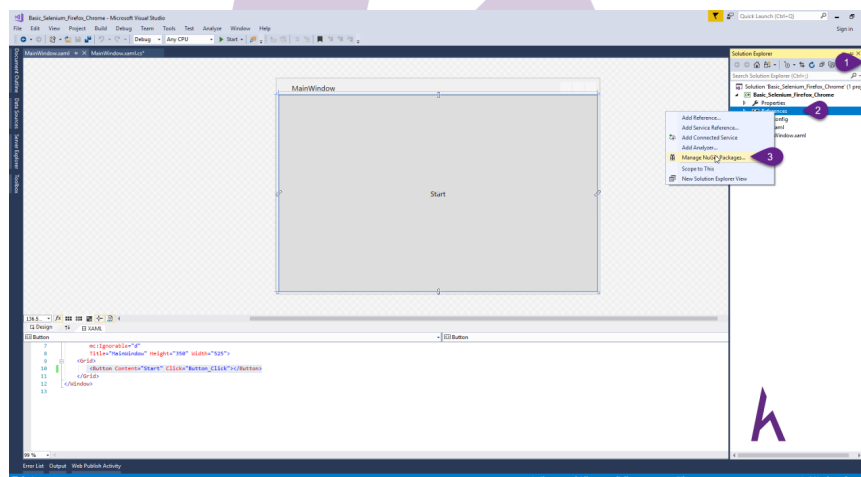
**Sẽ tự tạo một event ở code behind**

Bạn nào đã có kiến thức WPF thì có thể bỏ qua. Bạn nào chưa biết có thể xem qua Serial [LẬP TRÌNH WPF CƠ BẢN](#).

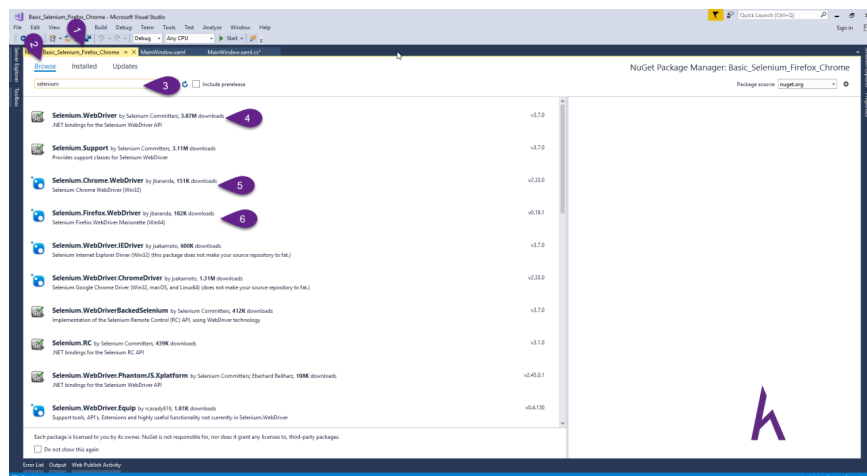
Chúng ta sẽ bắt đầu Import thư viện Selenium vào để dùng. Mình sẽ Import một lúc cả Firefox & Chrome nhé!

Cần 1 thư viện chính đó là IWebDriver còn lại tùy theo bạn dùng trình duyệt web nào mà chọn thư viện tương ứng.

Mở tab **Solution Explore** > Nhấp chuột phải vào **References** > Chọn **Manage Nuget Packages**.



Trong tab **Nuget** mở ra. Chọn **Browse** và gõ từ khóa **Selenium** vào thanh Search. Install **Selenium.WebDriver** và **Selenium.Chrome.WebDriver** hoặc **Selenium.Firefox.WebDriver** tùy theo trình duyệt bạn dùng.



Sau đó, qua phần code Behind là file **MainWindow.xaml.cs** để bắt đầu với phần chính.

## Navigate tới một URL với Selenium

Để thao tác với một trình duyệt thì mình phải thử vào trang web nào đó đúng không?

Vậy trước tiên mình sẽ khởi tạo trình duyệt đó đã! Ở đây mình sẽ chạy cùng lúc Firefox và Chrome cho các bạn thử nhé!

Selenium cũng là class như các class khác nên khởi tạo nó thì mình sẽ tạo một Instance của nó theo cách:

- **FirefoxDriver:**

:

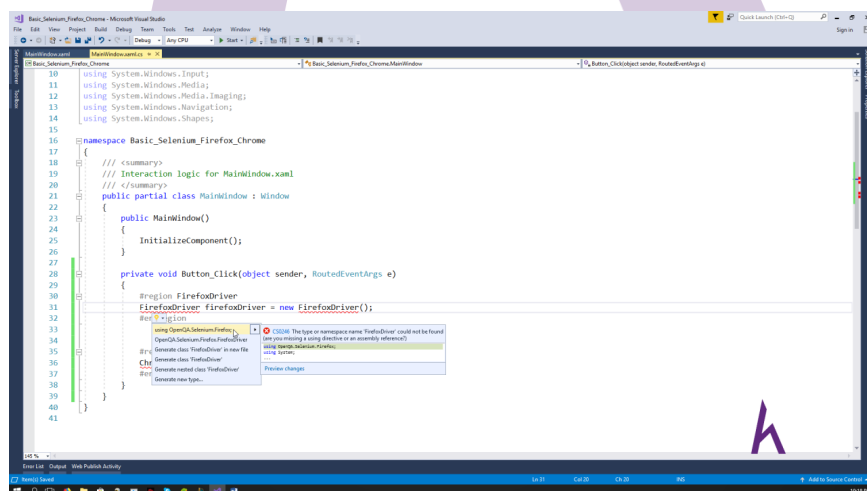
```
FirefoxDriver firefoxDriver = new FirefoxDriver();
```

- **ChromeDriver:**

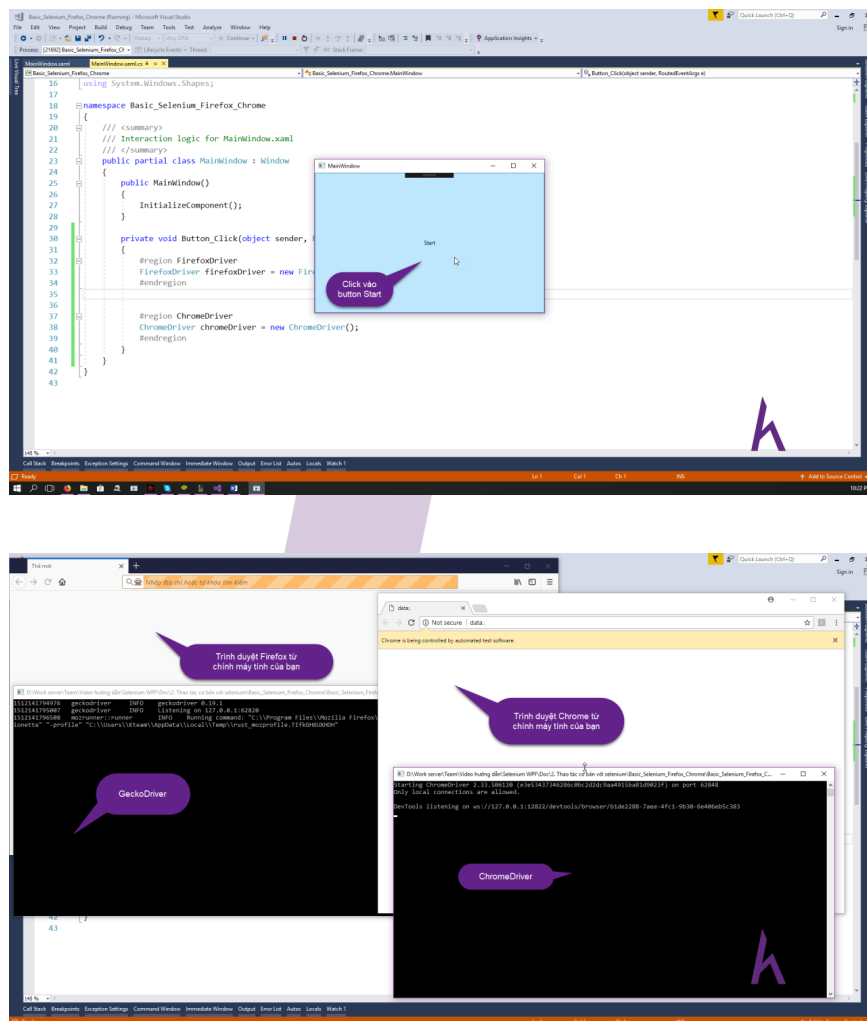
:

```
ChromeDriver chromeDriver = new ChromeDriver();
```

Các bạn thấy Visual báo lỗi ở **FirefoxDriver** và **ChromeDriver** vì các bạn chưa import thư viện vào. Cách làm nhanh là rê chuột vào tên class > chọn **biểu tượng bóng đèn** > Chọn **Using OpenQA.Selenium.Firefox** theo hình bên dưới (tương tự với Chrome nhé!):



Tới đây, chúng ta chạy thử chương trình. Nhấn button **Start** và sẽ thấy hai cửa sổ **CMD** hiện lên, mỗi cái tương ứng với một WebBrowser và sau đó là hai trình duyệt Firefox & Chrome. Sau này, bạn chỉ cần chọn dùng một trong hai WebDriver thôi. Ở đây, mình ví dụ cho các bạn thấy nên dùng một lúc cả hai.



Từ đây, bạn thấy ngay sau khi bạn khởi tạo WebDriver thì chương trình đã gọi trình duyệt lên cho bạn rồi. Nhưng giờ chỉ mới là mở trình duyệt. Cùng tìm hiểu cách chuyển trang nhé!

## Cách để Navigate tới một URL (Chrome và Firefox tương tự)

**Cách 1** (nên dùng cho rõ ràng):

:

```
firefoxDriver.Url = "http://www.howkteam.com/";
firefoxDriver.Navigate();

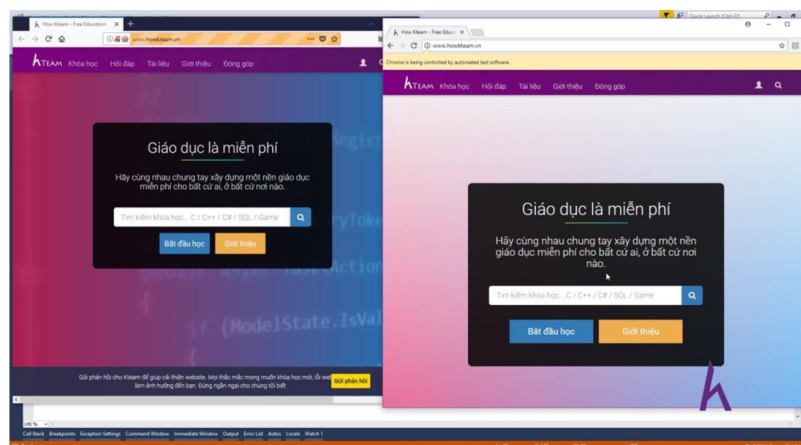
chromeDriver.Url = "http://www.howkteam.com/";
chromeDriver.Navigate();
```

**Cách 2:**

:

```
firefoxDriver.Navigate().GoToUrl("http://www.howkteam.com/");
chromeDriver.Navigate().GoToUrl("http://www.howkteam.com/");
```

Khi chạy thử, bạn sẽ thấy lúc này 2 trình duyệt đã chuyển đến trang **Howkteam.com**



## Một số thuộc tính và phương thức cơ bản trong Selenium

Kteam đã note khá nhiều thuộc tính và phương thức từ cơ bản đến nâng cao của Selenium trong bài ghi chú này:

### [Selenium FirefoxDriver ChromeDriver .Net | Notes](#)

Qua đây, mình cũng tổng hợp lại cho các bạn dễ hình dung:

- Lấy Source page hiện tại:

:

```
driver.PageSource
```

- Lấy địa chỉ URL hiện tại:

:

```
driver.Url
```

- Chuyển tới trang trước đó:

:

```
driver.Navigate().Back()
```

- Chuyển tới trang sau đó:

:

```
driver.Navigate().Forward()
```

- F5 (Refresh) lại trang:

:

```
driver.Navigate().Refresh()
```

- Handle arlet:

:

```
IAAlert a = driver.SwitchTo().Alert();  
  
a.Accept();  
  
a.Dismiss();
```

- Chuyển đổi giữa các cửa sổ hoặc tab:

:

```
ReadOnlyCollection<string> windowHandles = driver.WindowHandles;  
  
string firstTab = windowHandles.First();  
  
string lastTab = windowHandles.Last();  
  
driver.SwitchTo().Window(lastTab);
```

- Maximize window:

:

```
this.driver.Manage().Window.Maximize();
```

- Add cookies:

:

```
Cookie cookie = new OpenQA.Selenium.Cookie("key", "value");  
  
this.driver.Manage().Cookies.AddCookie(cookie);
```

- Get all cookies:

:

```
var cookies = this.driver.Manage().Cookies.AllCookies;
```

- Delete cookie:

:

```
this.driver.Manage().Cookies.DeleteCookieNamed("CookieName");
```

- Delete all cookies:

:

```
this.driver.Manage().Cookies.DeleteAllCookies();
```

- Chụp màn hình:

:

```
Screenshot screenshot = ((ITakesScreenshot)driver).GetScreenshot();  
  
screenshot.SaveAsFile(@"pathToImage", ImageFormat.Png);
```

**@pathToImage** là đường dẫn tới file hình sẽ lưu.

- Đợi đến khi Website load xong hết các đoạn JavaScript:

:

```
WebDriverWait wait = new WebDriverWait(this.driver, TimeSpan.FromSeconds(30));

wait.Until((x) =>

{

    return ((IJavaScriptExecutor)this.driver).ExecuteScript("return document.readyState").Equals("complete");

});
```

- Chuyển đổi giữa các frames:

:

```
this.driver.SwitchTo().Frame(1);

this.driver.SwitchTo().Frame("frameName");

IWebElement element = this.driver.FindElement(By.Id("id"));

this.driver.SwitchTo().Frame(element);
```

- Chuyển tới document mặc định:

:

```
this.driver.SwitchTo().DefaultContent();
```

## Get Element với Selenium

Cách lấy **WebElement** cơ bản nhất của Selenium là theo: id, class name, name và tag.

Ở phần này, mình sẽ giới thiệu những cách cơ bản này. Ở những phần sau, mình sẽ giới thiệu tới các cách lấy nâng cao hơn, khi các thuộc tính này không tồn tại hoặc khó xác định ra đối tượng.

Có những phương thức lấy đối tượng ra từ Selenium như:

```
driver.FindElementByld("ID của HTML element mà bạn muốn tìm")
```

Hàm này tương ứng với:

```
driver.FindElement(By.Id("id"))
```

**Ví dụ cụ thể:** Ở website **Howkteam.com**, mình muốn điền giá trị tìm kiếm tự động vào thanh tìm kiếm có ID là **searchUI**. Mình sẽ tìm đối tượng này ra và gọi hàm **SendKeys** để điền giá trị mong muốn vào.

:

```
// tìm đối tượng theo ID
var searchBar = firefoxDriver.FindElementById("searchUI");

// gõ giá trị Selenium Kteam vào thanh tìm kiếm
searchBar.SendKeys("Selenium Kteam");
```



Tới đây code tổng thể của event **Button\_Click** của chúng ta là:

- Mình bỏ code của Chrome ra vì tương tự Firefox

:

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    #region FirefoxDriver
    // khởi tạo WebDriver
    FirefoxDriver firefoxDriver = new FirefoxDriver();

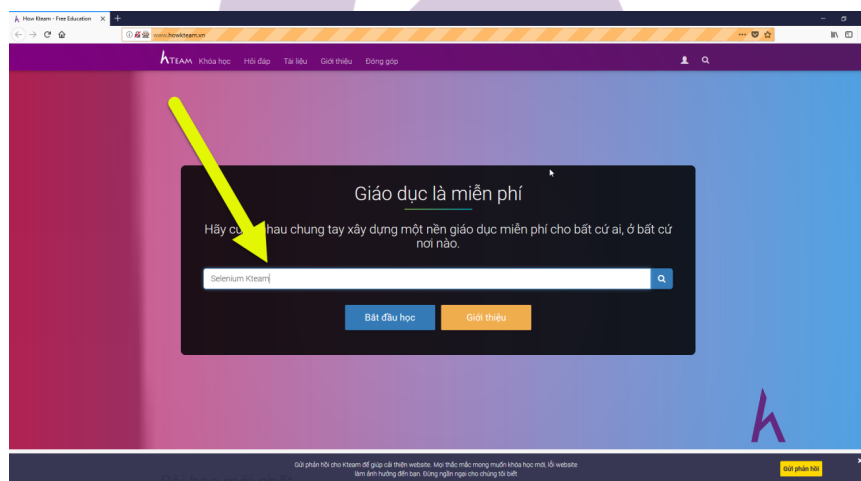
    // chuyển trang đến website howkteam.com
    firefoxDriver.Url = "http://www.howkteam.com/";
    firefoxDriver.Navigate();

    // tìm đối tượng theo ID
    var searchBar = firefoxDriver.FindElementById("searchUI");

    // gõ giá trị Selenium Kteam vào thanh tìm kiếm
    searchBar.SendKeys("Selenium Kteam");
    #endregion

    #region ChromeDriver
    //ChromeDriver chromeDriver = new ChromeDriver();
    //chromeDriver.Url = "http://www.howkteam.com/";
    //chromeDriver.Navigate();
    #endregion
}
```

Cùng chạy thử chương trình và tận hưởng kết quả nào!



Thanh tìm kiếm đã tự điền giá trị vào. WOW! Tự vỗ tay khen thưởng mình nào! Bạn đã đặt một chân vào con đường làm tool auto rồi đó! (ít nhất là còn đường kiểm thử tự động)

Khi đã nắm cách dùng của **FindElementById** thì các hàm tương tự khác cũng đơn giản với các bạn. Các bạn có thể tự tìm hiểu thêm các hàm này!

:

```
this.driver.FindElement(By.ClassName("className"));

this.driver.FindElement(By.CssSelector("css"));

this.driver.FindElement(By.Id("id"));

this.driver.FindElement(By.LinkText("text"));

this.driver.FindElement(By.Name("name"));

this.driver.FindElement(By.PartialLinkText("pText"));

this.driver.FindElement(By.TagName("input"));
```

### Lưu ý:

Có một lưu ý cho các bạn giữa hai hàm:

:

```
firefoxDriver.FindElementByClassName("");

firefoxDriver.FindElementsByClassName("");
```

- Hai hàm này sẽ cho ra kết quả khác nhau. Ở trên sẽ cho ra kết quả tìm thấy class tương ứng đầu tiên. Ở dưới sẽ cho ra một danh sách các class tương ứng được tìm thấy.

Và một lưu ý thứ hai: các element này đang được tìm ở Window hoặc frame đang active.

Chúng ta tiếp tục với hàm Click.

Các bạn hoàn toàn có thể click vào một element nào đó sau khi tìm thấy nó bằng hàm **Click()**

Cùng nhau thử click vào Button **Bắt đầu học** nhé!

Button **Bắt đầu học** không có thuộc tính id, tag, name hay cả link. Nên chúng ta hiện chỉ có thể dựa vào thuộc tính **class name**. Và Button này có **class name** là: **btn btn-primary**

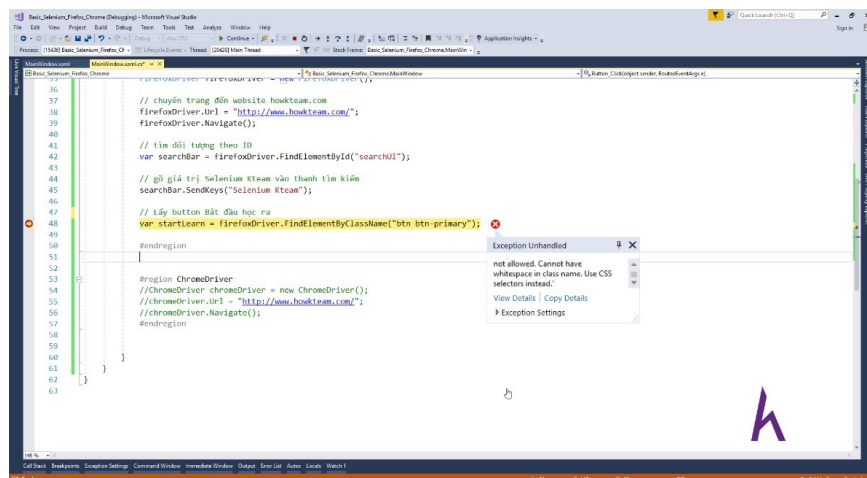
Code lấy element này ra:

:

```
// Lấy button Bắt đầu học ra

var startLearn = firefoxDriver.FindElementByClassName("btn btn-primary");
```

Khi chạy thử chương trình báo lỗi: **Class name không hợp lệ**. Hóa ra Selenium không hỗ trợ class name có khoảng trắng! @@. Cái này phải chấp nhận đau thương thôi!



Để đơn giản hơn chúng ta có thể thử một class viết liền. Nhưng mình sẽ để dành bài tập đó cho các bạn!

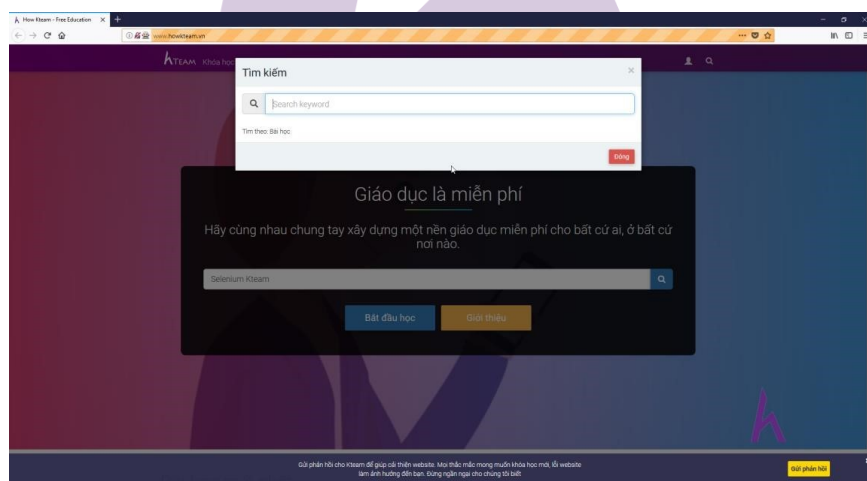
Mình có một element khác có ID và cũng để thử nghiệm hơn đó là Icon **Search** ở góc trên phải của website **Howkteam.com**. Khi nhấn vào icon đó thì một pop-up search sẽ hiện lên. Icon này có id là **tagssearch**. Mình sẽ thử nhấn vào nó:

:

```
// Lấy tagssearch ra
var tagssearch = firefoxDriver.FindElementById("tagssearch");

// click vào tagssearch
tagssearch.Click();
```

Như các bạn thấy search pop-up đã hiện lên một cách tự động



Bravo! À hi hi! Càng này càng thấy thích thú với cái này rồi đây! Đây là những thao tác cơ bản với Selenium mà bạn nên biết. Firefox và Chrome khá tương đồng nhau.

Đừng ngần ngại bật Website [Howkteam.com](https://www.howkteam.com) lên mà thử nghiệm những thứ bạn đã học được nhé!

## Kết luận

Qua bài viết này, bạn đã nắm được các thao tác cơ bản trong Selenium.

Ở bài sau, Kteam sẽ giới thiệu đến bạn [CÁCH EXECUTE MỘT ĐOẠN JAVASCRIPT](#) trong Selenium.

Cảm ơn bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó!"**

