## 

## 

## 

## 

## **Testing Management Tools**

## **Selenium**

Class: SE1617

Group: I

Subject: SWT301

Nguyễn Đăng Nhật (Leader)

Nguyễn Thị Lan Anh

Hà Gia Kính

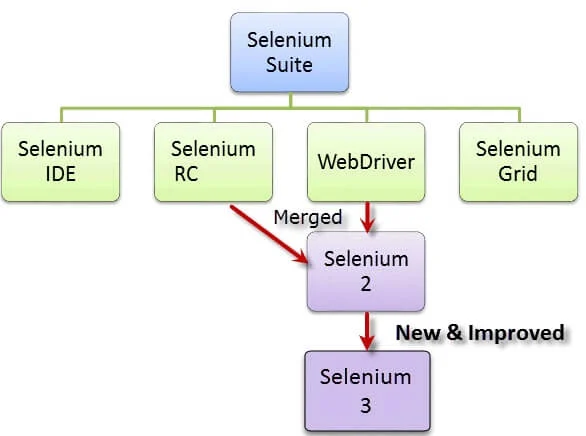
Đỗ Nhật Đức

# **I. Giới thiệu về Selenium**

# 1. Selenium là gì

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức. Nó có 4 thành phần:

* Selenium Integrated Development Environment (IDE):
* Selenium Remote Control (RC)
* WebDriver
* Selenium Grid



Hiện tại, Selenium RC và WebDriver được hợp nhất thành một framework duy nhất để tạo ra Selenium 2. Còn Selenium 1 thì tham chiếu đến Selenium RC.

# 2. Lịch sử của Selenium

# a. Người phát triển, tạo ra Selenium

Do Selenium là một tập hợp các công cụ khác nhau nên cũng có những người phát triển khác nhau. Dưới đây là những người chủ chốt đã có đóng góp đáng kể cho dự án Selenium. Chủ yếu Selenium được tạo ra bởi Jason Huggins năm 2004. Ông là một kỹ sư của ThoughtWorks, lúc đó đang làm việc trên một ứng dụng web đòi hỏi mức độ phải kiểm tra thường xuyên. Sau khi nhận ra rằng việc kiểm tra ứng dụng một cách thủ công và lặp đi lặp lại ngày càng trở nên không hiệu quả, ông đã tạo ra một chương trình JavaScript có thể tự động kiểm soát các hành động của trình duyệt. Ông đặt tên cho chương trình này là "JavaScript Test Runner." Nhìn thấy tiềm năng trong ý tưởng này để giúp tự động hóa các ứng dụng web khác, ông đã tạo ra mã nguồn mở JavaScript Runner, sau này được đặt tên lại là Selenium Core.

# **II. Thành phần chính của Selenium**

# 1. Selenium IDE

Môi trường phát triển tích hợp Selenium (IDE) là khuôn khổ đơn giản nhất trong bộ Selenium và là cách đơn giản nhất để học. Đây là một plugin của Firefox mà bạn có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì tính đơn giản của nó, Selenium IDE chỉ nên được sử dụng như là một công cụ tạo mẫu. Nếu bạn muốn tạo các trường hợp thử nghiệm nâng cao hơn, bạn sẽ cần phải sử dụng Selenium RC hoặc WebDriver.

| PROS | CONS |
| --- | --- |
| Sử dụng đơn giản và dễ cài đặt | Chỉ có sẵn trong Firefox |
| Không yêu cầu kinh nghiệm lập trình, mặc dù cần có đủ kiến thức về HTML và DOM | Được thiết kế để tạo ra nguyên mẫu của các lần test |
| Chỉ có thể xuất các lần test sang các định dạng có thể sử dụng được trong Selenium RC và WebDriver | Quá trình thực thi kiểm tra chậm hơn so với Selenium RC và WebDriver |

# 2. Selenium Remote Control(Selenium RC)

Selenium RC là framework thử nghiệm hàng đầu của toàn bộ dự án Selenium trong một thời gian dài. Đây là công cụ kiểm tra web tự động đầu tiên cho phép người dùng sử dụng ngôn ngữ lập trình mà họ thích. Tính đến phiên bản 2.25.0, RC có thể hỗ trợ các ngôn ngữ lập trình sau:

* Java
* C#
* PHP
* Python
* Peri
* Ruby

| PROS | CONS |
| --- | --- |
| Đa trình duyệt và đa nền tảng | Cài đặt phức tạp hơn IDE |
| Có thể thực hiện các hoạt động độc lập và có điều kiện | Phải có kiến thức lập trình |
| Có thể hỗ trợ kiểm tra theo hướng dữ liệu | Cần Selenium RC Server để chạy |
|  |  |

# 3. WebDriver

WebDriver chứng tỏ bản thân nó tốt hơn cả Selenium IDE và Selenium RC về nhiều mặt. Nó thực hiện tiếp cận hiện đại và ổn định hơn trong tự động hoá các hành động của trình duyệt. WebDriver, không giống như Selenium RC, không phụ thuộc vào JavaScript cho Tự động hóa. Nó điều khiển trình duyệt bằng cách liên lạc trực tiếp với nó. Các ngôn ngữ được hỗ trợ giống với Selenium RC.

* Java
* C#
* PHP
* Python
* Perl
* Ruby

| **Ưu điểm** | **Nhược điểm** |
| --- | --- |
| Cài đặt đơn giản hơn | Cài đặt phức tạp hơn Selenium IDE |
| Liên hệ trực tiếp với trình duyệt | Yêu cầu lập trình hiểu biết |
| Tương tác với trình duyệt nhiều hơn thực tế | Không thể sẵn sàng hỗ trợ mới trình duyệt |
| Không cần riêng thành phần chẳng hạn như RC Server | Không có cơ chế tích hợp ghi nhật ký thông báo thời gian chạy và tạo kết quả kiểm tra |
| Thời gian thực hiện nhanh hơn IDE và RC |  |

# 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 kiểm thử cùng một lúc. Tính năng:

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

## 

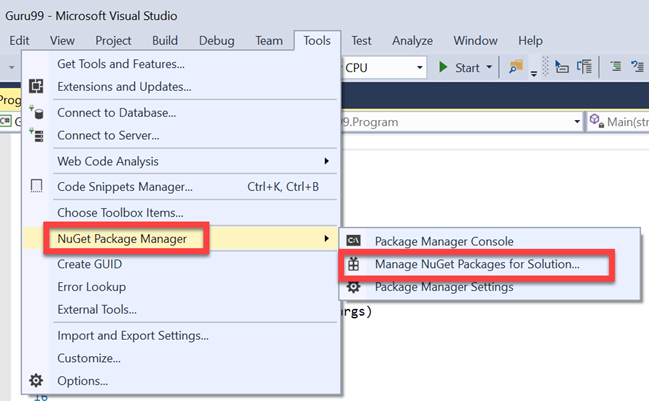
## \*\*\*Cách thức lựa chọn công cụ Selenium phù hợp với nhu cầu của bạn\*\*\*

| Công cụ | Tại sao phải chọn |
| --- | --- |
| Selenium IDE | Tìm hiểu về các khái niệm về kiểm thử tự động và Selenium, bao gồm:   * Các lệnh của Selenium như type, open, clickAndWait, assert, verify, etc. * Locators như id, tên, xpath, selector, css,... * Thực hiện mã JavaScript tùy chỉnh bằng cách sử dụng runScript * XUất các trường hợp thử nghiệm ở các định dạng khác nhau * Tạo ra kiểm thử có ít hoặc không có kiến thức về lập trình * Để tạo các trường hợp kiểm thử đơn giản và các bộ kiểm thử mã bạn có thể xuất sau đó đến RC hoặc WebDriver * Để kiểm tra một ứng dụng web chỉ với Firefox. |
| Selenium RC | * Thiết kế một kiểm thử sử dụng một nguồn ngôn ngữ biểu cảm hơn Selenese * Để chạy kiểm thử với các trình duyệt khác nhau (ngoại trừ HtmlUnit) trên các hệ điều hành khác nhau * Để triển khai kiểm thử trên nhiều môi trường sử dụng Selenium Grid. * Để kiểm tra ứng dụng của bạn trên một trình duyệt mới hỗ trợ JavaScript * Để kiểm tra các ứng dụng web với các kịch bản dựa trên AJAX phức tạp |
| WebDriver | * Để sử dụng một ngôn ngữ lập trình nhất định trong việc thiết kế các trường hợp kiểm thử * Để kiểm tra các ứng dụng có nhiều chức năng dựa trên AJAX * Để thực hiện các bài kiểm tra trên trình duyệt HtmlUnit * Tạo ra các kết quả kiểm tra tùy chỉnh |
| Selenium Grid | * Để chạy các kịch bản Selenium RC của bạn trong nhiều trình duyệt và hệ điều hành cùng một lúc * Để chạy một bộ phần mềm kiểm thử khổng lồ, cần phải hoàn thành trong thời gian sớm nhất có thể |
|  |  |

# **III. Cài đặt và sử dụng Selenium bằng ngôn ngữ C#**

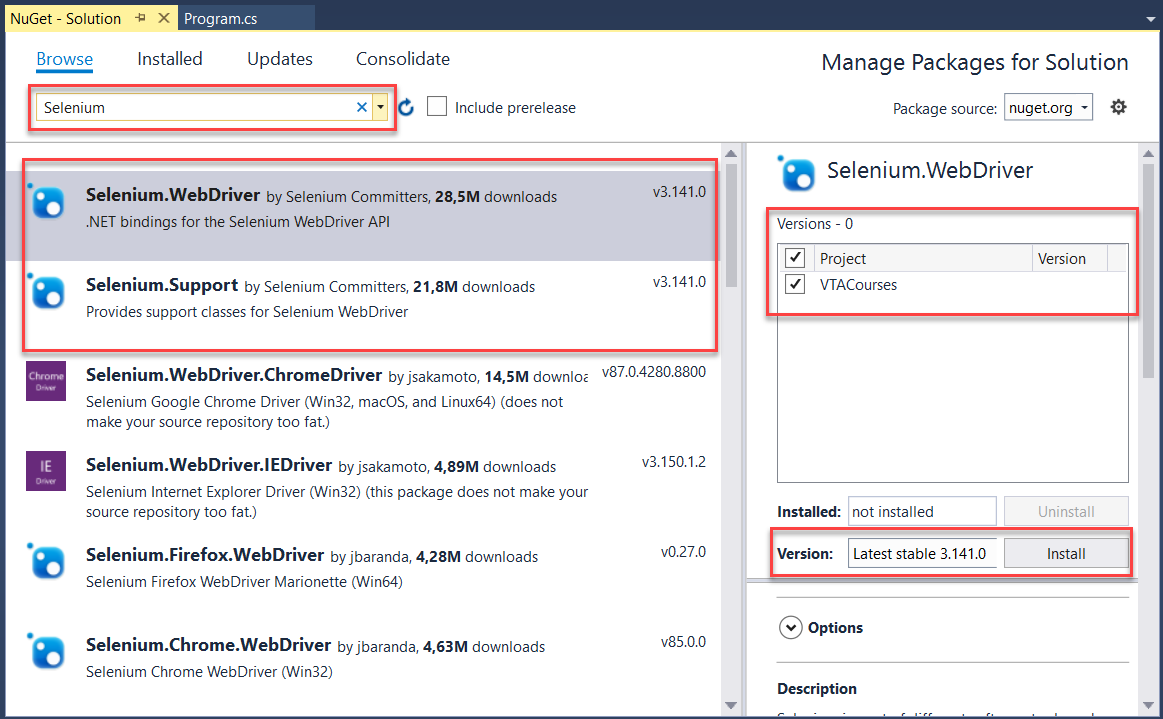
# 1. Cài đặt Selenium

# \* Bước 1: Điều hướng đến **Tools** => **NuGet Package Manager** => **Manager Nuget Packages for Solution**

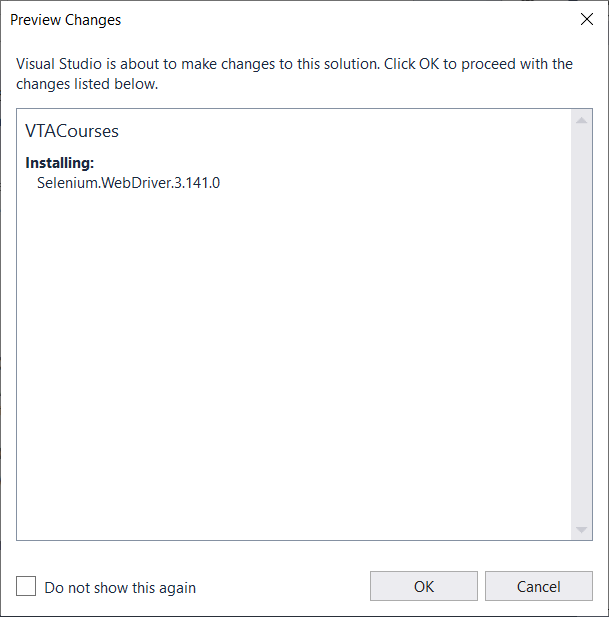


# \* Bước 2: Trong màn hình tiếp theo

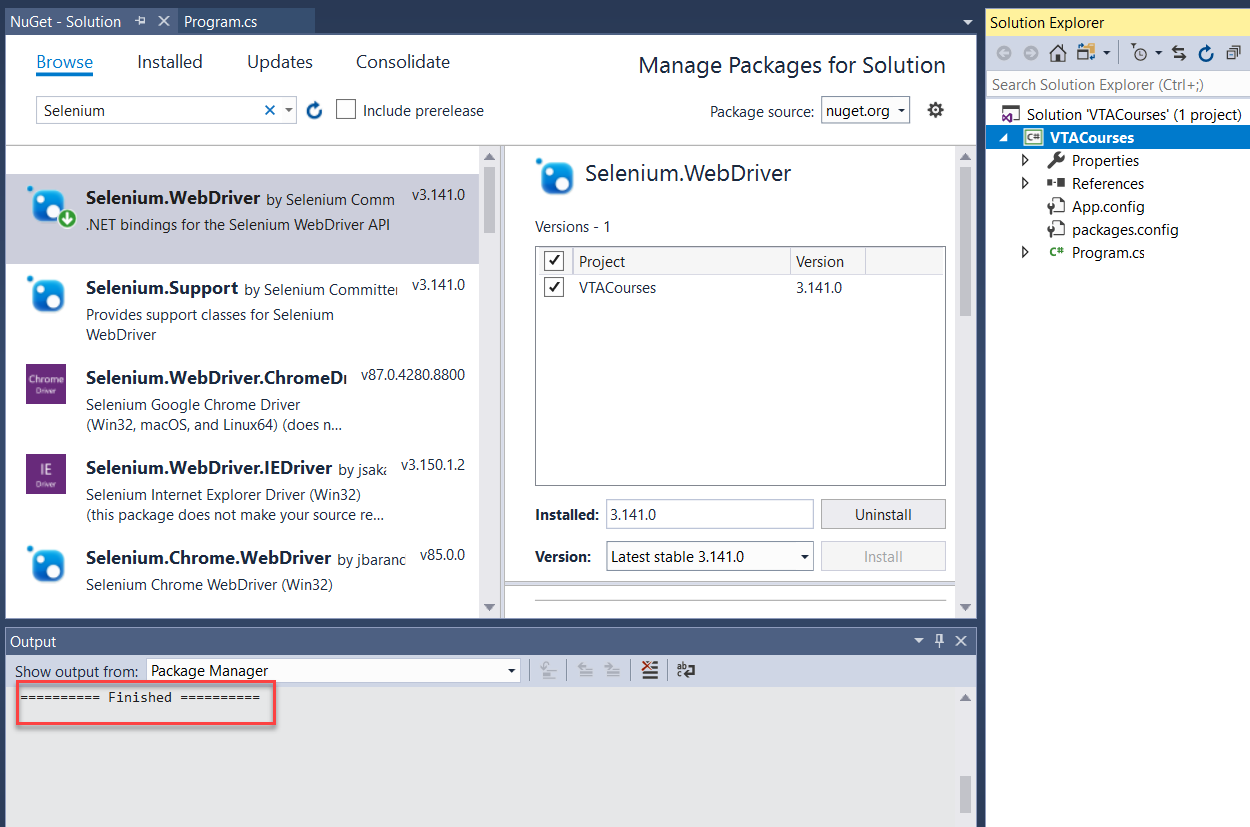
1. Tìm kiếm **Selenium** trên màn hình kết quả
2. Chọn kết quả tìm kiếm đầu tiên
3. Đánh dấu vào hộp kiểm dự án. **Chọn version mới nhất luôn cho máu :D**
4. Nhấn vào “**Install**”



# \* Bước 3: Nhấp nút ‘OK’ trong màn hình bật lên

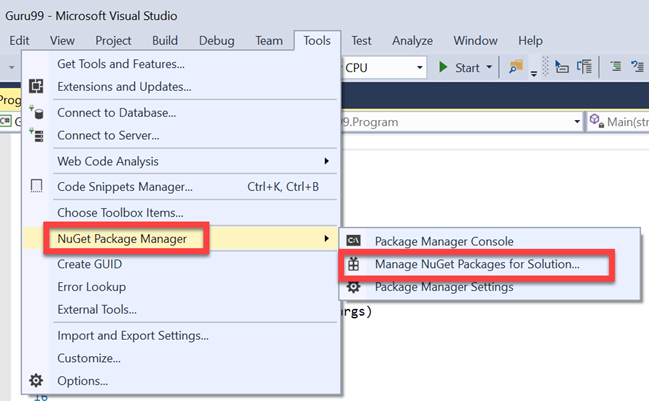


# \* Bước 4: Trong báo dưới đây sẽ được hiển thị khi gói được cài đặt thành công.



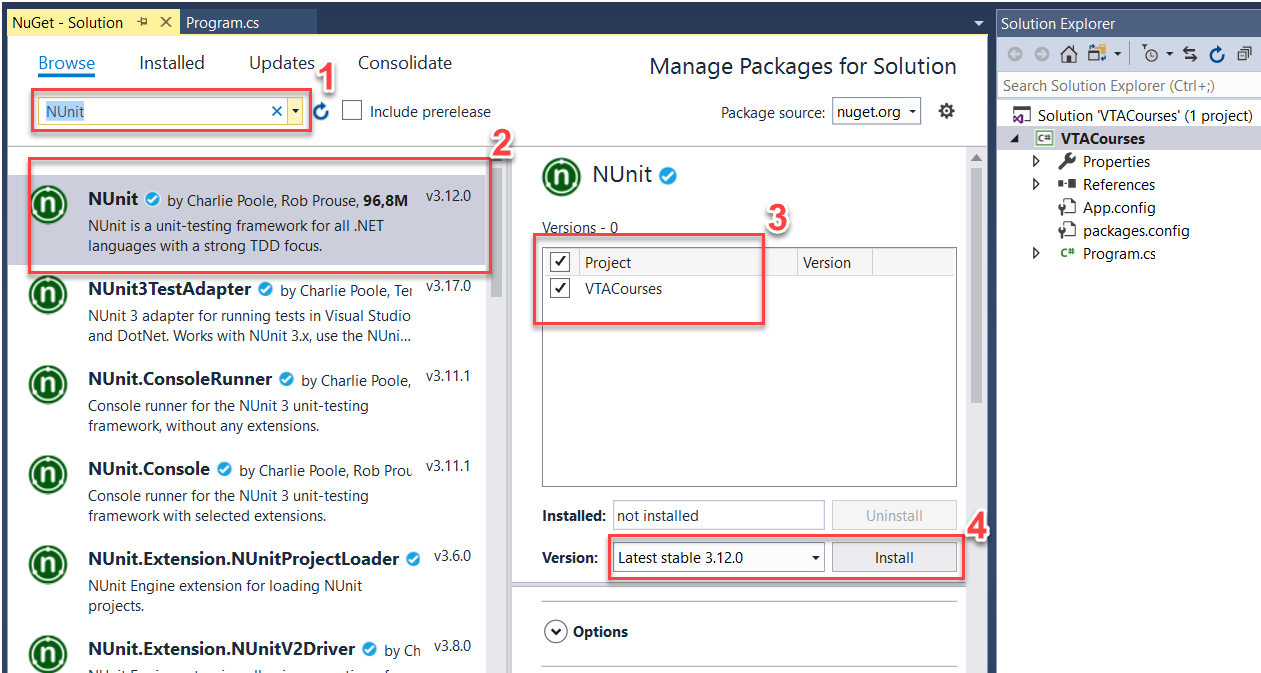
# 2. Cài đặt NUnit Framework:

# \* Bước 1: Điều hướng đến **Tools** => **NuGet Package Manager** => **Manager Nuget Packages for Solution**

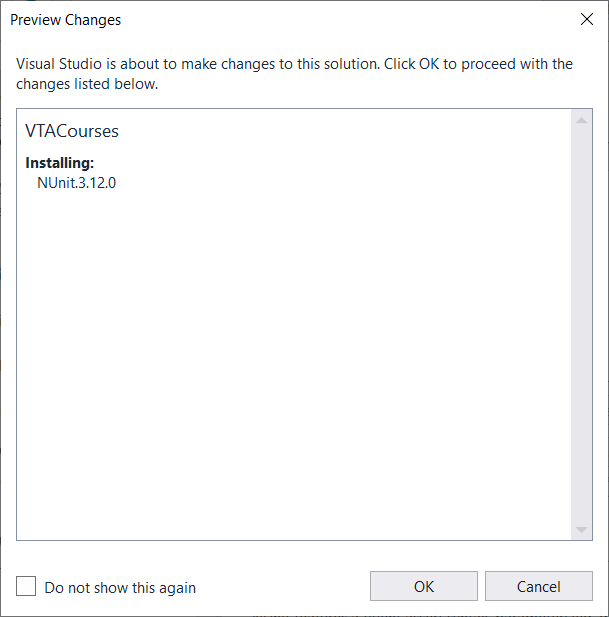


# \* Bước 2: Trong cửa sổ tiếp theo

1. Tìm kiếm **NUnit**
2. Chọn kết quả tìm kiếm
3. Chọn **Project**, Click **Version** chọn mới nhất
4. Nhấp vào “**Install**”



# \* Bước 3: Cửa sổ bật lên bên dưới sẽ xuất hiện. Nhập vào nút ‘OK’

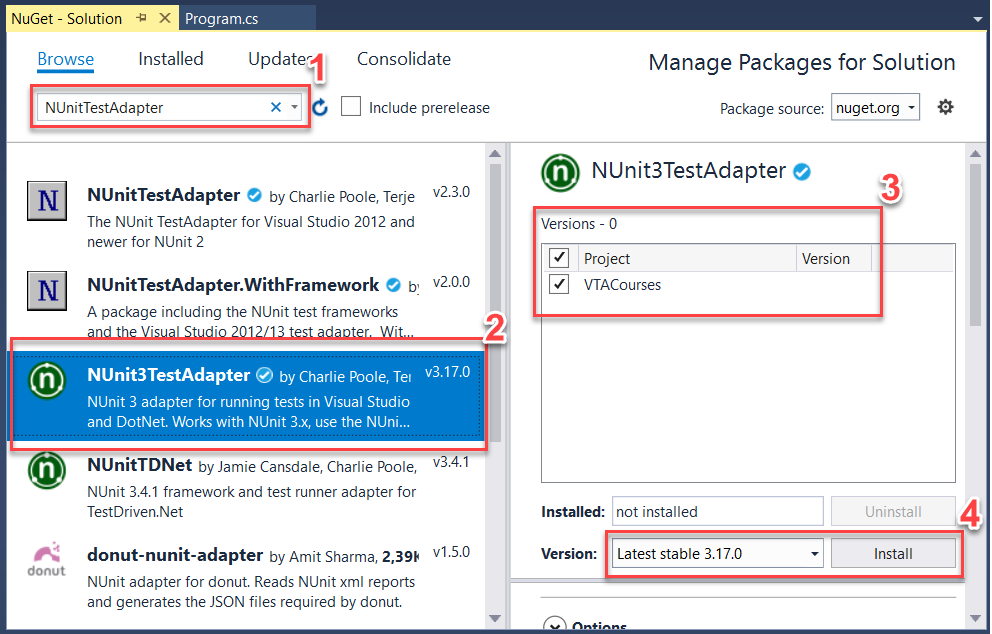


# \* Bước 4: Thông báo **Success** dưới Output xuất hiện khi quá trình cài đặt hoàn tất

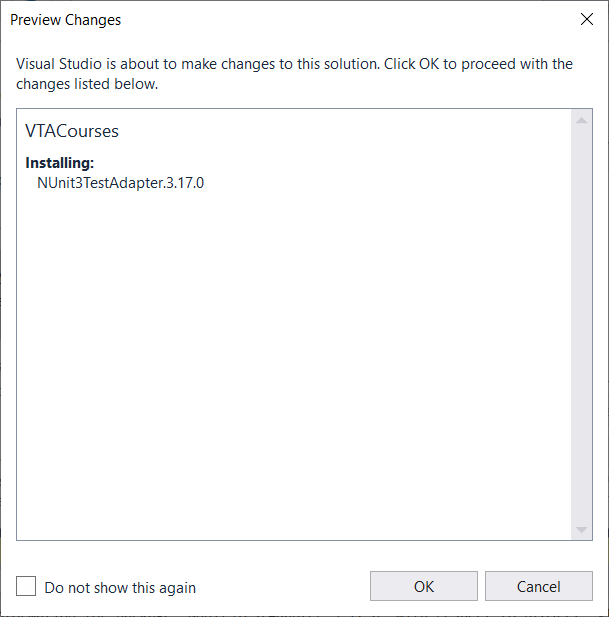
# 3. Cài đặt NUnit Test Adapter:

# \* Bước 1: Đều hướng đến Tools => NuGet Package Manager => Manage NuGet Packages for Solution. Trong màn hình đó:

1. Tìm kiếm **UnitTestAdapter**
2. Nhấp vào Kết quả Tìm Kiếm
3. Chọn **Project**. Click chọn Version mới nhất
4. Nhấp vào “**Install**”



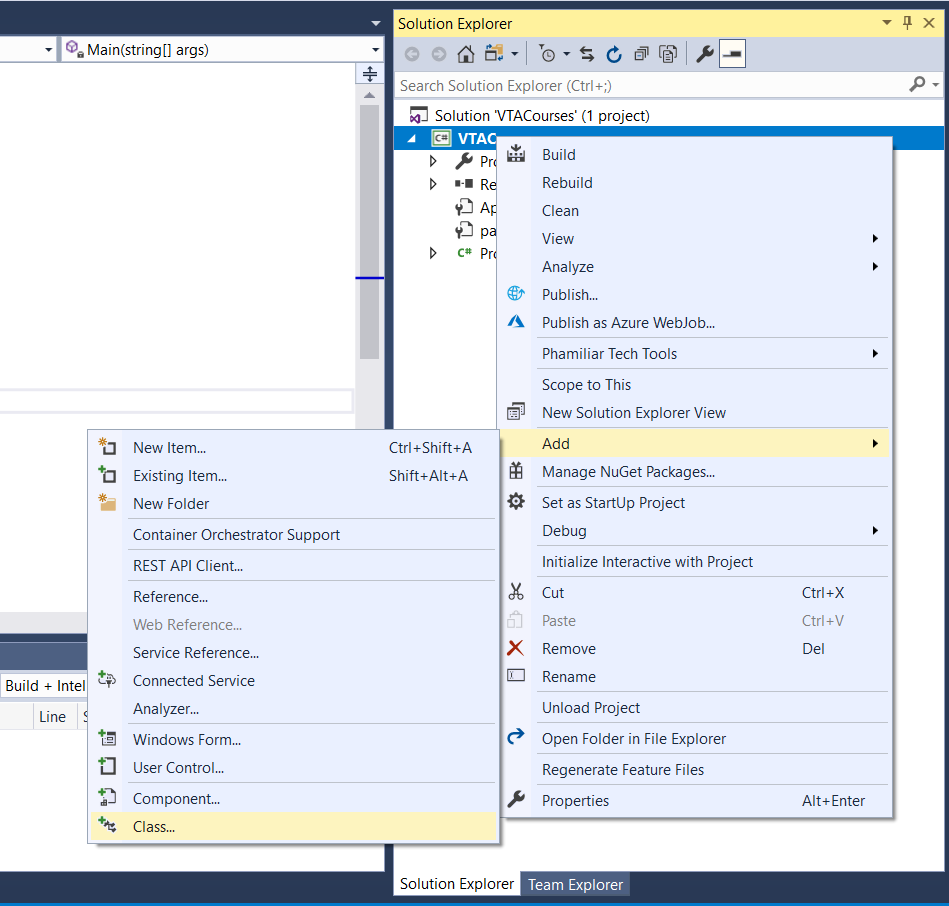
# \* Bước 2: Nhấp vào **OK** trên cửa sổ bật lên xác nhận



# 4. Các bước để tạo một lớp NUnit Test trong Selenium:

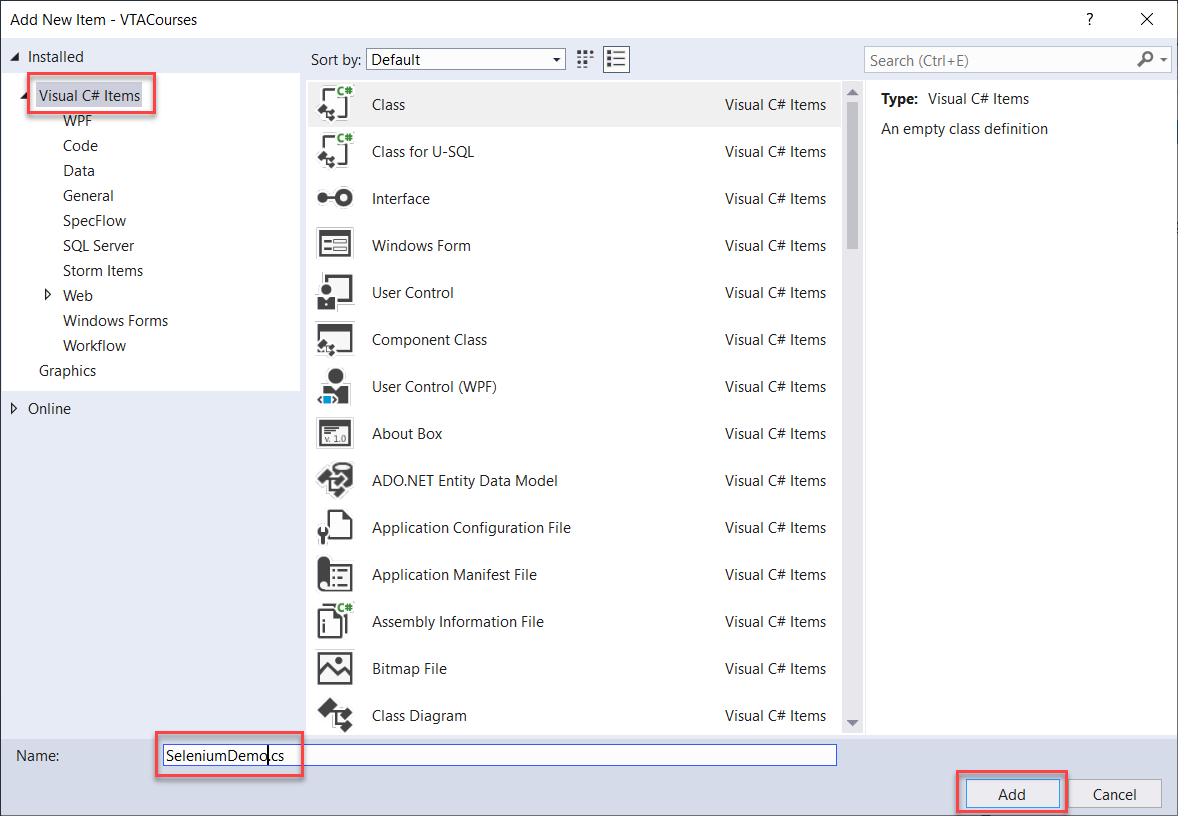
# \* Bước 1: Trong trình khám phá Solution, Nhấp chuột

Bước 1) Trong Trình khám phá Solution, Nhấp chuột phải vào tên Project > Add > Class

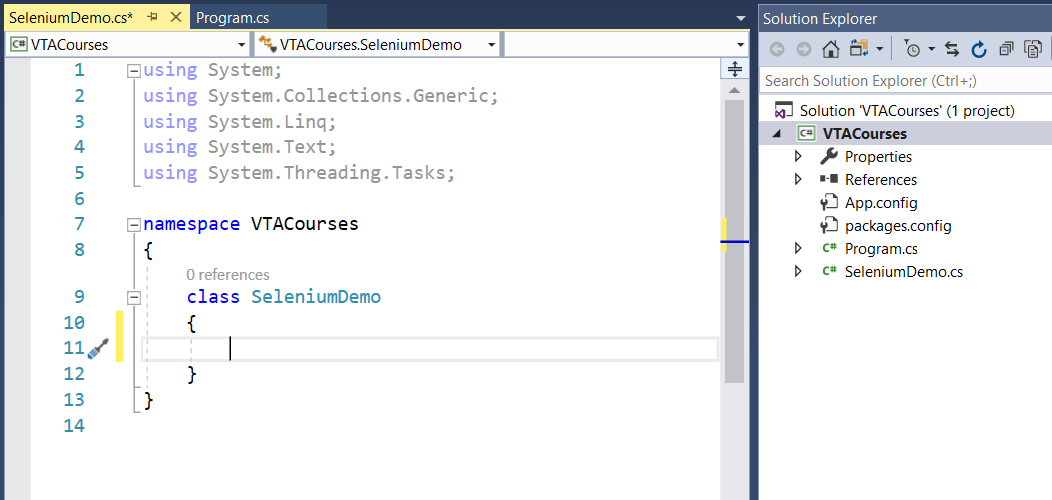


Bước 2) Cửa sổ tạo Class sẽ xuất hiện.

1. Cung cấp tên cho Class
2. Bấm vào nút "Add"



Màn hình dưới đây sẽ xuất hiện.



Bước 3) Thêm mã sau vào Class đã tạo.

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Firefox;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace VTACourses

{

class SeleniumDemo

{

IWebDriver driver;

[SetUp]

public void startBrowser()

{

driver = new ChromeDriver("D:\\TESTER\\Automation Testing\\Project");

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

}

[Test]

public void Demo1()

{

driver.Url = "https://testing.devforum.info/";

IWebElement element = driver.FindElement(By.XPath("//a[normalize-space()='Add/Remove Elements']"));

element.Click();

IWebElement button = driver.FindElement(By.XPath("//button[normalize-space()='Add Element']"));

button.Click();

}

[TearDown]

public void closeBrowser()

{

driver.Close();

}

}

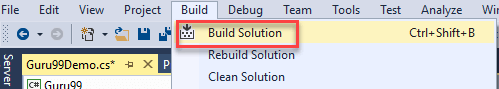
}

Xin lưu ý rằng cần chỉ định vị trí của tệp 'chromdriver.exe' trong quá trình khởi tạo trình điều khiển Chrome.

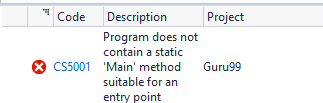
- Download version bằng với Version của Chrome đang dùng

- <http://chromedriver.storage.googleapis.com/index.html>

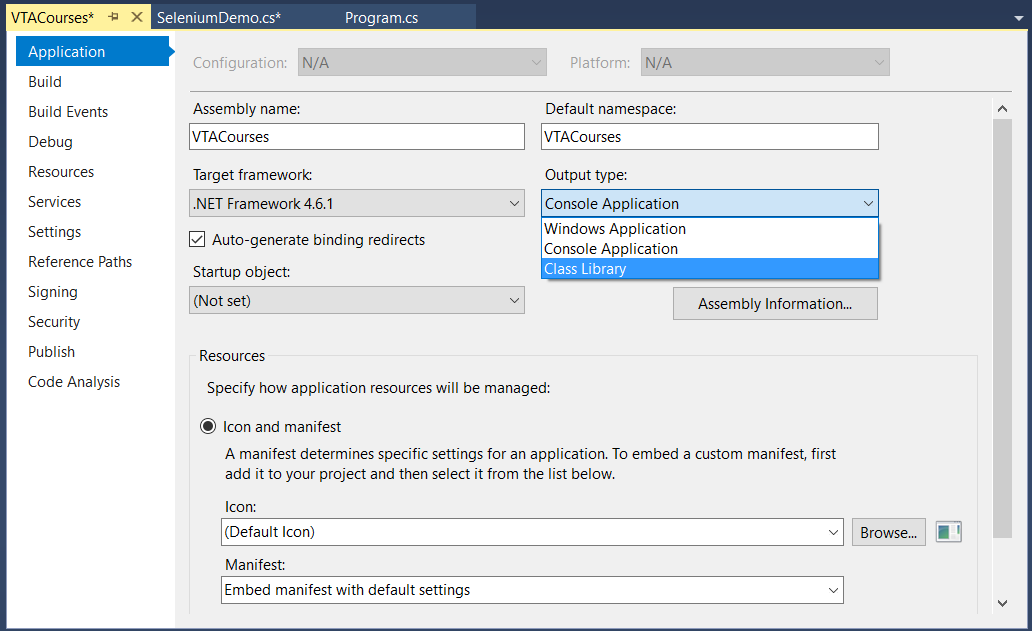
Bước 4) Nhấp vào 'Build' -> 'Build Solution'



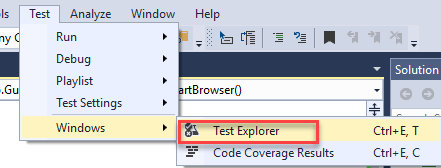
LƯU Ý: Bạn có thể gặp lỗi như sau



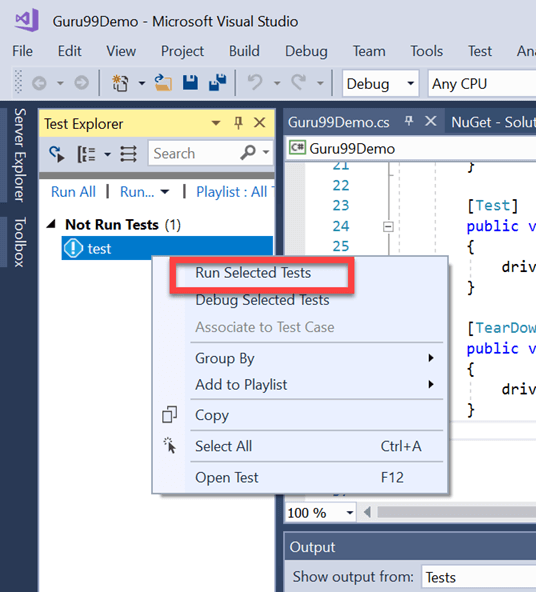
Để giải quyết điều này, Click chuột phải vào tên Project > Properties và thay đổi Loại đầu ra thành "Class Library". Vì mặc định là "Console Application".



Bước 5) Sau khi xây dựng thành công, chúng ta cần mở cửa sổ Test Explorer. Nhấp vào Test -> Windows -> Test Explorer



Bước 6) Cửa sổ Test Explorer mở ra với danh sách các bài kiểm tra có sẵn. Nhấp chuột phải vào Test Explorer và chọn Run Selected Tests



Bước 7) Selenium phải mở trình duyệt với URL được chỉ định và đóng trình duyệt. Trạng thái trường hợp kiểm tra sẽ được thay đổi thành 'Passed' trên cửa sổ Test Explorer.

