**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**uuu**

**BÁO CÁO BÀI TẬP LỚN KIỂM THỬ PHẦN MỀM**

***ĐỀ TÀI:* KIỂM THỬ WEBDRIVER PINTEREST**

**KIỂM THỬ API SPOTIFY**

**KIỂM THỬ NUNIT PT BẬC 2**

**GVHD:** Nguyễn Thị Ngọc Thanh

**SINH VIÊN THỰC HIỆN:** Nguyễn Vĩnh Khiêm

**MÃ SỐ SINH VIÊN :** 2151050192

**KHOA :** Công Nghệ Công Tin

**LỚP:** DH21IT03

**TPHCM, 05/2024**

**LỜI MỞ ĐẦU**

Pinterest là một công cụ tuyệt vời để khám phá và chia sẻ ý tưởng trực quan, tạo nên một không gian sáng tạo và đa dạng. Với hàng tỷ Ghim trên nền tảng này, người dùng có thể dễ dàng tìm thấy cảm hứng cho mọi lĩnh vực trong cuộc sống.

Nếu bạn đang tìm kiếm công thức nấu ăn mới, ý tưởng trang trí nhà cửa, hoặc cảm hứng về thời trang và phong cách, Pinterest sẽ là địa điểm lý tưởng. Người dùng có thể tìm kiếm theo từ khóa, duyệt qua các bảng ghim của người khác, và lưu lại những ý tưởng yêu thích vào bảng cá nhân của mình.

Với giao diện trực quan và dễ sử dụng, Pinterest không chỉ là một nguồn thông tin hữu ích mà còn là cộng đồng nơi mọi người có thể chia sẻ ý tưởng và tìm kiếm sự sáng tạo từ cộng đồng toàn cầu. Hãy để Pinterest làm nổi bật sự sáng tạo và khám phá trong cuộc sống hàng ngày của bạn!

**MỤC LỤC**

[**1.** **Tạo Một Project Unit Test** 5](#_Toc165553886)

[Bước 1: 5](#_Toc165553887)

[Bước 2: 5](#_Toc165553888)

[Bước 3: 6](#_Toc165553889)

[Bước 4: 7](#_Toc165553890)

[**2.** **Kiểm thử chức năng “Tạo” hình ảnh hoặc video** 9](#_Toc165553891)

[2.1. Mô tả chức năng 9](#_Toc165553892)

[2.2. Kiểm thử hộp đen 9](#_Toc165553893)

[2.2.1. Decision Table 9](#_Toc165553894)

[2.2.2. Giải thích code Test 10](#_Toc165553895)

[**3.** **Kiểm thử chức năng nhắn tin** 22](#_Toc165553896)

[3.1. Mô tả chức năng 22](#_Toc165553897)

[3.2. Kiểm thử hộp đen 23](#_Toc165553898)

[3.2.1. Decision Table 23](#_Toc165553899)

[3.2.2. Giải thích code Test 23](#_Toc165553900)

[**4.** **Test API trên Postman** 33](#_Toc165553901)

[4.1. Lấy Quyền trang website để test API 37](#_Toc165553902)

[4.2. Thực hiện test API 43](#_Toc165553903)

[4.2.1. Test API Movie 43](#_Toc165553904)

[4.2.2. Test API Favroite 49](#_Toc165553905)

[**5.** **Test NUnit phương trình bậc 2** 52](#_Toc165553906)

[5.1. Chuẩn bị môi trường 52](#_Toc165553907)

[Bước 1: 52](#_Toc165553908)

[Bước 2: 53](#_Toc165553909)

[Bước 3: 54](#_Toc165553910)

[Bước 4: 55](#_Toc165553911)

[Bước 5: 56](#_Toc165553912)

[Bước 6: 57](#_Toc165553913)

[Bước 7: 58](#_Toc165553914)

[Bước 8: 58](#_Toc165553915)

[5.2. Giải thích code 59](#_Toc165553916)

[Bước 1: 59](#_Toc165553917)

[Bước 2: 60](#_Toc165553918)

[Bước 3: 61](#_Toc165553919)

[Bước 4: Test case 1 – PT có 2 nghiệm phân biệt 63](#_Toc165553920)

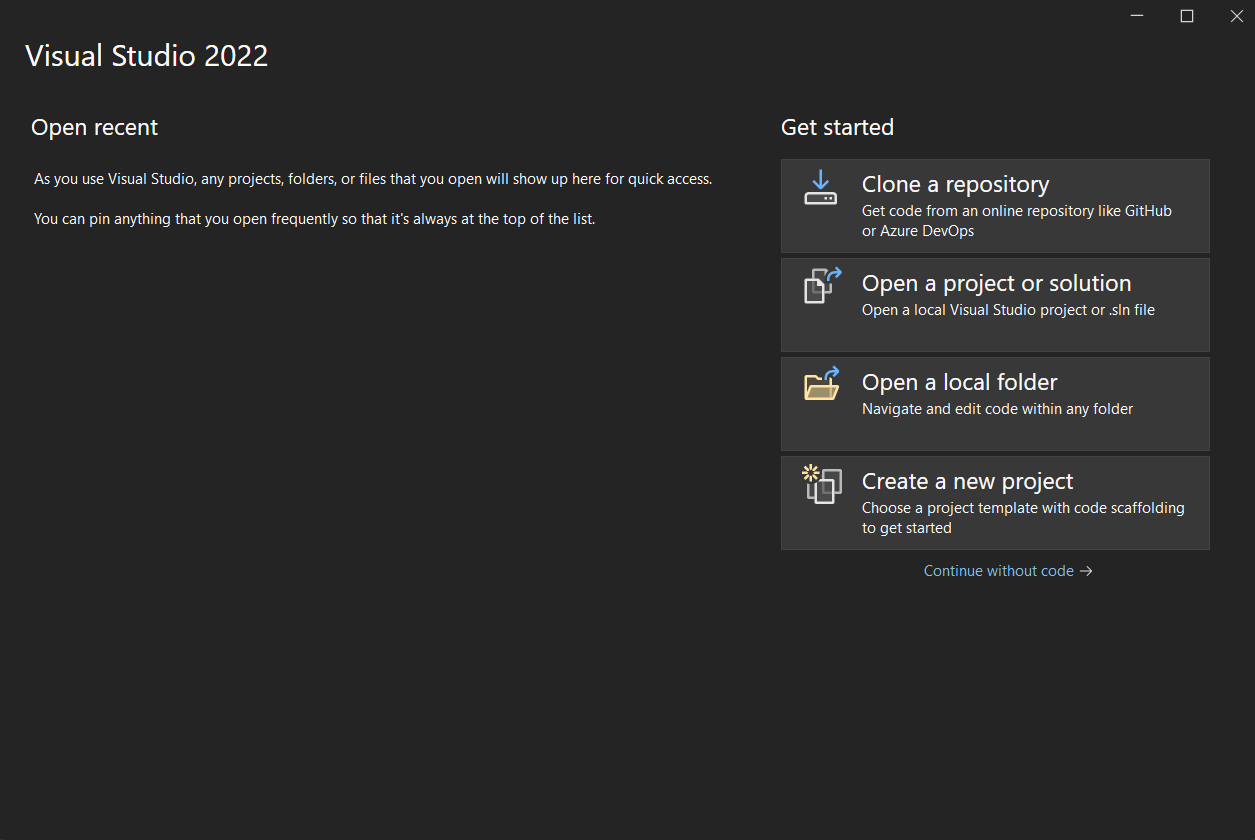
[Bước 5: Test case 2 - PT có nghiệm kép 64](#_Toc165553921)

[Bước 6: Test case 3 – PT vô nghiệm 65](#_Toc165553922)

1. **Tạo Một Project Unit Test**

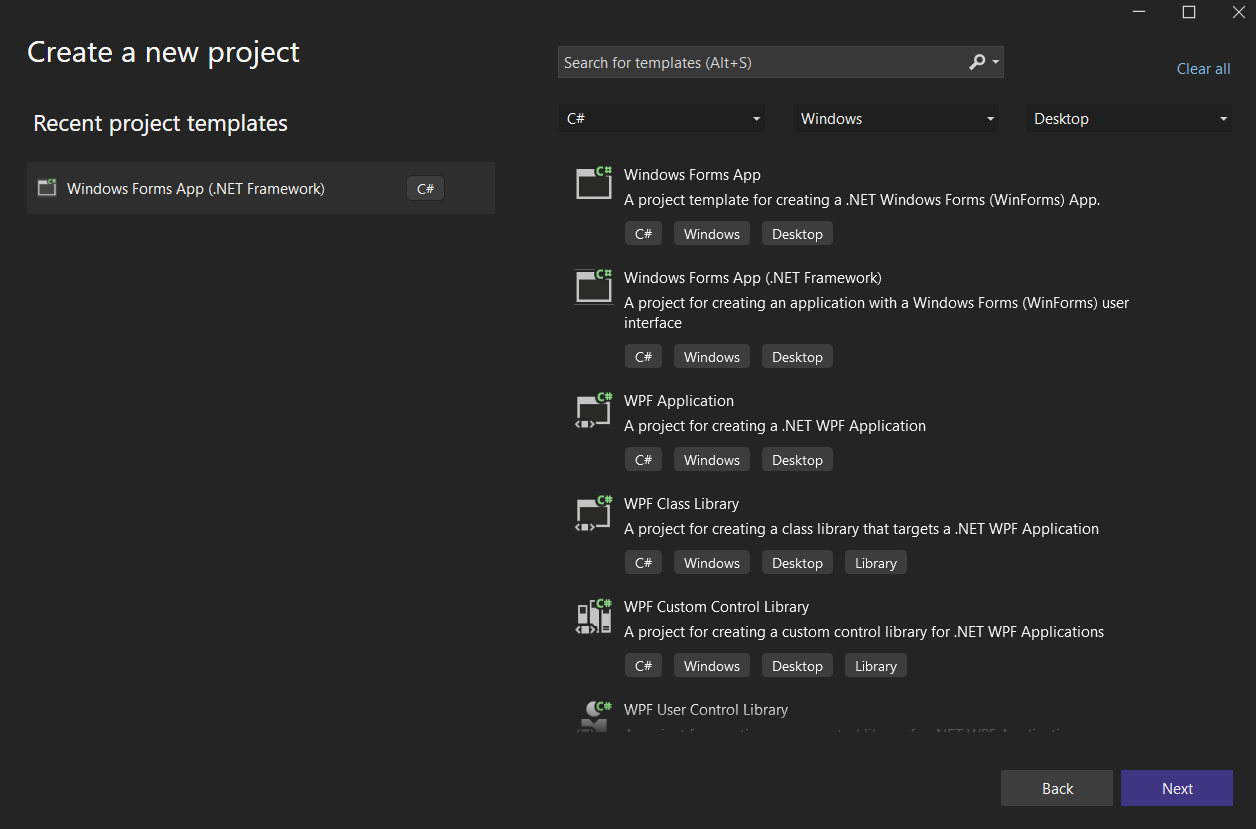
Bước 1:

Chọn vào create a new project



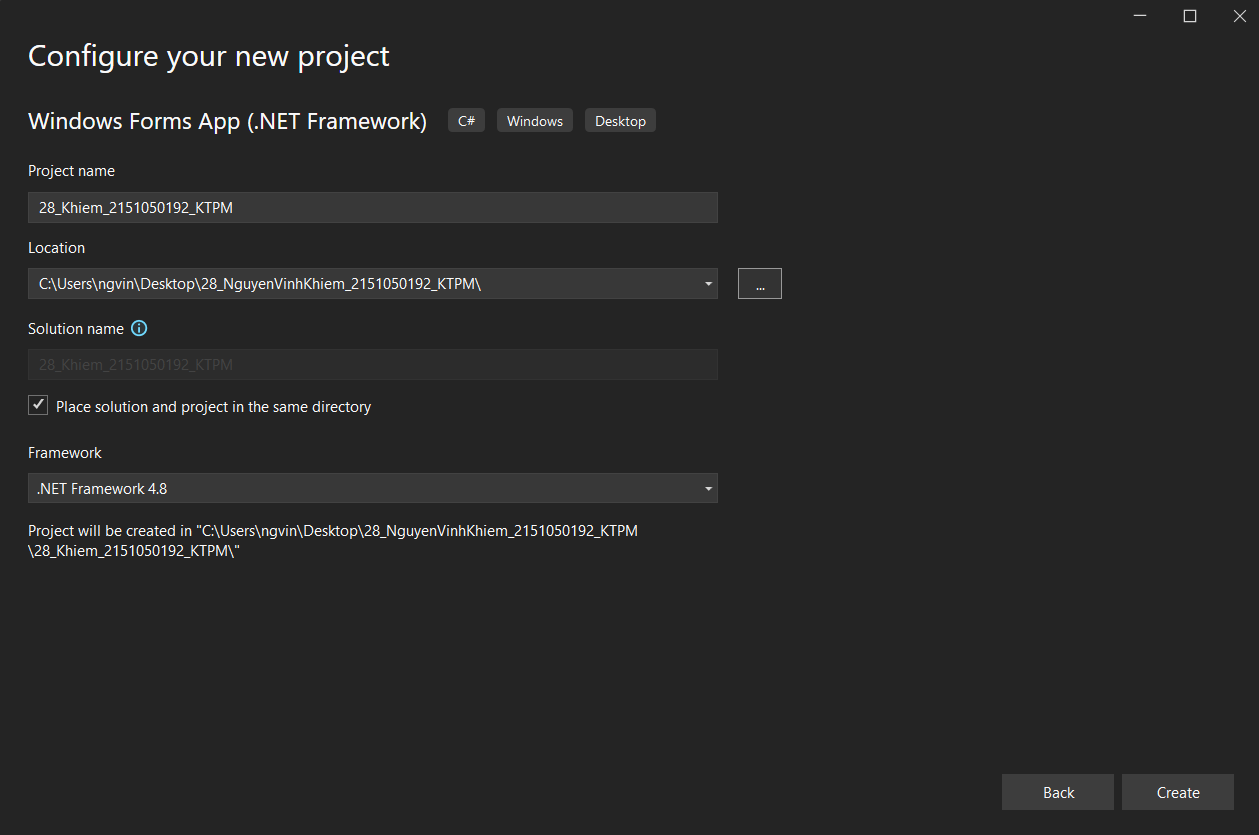
Bước 2:

Chọn vào lần lượt vào C# - Windows – Desktop -> Chọn project Unit TestProject (.Net Framework) và ấn next



Bước 3:

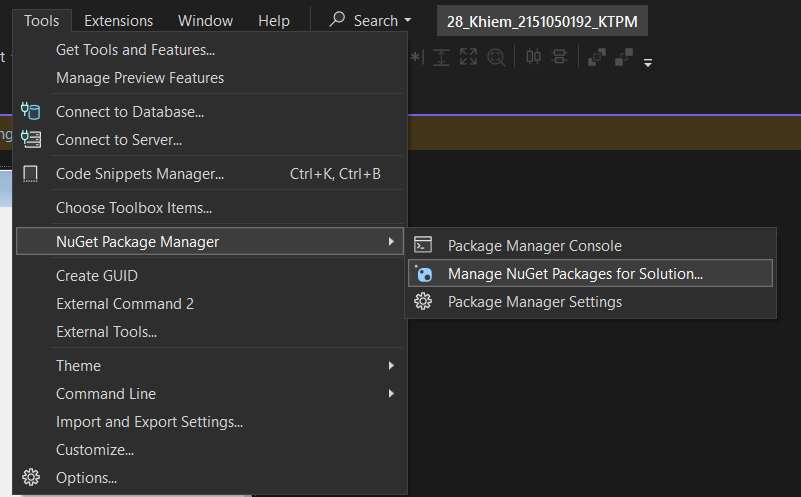
Đặt tên project và lưu địa chỉ muốn lưu và sau đó ấn create



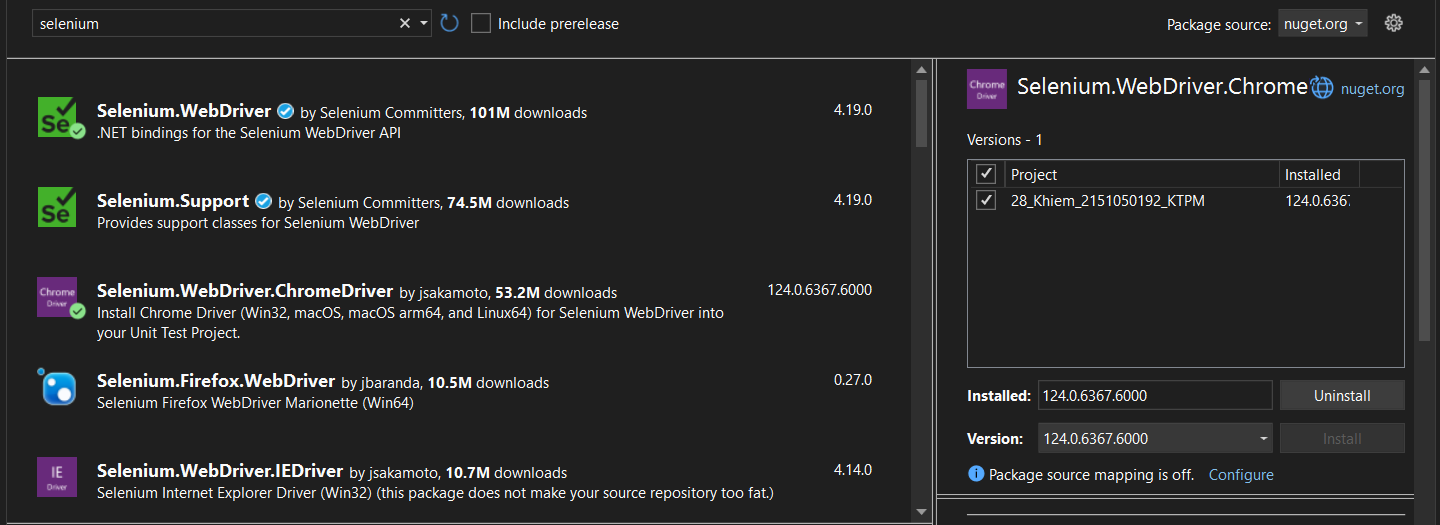
Bước 4:

Thêm thư viện selenium vào project

* Ấn vào tool 🡪 chọn NuGet Package Manager 🡪 Chọn NuGet Package Manager for Solution



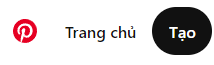
Tìm từ khóa “Selenium” -> Chọn 2 chức năng như hình -> Chọn Project -> Install

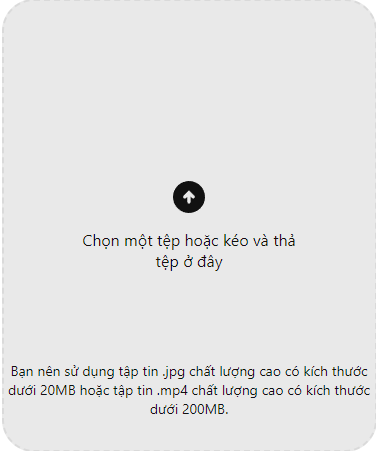


1. **Kiểm thử chức năng “Tạo” hình ảnh hoặc video**
   1. Mô tả chức năng

Chức năng kiểm tra hình hoặc video có thỏa điều kiện đăng lên Pinterest hay không.

Điều kiện: Sử dụng tập tin .jpg chất lượng cao có kích thước dưới 20MB hoặc tập tin .mp4 chất lượng cao có kích thước dưới 200MB.





* 1. Kiểm thử hộp đen
     1. Decision Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Điều kiện | TC1 | | TC2 | TC3 | TC4 | TC5 |
| Loại file hình | T | - | F | T | - | - |
| Dung lượng hình | T | - | - | F | - | - |
| Loại file video | - | T | - | - | F | T |
| Dung lượng video | - | T | - | - | - | F |
| Hành động | Cho đăng bài | Cho đăng bài | Ảnh định dạng sai | Ảnh lớn hơn 20MB | Video sai định dạng | Video lớn hơn 200MB |
| Kết quả | Pass | Pass | Pass | Pass | Pass | Pass |

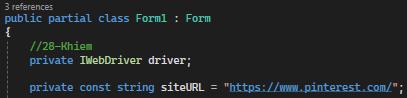
* + 1. Giải thích code Test

Bước 1: Khởi tạo driver và gán url web pinterest

//28-Khiem

private IWebDriver driver;

private const string siteURL = "https://www.pinterest.com/";



Bước 2: Tạo hàm dẫn đến url pinterest và click vào chức năng “Tạo”

private void Open\_Tao\_28\_Khiem()

{

//28-Khiem

driver.Navigate().GoToUrl(siteURL);

//Nút "Tạo"

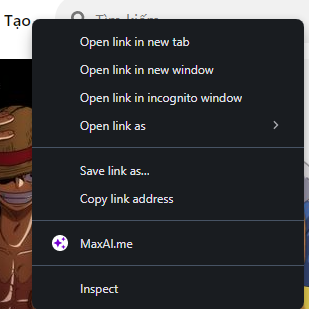
driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[1]/div[2]/div/div/div[2]/div/div/div/div[3]")).Click();

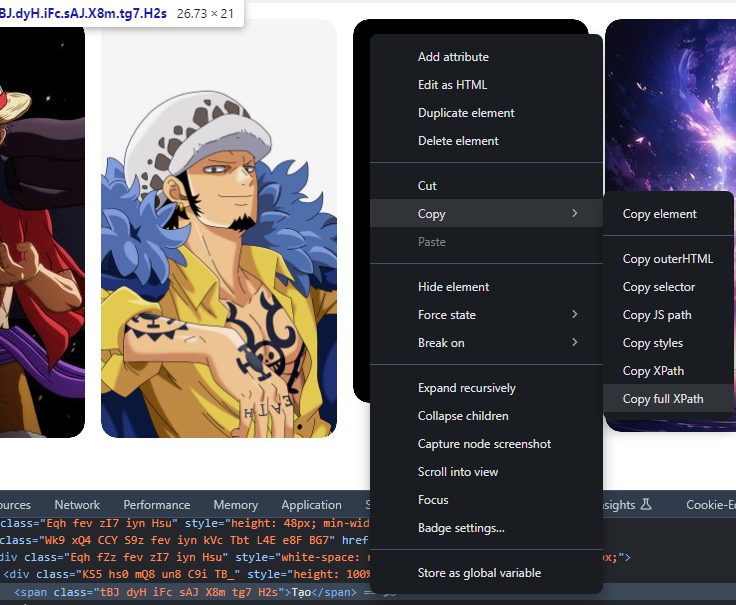
Thread.Sleep(5000);

}



\*Cách lấy đường dẫn XPath:





Bước 3: Tạo phương thức chromeconsole gán đường dẫn đến tài khoản Chrome mong muốn (ở đây là Default)

private void chromeconsole()

{

//28-Khiem

var options = new ChromeOptions();

options.AddArgument("user-data- dir=C:\\Users\\ngvin\\AppData\\Local\\Google\\Chrome\\User Data\\");

options.AddArgument("profile-directory=Default"); //Chọn profile Default

options.AddArgument("--start-maximized");

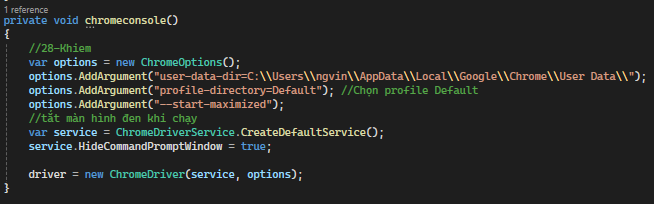
//tắt màn hình đen khi chạy

var service = ChromeDriverService.CreateDefaultService();

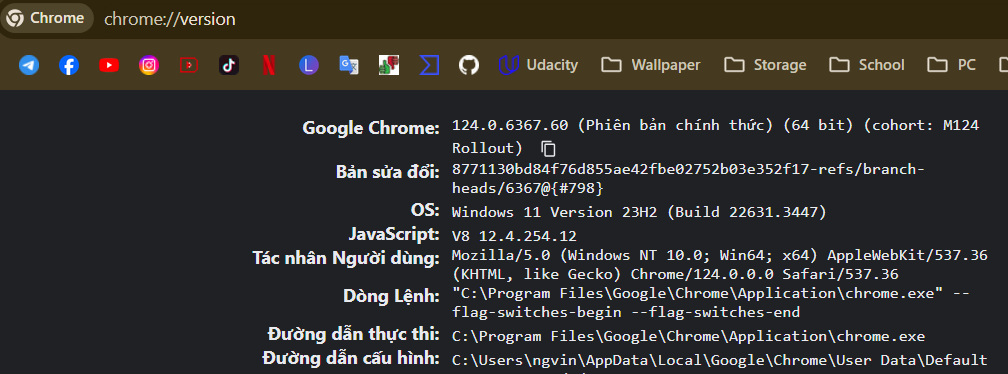
service.HideCommandPromptWindow = true;

driver = new ChromeDriver(service, options);

}



\*Cách lấy đường dẫn:



Bước 4: Gọi phương thức chromeconsole trong Form1

public Form1()

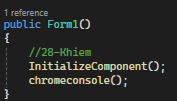
{

//28-Khiem

InitializeComponent();

chromeconsole();

}



Bước 5: Test case 1 - Đăng nhập thành công

private void TC1\_DangBai\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_Tao\_28\_Khiem();

driver.FindElement(By.CssSelector("input[type='file']")).SendKeys(@"C:\Users\ngvin\OneDrive\28\_NguyenVinhKhiem\_2151050192\_KTPM\28\_khiem\_file\_hinh\_video\anhcun.jpg");

Thread.Sleep(3000);

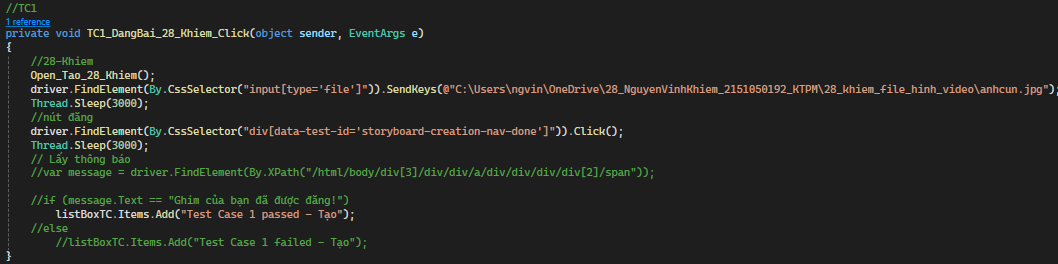
//nút đăng

driver.FindElement(By.CssSelector("div[data-test-id='storyboard-creation-nav-done']")).Click();

Thread.Sleep(3000);

listBoxTC.Items.Add("Test Case 1 passed - Tạo");

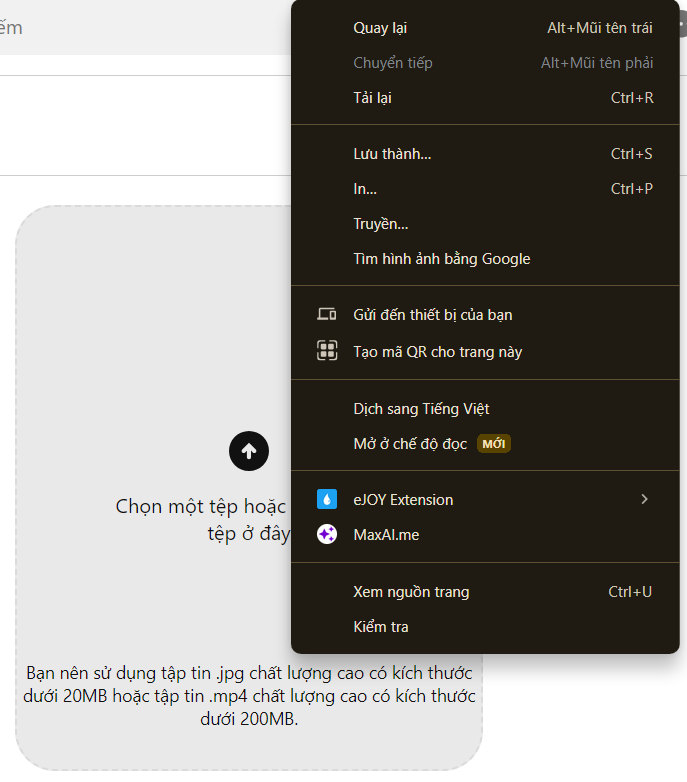
}



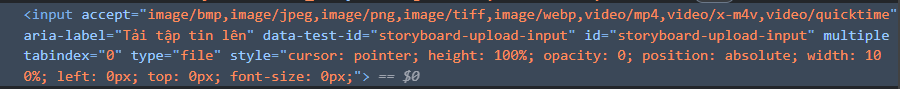
\*Chọn đường dẫn tới file ảnh đúng



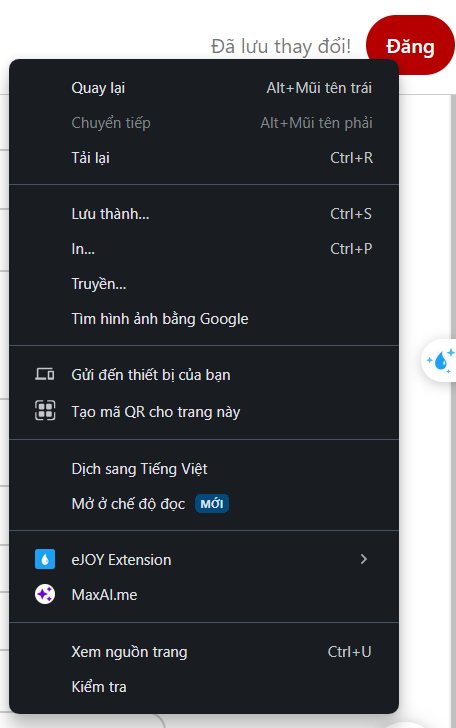
\*Cách lấy nút chọn hình:



Sử dụng phương thức CssSelector trong thẻ input chứa thuộc tính type=”file"



\*Cách lấy nút đăng bài:



Sử dụng phương thức CssSelector trong thẻ div có data-test-id=”storyboard-creation-nav-done"



Bước 6: Test case 2 – Sai định dạng ảnh

private void TC2\_SaiDinhDang\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_Tao\_28\_Khiem();

driver.FindElement(By.CssSelector("input[type='file']")).SendKeys(@"C:\Users\ngvin\OneDrive\28\_NguyenVinhKhiem\_2151050192\_KTPM\28\_khiem\_file\_hinh\_video\anhmeo.eps");

Thread.Sleep(3000);

// Lấy thông báo

var message = driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[2]/div/div/div[2]/div/div/div"));

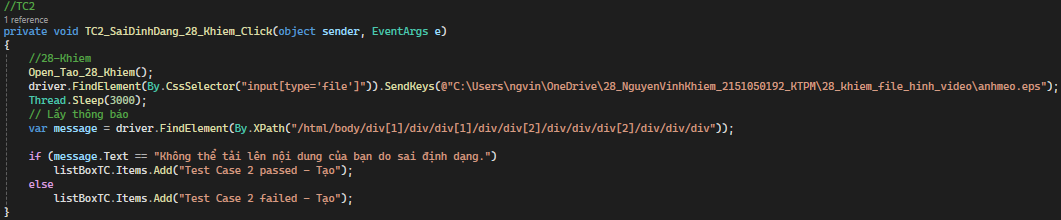
if (message.Text == "Không thể tải lên nội dung của bạn do sai định dạng.")

listBoxTC.Items.Add("Test Case 2 passed - Tạo");

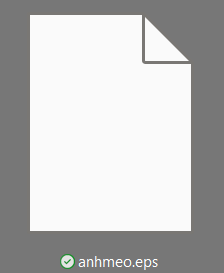
else

listBoxTC.Items.Add("Test Case 2 failed - Tạo");

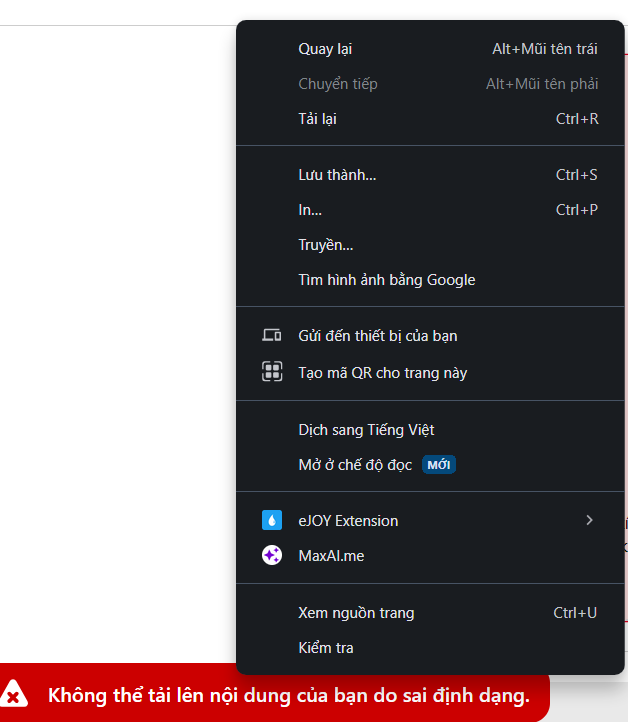
}

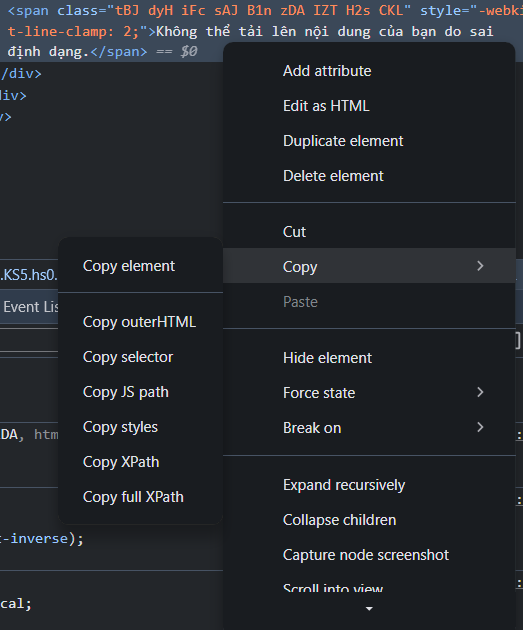


\*Chọn đường dẫn tới file ảnh sai định dạng



\*Cách lấy đường dẫn thông báo lỗi:





Bước 7: Test case 3 – Ảnh vượt dung lượng

private void TC3\_VuotDungLuong\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_Tao\_28\_Khiem();

driver.FindElement(By.CssSelector("input[type='file']")).SendKeys(@"C:\Users\ngvin\OneDrive\28\_NguyenVinhKhiem\_2151050192\_KTPM\28\_khiem\_file\_hinh\_video\vuotdungluong.png");

Thread.Sleep(3000);

// Lấy thông báo

var message = driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[2]/div/div/div[2]/div/div/div"));

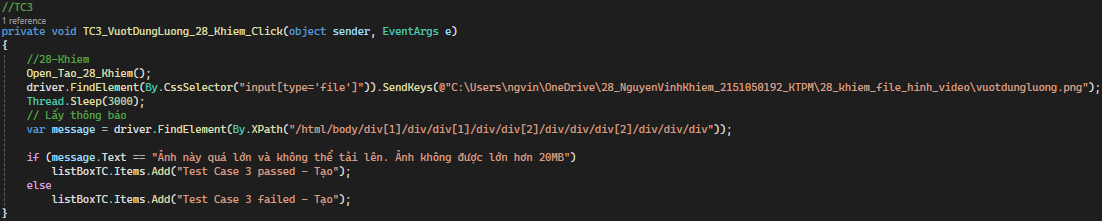
if (message.Text == "Ảnh này quá lớn và không thể tải lên. Ảnh không được lớn hơn 20MB")

listBoxTC.Items.Add("Test Case 3 passed - Tạo");

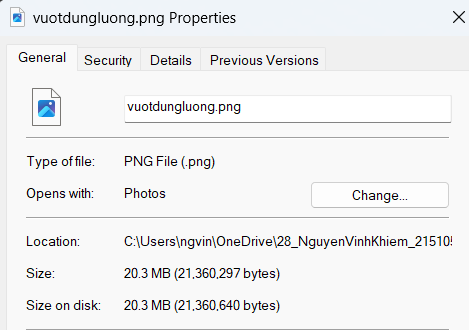
else

listBoxTC.Items.Add("Test Case 3 failed - Tạo");

}



\*Chọn đường dẫn tới file ảnh vượt dung lượng



Bước 8: Test case 4 – Video sai định dạng

private void TC4\_VidSaiDinhDang\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_Tao\_28\_Khiem();

driver.FindElement(By.CssSelector("input[type='file']")).SendKeys(@"C:\Users\ngvin\OneDrive\28\_NguyenVinhKhiem\_2151050192\_KTPM\28\_khiem\_file\_hinh\_video\videosai.vob");

Thread.Sleep(3000);

// Lấy thông báo

var message = driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[2]/div/div/div[2]/div/div/div"));

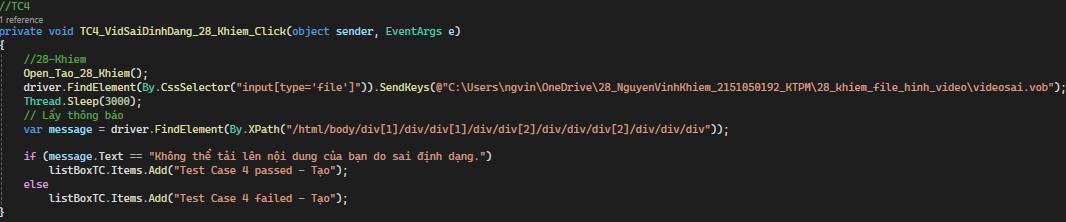
if (message.Text == "Không thể tải lên nội dung của bạn do sai định dạng.")

listBoxTC.Items.Add("Test Case 4 passed - Tạo");

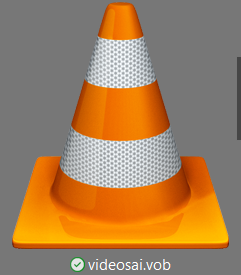
else

listBoxTC.Items.Add("Test Case 4 failed - Tạo");

}



\*Chọn đường dẫn tới file video sai định dạng



Bước 9: Test case 5 – Video vượt dung lượng

private void TC5\_VidVuotDungLuong\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_Tao\_28\_Khiem(); driver.FindElement(By.CssSelector("input[type='file']")).SendKeys(@"C:\Users\ngvin\OneDrive\28\_NguyenVinhKhiem\_2151050192\_KTPM\28\_khiem\_file\_hinh\_video\outsize.mp4");

Thread.Sleep(3000);

// Lấy thông báo

var message = driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[2]/div/div/div[2]/div/div/div"));

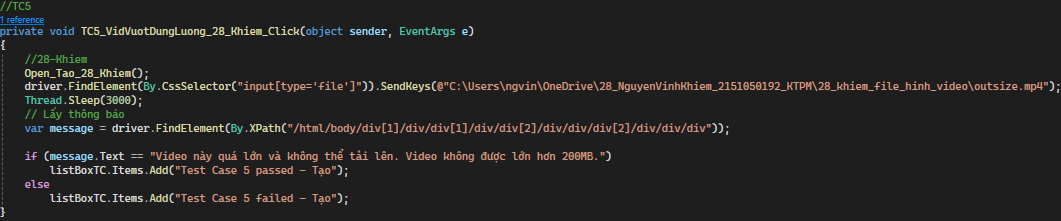
if (message.Text == "Video này quá lớn và không thể tải lên. Video không được lớn hơn 200MB.")

listBoxTC.Items.Add("Test Case 5 passed - Tạo");

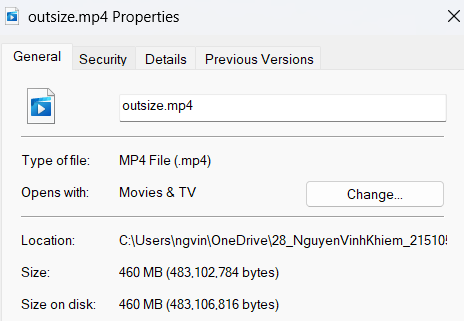
else

listBoxTC.Items.Add("Test Case 5 failed - Tạo");

}



\*Chọn đường dẫn tới file video vượt dung lượng

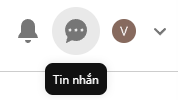


1. **Kiểm thử chức năng nhắn tin**
   1. Mô tả chức năng

Chức năng kiểm tra người nhận hay nội dung tin nhắn có gửi được không.

Điều kiện:

* Phải có người nhận hợp lệ
* Nội dung không quá 1000 kí tự



* 1. Kiểm thử hộp đen
     1. Decision Table

|  |  |  |  |
| --- | --- | --- | --- |
| Điều kiện | TC1 | TC2 | TC3 |
| Chọn người dùng | F | T | T |
| Nội dung hợp lệ (<1000 kí tự) | - | - | F |
| Hành động | Không gửi được vì bị chặn | Gửi thành công | Không gửi được vì quá 1000 kí tự |
| Kết quả | Pass | Pass | Pass |

* + 1. Giải thích code Test

Buớc 1: Tạo hàm truy cập vào web Pinterest chọn nút nhắn tin và ô tìm kiếm

private void Open\_NhanTin\_28\_Khiem()

{

//28-Khiem

driver.Navigate().GoToUrl(siteURL);

Thread.Sleep(5000);

driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[1]/div[2]/div/div/div[2]/div/div/div/div[5]/div[3]/div/div/div/div/div/div")).Click();

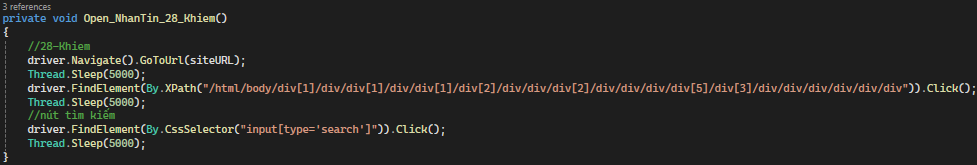
Thread.Sleep(5000);

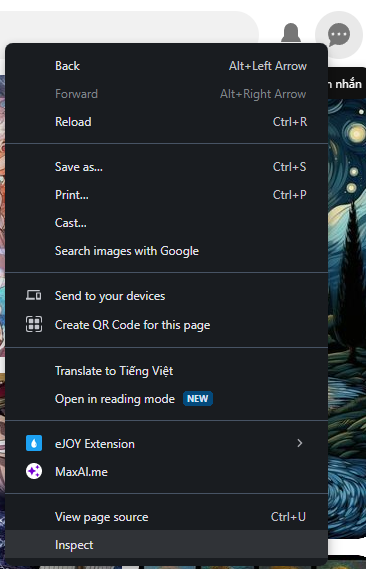
//nút tìm kiếm

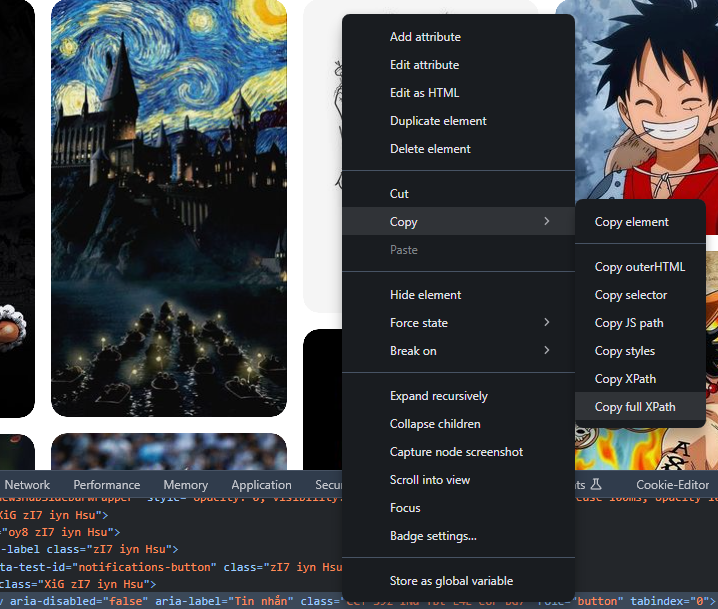
driver.FindElement(By.CssSelector("input[type='search']")).Click();

Thread.Sleep(5000);

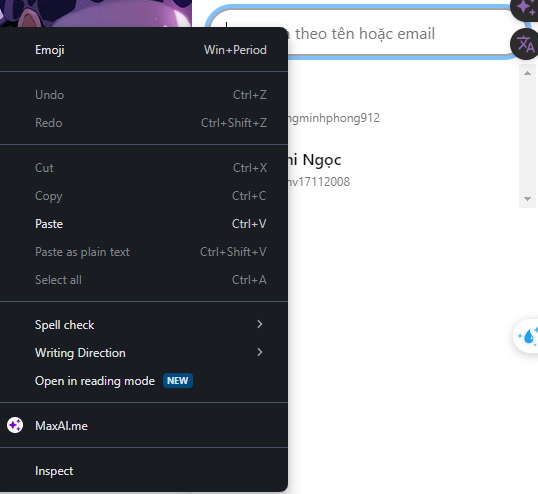
}

\*Cách lấy nút nhắn tin:





\*Cách lấy ô tìm kiếm:



Sử dụng phương thức CssSelector để lấy type=”search” trong thẻ input



Bước 2: Test case 1- Tin nhắn không thành công vì bị người dùng chặn

private void TC1\_TinNhanThatBai\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_NhanTin\_28\_Khiem();

var titleBox = driver.FindElement(By.CssSelector("input[id='contactSearch']"));

titleBox.SendKeys("nnnv17112008");

Thread.Sleep(500);

// chọn tài khoản đầu tiên được tìm thấy

driver.FindElement(By.CssSelector("div[class='Jea KS5 mQ8 wsz zI7 iyn Hsu']")).Click();

Thread.Sleep(5000);

//gửi trái tim để check hộp thoại - người dùng hợp lệ hay không driver.FindElement(By.CssSelector("button[aria-label='Thêm biểu tượng cảm xúc trái tim vào cuộc trò chuyện']")).Click();

Thread.Sleep(3000);

//nút cảnh báo

var message = driver.FindElement(By.CssSelector("div[class='KS5 hs0 un8 HOt BxF']"));

//28-Khiem

if (message.Text == "Không thể tạo hội thoại.")

listBoxTC.Items.Add("Test Case 1 passed - Nhắn tin");

else

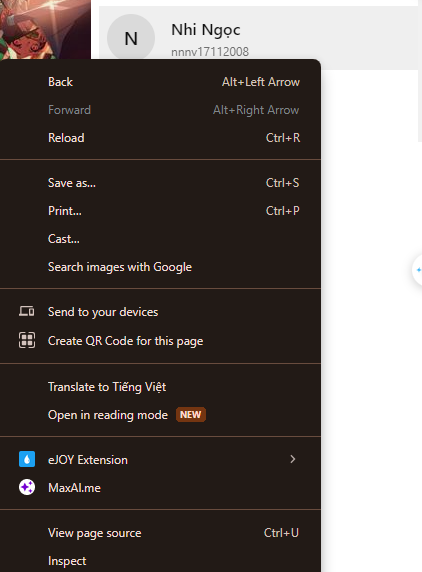
listBoxTC.Items.Add("Test Case 1 failed - Nhắn tin");

}



\*Tại ô input nhập vào SendKeys là tên hoặc gmail của người dùng, ở đây là ‘nnnv17112008’ – người dùng đã chặn

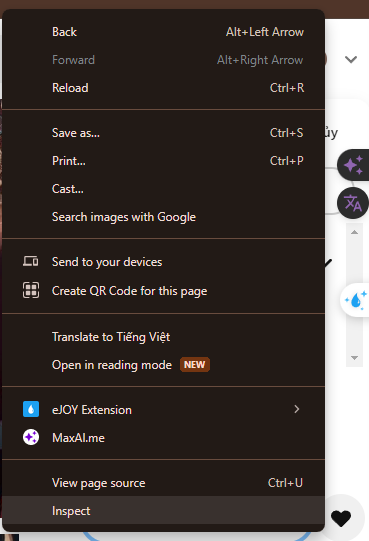
\*Cách lấy ô chọn tài khoản đầu tiên:



Sử dụng phương thức CssSelector để lấy class



\*Cách lấy nút trái tim để click:



Sử dụng phương thức CssSelector trong thẻ button để lấy aria-label=”Thêm biểu tượng cảm xúc trái tim vào cuộc trò chuyện”



Bước 3: Test case 2 – Gửi tin nhắn thành công

private void TC2\_TinNhanThanhCong\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_NhanTin\_28\_Khiem();

var titleBox = driver.FindElement(By.CssSelector("input[id='contactSearch']"));

titleBox.SendKeys("dangminhphong912");

Thread.Sleep(500);

// chọn tài khoản đầu tiên được tìm thấy

driver.FindElement(By.CssSelector("div[class='Jea KS5 mQ8 wsz zI7 iyn Hsu']")).Click();

Thread.Sleep(5000);

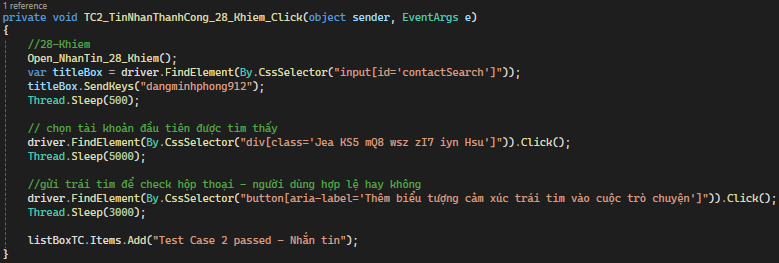
//gửi trái tim để check hộp thoại - người dùng hợp lệ hay không

driver.FindElement(By.CssSelector("button[aria-label='Thêm biểu tượng cảm xúc trái tim vào cuộc trò chuyện']")).Click();

Thread.Sleep(3000);

listBoxTC.Items.Add("Test Case 2 passed - Nhắn tin");

}



\* Tại ô input nhập vào SendKeys là tên hoặc gmail của người dùng, ở đây là ‘dangminhphong912’

\*Vì người dùng hợp lệ nên đã gửi được biểu tượng trái tim

Bước 4: Test case 3 - Tin nhắn không thành công vì nội dung vượt quá kí tự

private void TC3\_SaiNoiDung\_28\_Khiem\_Click(object sender, EventArgs e)

{

//28-Khiem

Open\_NhanTin\_28\_Khiem();

var titleBox = driver.FindElement(By.CssSelector("input[id='contactSearch']"));

titleBox.SendKeys("dangminhphong912");

Thread.Sleep(500);

// chọn tài khoản đầu tiên được tìm thấy

driver.FindElement(By.CssSelector("div[class='Jea KS5 mQ8 wsz zI7 iyn Hsu']")).Click();

Thread.Sleep(5000);

// nhấn vào ô gửi tin nhắn

driver.FindElement(By.CssSelector("div[class='Jea XiG zI7 iyn Hsu']")).Click();

Thread.Sleep(5000);

//nhập số lượng kí tự cho trước vượt quá 1000 từ

driver.FindElement(By.CssSelector("textarea[class='ExpandingTextArea\_\_textArea ExpandingTextArea\_\_enabled ExpandingTextArea\_\_grow ExpandingTextArea\_\_large']")).SendKeys("1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111");

Thread.Sleep(5000);

//check nút send

var postButton = driver.FindElement(By.XPath("/html/body/div[1]/div/div[1]/div/div[2]/div[2]/div/div/div[2]/div[2]/div/div/div/div[2]/button"));

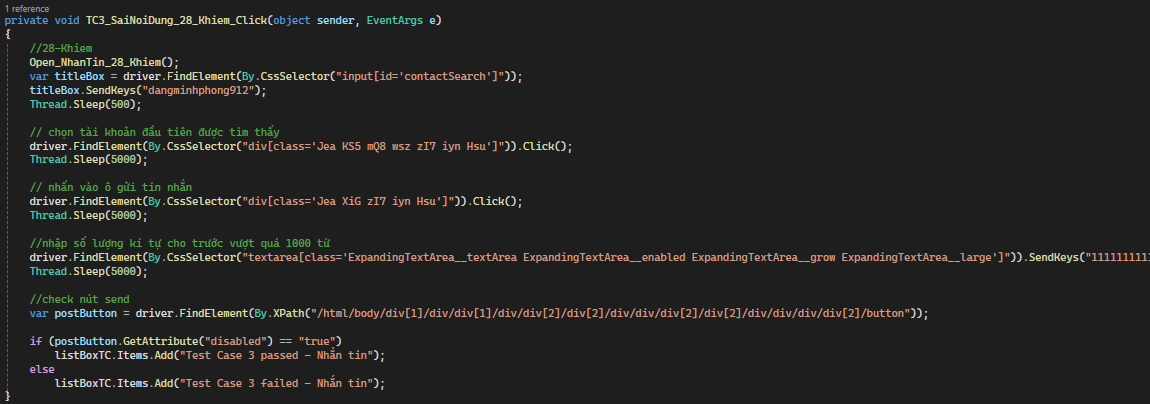
if (postButton.GetAttribute("disabled") == "true")

listBoxTC.Items.Add("Test Case 3 passed - Nhắn tin");

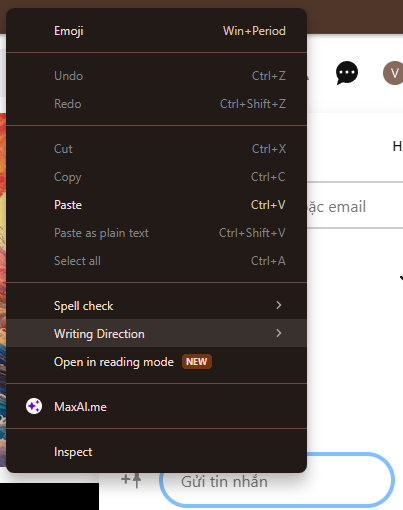
else

listBoxTC.Items.Add("Test Case 3 failed - Nhắn tin");

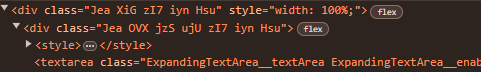
}



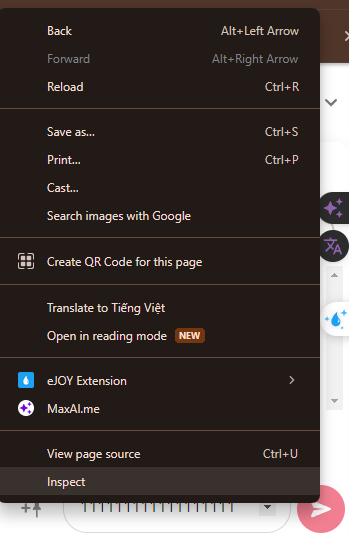
\*Cách lấy ô nhập tin nhắn:



Sử dụng phương thức CssSelector để lấy class



\*Cách lấy nút send bị disabled:



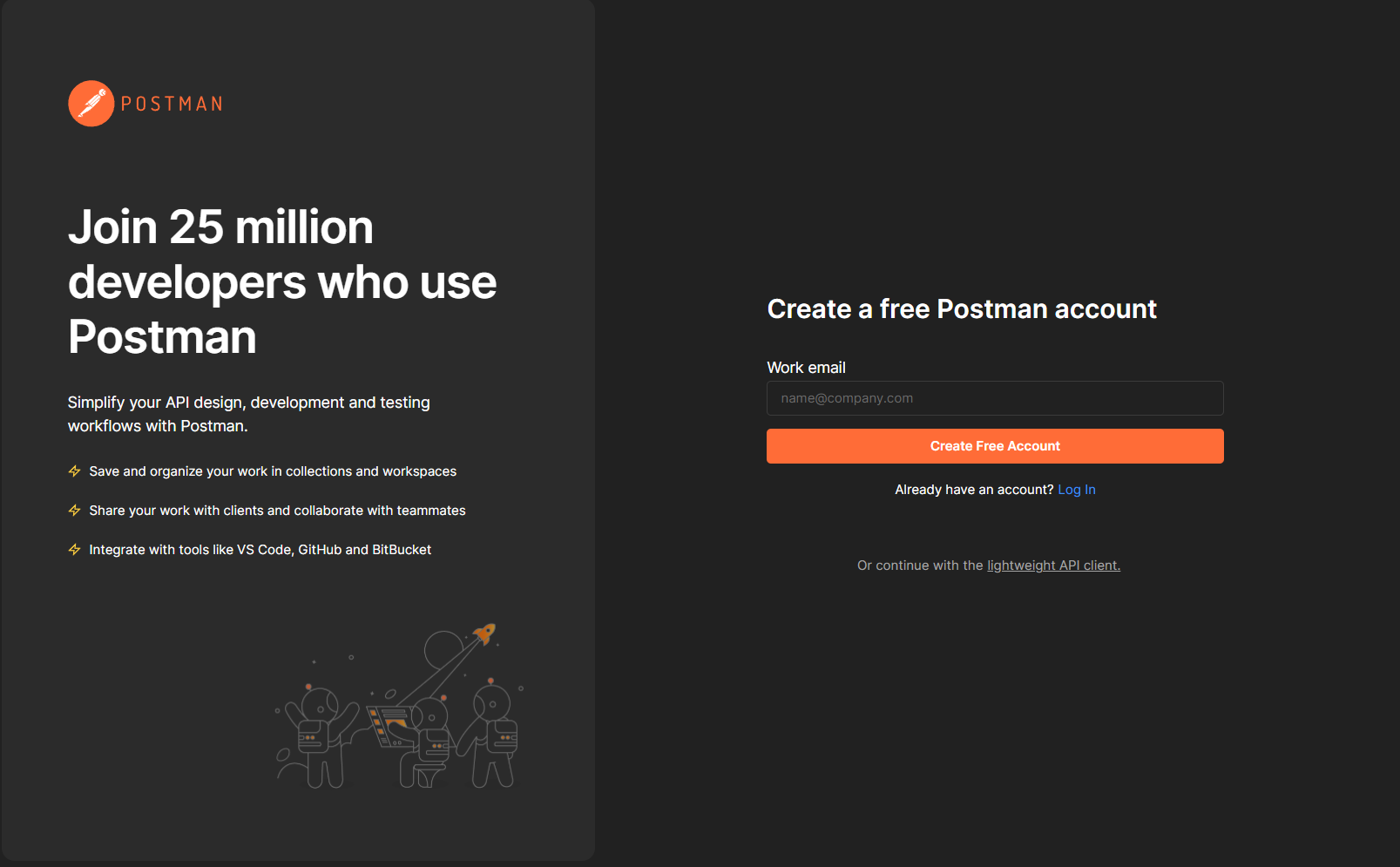
Sử dụng Xpath để lấy đường dẫn và sử dụng phương thức GetAttribute để lấy thuộc tính “disabled” để kiểm tra xem có không

->Nếu có thì test case pass

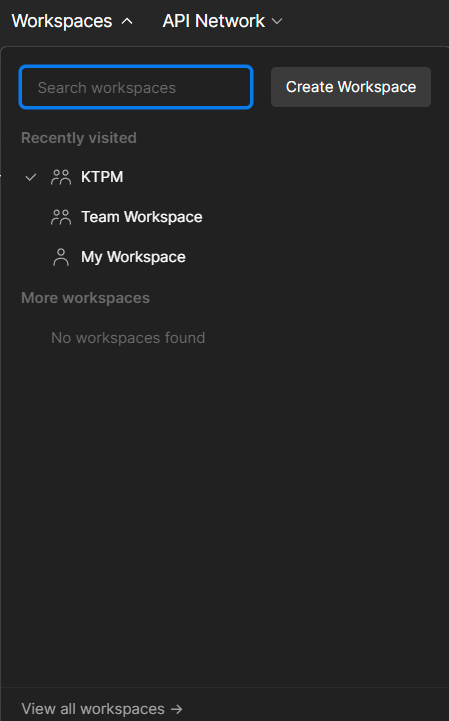


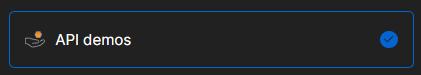
1. **Test API trên Postman**
   1. Lấy quyền trang web để test API
      1. Cài đặt Postman

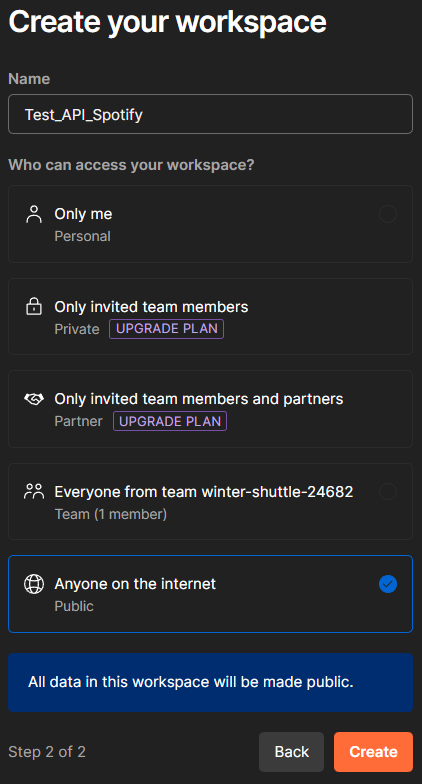
Bước 1: Tải Postman từ Google Chrome và tạo tài khoản để sử dụng



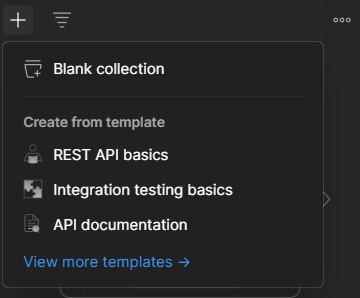
Bước 2: Tạo Workspace mới



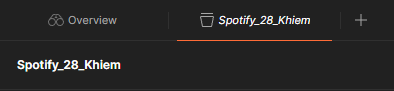




Bước 3: Tạo Collection

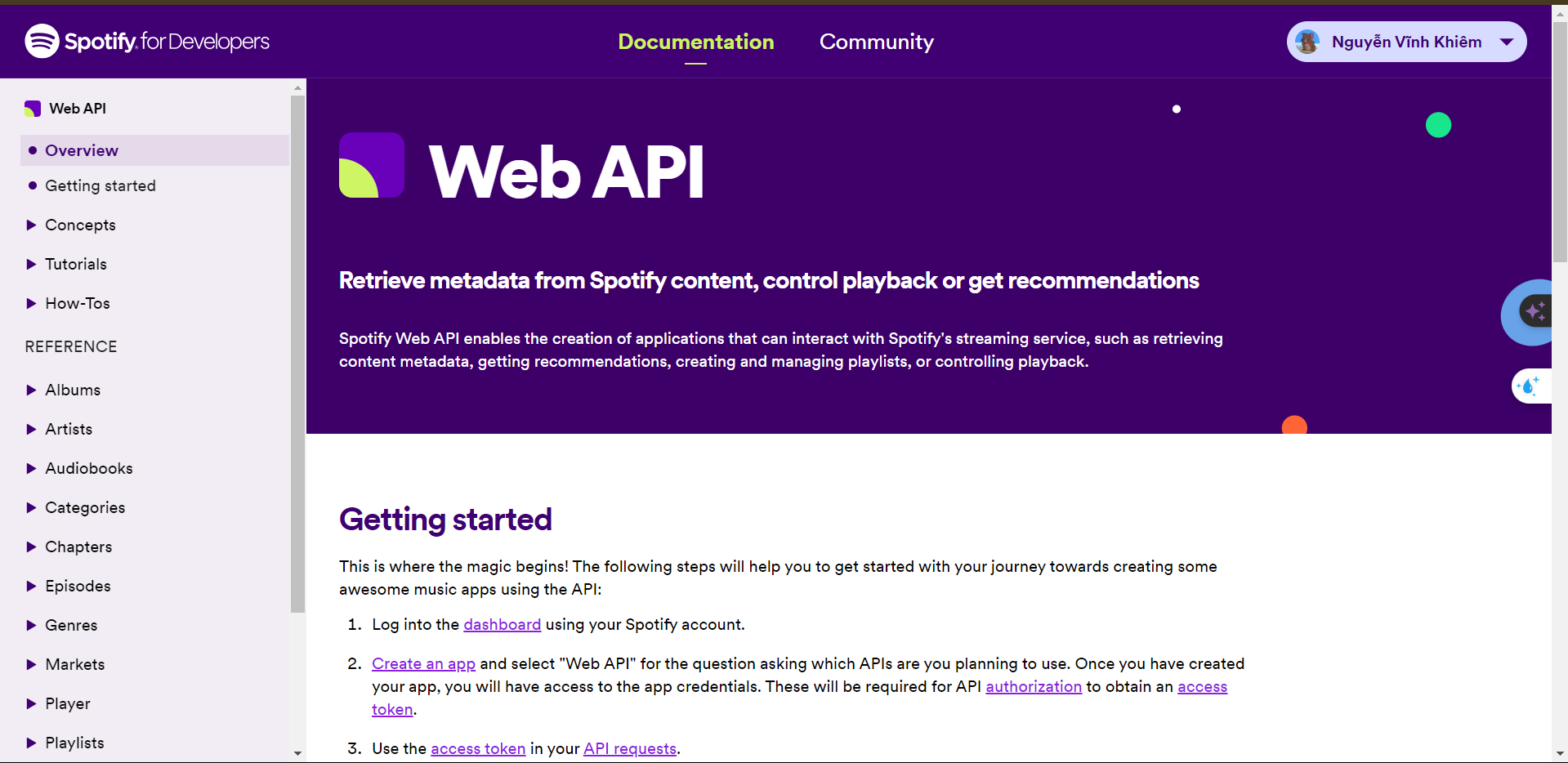


Bước 4: Đặt tên Collection và tạo phương thức test

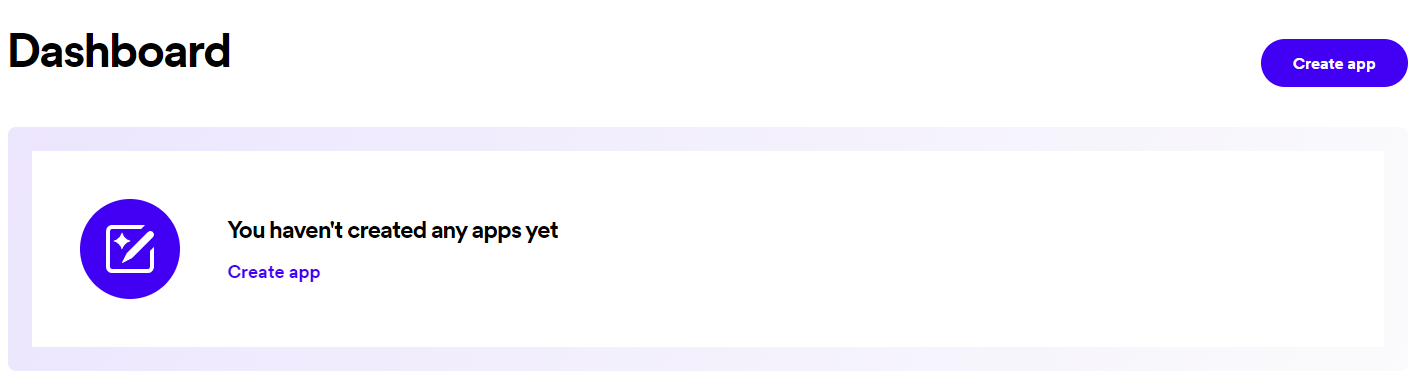


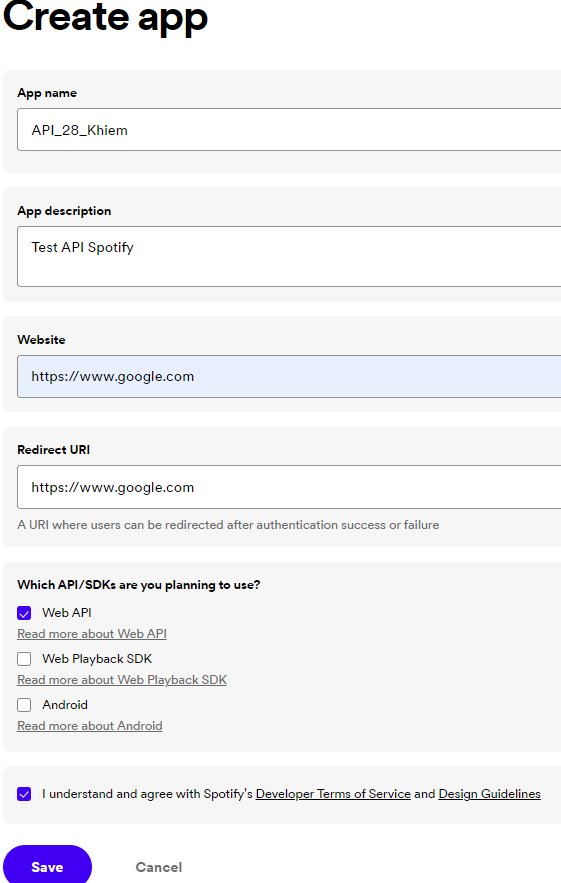
* + 1. Chứng thực và phân quyền trong Collection

Bước 1: Tìm từ khóa “Spotify API” ở Google Chrome -> Chọn trang “Spotify for Developers” -> Chọn “dashboard”

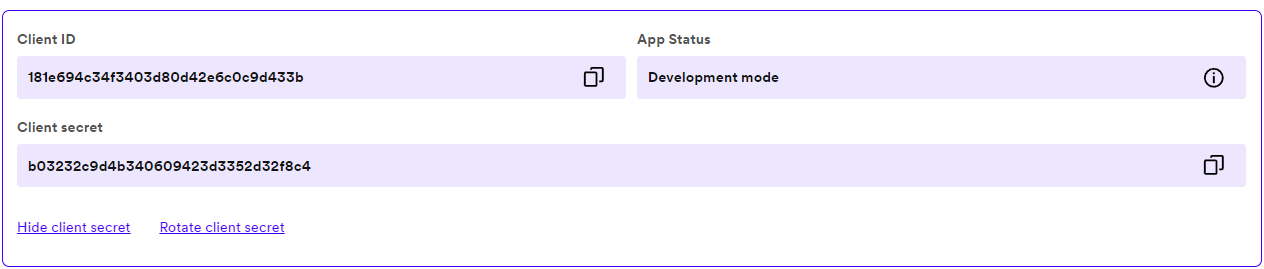


Bước 2: Click Create app và đặt tên như hình





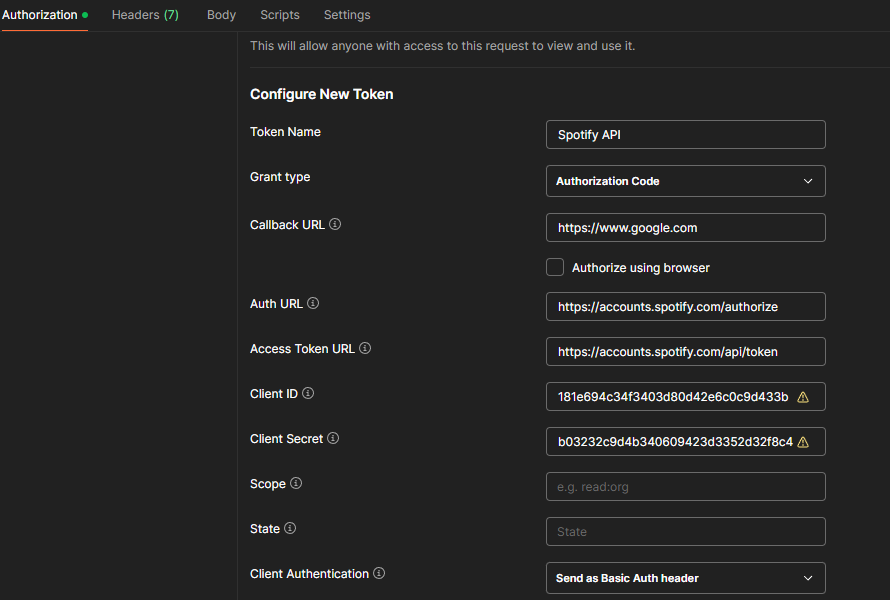
Bước 3: Ta sẽ được cung cấp Client ID VÀ Client Secret



Client ID: 181e694c34f3403d80d42e6c0c9d433b

Client secret: b03232c9d4b340609423d3352d32f8c4

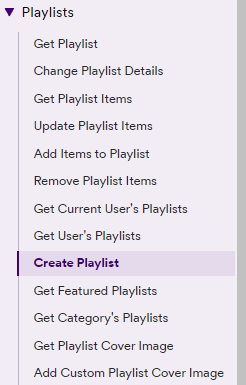
Bước 4: Copy Client ID và Client secret vào thẻ Authorization



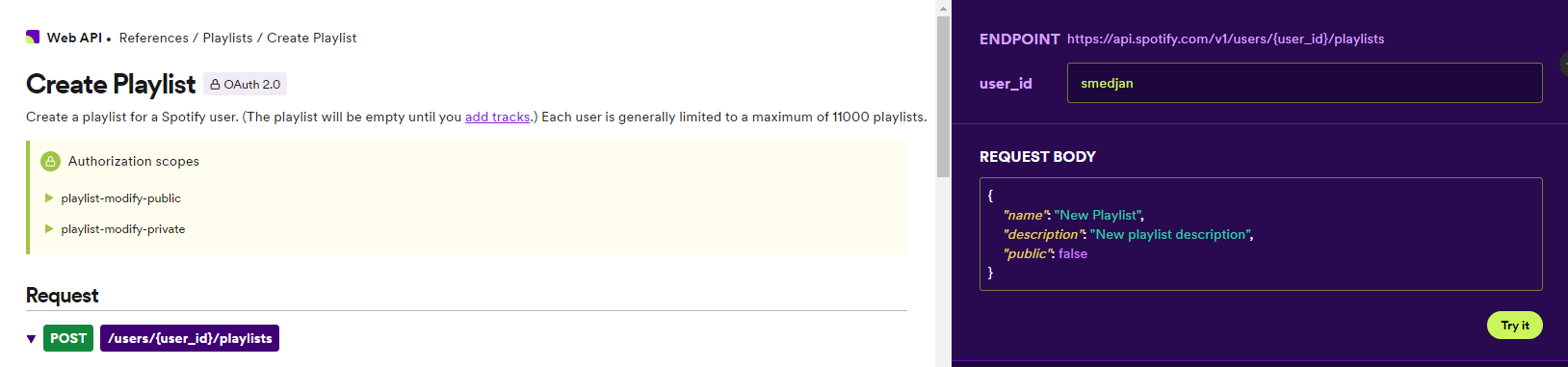
Điền thêm Scope nếu như phương thức cần dùng yêu cầu.

* 1. Thực hiện test API
     1. Phương thức POST – Tạo 1 playlist mới

Bước 1: Chọn mục Create Playlist trong trang Spotify for Developers



Bước 2: Copy ENDPOINT và thay id vào Postman

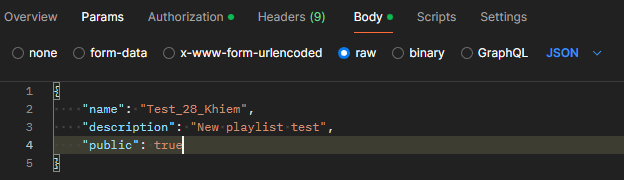




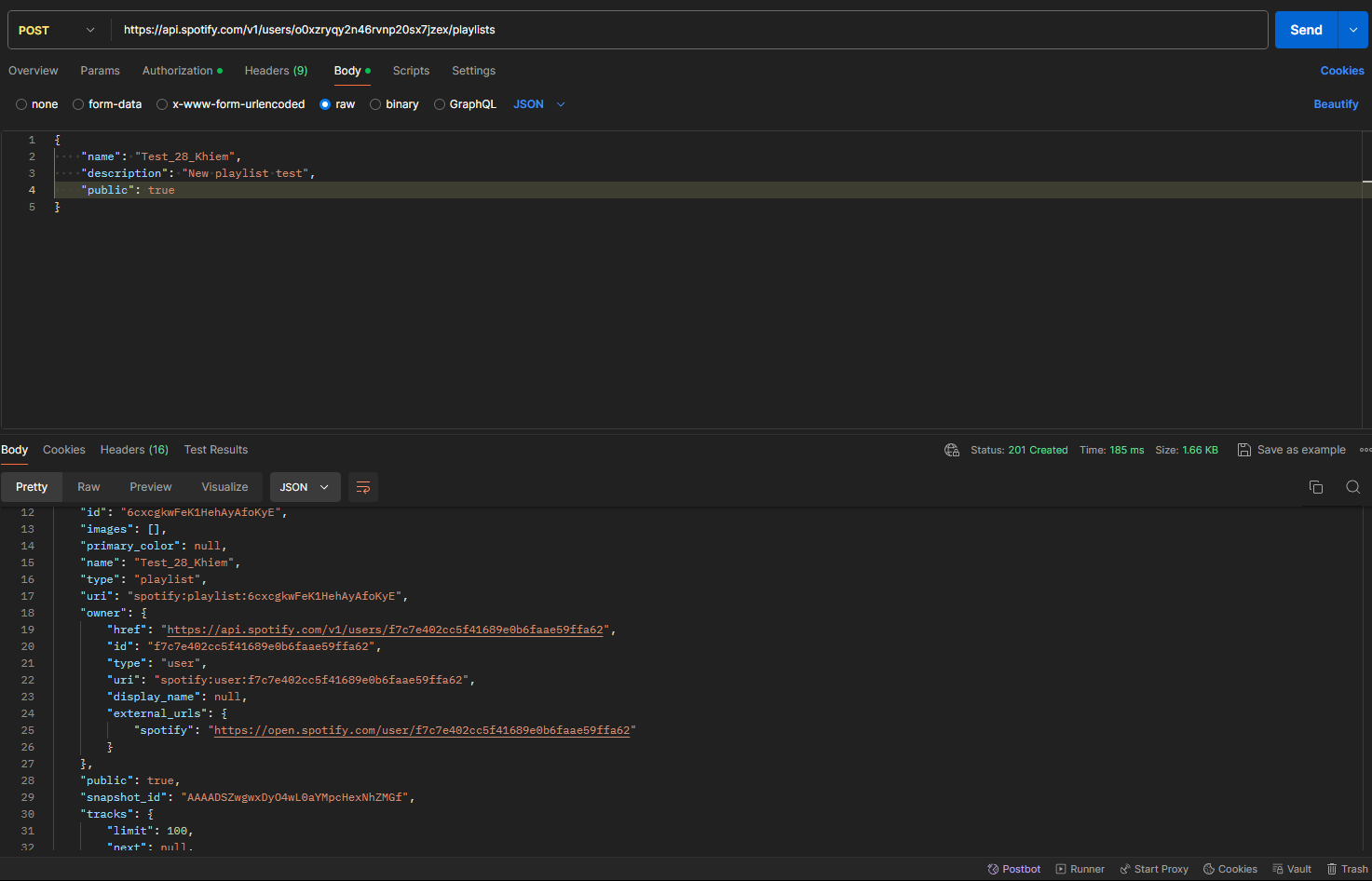
Bước 3: Vì phương thức có yêu cầu nên ta thêm Scopes vào Authorization



Bước 4: Vào body đặt tên, mô tả, quyền riêng tư Playlist theo ý muốn



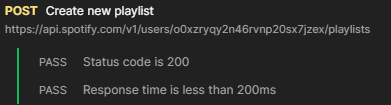
Bước 5: Ấn send và được kết quả



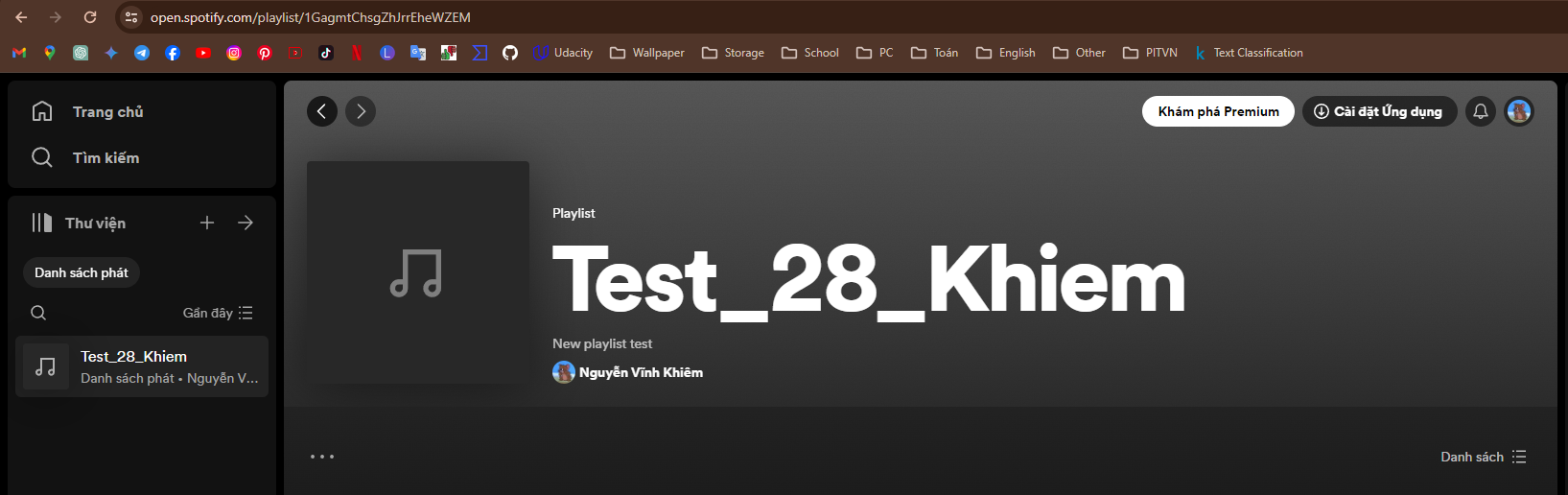
Bước 6: Viết Test case cho phương thức



Và nhận được kết quả

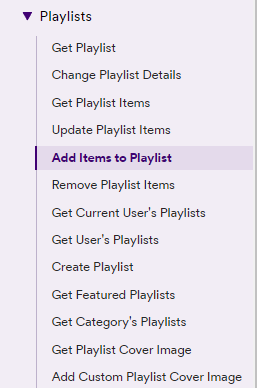


Bước 7: Xem kết quả thật ở Spotify

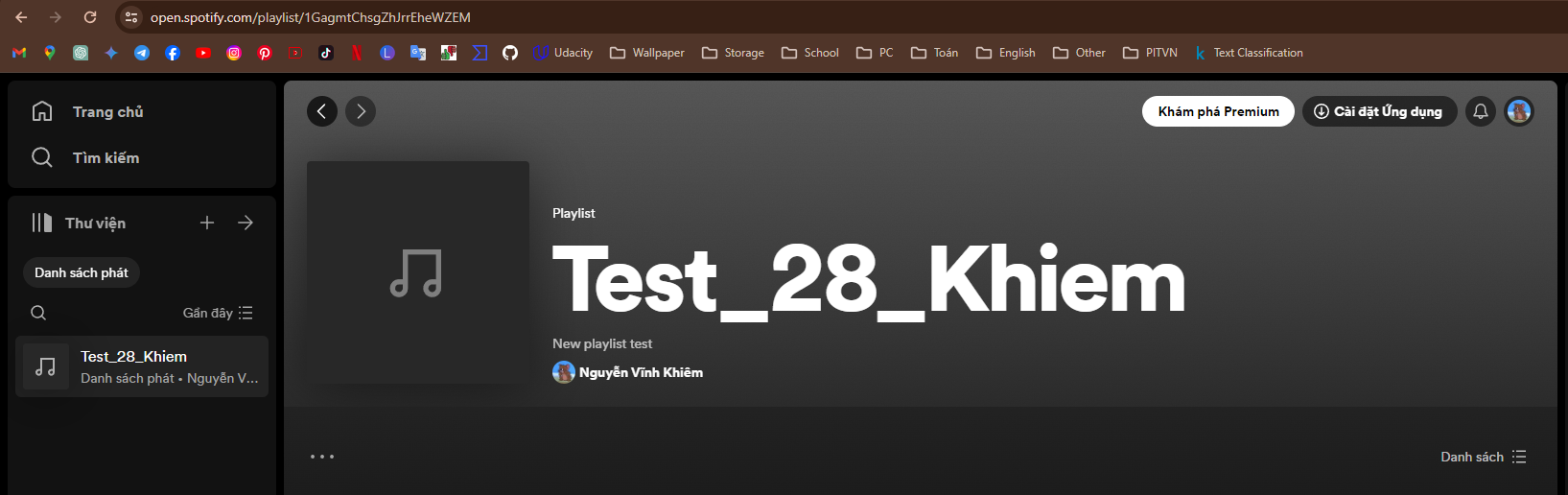


* + 1. Phương thức POST – Tạo 1 item trong Playlist vừa tạo

Bước 1: Chọn mục Add Items to Playlist trong trang Spotify for Developers



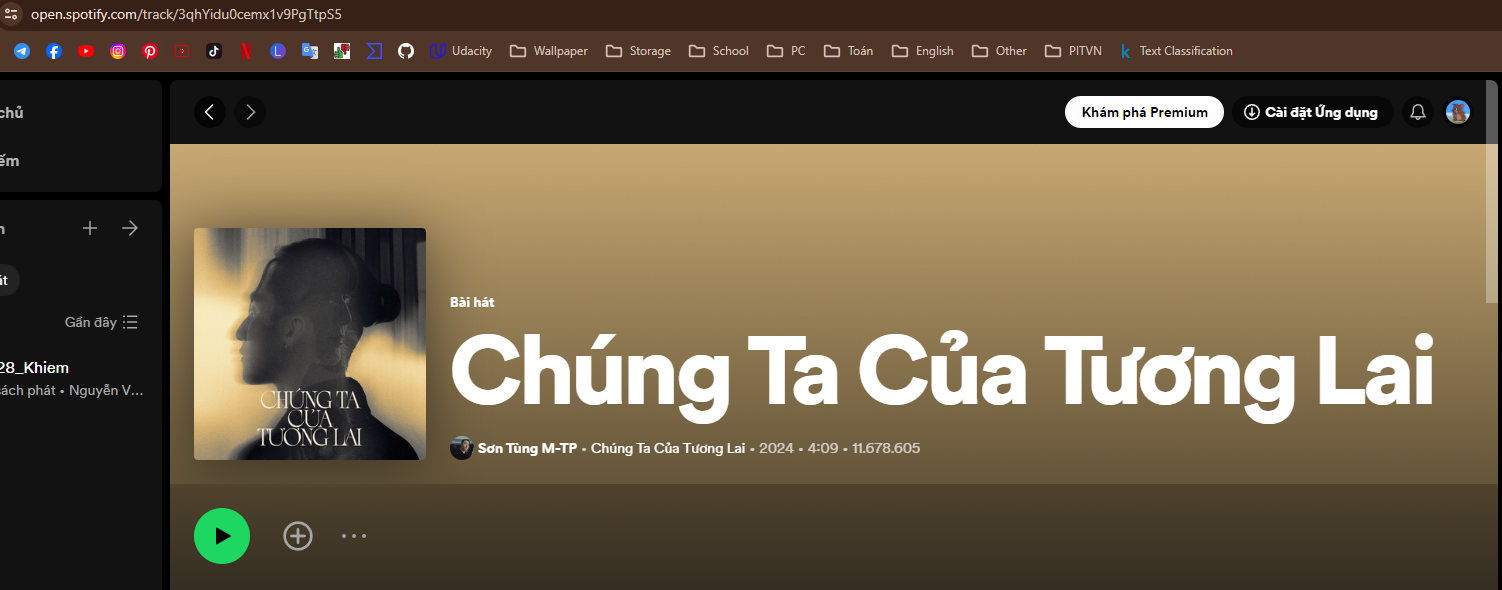
Bước 2: Copy ENDPOINT và thay id của playlist vừa tạo vào Postman

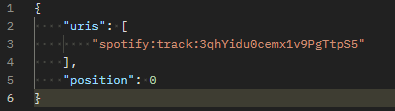




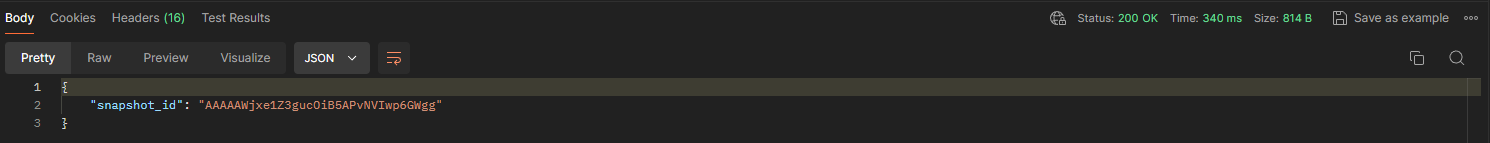


Bước 3: Vào body gắn id của item muốn thêm vào playlist

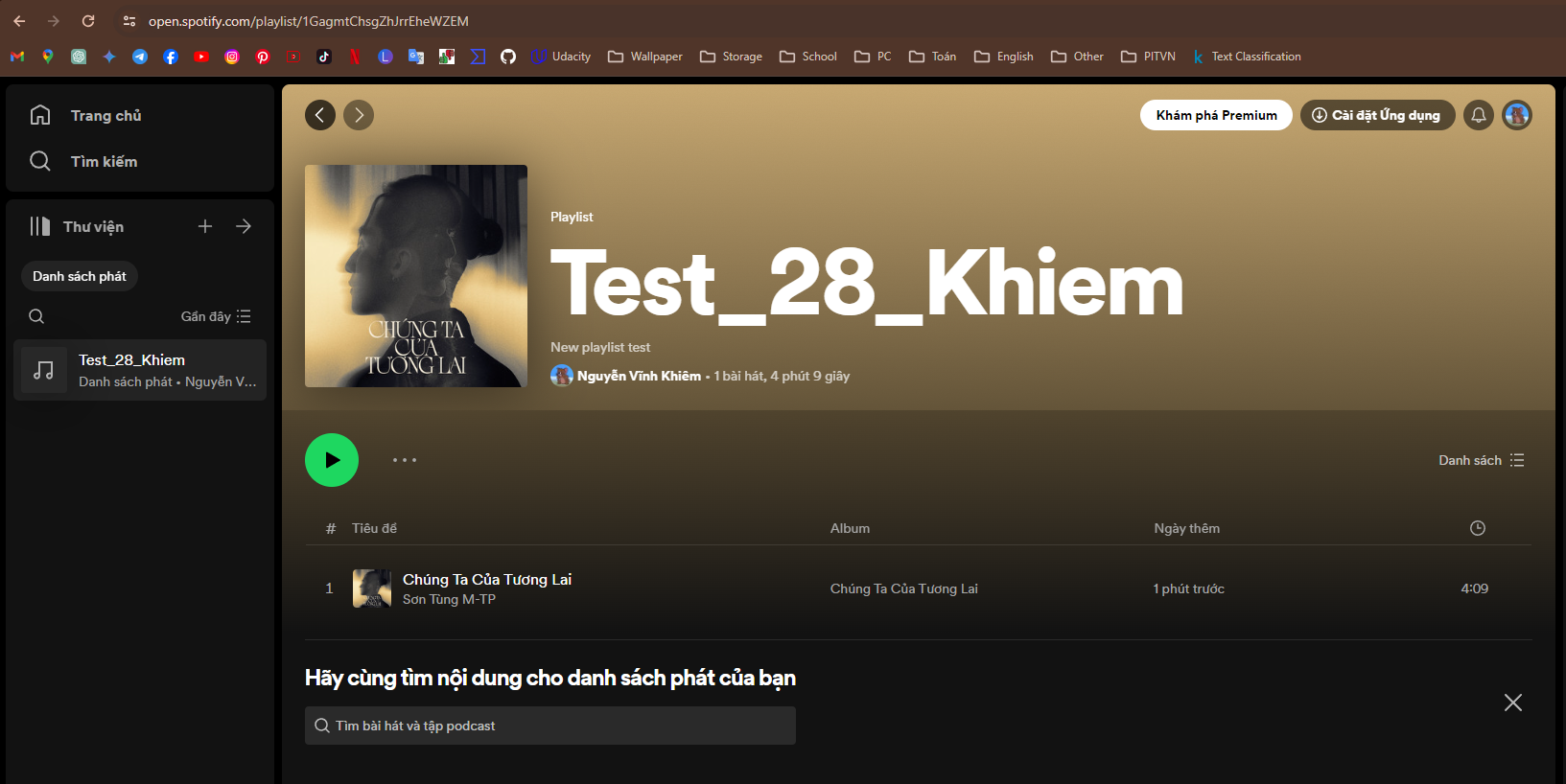




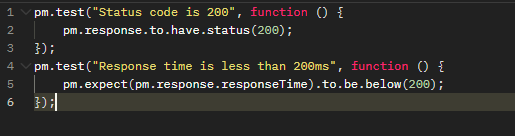
Bước 4: Xem thành quả



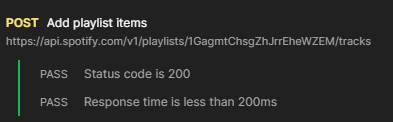
Postman sẽ trả về cho chúng ta snapshot\_id của Item để dùng cho các phương thức khác



Bước 5: Viết Test case cho phương thức

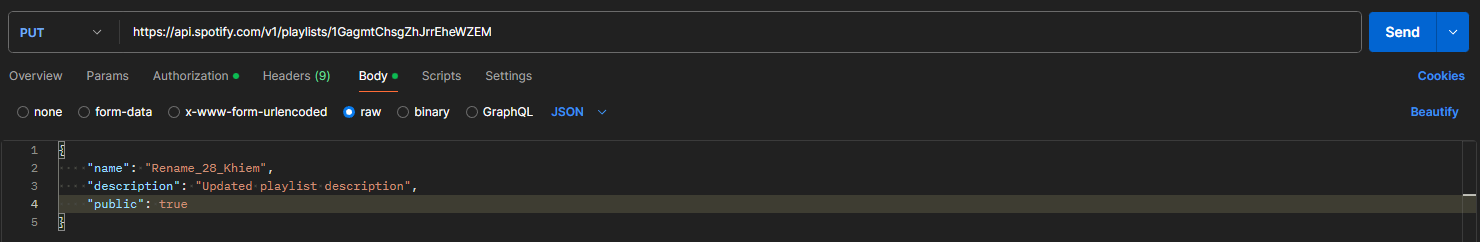


Và nhận được kết quả

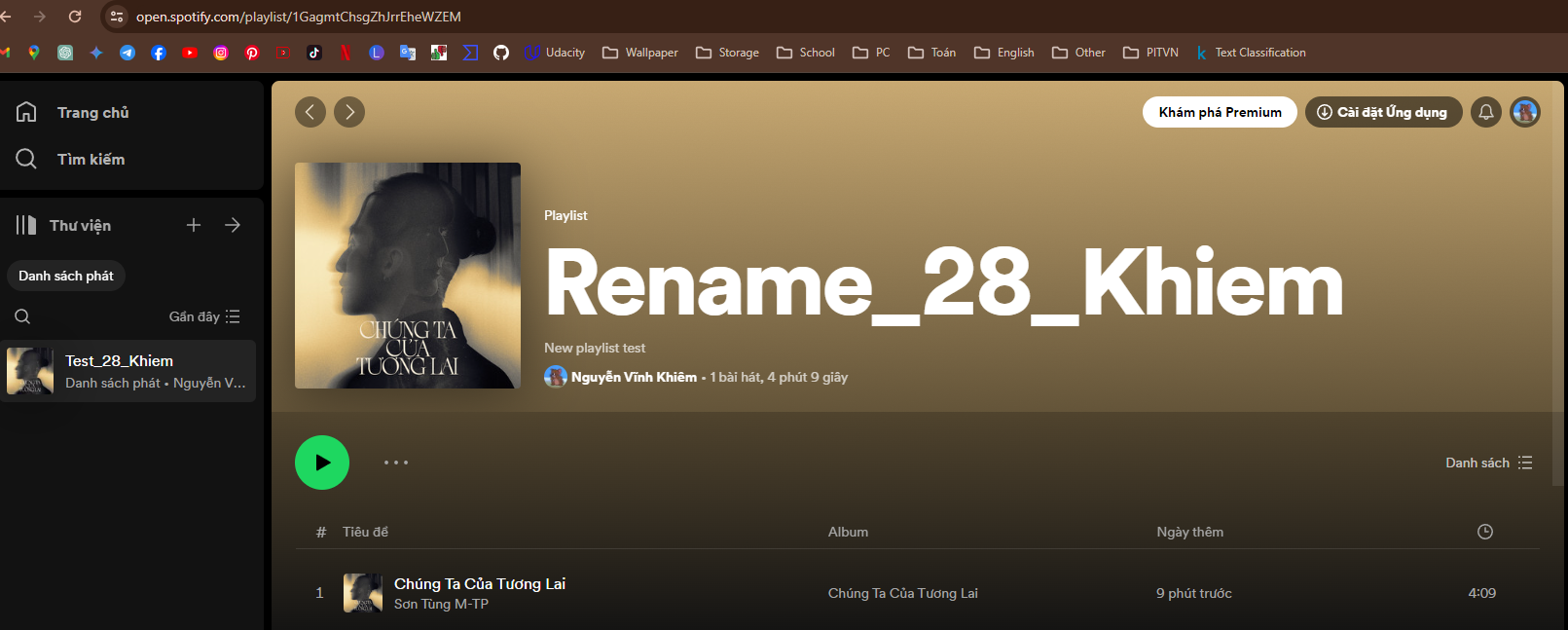


* + 1. Phương thức PUT – Đổi tên playlist

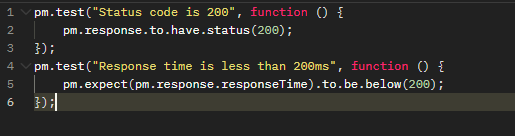
Bước 1, 2 và 3 làm tương tự như phương thức ở trên

Thay id playlist vào đường dẫn và đặt tên playlist mới

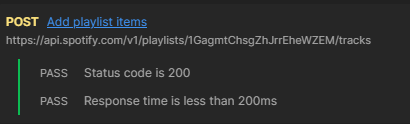
Bước 4: Xem thành quả



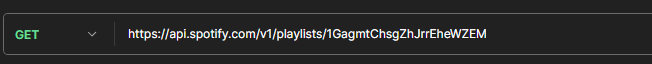
Bước 5: Viết Test case cho phương thức



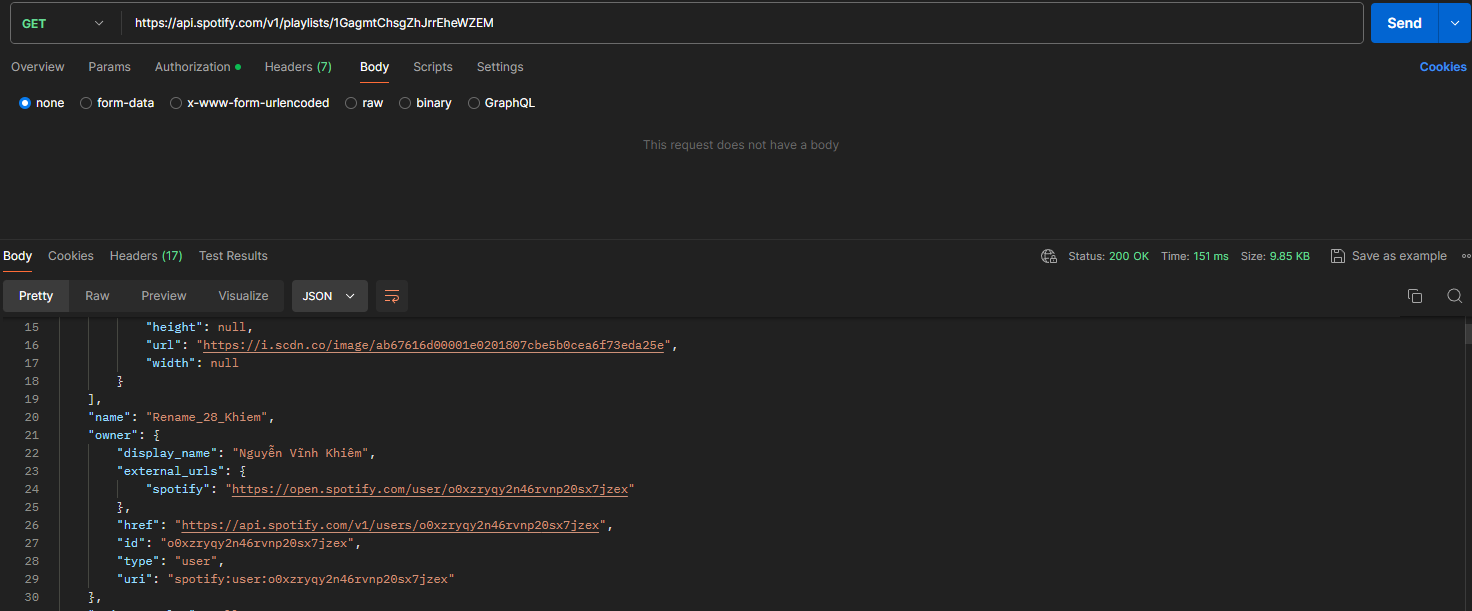
Và nhận được kết quả



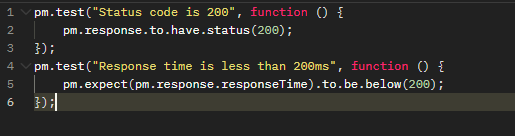
* + 1. Phương thức GET – Lấy playlist

Bước 1: Copy playlist vào đường dẫn

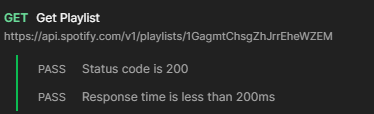
Bước 2: Nhận được kết quả



Bước 3: Viết Test case cho phương thức



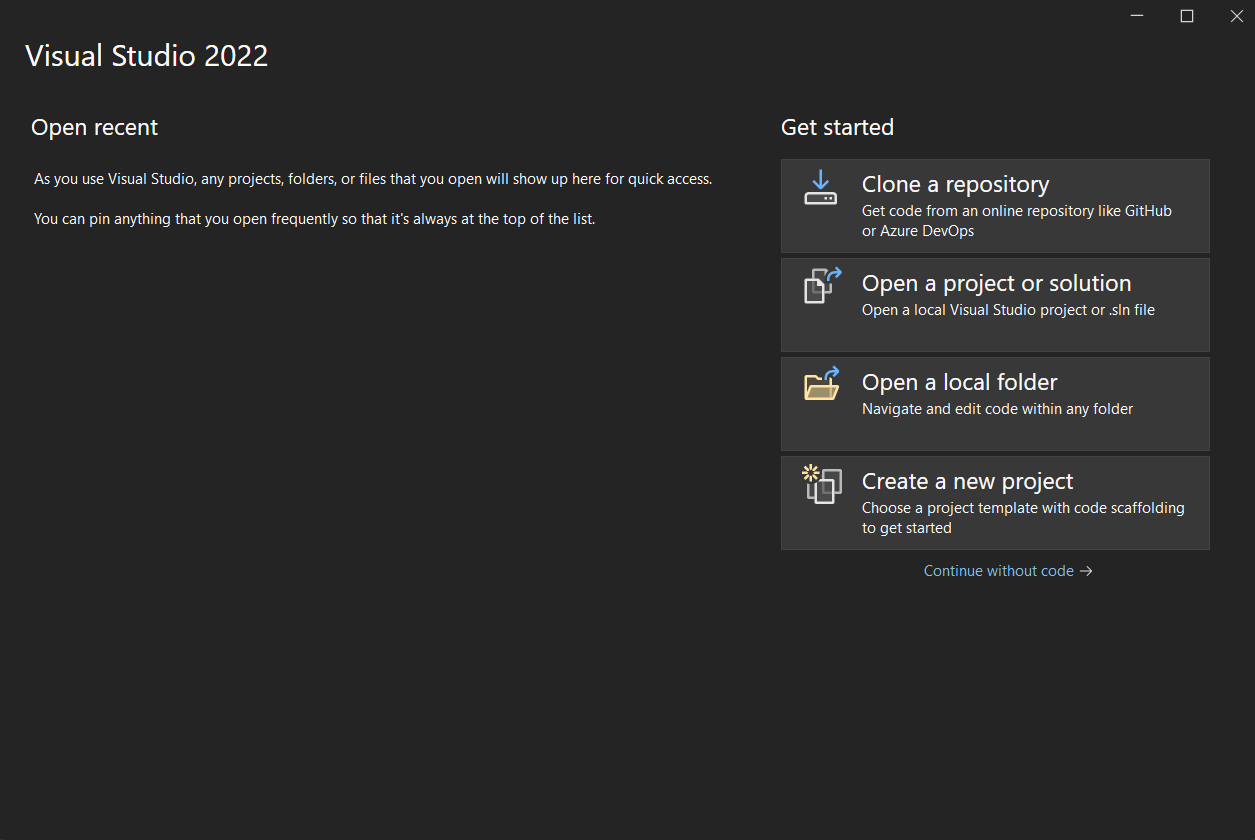
Và nhận được kết quả



1. **Test NUnit phương trình bậc 2**
   1. Chuẩn bị môi trường

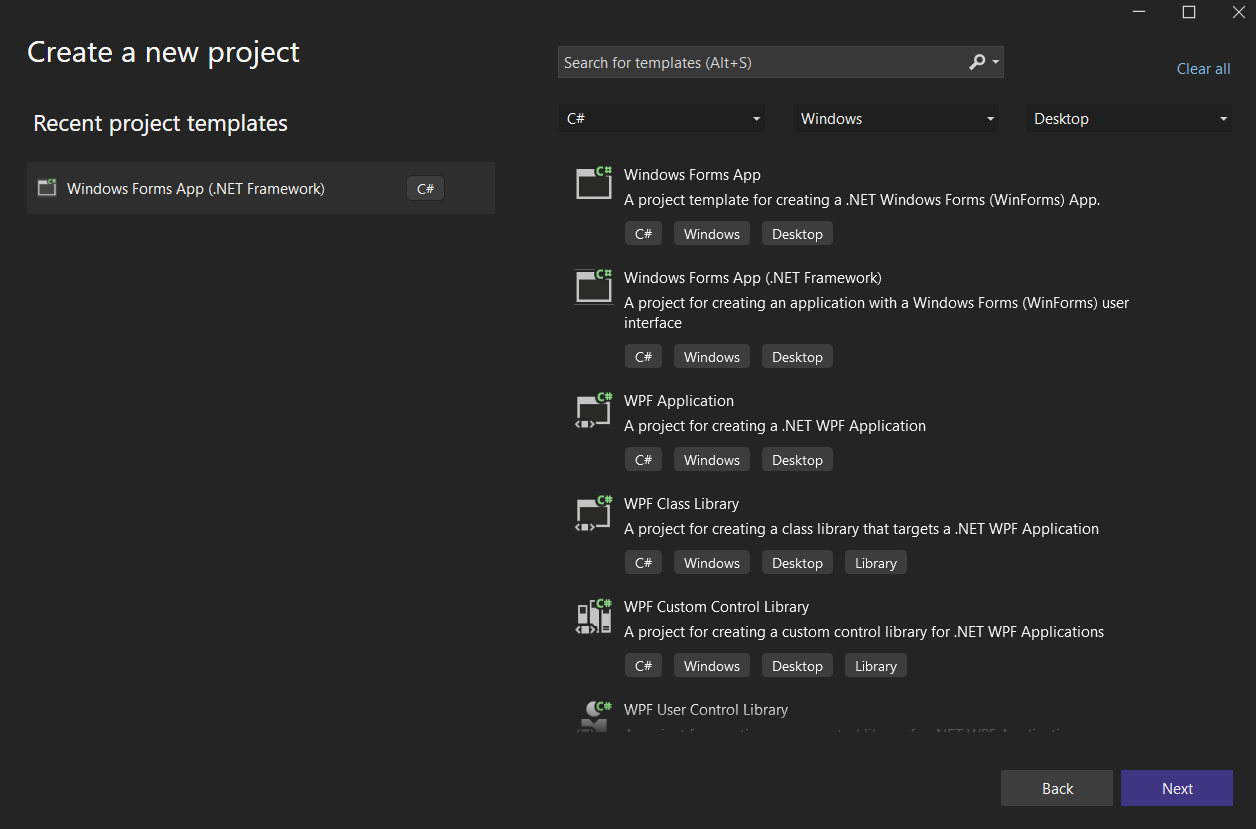
Bước 1:

Chọn vào create a new project



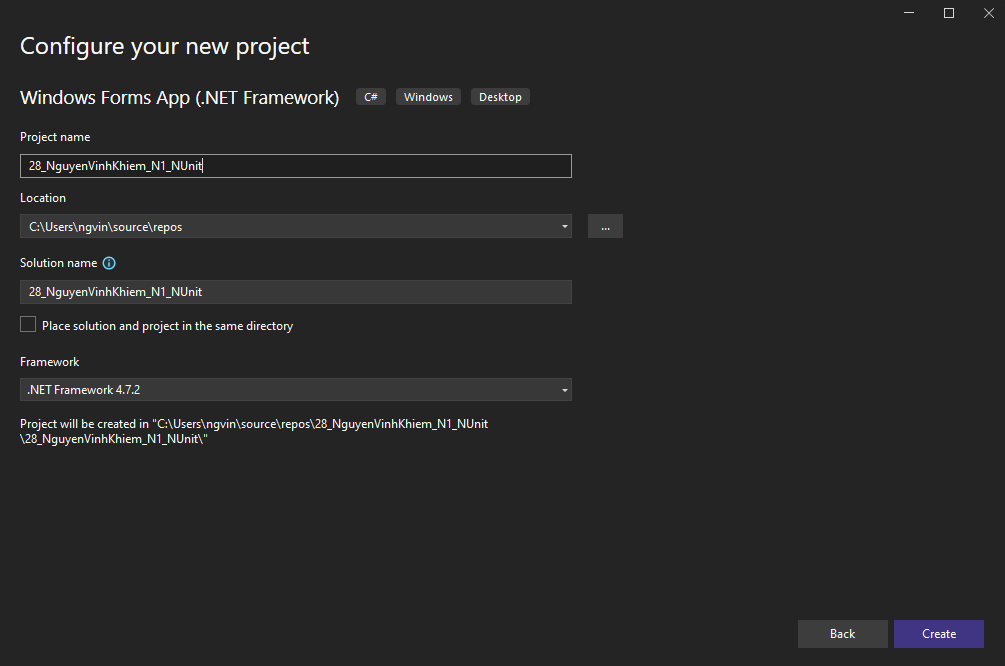
Bước 2:

Chọn vào lần lượt vào C# - Windows – Desktop -> Chọn project Unit TestProject (.Net Framework) và ấn next



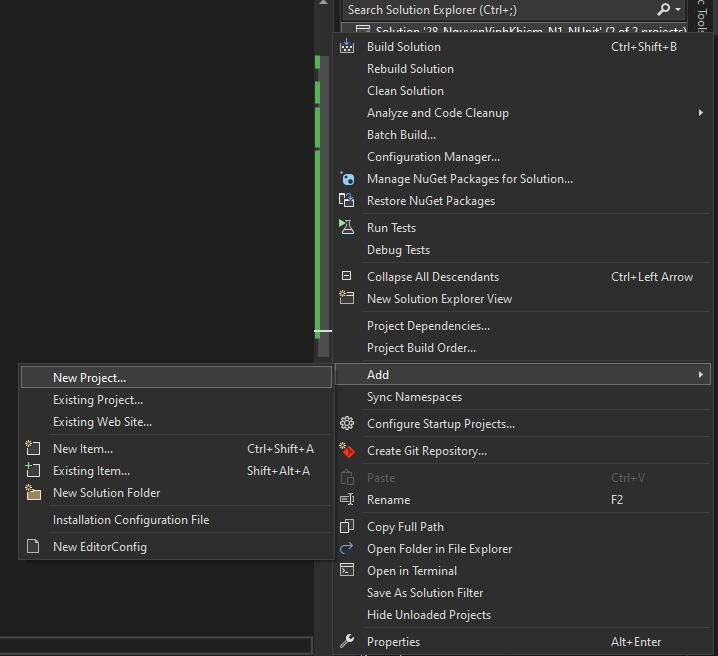
Bước 3:

Đặt tên project và lưu địa chỉ muốn lưu và sau đó ấn create



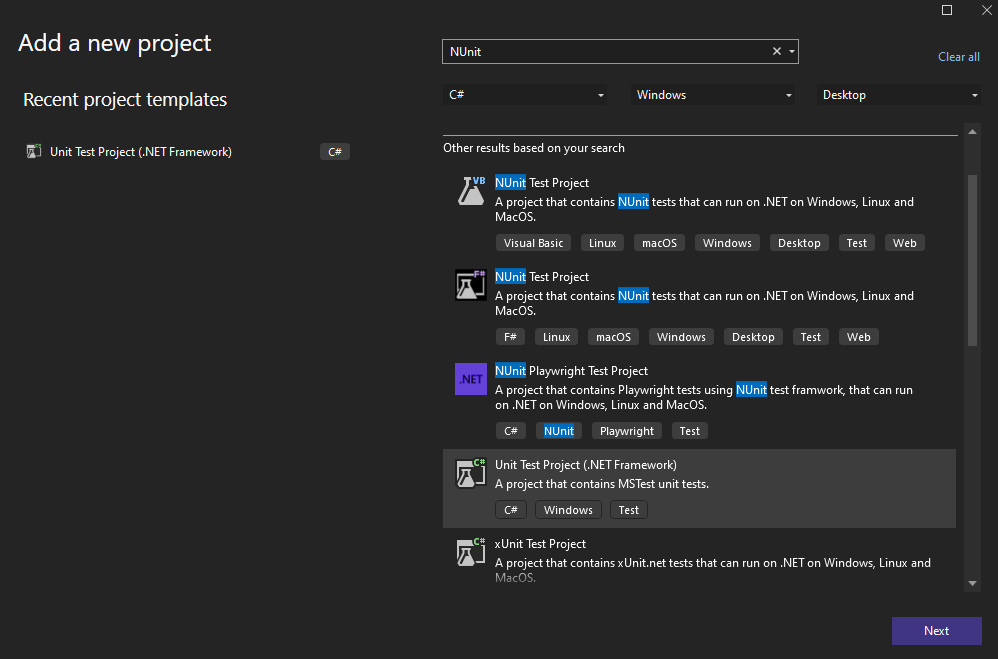
Bước 4:

Click chuột phải vào Solution ở thanh Solution Explorer -> Add -> New Project



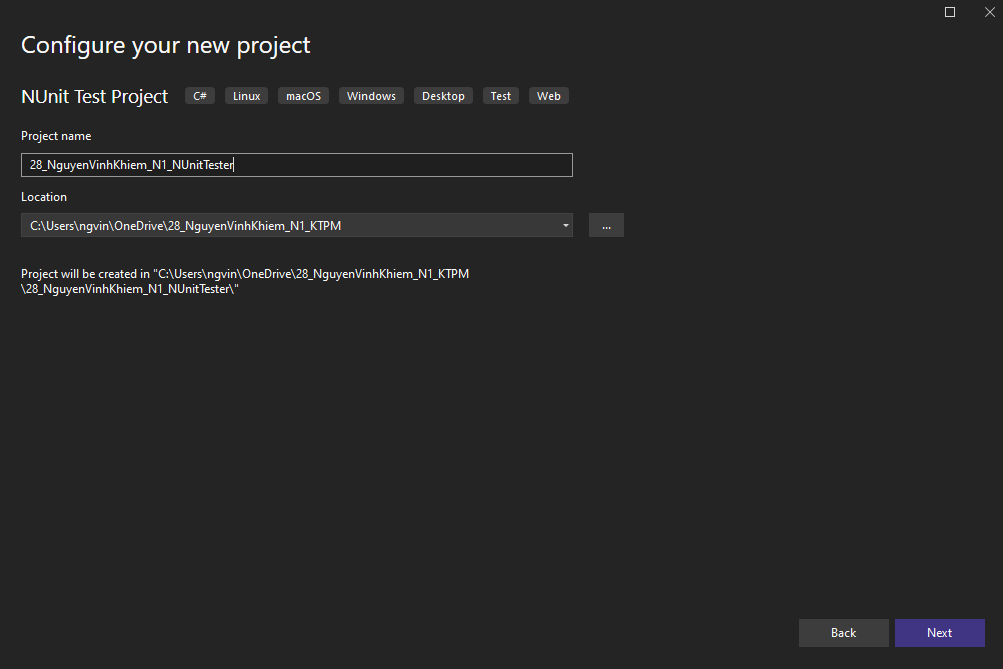
Bước 5:

Tìm từ khóa “NUnit” -> Chọn project .NET sau đó ấn next



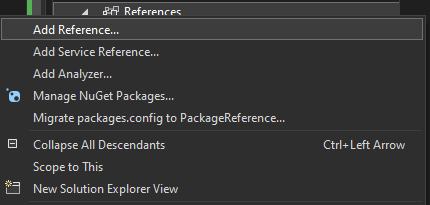
Bước 6:

Đặt tên Project



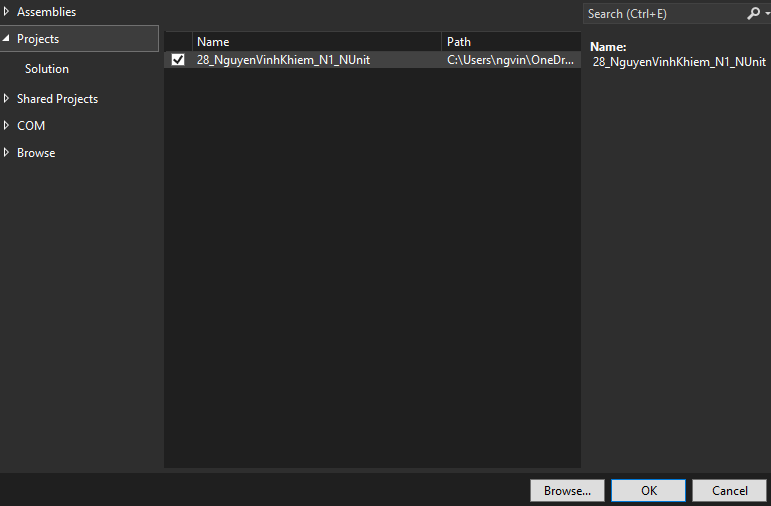
Bước 7:

Chọn References -> Add Reference..



Bước 8:

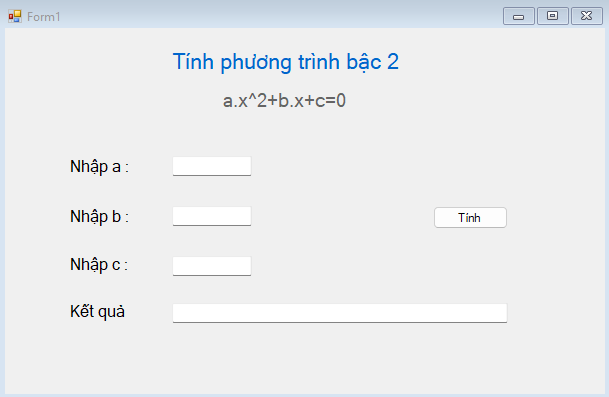
Tick vào Project của Form sau đó ấn OK



* 1. Giải thích code

Bước 1:

Tạo form, cho người dùng nhập vào 2 số a, b, c



Bước 2:

Tạo 1 class mới tên PTBac2 và viết công thức tính delta trong đó

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_28\_NguyenVinhKhiem\_N1\_NUnit

{

public class PTBac2

{

//28-Khiem

private int a, b, c;

public PTBac2(int a, int b, int c)

{

this.a = a;

this.b = b;

this.c = c;

}

//28-Khiem

public int Execute(string CalSymbol)

{

int result = 0;

switch (CalSymbol)

{

case "delta":

result = this.b \* this.b - 4 \* this.a \* this.c;

break;

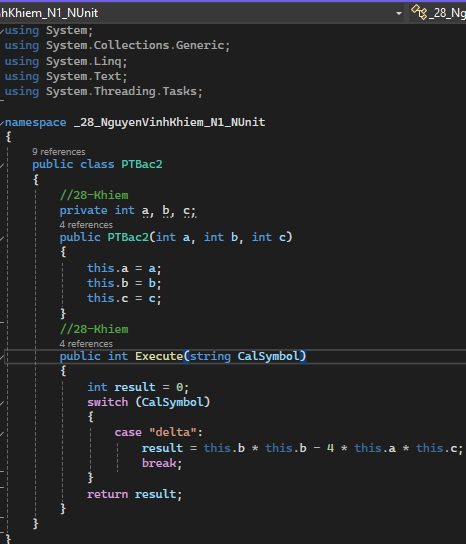
}

return result;

}

}

}



Bước 3:

Viết code nút Tính phường trình bậc 2

public void PTBac2\_btn\_Click(object sender, EventArgs e)

{

//28-Khiem

int a, b, c, delta;

string ketqua;

if (!int.TryParse(textBox1.Text, out a) || !int.TryParse(textBox2.Text, out b) || !int.TryParse(textBox3.Text, out c))

{

textBox4.Text = "Các hệ số a, b, c phải là các số nguyên hợp lệ.";

return;

}

PTBac2 d = new PTBac2(a, b, c);

delta = d.Execute("delta");

if (a == 0)

{

textBox4.Text = "Hệ số a phải khác 0";

}

else

{

if (delta > 0)

{

double x1 = (-b + Math.Sqrt(delta)) / (2 \* a);

double x2 = (-b - Math.Sqrt(delta)) / (2 \* a);

ketqua = "Phương trình có hai nghiệm phân biệt: ";

ketqua += "x1 = " + x1.ToString() + " và ";

ketqua += "x2 = " + x2.ToString();

textBox4.Text = ketqua;

}

else if (delta == 0)

{

double x = -b / (2 \* a);

ketqua = "Phương trình có một nghiệm kép: ";

ketqua += "x = " + x.ToString();

textBox4.Text = ketqua;

}

else

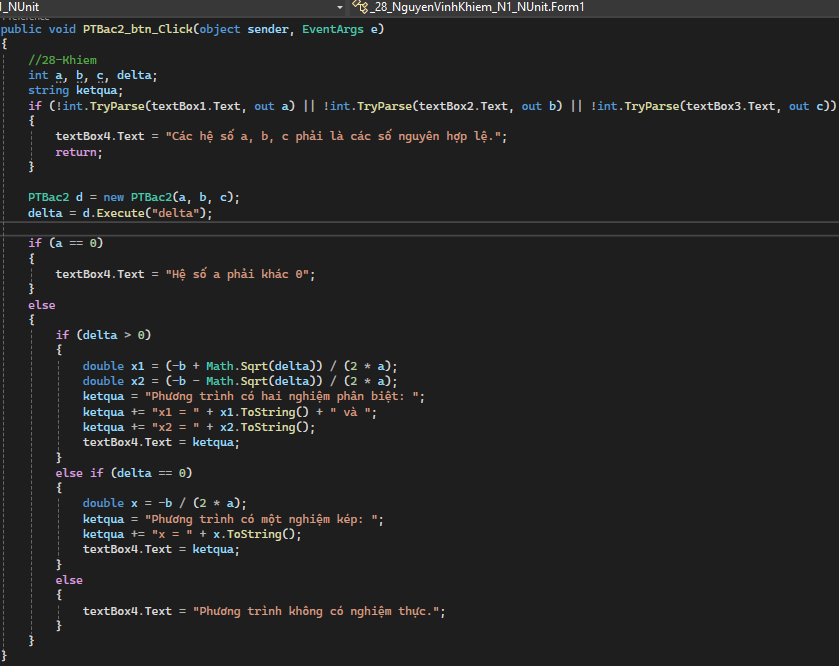
{

textBox4.Text = "Phương trình không có nghiệm thực.";

}

}

}



Khi đó chương trình tính phương trình bậc 2 đã hoàn thành.

Bước 4: Test case 1 – PT có 2 nghiệm phân biệt

[TestMethod] //TC1: a=1, b=3, c=-4 => delta = 25 > 0 => PT có 2 nghiệm phân biệt

public void Test2N()

{

double expected, actual;

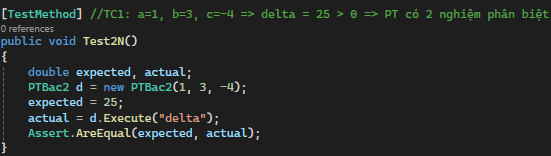
PTBac2 d = new PTBac2(1, 3, -4);

expected = 25;

actual = d.Execute("delta");

Assert.AreEqual(expected, actual);

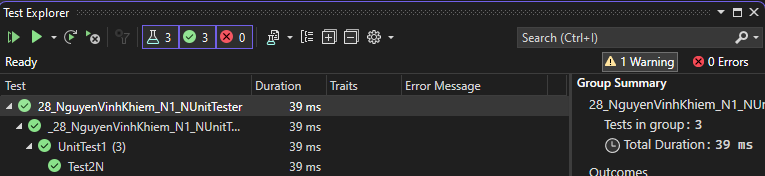
}



Tạo 2 biến expected là số mong muốn và actual là số thực tế.

Sau đó đưa vào 3 số 1, 3, 4 để tính delta.

Nếu delta > 0, cụ thể là 25 như mong muốn thì test case đúng.



Bước 5: Test case 2 - PT có nghiệm kép

[TestMethod] //TC1: a=1, b=2, c=-1 => delta = 0 => PT có 2 nghiệm kép

public void TestNK()

{

double expected, actual;

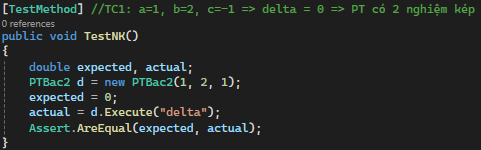
PTBac2 d = new PTBac2(1, 2, 1);

expected = 0;

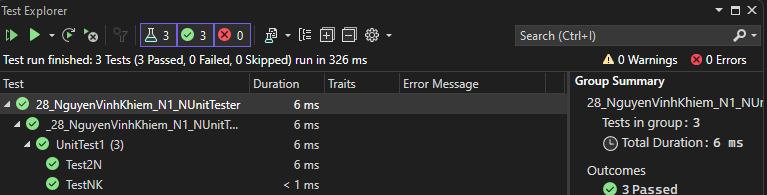
actual = d.Execute("delta");

Assert.AreEqual(expected, actual);

}



Tương tự như test case 1, nếu delta = 0 thì test case đúng



Bước 6: Test case 3 – PT vô nghiệm

[TestMethod] //TC1: a=1, b=1, c=-1 => delta = -3 < 0 => PT vô nghiệm

public void TestVN()

{

double expected, actual;

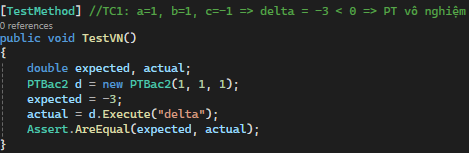
PTBac2 d = new PTBac2(1, 1, 1);

expected = -3;

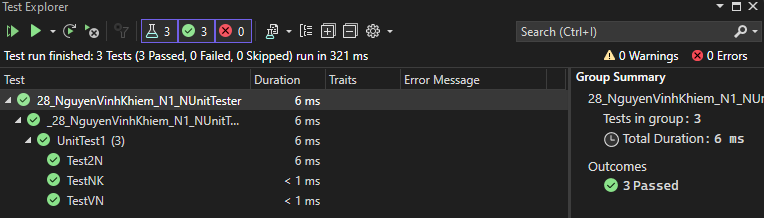
actual = d.Execute("delta");

Assert.AreEqual(expected, actual);

}



Tương tự như 2 test case trên, nếu delta < 0, cụ thể là -3 thì test case đúng



Hết