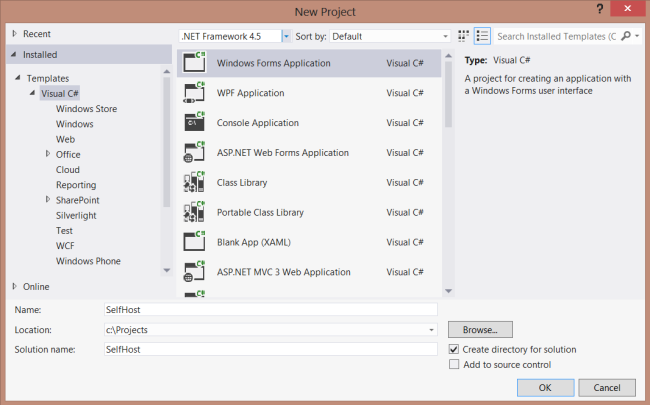
**Self-Host ASP.NET Web API**

## Tạo dự án mới, chọn loại **Console Application Project**

Start Visual Studio and select **New Project** from the **Start** page. Or, from the **File** menu, select **New** and then **Project**.

In the **Templates** pane, select **Installed Templates** and expand the **Visual C#** node. Under **Visual C#**, select **Windows**. In the list of project templates, select **Console Application**. Name the project "SelfHost" and click **OK**.

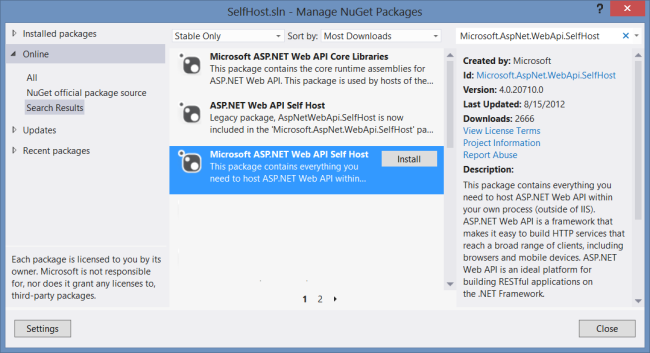


**Cài package WebApi Self-Host** - Add the Web API Self-Host package to your project.

1. From the **Tools** menu, select **NuGet Package Manager**. Note: If do you not see this menu item, make sure that NuGet Package Manager installed correctly.
2. Select **Manage NuGet Packages for Solution**
3. In the **Manage NugGet Packages** dialog, select **Online**.
4. In the search box, type "Microsoft.AspNet.WebApi.SelfHost".
5. Select the ASP.NET Web API Self Host package and click **Install**.
6. After the package installs, click **Close** to close the dialog.

**Note**

Make sure to install the package named **Microsoft.AspNet.WebApi.SelfHost**, not AspNetWebApi.SelfHost.



## Create the Model and Controller

This tutorial uses the same model and controller classes as the [Getting Started](https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api) tutorial.

Add a public class named Product.

namespace SelfHost

{

public class Product

{

public int Id { get; set; }

public string Name { get; set; }

public string Category { get; set; }

public decimal Price { get; set; }

}

}

Add a public class named ProductsController. Derive this class from **System.Web.Http.ApiController**.

namespace SelfHost

{

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Web.Http;

public class ProductsController : ApiController

{

Product[] products = new Product[]

{

new Product { Id = 1, Name = "Tomato Soup", Category = "Groceries", Price = 1 },

new Product { Id = 2, Name = "Yo-yo", Category = "Toys", Price = 3.75M },

new Product { Id = 3, Name = "Hammer", Category = "Hardware", Price = 16.99M }

};

public IEnumerable<Product> GetAllProducts()

{

return products;

}

public Product GetProductById(int id)

{

var product = products.FirstOrDefault((p) => p.Id == id);

if (product == null)

{

throw new HttpResponseException(HttpStatusCode.NotFound);

}

return product;

}

public IEnumerable<Product> GetProductsByCategory(string category)

{

return products.Where(p => string.Equals(p.Category, category,

StringComparison.OrdinalIgnoreCase));

}

}

}

| **TABLE 1** | |
| --- | --- |
| **URI** | **Description** |
| /api/products | Get a list of all products. |
| /api/products/id | Get a product by ID. |
| /api/products/?category=category | Get a list of products by category. |

## **Host the Web API**

Open the file Program.cs and add the following using statements:

using System.Web.Http;

using System.Web.Http.SelfHost;

Add the following code to the **Program** class.

var config = new HttpSelfHostConfiguration("http://localhost:8080");

config.Routes.MapHttpRoute(

"API Default", "api/{controller}/{id}",

new { id = RouteParameter.Optional });

using (HttpSelfHostServer server = new HttpSelfHostServer(config))

{

server.OpenAsync().Wait();

Console.WriteLine("Press Enter to quit.");

Console.ReadLine();

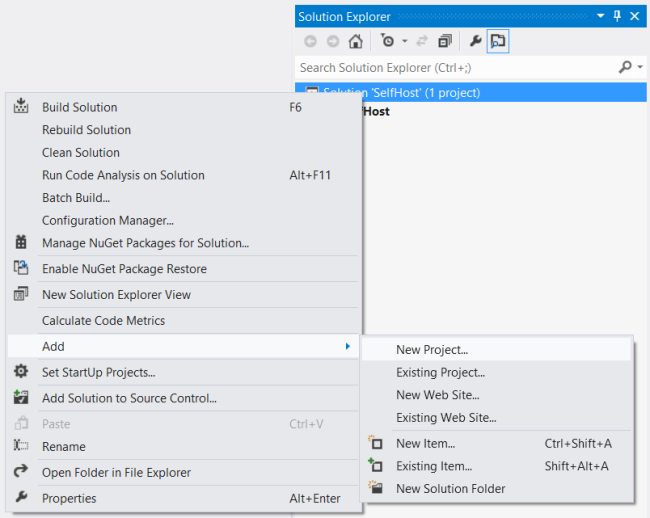
}

## **Call the Web API from a Client Application (C#)**

Let's write a simple console application that calls the web API.

Add a new console application project to the solution:

* In Solution Explorer, right-click the solution and select **Add New Project**.
* Create a new console application named "ClientApp".

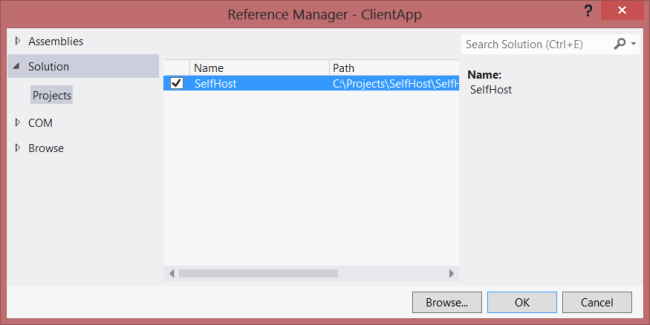


Use NuGet Package Manager to add the ASP.NET Web API Core Libraries package:

* From the Tools menu, select **NuGet Package Manager**.
* Select **Manage NuGet Packages for Solution**
* In the **Manage NuGet Packages** dialog, select **Online**.
* In the search box, type "Microsoft.AspNet.WebApi.Client".
* Select the Microsoft ASP.NET Web API Client Libraries package and click **Install**.

Add a reference in ClientApp to the SelfHost project:

* In Solution Explorer, right-click the ClientApp project.
* Select **Add Reference**.
* In the **Reference Manager** dialog, under **Solution**, select **Projects**.
* Select the **SelfHost** project.
* Click **OK**.



Open the Client/Program.cs file. Add the following **using** statement:

using System.Net.Http;

Add a static **HttpClient** instance:

namespace Client

{

class Program

{

static HttpClient client = new HttpClient();

}

}

Add the following methods to list all products, list a product by ID, and list products by category.

static void ListAllProducts()

{

HttpResponseMessage resp = client.GetAsync("api/products").Result;

resp.EnsureSuccessStatusCode();

var products = resp.Content.ReadAsAsync<IEnumerable<SelfHost.Product>>().Result;

foreach (var p in products)

{

Console.WriteLine("{0} {1} {2} ({3})", p.Id, p.Name, p.Price, p.Category);

}

}

static void ListProduct(int id)

{

var resp = client.GetAsync(string.Format("api/products/{0}", id)).Result;

resp.EnsureSuccessStatusCode();

var product = resp.Content.ReadAsAsync<SelfHost.Product>().Result;

Console.WriteLine("ID {0}: {1}", id, product.Name);

}

static void ListProducts(string category)

{

Console.WriteLine("Products in '{0}':", category);

string query = string.Format("api/products?category={0}", category);

var resp = client.GetAsync(query).Result;

resp.EnsureSuccessStatusCode();

var products = resp.Content.ReadAsAsync<IEnumerable<SelfHost.Product>>().Result;

foreach (var product in products)

{

Console.WriteLine(product.Name);

}

}

Each of these methods follows the same pattern:

1. Call **HttpClient.GetAsync** to send a GET request to the appropriate URI.
2. Call **HttpResponseMessage.EnsureSuccessStatusCode**. This method throws an exception if the HTTP response status is an error code.
3. Call **ReadAsAsync<T>** to deserialize a CLR type from the HTTP response. This method is an extension method, defined in **System.Net.Http.HttpContentExtensions**.

The **GetAsync** and **ReadAsAsync** methods are both asynchronous. They return **Task** objects that represent the asynchronous operation. Getting the **Result** property blocks the thread until the operation completes.

static void Main(string[] args)

{

client.BaseAddress = new Uri("http://localhost:8080");

ListAllProducts();

ListProduct(1);

ListProducts("toys");

Console.WriteLine("Press Enter to quit.");

Console.ReadLine();

}

**This should output the following**. (Remember to run the SelfHost application first.)

1 Tomato Soup 1.0 (Groceries)

2 Yo-yo 3.75 (Toys)

3 Hammer 16.99 (Hardware)

ID 1: Tomato Soup

Products in 'toys':

Yo-yo

Press Enter to quit.