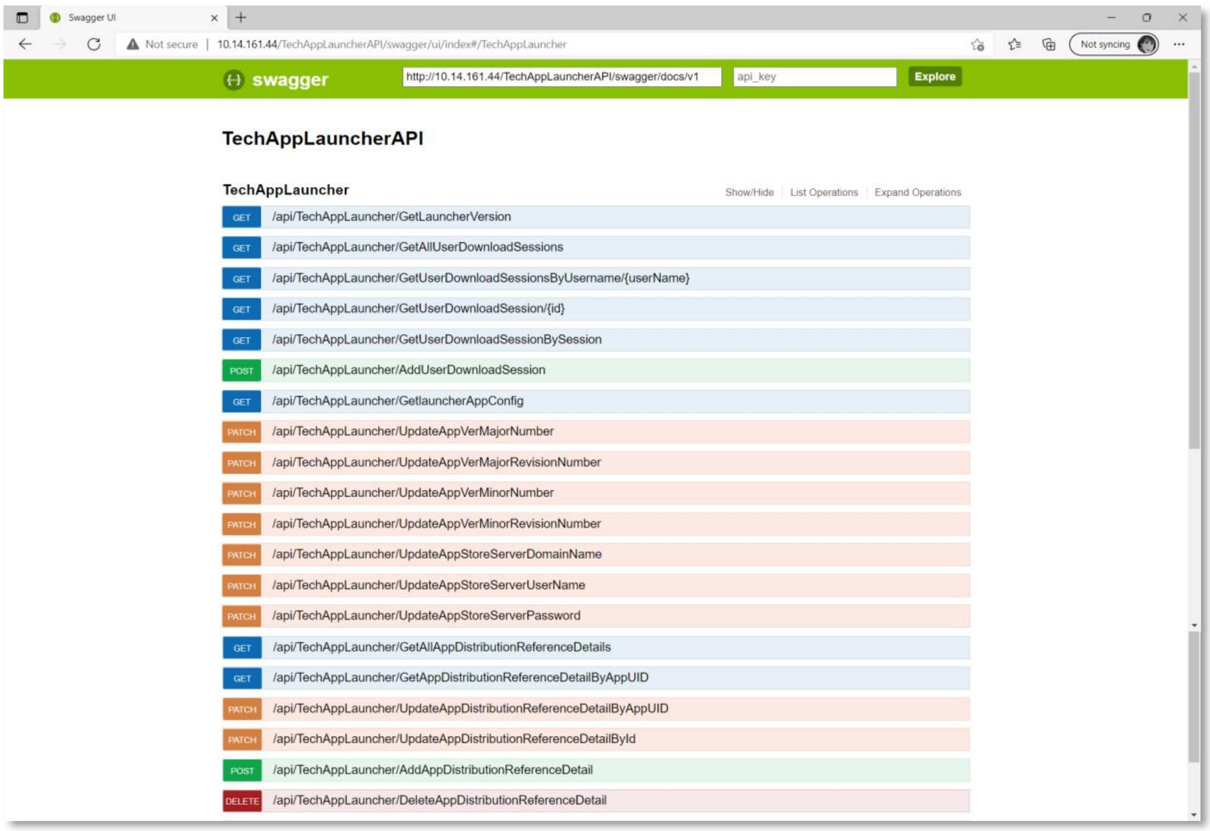
TechAppLauncherAPI

Application Name	**TechAppLauncherAPI**
Application Framework	.Net Framework 4.7.2
UI Framework	ASP.net WebAPI
IDE	Visual Studio 2019
Repository	https://github.com/azzulhisham/Petronas_TechAppLauncherAPI.git
Application Url	http://10.14.161.44/TechAppLauncherAPI/swagger/ui/index
Host	IIS

Overview



Swagger UI
Not secure
10.14.161.44/TechAppLauncherAPI/swagger/ui/index#/TechAppLauncher
Not syncing
swagger
http://10.14.161.44/TechAppLauncherAPI/swagger/docs/v1
api_key
Explore
Show/Hide
List Operations
Expand Operations

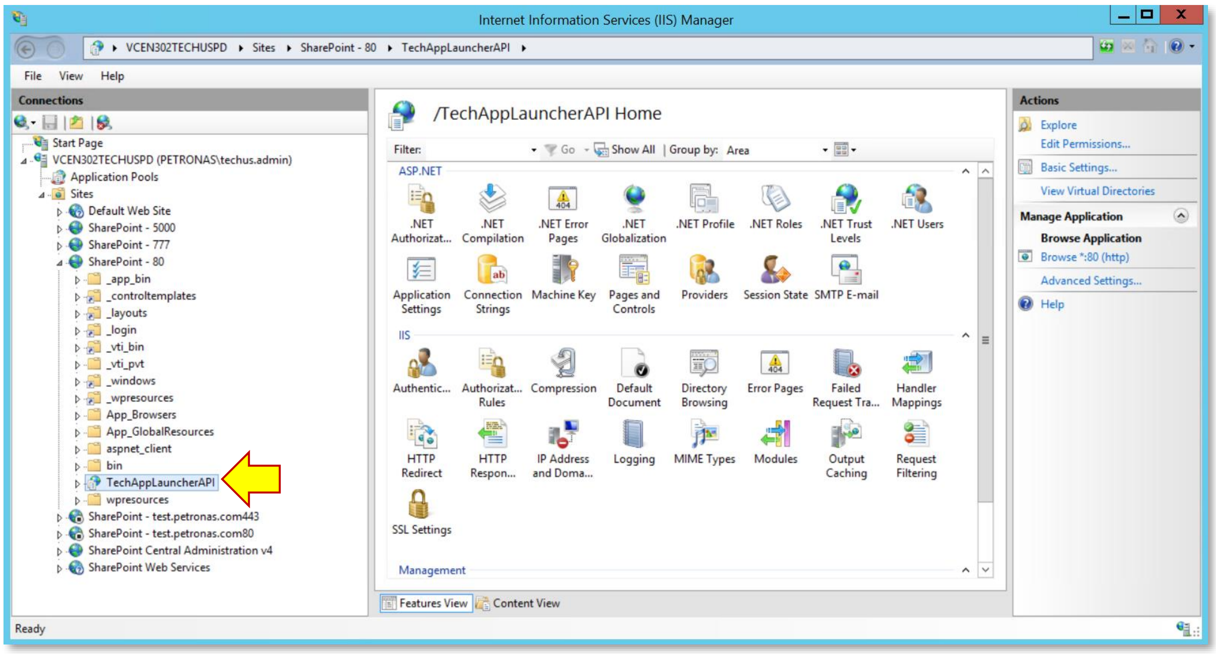
TechAppLauncherAPI

TechAppLauncher

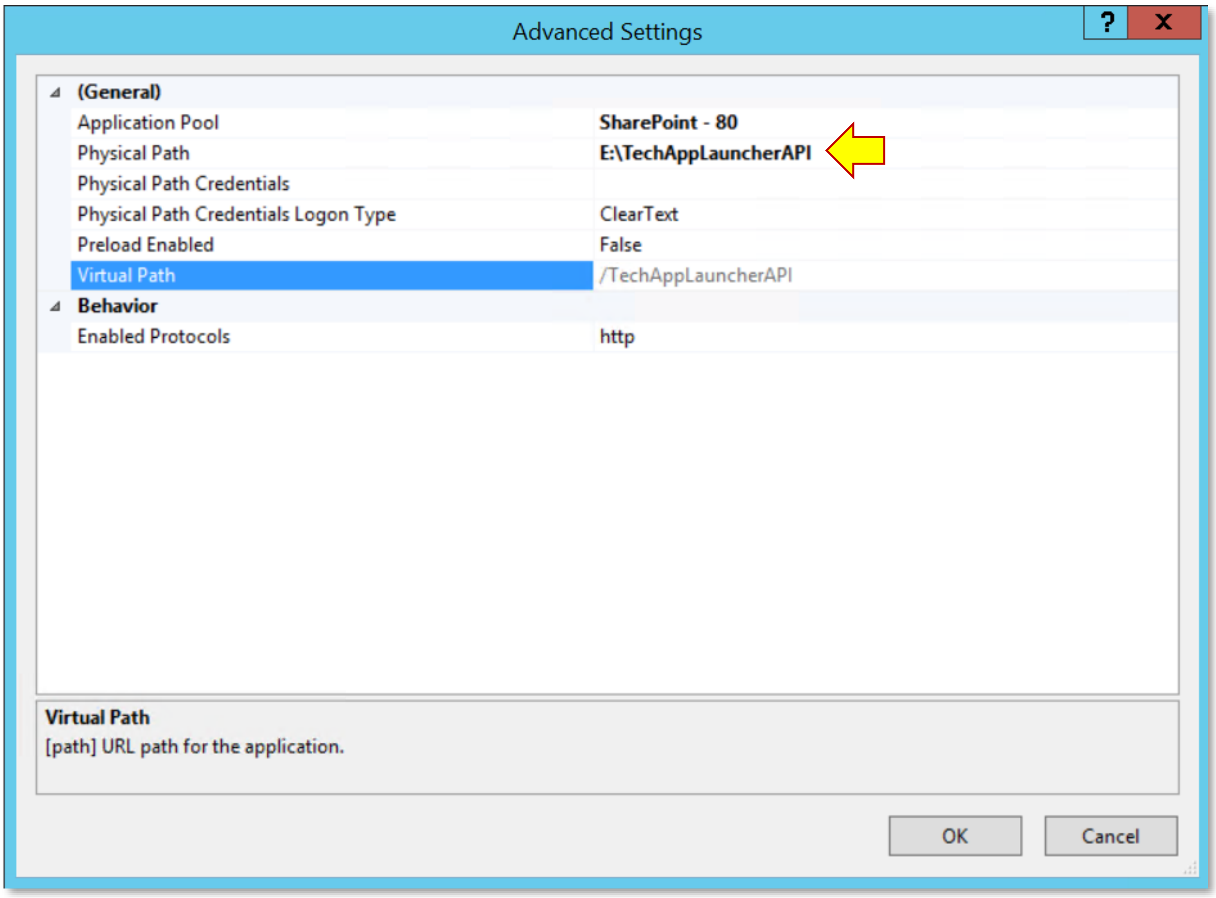
GET	/api/TechAppLauncher/GetLauncherVersion
GET	/api/TechAppLauncher/GetAllUserDownloadSessions
GET	/api/TechAppLauncher/GetUserDownloadSessionsByUsername/{userName}
GET	/api/TechAppLauncher/GetUserDownloadSession/{id}
GET	/api/TechAppLauncher/GetUserDownloadSessionBySession
POST	/api/TechAppLauncher/AddUserDownloadSession
GET	/api/TechAppLauncher/GetlauncherAppConfig
PATCH	/api/TechAppLauncher/UpdateAppVerMajorNumber
PATCH	/api/TechAppLauncher/UpdateAppVerMajorRevisionNumber
PATCH	/api/TechAppLauncher/UpdateAppVerMinorNumber
PATCH	/api/TechAppLauncher/UpdateAppVerMinorRevisionNumber
PATCH	/api/TechAppLauncher/UpdateAppStoreServerDomainName
PATCH	/api/TechAppLauncher/UpdateAppStoreServerUserName
PATCH	/api/TechAppLauncher/UpdateAppStoreServerPassword
GET	/api/TechAppLauncher/GetAllAppDistributionReferenceDetails
GET	/api/TechAppLauncher/GetAppDistributionReferenceDetailByAppUID
PATCH	/api/TechAppLauncher/UpdateAppDistributionReferenceDetailByAppUID
PATCH	/api/TechAppLauncher/UpdateAppDistributionReferenceDetailById
POST	/api/TechAppLauncher/AddAppDistributionReferenceDetail
DELETE	/api/TechAppLauncher/DeleteAppDistributionReferenceDetail

The link Swagger UI provided above is the application API content which can be viewed through Swagger UI.

It is hosted by IIS Server at **10.14.161.44**. On top of that it is connecting to the MS-SQL server at **10.14.162.74**.



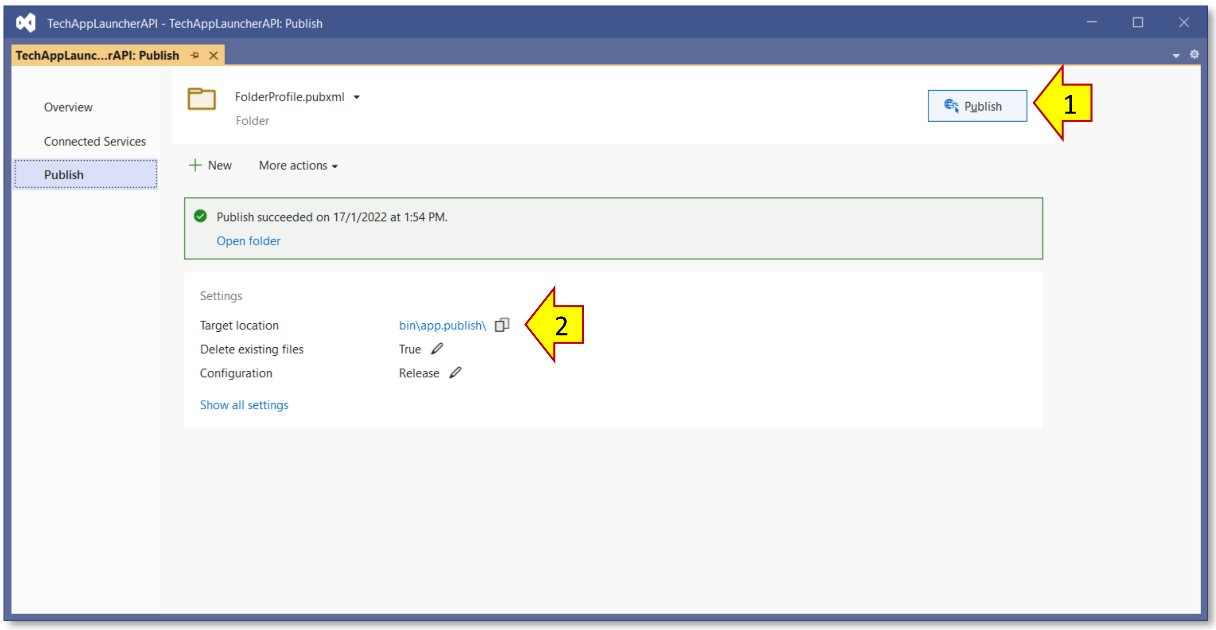
The physical path of the application is located at ‘**E:\TechAppLauncherAPI**’


Advanced Settings

(General)	
Application Pool	SharePoint - 80
Physical Path	E:\TechAppLauncherAPI
Physical Path Credentials	
Physical Path Credentials Logon Type	ClearText
Preload Enabled	False
Virtual Path	/TechAppLauncherAPI
Behavior	
Enabled Protocols	http

Virtual Path
[path] URL path for the application.

Deploying New Version



Publish succeeded on 17/1/2022 at 1:54 PM.

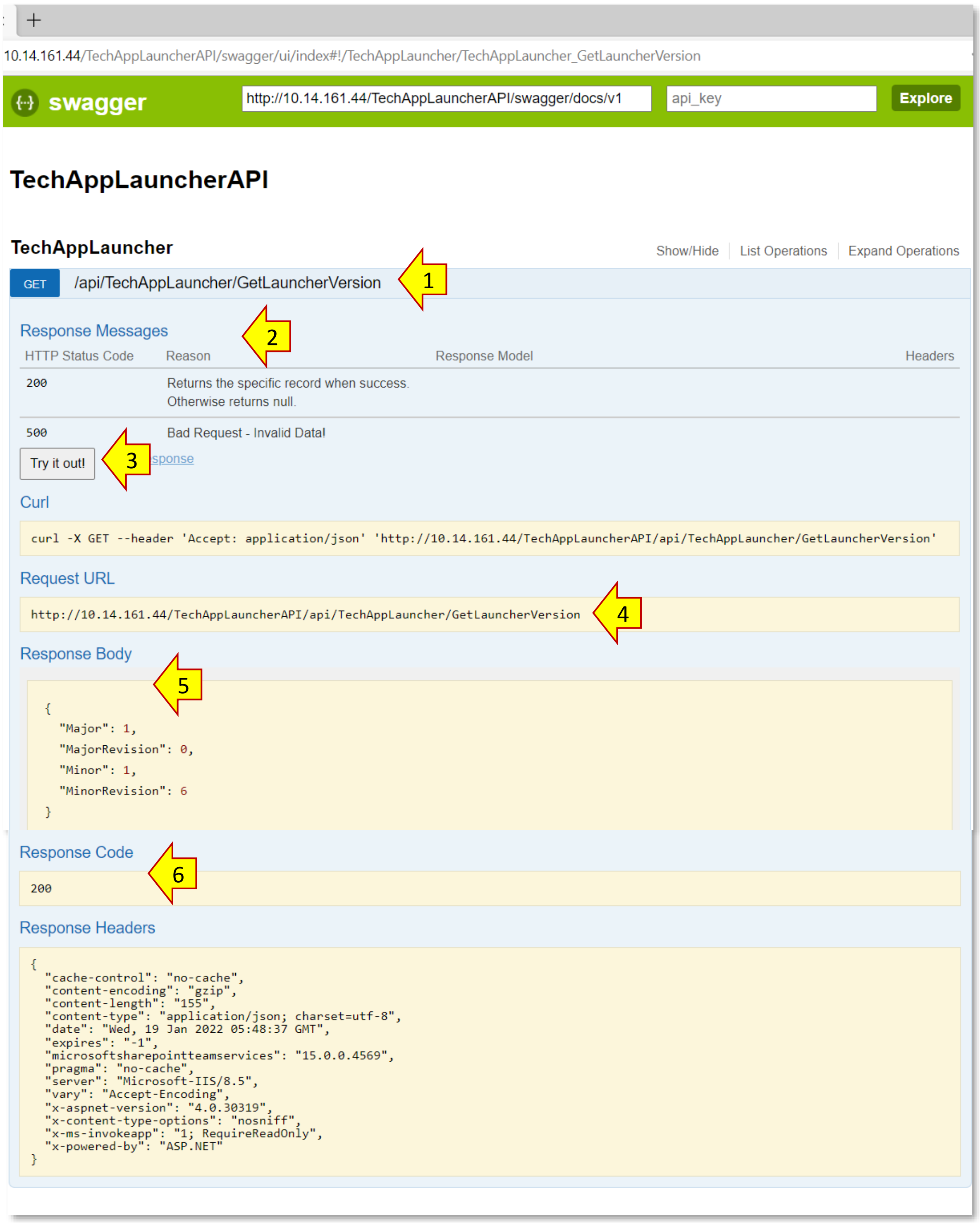
Settings

Target location	bin\app.publish\
Delete existing files	True
Configuration	Release

To publish a newer version of the app, right-click on the project in the solution explorer and then choose ‘**Publish…**’ button[1]. The wizard of publishing a newer version of the application will appear. Click on the ‘**Publish**’ button to proceed. Upon successful, the package of the application is located at ‘**bin\app.publish**’[2].

To deploy the API to the server – copy the entire content of the folder ‘**bin\app.publish**’ to the folder ‘**E:\TechAppLauncherAPI**’ at the server ’**10.14.161.44**’.

About Swagger UI



10.14.161.44/TechAppLauncherAPI/swagger/ui/index#!/TechAppLauncher/TechAppLauncher_GetLauncherVersion

http://10.14.161.44/TechAppLauncherAPI/swagger/docs/v1 api_key

TechAppLauncherAPI

TechAppLauncher

GET /api/TechAppLauncher/GetLauncherVersion

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Returns the specific record when success. Otherwise returns null.		
500	Bad Request - Invalid Data!		

Curl

```
curl -X GET --header 'Accept: application/json' 'http://10.14.161.44/TechAppLauncherAPI/api/TechAppLauncher/GetLauncherVersion'
```

Request URL

```
http://10.14.161.44/TechAppLauncherAPI/api/TechAppLauncher/GetLauncherVersion
```

Response Body

```
{
  "Major": 1,
  "MajorRevision": 0,
  "Minor": 1,
  "MinorRevision": 6
}
```

Response Code

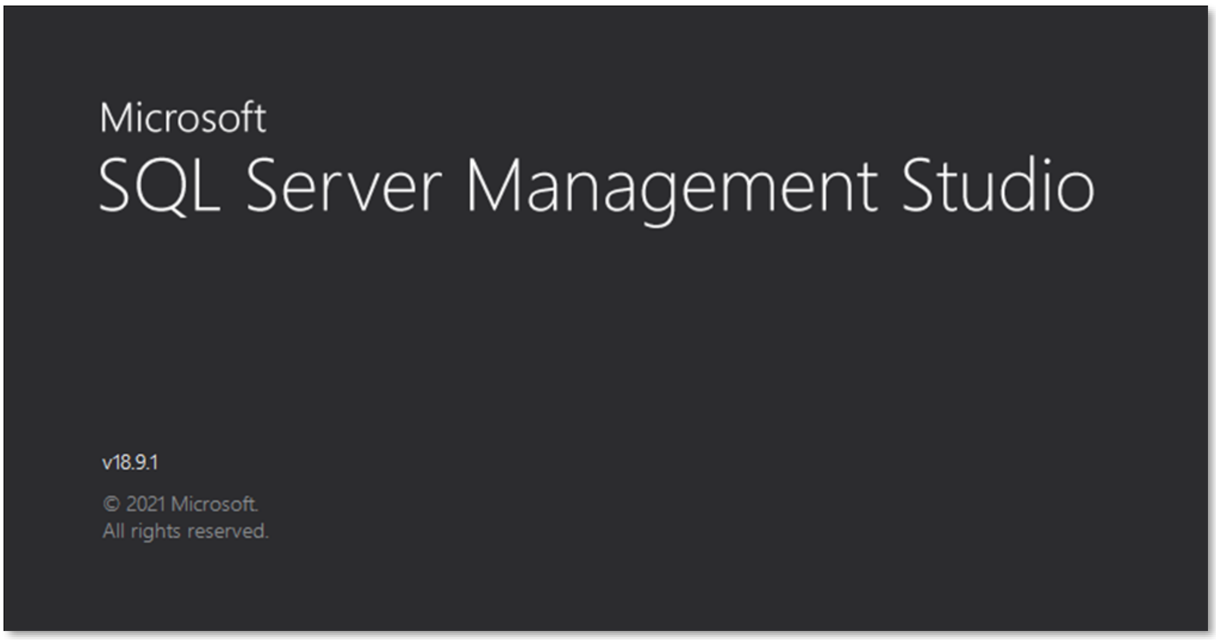
```
200
```

Response Headers

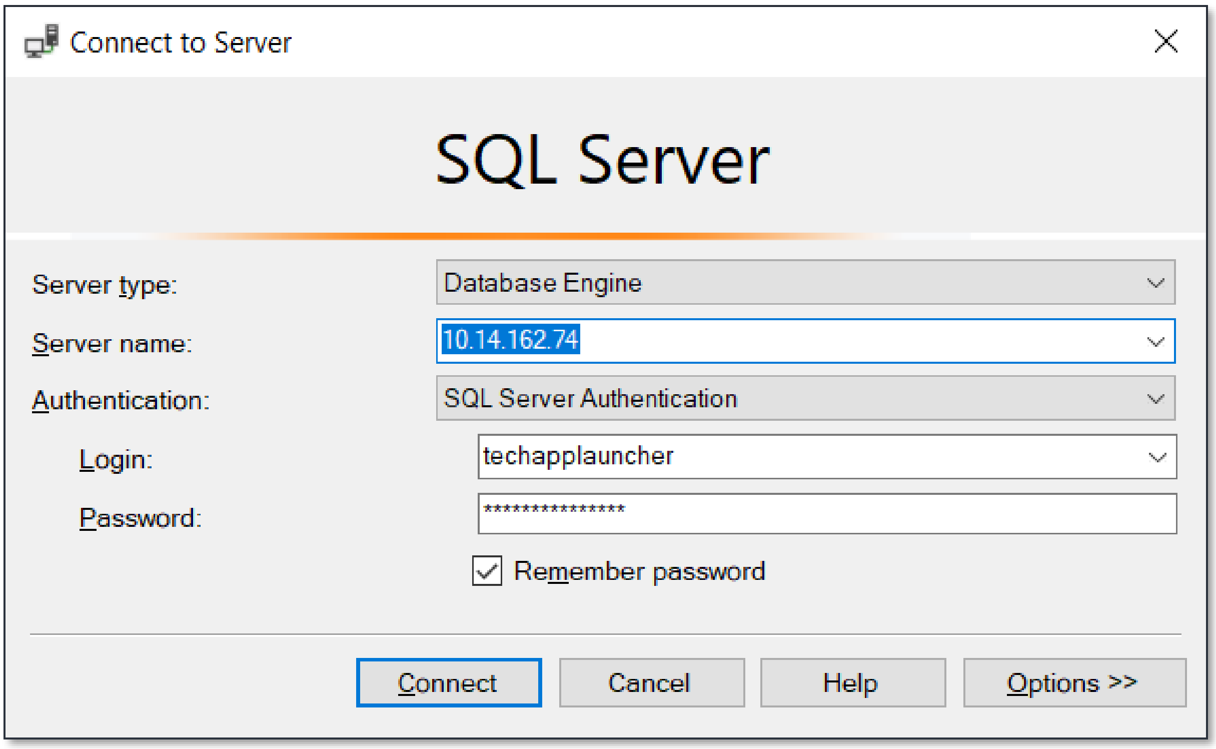
```
{
  "cache-control": "no-cache",
  "content-encoding": "gzip",
  "content-length": "155",
  "content-type": "application/json; charset=utf-8",
  "date": "Wed, 19 Jan 2022 05:48:37 GMT",
  "expires": "-1",
  "microsoftsharepointteamservices": "15.0.0.4569",
  "pragma": "no-cache",
  "server": "Microsoft-IIS/8.5",
  "vary": "Accept-Encoding",
  "x-aspnet-version": "4.0.30319",
  "x-content-type-options": "nosniff",
  "x-ms-invokeapp": "1; RequireReadOnly",
  "x-powered-by": "ASP.NET"
}
```

1. Shows the endpoint of the API
2. Provides the summary of the HTTP Response Message.
3. The button provides the feature to test the endpoint.
4. Shows the fully qualified URL for the specific endpoint.
5. Shows the HTTP Response body when success.
6. Shows the HTTP Response Status Code.

Managing SQL Server

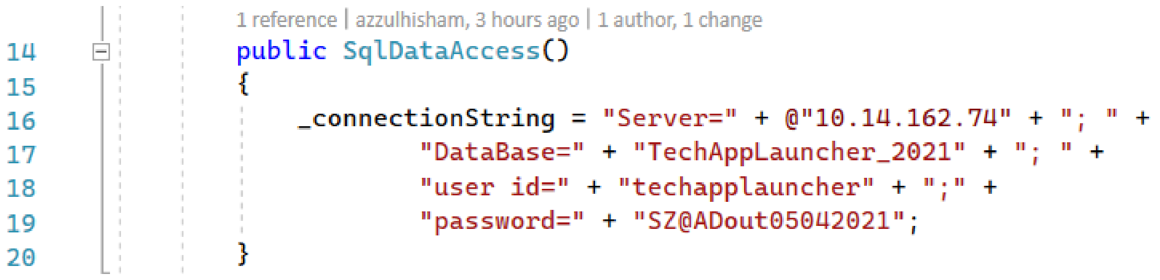


It is recommended to use the **Microsoft SQL Server Management Studio** in order to manage the SQL database for TechAppLauncher. It can download from the Microsoft official website: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

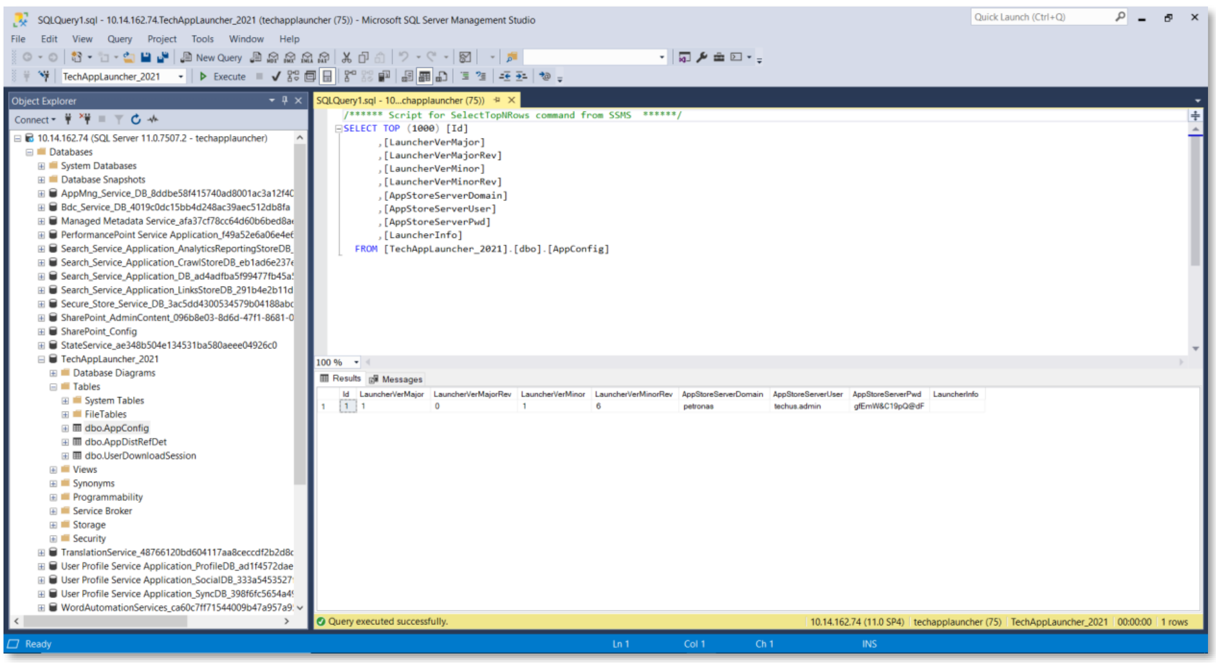


Server type: Database Engine
Server name: 10.14.162.74
Authentication: SQL Server Authentication
Login: techapplauncher
Password: ***************
☑ Remember password

The **username** and the **password** as well as the database being used by the application can be found in the repository of the application.



```
1 reference | azzulhisham, 3 hours ago | 1 author, 1 change
public SqlDataAccess()
{
    _connectionString = "Server=" + @"10.14.162.74" + "; " +
              "DataBase=" + "TechAppLauncher_2021" + "; " +
              "user id=" + "techapplauncher" + ";" +
              "password=" + "SZ@ADout05042021";
}
```



SQLQuery1.sql - 10.14.162.74.TechAppLauncher_2021 (techapplauncher (75)) - Microsoft SQL Server Management Studio

```
10.14.162.74 (SQL Server 11.0.7507.2 - techapplauncher)
  Databases
    System Databases
    Database Snapshots
    AppMng_Service_DB_8ddbe58f415740ad8001ac3a12f4c
    Bdc_Service_DB_4019c0dc15bb4d248ac39aec512db8fa
    Managed Metadata Service_afa37cf78cc64d60b6bed8a
    PerformancePoint Service Application_f49a52e6a06e4e
    Search_Service_Application_AnalyticsReportingStoreDB_
    Search_Service_Application_CrawlStoreDB_eb1ad6e237
    Search_Service_Application_DB_ad4adfba5f99477fb45a
    Search_Service_Application_LinksStoreDB_291b4e2b11d
    Secure_Store_Service_DB_3ac5dd4300534579b04188abc
    SharePoint_AdminContent_096b8e03-8d6d-47f1-8681-0
    SharePoint_Config
    StateService_ae348b504e134531ba580aeee04926c0
    TechAppLauncher_2021
      Database Diagrams
      Tables
        System Tables
        FileTables
        dbo.AppConfig
        dbo.AppDistRefDet
        dbo.UserDownloadSession
      Views
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
    TranslationService_48766120bd604117aa8ceccdf2b2d8c
    User Profile Service Application_ProfileDB_ad1f4572dae
    User Profile Service Application_SocialDB_333a5453527
    User Profile Service Application_SyncDB_398f6fc5654a4
    WordAutomationServices_ca60c7ff71544009b47a957a9
```

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Id]
      ,[LauncherVerMajor]
      ,[LauncherVerMajorRev]
      ,[LauncherVerMinor]
      ,[LauncherVerMinorRev]
      ,[AppStoreServerDomain]
      ,[AppStoreServerUser]
      ,[AppStoreServerPwd]
      ,[LauncherInfo]
  FROM [TechAppLauncher_2021].[dbo].[AppConfig]
```

	Id	LauncherVerMajor	LauncherVerMajorRev	LauncherVerMinor	LauncherVerMinorRev	AppStoreServerDomain	AppStoreServerUser	AppStoreServerPwd	LauncherInfo
1	1	1	0	1	6	petronas	techus.admin	gfEmW&C19pQ@dF	

Query executed successfully. 10.14.162.74 (11.0 SP4) | techapplauncher (75) | TechAppLauncher_2021 | 00:00:00 | 1 rows

Microsoft SQL Database Server

You try to download a free version of SQL server (**SQL Express**) and set it up on your system in order to get familiar with it. The link below is the official website to download the **SQL Express**.

https://www.microsoft.com/en-gb/download/details.aspx?id=101064

Database Tables

There are 3 tables currently in used in this project.

1. AppConfig
2. AppDistRefDet
3. UserDownloadSession

The script below is to create '**AppConfig**' table:

```
CREATE TABLE [AppConfig](
      [Id] [bigint] IDENTITY(1,1) NOT NULL,
      [LauncherVerMajor] [int] NOT NULL,
      [LauncherVerMajorRev] [int] NOT NULL,
      [LauncherVerMinor] [int] NOT NULL,
      [LauncherVerMinorRev] [int] NOT NULL,
      [AppStoreServerDomain] [nvarchar](32) NOT NULL,
      [AppStoreServerUser] [nvarchar](32) NOT NULL,
      [AppStoreServerPwd] [nvarchar](16) NOT NULL,
      [LauncherInfo] [nvarchar](50) NULL
) ON [PRIMARY]
```

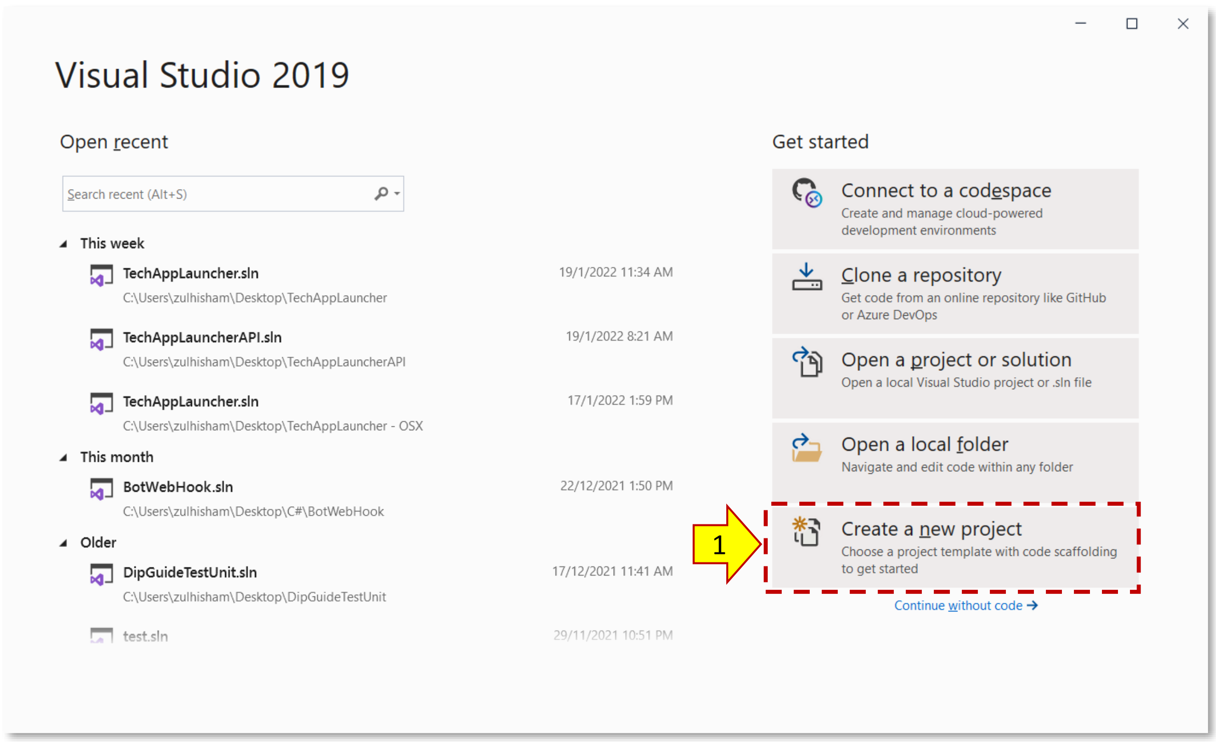
The script below is used to create '**AppDistRefDet**':

```
CREATE TABLE [AppDistRefDet](
      [Id] [bigint] IDENTITY(1,1) NOT NULL,
      [AppUID] [nvarchar](64) NOT NULL,
      [LinkID] [nvarchar](150) NOT NULL,
      [AgentName] [nvarchar](50) NOT NULL,
      [Description] [nvarchar](250) NULL,
 CONSTRAINT [uc_AppUID] UNIQUE NONCLUSTERED
(
      [AppUID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

And the script below is used to create the '**UserDownloadSession**':

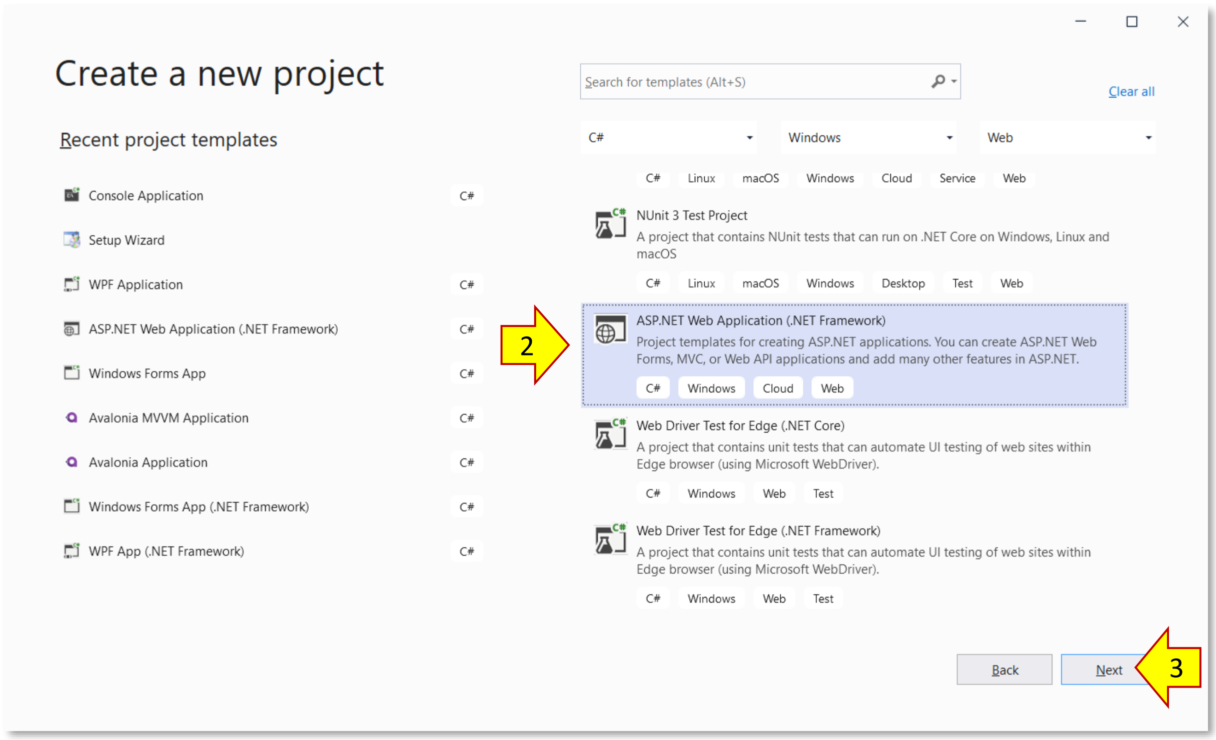
```
CREATE TABLE [UserDownloadSession](
      [Id] [bigint] IDENTITY(1,1) NOT NULL,
      [AppId] [bigint] NULL,
      [AppUID] [nvarchar](50) NOT NULL,
      [Title] [nvarchar](250) NULL,
      [Status] [nvarchar](50) NULL,
      [UserName] [nvarchar](150) NOT NULL,
      [InstallTimeStamp] [datetime] NOT NULL,
      [Remark] [nvarchar](50) NULL
) ON [PRIMARY]
```

A Quick Tutorial To Get The WebAPI Project Start Up By Using Visual Studio 2019



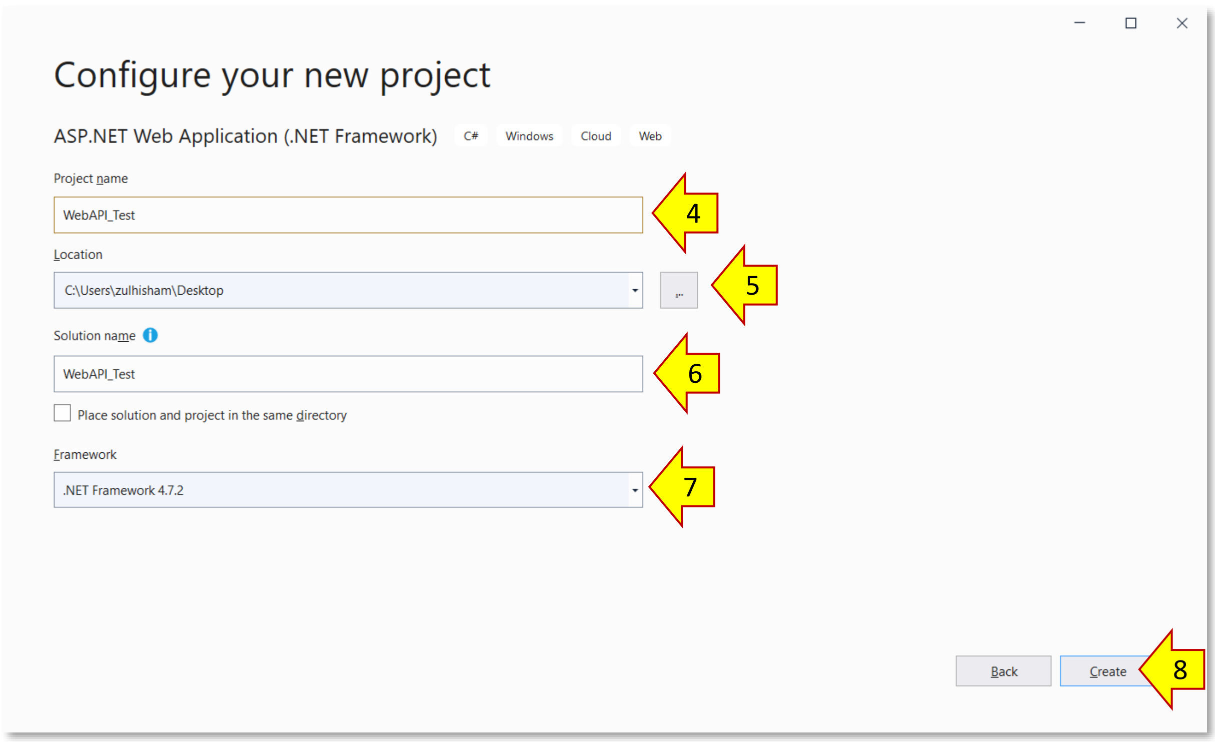
Visual Studio 2019
Open recent
Search recent (Alt+S)
This week
TechAppLauncher.sln 19/1/2022 11:34 AM
C:\Users\zulhisham\Desktop\TechAppLauncher
TechAppLauncherAPI.sln 19/1/2022 8:21 AM
C:\Users\zulhisham\Desktop\TechAppLauncherAPI
TechAppLauncher.sln 17/1/2022 1:59 PM
C:\Users\zulhisham\Desktop\TechAppLauncher - OSX
This month
BotWebHook.sln 22/12/2021 1:50 PM
C:\Users\zulhisham\Desktop\C#\BotWebHook
Older
DipGuideTestUnit.sln 17/12/2021 11:41 AM
C:\Users\zulhisham\Desktop\DipGuideTestUnit
test.sln 29/11/2021 10:51 PM
Get started
Connect to a codespace
Create and manage cloud-powered development environments
Clone a repository
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
1
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →

Now… let get your Visual Studio 2019 up and run. In the '**Get started**' page, choose the '**Create a new project**'[1] as shown in the screenshot above.



Create a new project
Recent project templates
Console Application C#
Setup Wizard
WPF Application C#
ASP.NET Web Application (.NET Framework) C#
Windows Forms App C#
Avalonia MVVM Application C#
Avalonia Application C#
Windows Forms App (.NET Framework) C#
WPF App (.NET Framework) C#
Search for templates (Alt+S)
Clear all
C#
Windows
Web
C# Linux macOS Windows Cloud Service Web
NUnit 3 Test Project
A project that contains NUnit tests that can run on .NET Core on Windows, Linux and macOS
C# Linux macOS Windows Desktop Test Web
2
ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Cloud Web
Web Driver Test for Edge (.NET Core)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft WebDriver).
C# Windows Web Test
Web Driver Test for Edge (.NET Framework)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft WebDriver).
C# Windows Web Test
Back
Next
3

In the '**Create a new project**' page, search for '**ASP.NET Web Application (.Net Framework)**'[2], then click the '**Next**' button[3] to proceed.



Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name: WebAPI_Test

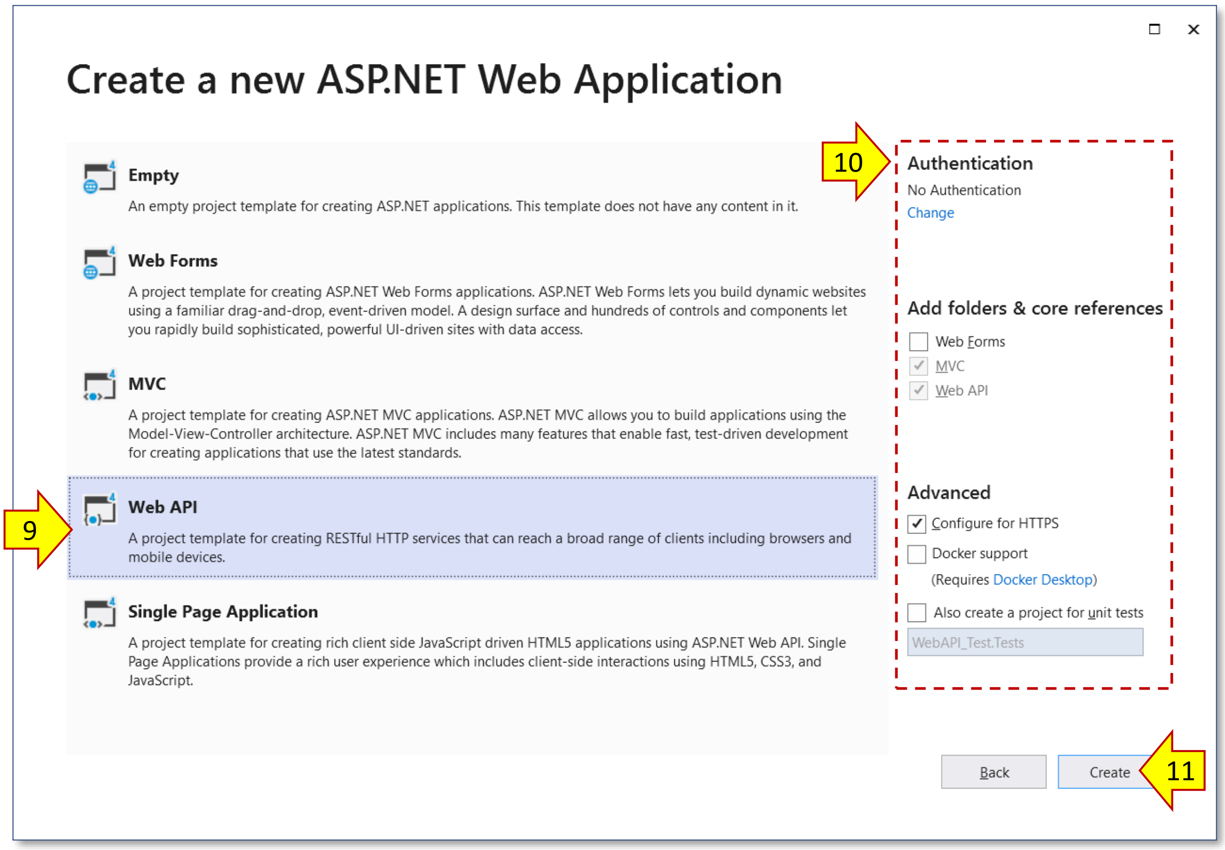
Location: C:\Users\zulhisham\Desktop

Solution name: WebAPI_Test

☐ Place solution and project in the same directory

Framework: .NET Framework 4.7.2

In the '**Configure your new project**' page, give a name for your project[4]. Then choose a location or path to place your project files[5]. You can leave the **Solution Name**[6] as the Visual Studio provided to you. Finally, select the appropriate **.net framework**[7] for your project. In this example, we will use '**.Net Framework 4.7.2**'. Once done, click the '**Create**' button[8] to let Visual Studio to start creating your project.



Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

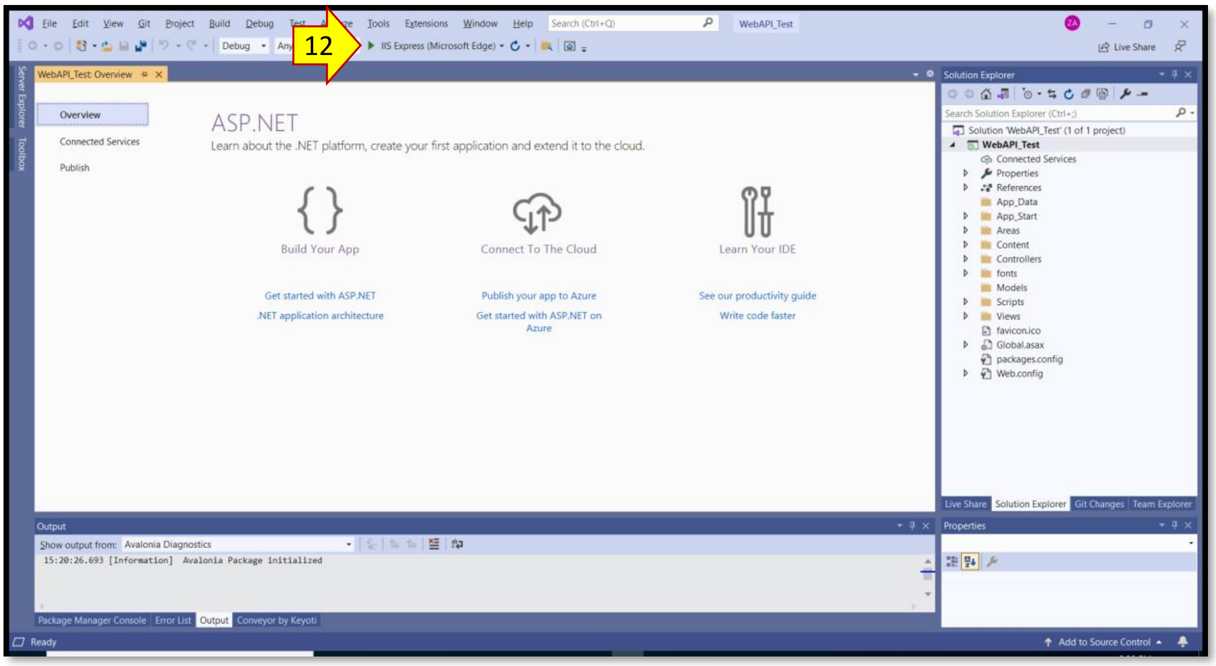
Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
Change

Add folders & core references
☐ Web Forms
✓ MVC
✓ Web API

Advanced
✓ Configure for HTTPS
☐ Docker support
(Requires Docker Desktop)
☐ Also create a project for unit tests
WebAPI_Test.Tests

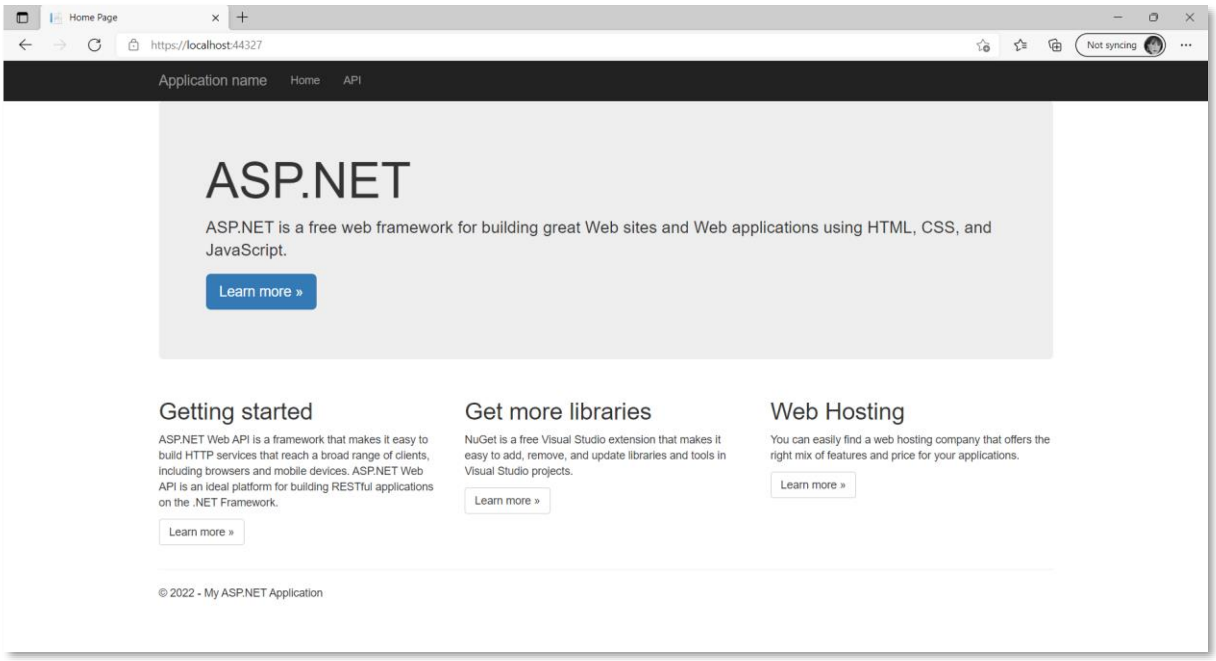
In the ‘**Create a new ASP.NET Web Application**’ page, choose ‘**WebAPI**’ template[9]. And leave the **Authentication** with ‘**No Authentication**’ as well as others setting remain as default[10]. The authentication is another huge topic to discuss. Click ‘**Create**’ button[11] to start create your project.



```
15:20:26.693 [Information] Avalonia Package initialized
```

Visual Studio will takes awhile to generate your project files. Once it is done, you should see something similar to the screenshot above.

Now, try to run your first ASP.NET application by click on the ‘**Run**’ button[12] as shown in the screenshot above or press ‘**F5**’ to start the Web Application.



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Getting started

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework.

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

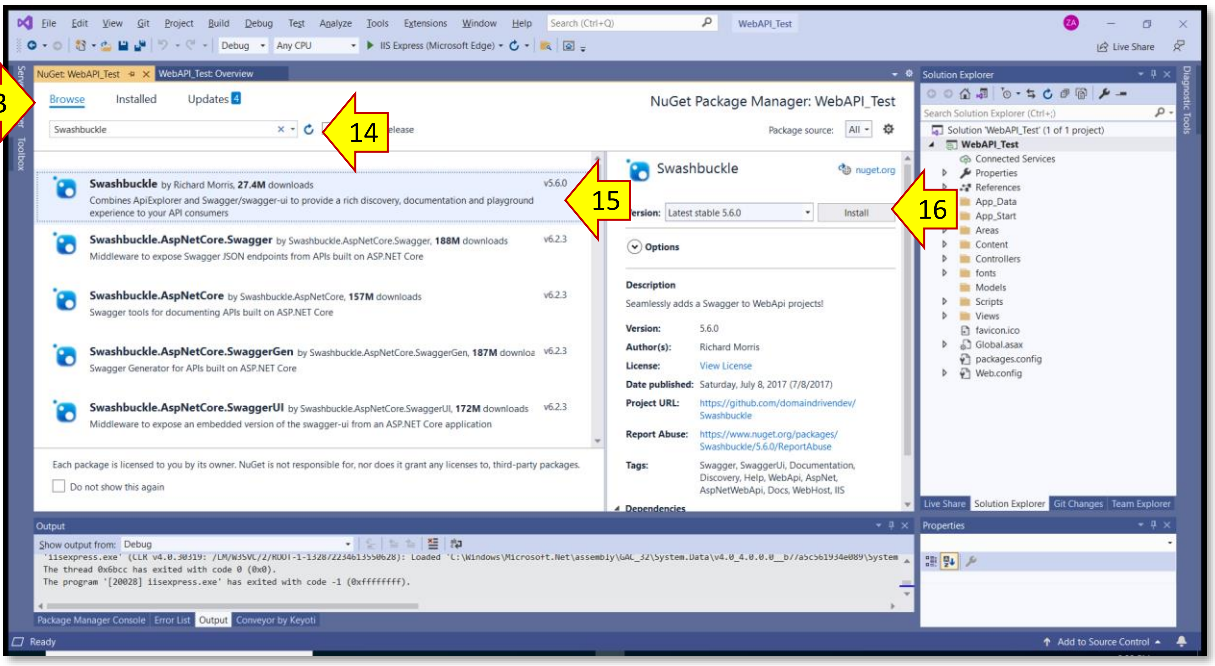
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

© 2022 - My ASP.NET Application

And if you see something similar shown in the screenshot just above in your internet browser, that’s mean your ASP.NET application is up and run. **Hurray……. Congratulation……………!**

Now, let add the **Swagger** page into your project. Go to the '**NuGet Package Manager**' by right-click on your project in the '**Solution Explorer**', then choose '**Manage NuGet Packages…**'.



Swashbuckle by Richard Morris, **27.4M** downloads v5.6.0
Combines ApiExplorer and Swagger/swagger-ui to provide a rich discovery, documentation and playground experience to your API consumers

Swashbuckle.AspNetCore.Swagger by Swashbuckle.AspNetCore.Swagger, **188M** downloads v6.2.3
Middleware to expose Swagger JSON endpoints from APIs built on ASP.NET Core

Swashbuckle.AspNetCore by Swashbuckle.AspNetCore, **157M** downloads v6.2.3
Swagger tools for documenting APIs built on ASP.NET Core

Swashbuckle.AspNetCore.SwaggerGen by Swashbuckle.AspNetCore.SwaggerGen, **187M** downloa v6.2.3
Swagger Generator for APIs built on ASP.NET Core

Swashbuckle.AspNetCore.SwaggerUI by Swashbuckle.AspNetCore.SwaggerUI, **172M** downloads v6.2.3
Middleware to expose an embedded version of the swagger-ui from an ASP.NET Core application

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again

Swashbuckle nuget.org
Version: Latest stable 5.6.0

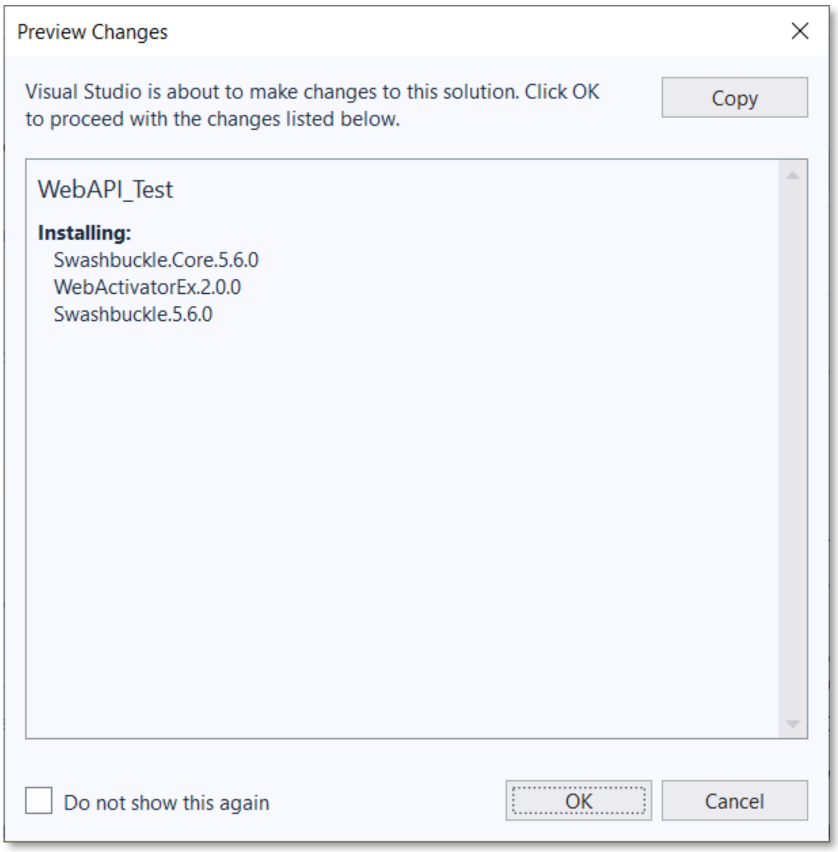
Description
Seamlessly adds a Swagger to WebApi projects!

Version: 5.6.0
Author(s): Richard Morris
License: View License
Date published: Saturday, July 8, 2017 (7/8/2017)
Project URL: https://github.com/domaindrivendev/Swashbuckle
Report Abuse: https://www.nuget.org/packages/Swashbuckle/5.6.0/ReportAbuse
Tags: Swagger, SwaggerUi, Documentation, Discovery, Help, WebApi, AspNet, AspNetWebApi, Docs, WebHost, IIS

Solution 'WebAPI_Test' (1 of 1 project)
WebAPI_Test: Connected Services, Properties, References, App_Data, App_Start, Areas, Content, Controllers, fonts, Models, Scripts, Views, favicon.ico, Global.asax, packages.config, Web.config

```
Show output from: Debug
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132872234613550628): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_32\System.Data\v4.0_4.0.0.0__b77a5c561934e089\System
The thread 0x6bcc has exited with code 0 (0x0).
The program '[20028] iisexpress.exe' has exited with code -1 (0xffffffff).
```

In the '**NuGet Package Manager**', select '**Browse**'[13] tab and search for '**Swashbuckle**'[14]. Then select the '**Swashbuckle**' package[15] to install it by click on the '**Install**' button[16].



Preview Changes

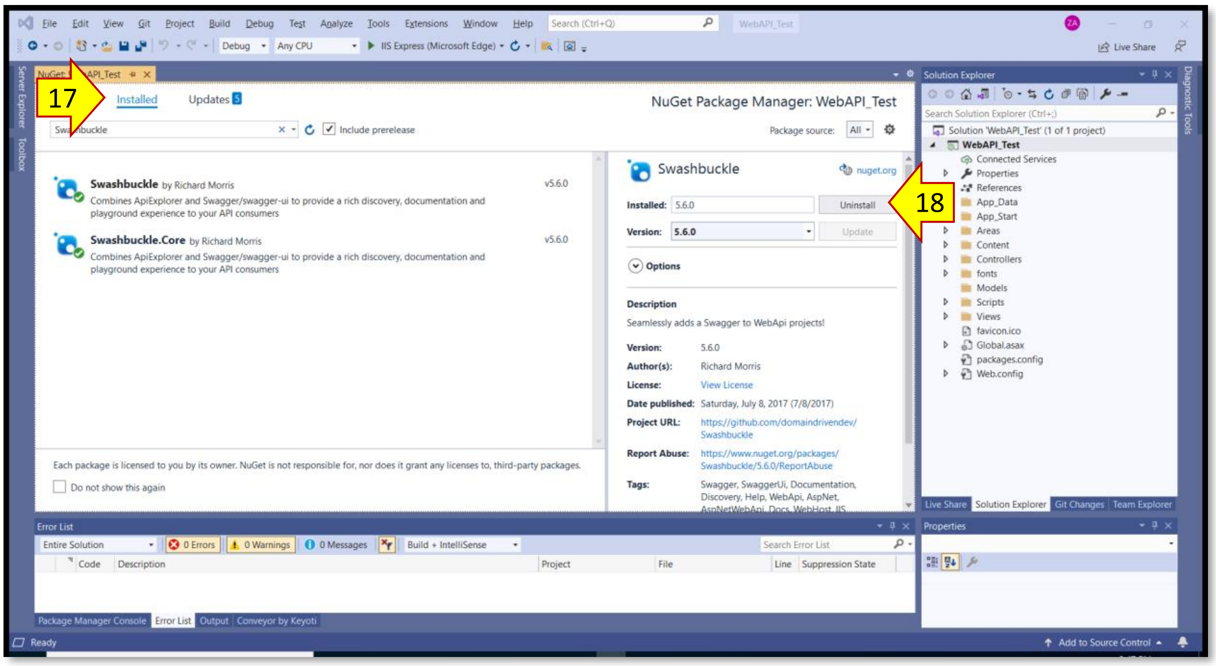
Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

WebAPI_Test

Installing:
Swashbuckle.Core.5.6.0
WebActivatorEx.2.0.0
Swashbuckle.5.6.0

Do not show this again

Visual Studio will prompt you the changes that it is going to make to your project file as shown in the screenshot above. Click ‘**OK**’ to proceed.



NuGet Package Manager: WebAPI_Test

Package source: All

Swashbuckle by Richard Morris v5.6.0
Combines ApiExplorer and Swagger/swagger-ui to provide a rich discovery, documentation and playground experience to your API consumers

Swashbuckle.Core by Richard Morris v5.6.0
Combines ApiExplorer and Swagger/swagger-ui to provide a rich discovery, documentation and playground experience to your API consumers

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

Swashbuckle

Installed: 5.6.0

Version: 5.6.0

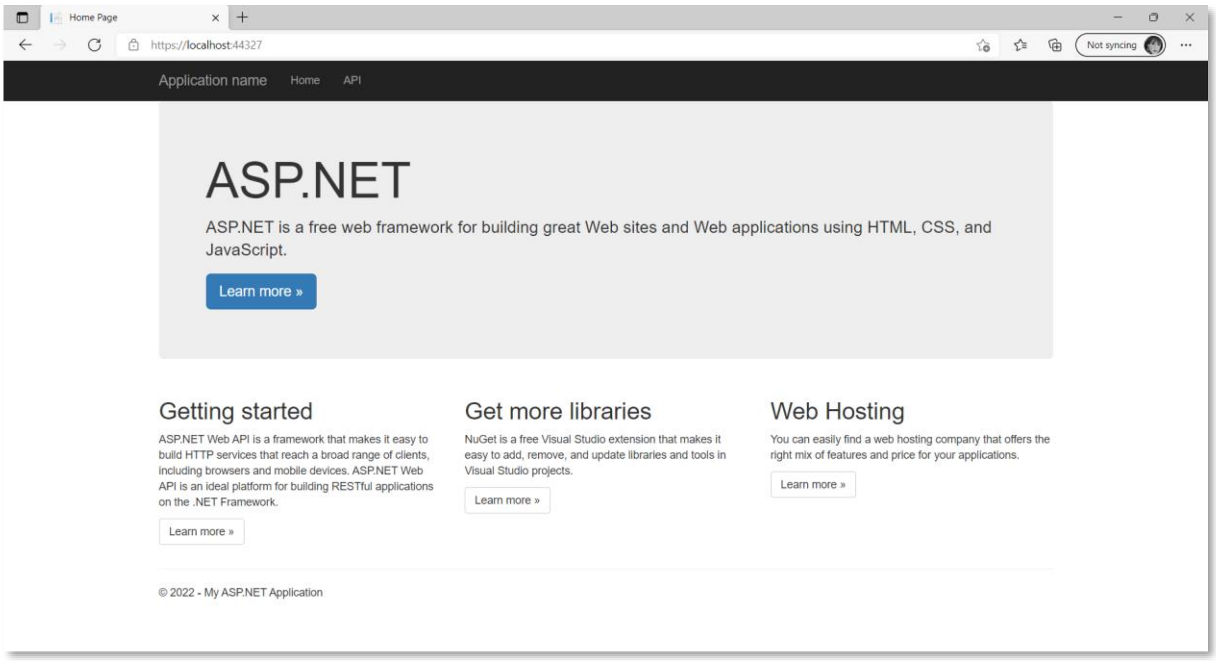
Description
Seamlessly adds a Swagger to WebApi projects!

Version: 5.6.0
Author(s): Richard Morris
License: View License
Date published: Saturday, July 8, 2017 (7/8/2017)
Project URL: https://github.com/domaindrivendev/Swashbuckle
Report Abuse: https://www.nuget.org/packages/Swashbuckle/5.6.0/ReportAbuse
Tags: Swagger, SwaggerUi, Documentation, Discovery, Help, WebApi, AspNet, AspNetWebApi, Docs, WebHost, IIS

Solution 'WebAPI_Test' (1 of 1 project)
WebAPI_Test
Connected Services
Properties
References
App_Data
App_Start
Areas
Content
Controllers
fonts
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config

Once done, switch to ‘**Installed**’ tab[17], the ‘**Swashbuckle**’ should have installed to your project. From here, you can remove it if you don’t like Swagger page later on. Just select the package and then click the ‘**Uninstall**’ button[18] in order to remove it.

Now, you can close the ‘**NuGet Package Manager**’ and then try to run the project again.



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Getting started

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework.

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

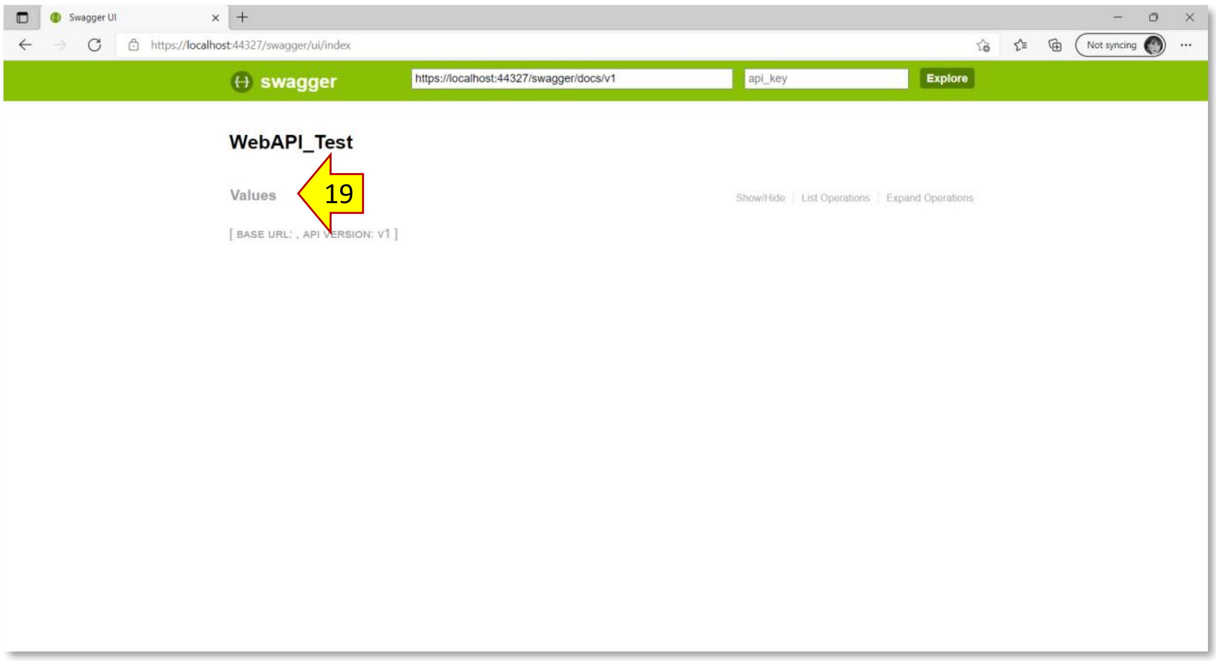
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

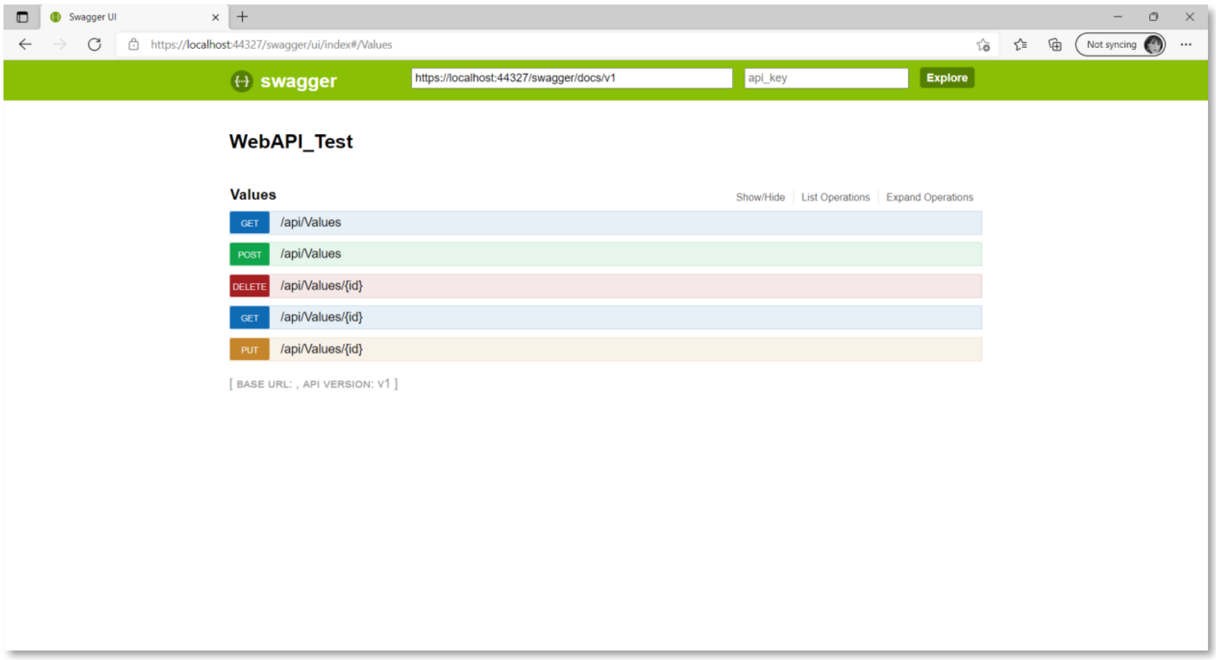
© 2022 - My ASP.NET Application

Well, nothing has been changed. You will see the web page just as before. **Damn…….! Where is the Swagger page?**

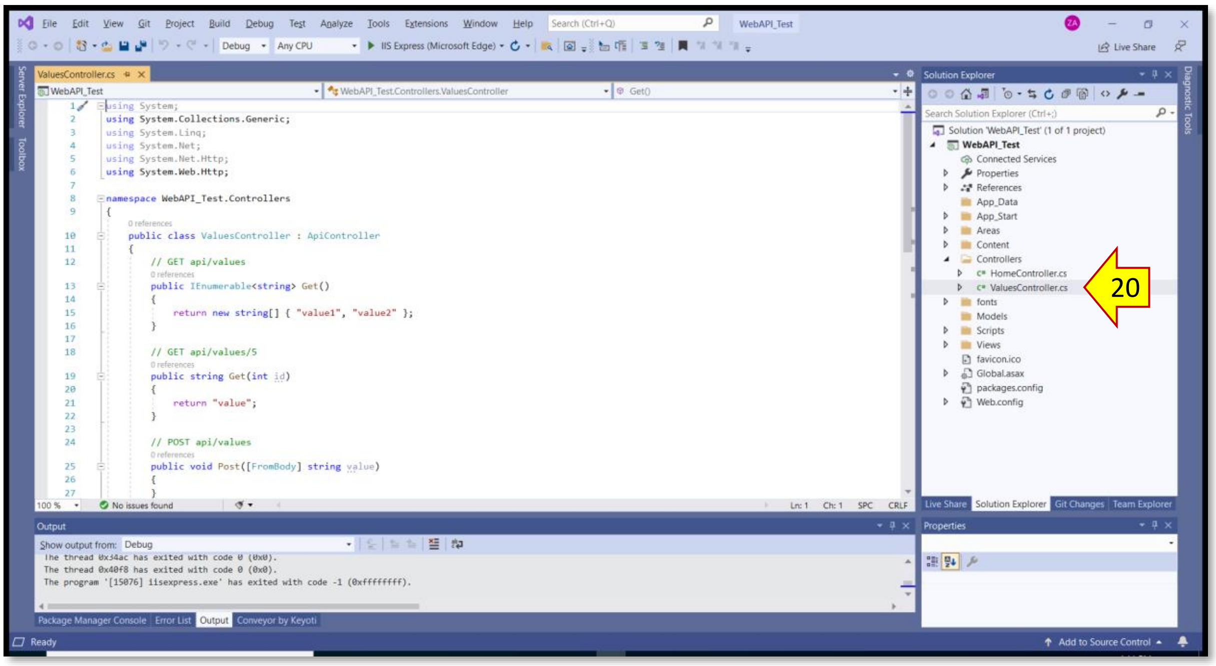
Well, don't panic….! Try to add '**/swagger**' after the '**https://localhost:xxxxx**' at the address bar and then hit '**Enter**' key. **Bomb…. The swagger page is loaded!**


Swagger UI
https://localhost:44327/swagger/ui/index
Not syncing
swagger
https://localhost:44327/swagger/docs/v1
api_key
Explore
WebAPI_Test
Values
19
Show/Hide | List Operations | Expand Operations
[BASE URL: , API VERSION: v1]

Now, expend the '**Values**' API[19]. You notice some endpoints is already exists. But where does the C# code for these endpoints?


Swagger UI
https://localhost:44327/swagger/ui/index#/Values
Not syncing
swagger
https://localhost:44327/swagger/docs/v1
api_key
Explore
WebAPI_Test
Values
Show/Hide | List Operations | Expand Operations
GET /api/Values
POST /api/Values
DELETE /api/Values/{id}
GET /api/Values/{id}
PUT /api/Values/{id}
[BASE URL: , API VERSION: v1]

Go back to Visual Studio……You can also close the browser in order to stop the application or press the '**Stop**' button at Visual Studio to stop it.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace WebAPI_Test.Controllers
{
    0 references
    public class ValuesController : ApiController
    {
        // GET api/values
        0 references
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/values/5
        0 references
        public string Get(int id)
        {
            return "value";
        }

        // POST api/values
        0 references
        public void Post([FromBody] string value)
        {
        }
```

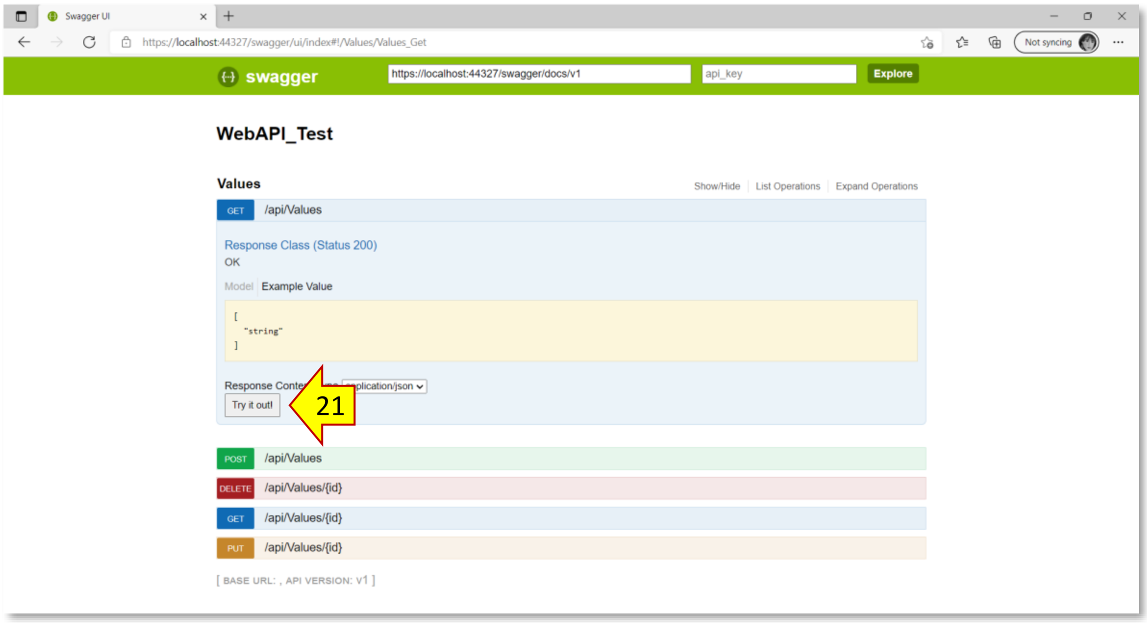
Output — Show output from: Debug

```
The thread 0x34ac has exited with code 0 (0x0).
The thread 0x40f8 has exited with code 0 (0x0).
The program '[15076] iisexpress.exe' has exited with code -1 (0xffffffff).
```

Expend the '**Controllers**' folder[20] in the '**Solution Explorer**' as shown above. And then double-click on the '**ValuesController.cs**' file, all the C# code for the '**Values**' API are located in this C# file.

```
public class ValuesController : ApiController
{
    // GET api/values
    0 references
    public IEnumerable<string> Get()
    {
        return new string[] { "value1", "value2" };
    }
```

Let look at the code above, which is a '**Get**' endpoint. The code in this endpoint is just returning an array of string whenever it is being hit. Now, let examine this by starting up your web application.



WebAPI_Test

Values

GET /api/Values

Response Class (Status 200)
OK

Model Example Value

```
[
  "string"
]
```

POST /api/Values

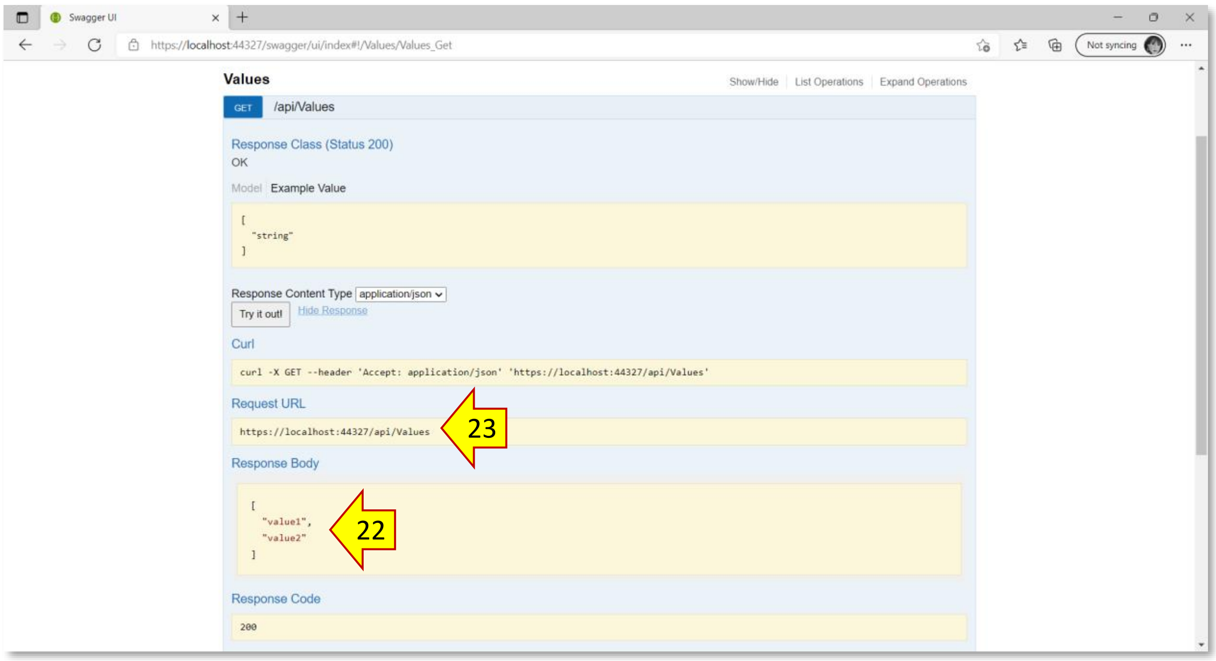
DELETE /api/Values/{id}

GET /api/Values/{id}

PUT /api/Values/{id}

[BASE URL: , API VERSION: v1]

Once the web application is up and run, go to its **swagger page** and then hit the '**Try it out**' button[21] as shown in the screen above.



Values

GET /api/Values

Response Class (Status 200)
OK

Model Example Value

```
[
  "string"
]
```

Response Content Type application/json

Curl

```
curl -X GET --header 'Accept: application/json' 'https://localhost:44327/api/Values'
```

Request URL

```
https://localhost:44327/api/Values
```

Response Body

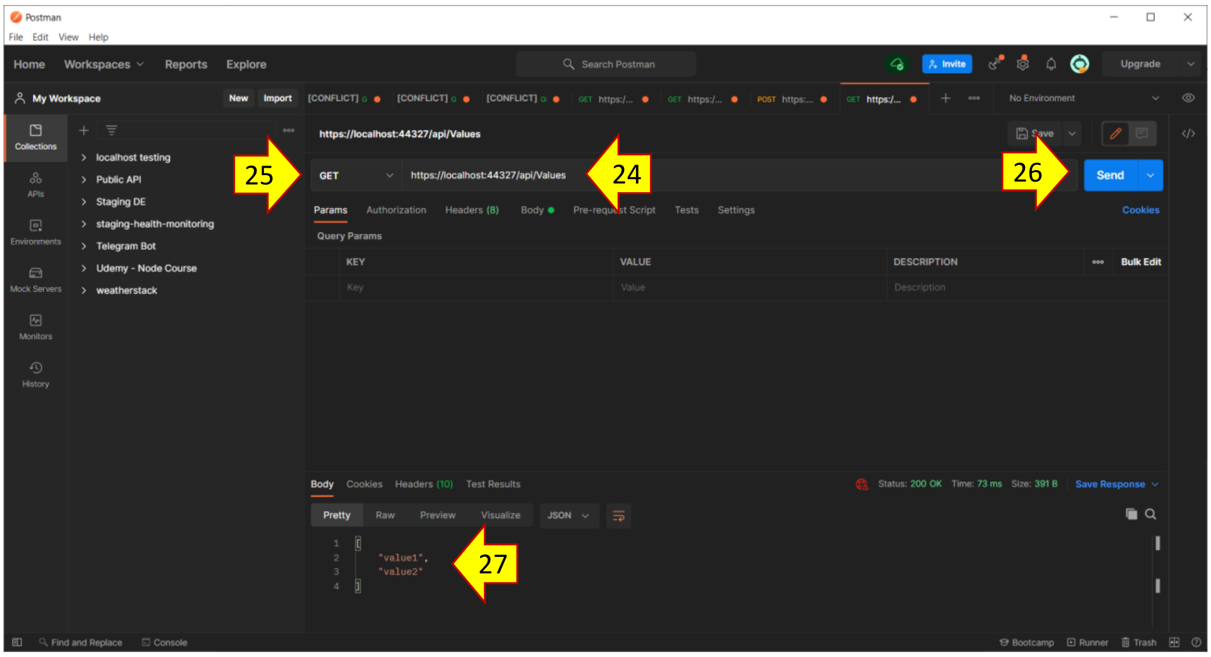
```
[
  "value1",
  "value2"
]
```

Response Code

```
200
```

You will notice that the response body is an **array of string which is in json format**[22]. Compare to the C# code, the response body is exactly what the C# code returns.

While the web application is still running, you can try test the endpoint using '**Postman**'. Before start up your '**Postman**', make a copy of the Request URL[23].



GET https://localhost:44327/api/Values

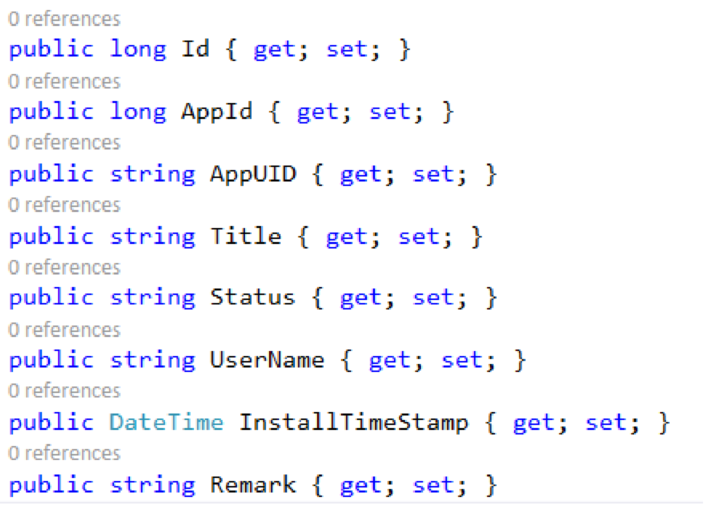
Status: 200 OK Time: 73 ms Size: 391 B

```
[
  "value1",
  "value2"
]
```

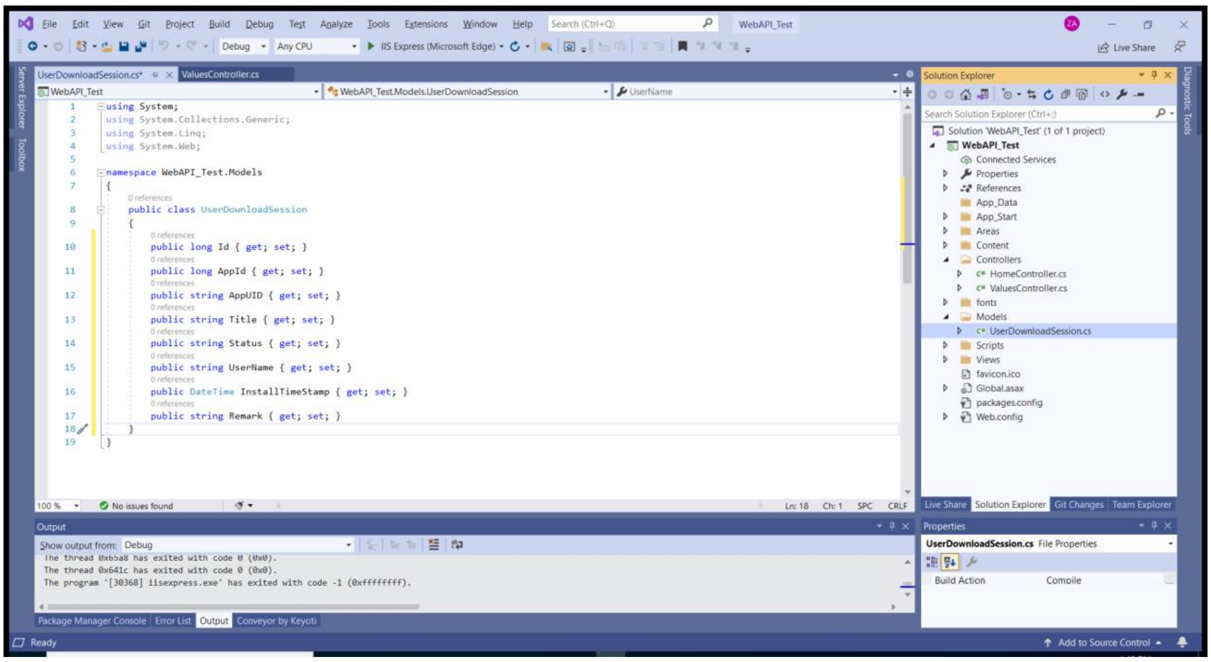
Paste the URL to the address bar[24]. And change the HTTP request to '**Get**'[25] method, then press '**Send**' button[26]. You will notice the response body[27] is same as the swagger page shows to you. This is because when you press the '**Send**' button, the same endpoint is being hit, and the same C# code is running at the backend. Hence you got the same response as in the swagger page.

Now… Let do something interesting! Stop the application if the Web application is still running. Right-click the project in the solution explorer and then create a new folder call ‘**Models**’. If it is already exists, then you can skip this step to create the folder.

Add a new C# class file into the ‘**Models**’ folder and name the file as ‘**UserDownloadSession.cs**’. Then add the following C# code into the file.



```
0 references
public long Id { get; set; }
0 references
public long AppId { get; set; }
0 references
public string AppUID { get; set; }
0 references
public string Title { get; set; }
0 references
public string Status { get; set; }
0 references
public string UserName { get; set; }
0 references
public DateTime InstallTimeStamp { get; set; }
0 references
public string Remark { get; set; }
```



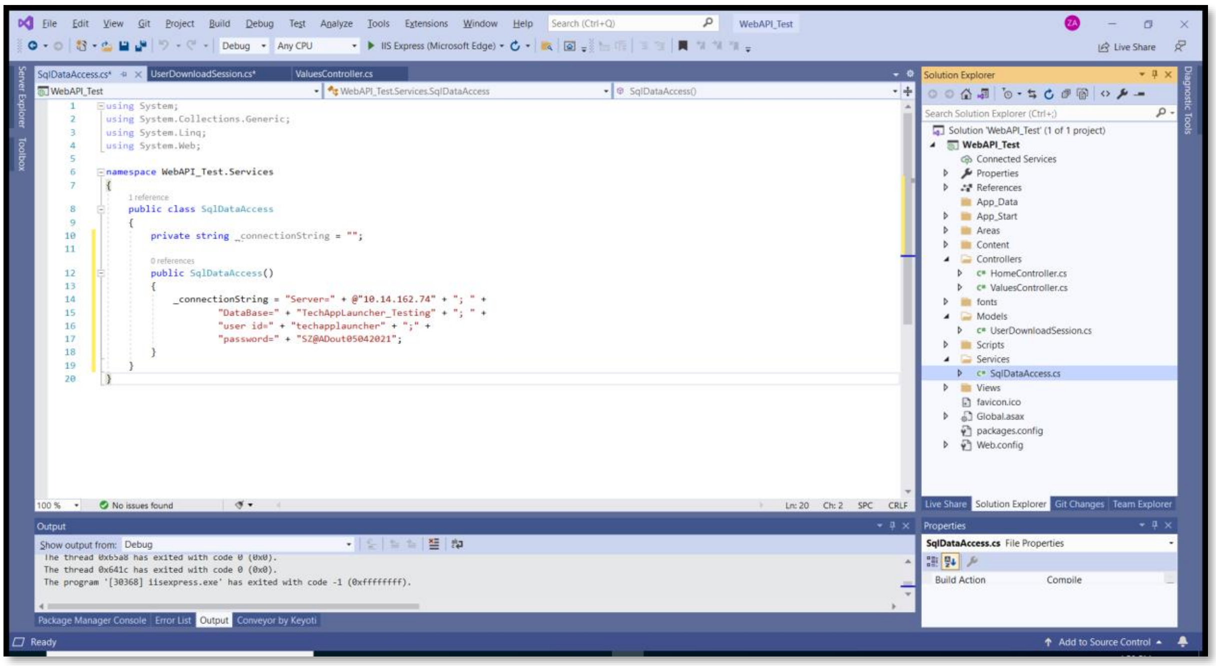
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebAPI_Test.Models
{
    0 references
    public class UserDownloadSession
    {
        0 references
        public long Id { get; set; }
        0 references
        public long AppId { get; set; }
        0 references
        public string AppUID { get; set; }
        0 references
        public string Title { get; set; }
        0 references
        public string Status { get; set; }
        0 references
        public string UserName { get; set; }
        0 references
        public DateTime InstallTimeStamp { get; set; }
        0 references
        public string Remark { get; set; }
    }
}
```

```
The thread 0x65a8 has exited with code 0 (0x0).
The thread 0x641c has exited with code 0 (0x0).
The program '[30368] iisexpress.exe' has exited with code -1 (0xffffffff).
```

Your entire project files should have something similar to the screenshot above.

Next, right-click the project in the solution explorer and then create a new folder call ‘**Services**’. Then add a new C# class file into the ‘**Services**’ folder and name the file as ‘**SqlDataAccess.cs**’. Also, add the following code to the file. Please note that the connection string is redirecting to the testing database ‘**TechAppLauncher_Testing**’.

```
private string _connectionString = "";

0 references
public SqlDataAccess()
{
    _connectionString = "Server=" + @"10.14.162.74" + "; " +
                "DataBase=" + "TechAppLauncher_Testing" + "; " +
                "user id=" + "techapplauncher" + ";" +
                "password=" + "SZ@ADout05042021";
}
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebAPI_Test.Services
{
    1 reference
    public class SqlDataAccess
    {
        private string _connectionString = "";

        0 references
        public SqlDataAccess()
        {
            _connectionString = "Server=" + @"10.14.162.74" + "; " +
                        "DataBase=" + "TechAppLauncher_Testing" + "; " +
                        "user id=" + "techapplauncher" + ";" +
                        "password=" + "SZ@ADout05042021";
        }
    }
}
```

```
The thread 0x65a8 has exited with code 0 (0x0).
The thread 0x641c has exited with code 0 (0x0).
The program '[30368] iisexpress.exe' has exited with code -1 (0xffffffff).
```

Your entire project files now should have something similar to the screenshot above.

Now, copy the following C# code to the '**SqlDataAccess.cs**' file. It is also necessary for you to add the 2 lines of codes below at top of the C# file.

```
using System.Data.SqlClient;
using WebAPI_Test.Models;
```

```
        public List<UserDownloadSession> GetUserDownloadSessions()
        {
            SqlConnection dbConnection = new SqlConnection(_connectionString);
            string _qry = "SELECT * FROM UserDownloadSession ORDER BY 
InstallTimeStamp DESC";

            List<UserDownloadSession> userDownloadSessions = null;

            try
            {
                dbConnection.Open();
                SqlCommand _qrycmd = new SqlCommand(_qry, dbConnection);
                //_qrycmd.ExecuteNonQuery();

                SqlDataReader Reader = _qrycmd.ExecuteReader();

                if (Reader.HasRows)
                {
                    userDownloadSessions = new List<UserDownloadSession>();

                    while (Reader.Read())
                    {
                        userDownloadSessions.Add(new UserDownloadSession
                        {
                            Id = Reader.GetInt64(0),
                            AppId = Reader.IsDBNull(1) ? 0 : Reader.GetInt64(1),
                            AppUID = Reader.GetString(2),
                            Title = Reader.IsDBNull(3) ? "" : 
Reader.GetString(3),
                            Status = Reader.IsDBNull(4) ? "" : 
Reader.GetString(4),
                            UserName = Reader.GetString(5),
                            InstallTimeStamp = Reader.GetDateTime(6),
                            Remark = Reader.IsDBNull(7) ? "" : 
Reader.GetString(7)
                        });
                    }
                }

            }
            catch (Exception Ex)
            {
                string msg = Ex.Message;
            }
            finally
            {
                dbConnection.Close();
            }

            return userDownloadSessions;
        }
```

The code above is actually add a function ‘`GetUserDownloadSessions()`’ which return a list of ‘`UserDownloadSession`’ in the ‘**SqlDataAccess.cs**’ file.

Again… copy the entire code below to the ‘**SqlDataAccess.cs**’ file.

```
        public List<UserDownloadSession> AddUserDownloadSession(UserDownloadSession
userDownloadSession)
        {
            int _ret = 0;

            SqlConnection dbConnection = new SqlConnection(_connectionString);
            string _qry = "INSERT INTO UserDownloadSession VALUES (@AppId, @AppUID,
@Title, @Status, @UserName, @InstallTimeStamp, @Remark)";

            List<UserDownloadSession> userDownloadSessions = null;

            try
            {
                dbConnection.Open();
                SqlCommand _qrycmd = new SqlCommand(_qry, dbConnection);
                _qrycmd.CommandText = _qry;
                _qrycmd.Parameters.AddWithValue("@AppId", userDownloadSession.AppId);
                _qrycmd.Parameters.AddWithValue("@AppUID",
userDownloadSession.AppUID);
                _qrycmd.Parameters.AddWithValue("@Title", userDownloadSession.Title);
                _qrycmd.Parameters.AddWithValue("@Status",
userDownloadSession.Status);
                _qrycmd.Parameters.AddWithValue("@UserName",
userDownloadSession.UserName);
                _qrycmd.Parameters.AddWithValue("@InstallTimeStamp",
userDownloadSession.InstallTimeStamp);
                _qrycmd.Parameters.AddWithValue("@Remark",
userDownloadSession.Remark);
                //_qrycmd.ExecuteNonQuery();

                SqlDataReader Reader = _qrycmd.ExecuteReader();
                _ret = Reader.RecordsAffected;

                if (_ret > 0)
                {
                    userDownloadSessions = GetUserDownloadSessions();
                }
            }
            catch (Exception Ex)
            {
                string msg = Ex.Message;
            }
            finally
            {
                dbConnection.Close();
            }

            return userDownloadSessions;
        }
```

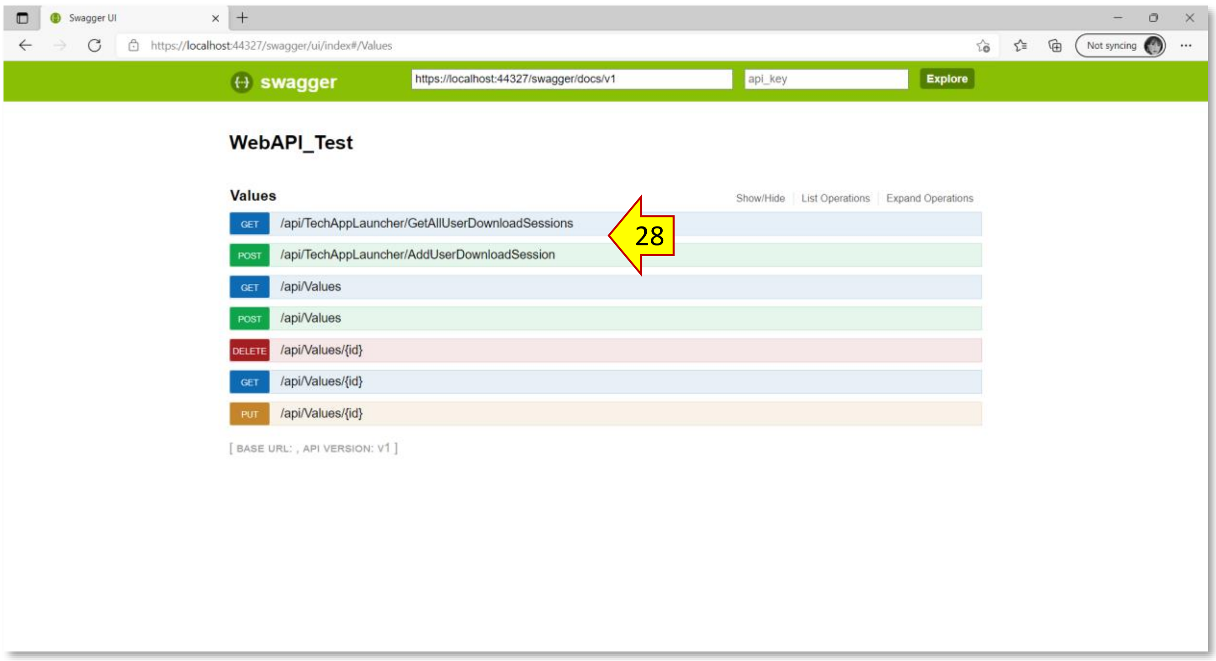
Last but not least, add the following code into the ‘**ValuesController.cs**’ file.

```
        private SqlDataAccess _sqlDataAccess;

        public ValuesController()
        {
            _sqlDataAccess = new SqlDataAccess();
        }

        [HttpGet]
        [Route("api/TechAppLauncher/GetAllUserDownloadSessions")]
        [SwaggerResponse(HttpStatusCode.OK, "Returns the list of records when success. 
Otherwise returns null.")]
        [SwaggerResponse(HttpStatusCode.InternalServerError, "Bad Request - Invalid 
Data!")]
        public IEnumerable<UserDownloadSession> GetAllUserDownloadSessions()
        {
            return _sqlDataAccess.GetUserDownloadSessions();
        }


        [HttpPost]
        [Route("api/TechAppLauncher/AddUserDownloadSession")]
        [SwaggerResponse(HttpStatusCode.OK, "Returns the specific record when success. 
Otherwise returns null.")]
        [SwaggerResponse(HttpStatusCode.InternalServerError, "Bad Request - Invalid 
Data!")]
        public IEnumerable<UserDownloadSession> AddUserDownloadSession([FromBody] 
UserDownloadSession userDownloadSession)
        {
            return _sqlDataAccess.AddUserDownloadSession(userDownloadSession);
        }
```

You may facing a number of errors after copy the code into the controller file. What you need to do is just add the 3 lines of code below to the top of the file.

```
using WebAPI_Test.Models;
using WebAPI_Test.Services;
using Swashbuckle.Swagger.Annotations;
```

Now, you can start the Web Application. And then go to the swagger page.


Swagger UI
https://localhost:44327/swagger/ui/index#/Values
Not syncing
swagger
https://localhost:44327/swagger/docs/v1
api_key
Explore
28

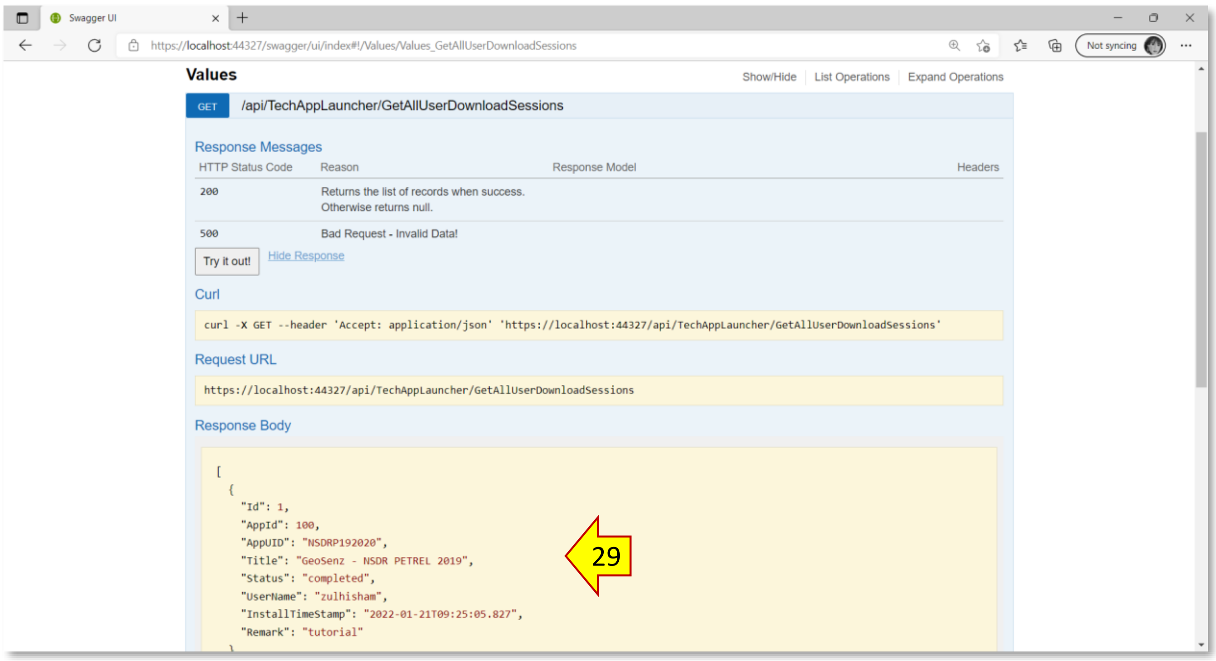
WebAPI_Test

Values — Show/Hide | List Operations | Expand Operations

GET	/api/TechAppLauncher/GetAllUserDownloadSessions
POST	/api/TechAppLauncher/AddUserDownloadSession
GET	/api/Values
POST	/api/Values
DELETE	/api/Values/{id}
GET	/api/Values/{id}
PUT	/api/Values/{id}

[BASE URL: , API VERSION: v1]

You will notice that there are 2 newly added endpoints[28] as shown in the screen shot above. You can expend it and then click on the '**Try it out**' button in order to test the '**/api/TechAppLauncher/GetAllUserDownloadSessions**' API. It will returns a list of '**UserDownloadSession**' in the response body[29].



Values

GET /api/TechAppLauncher/GetAllUserDownloadSessions

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Returns the list of records when success. Otherwise returns null.		
500	Bad Request - Invalid Data!		

Curl

```
curl -X GET --header 'Accept: application/json' 'https://localhost:44327/api/TechAppLauncher/GetAllUserDownloadSessions'
```

Request URL

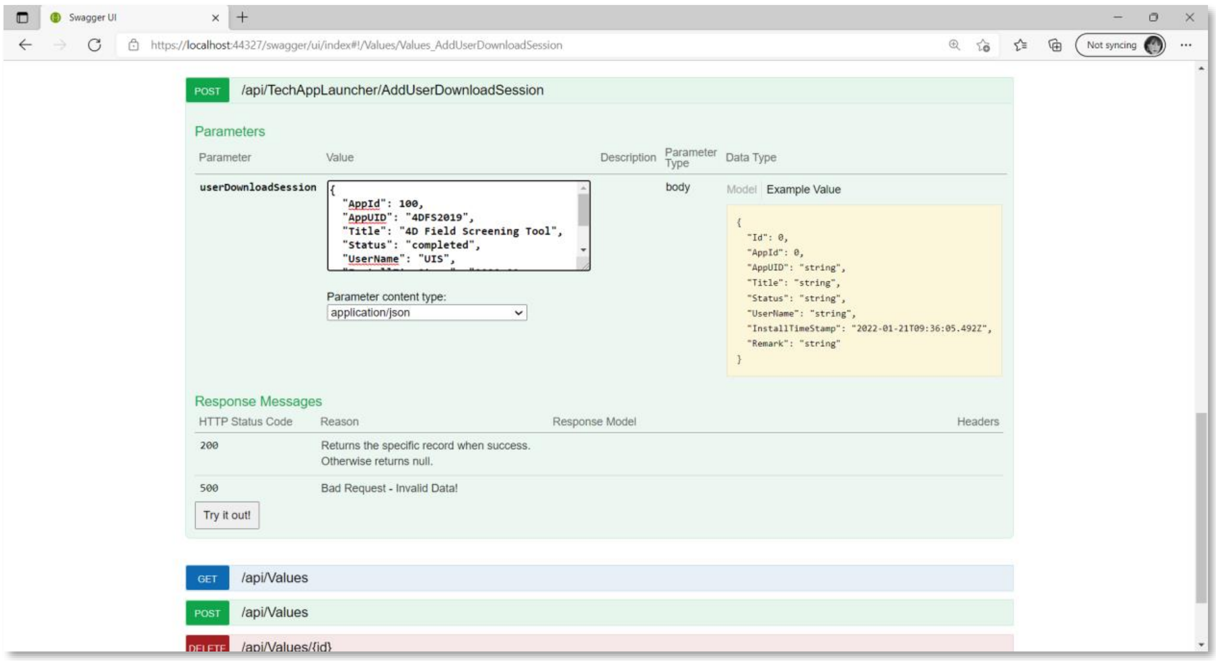
```
https://localhost:44327/api/TechAppLauncher/GetAllUserDownloadSessions
```

Response Body

```
[
  {
    "Id": 1,
    "AppId": 100,
    "AppUID": "NSDRP192020",
    "Title": "GeoSenz - NSDR PETREL 2019",
    "Status": "completed",
    "UserName": "zulhisham",
    "InstallTimeStamp": "2022-01-21T09:25:05.827",
    "Remark": "tutorial"
```

To add a new record… Expend the '**/api/TechAppLauncher/AddUserDownloadSession**' API, then add the following payload as a **parameter of** '**userDownloadSession**'.

```
{
  "AppId": 100,
  "AppUID": "NSDRP192020",
  "Title": "GeoSenz - NSDR PETREL 2019",
  "Status": "completed",
  "UserName": "zulhisham",
  "InstallTimeStamp": "2022-01-21T09:25:05.828Z",
  "Remark": "tutorial"
}
```



https://localhost:44327/swagger/ui/index#!/Values/Values_AddUserDownloadSession

POST /api/TechAppLauncher/AddUserDownloadSession

Parameters

Parameter	Value	Description	Parameter Type	Data Type
userDownloadSession	{ "AppId": 100, "AppUID": "4DFS2019", "Title": "4D Field Screening Tool", "Status": "completed", "UserName": "UIS",		body	Model / Example Value

```
{
  "AppId": 100,
  "AppUID": "4DFS2019",
  "Title": "4D Field Screening Tool",
  "Status": "completed",
  "UserName": "UIS",
```

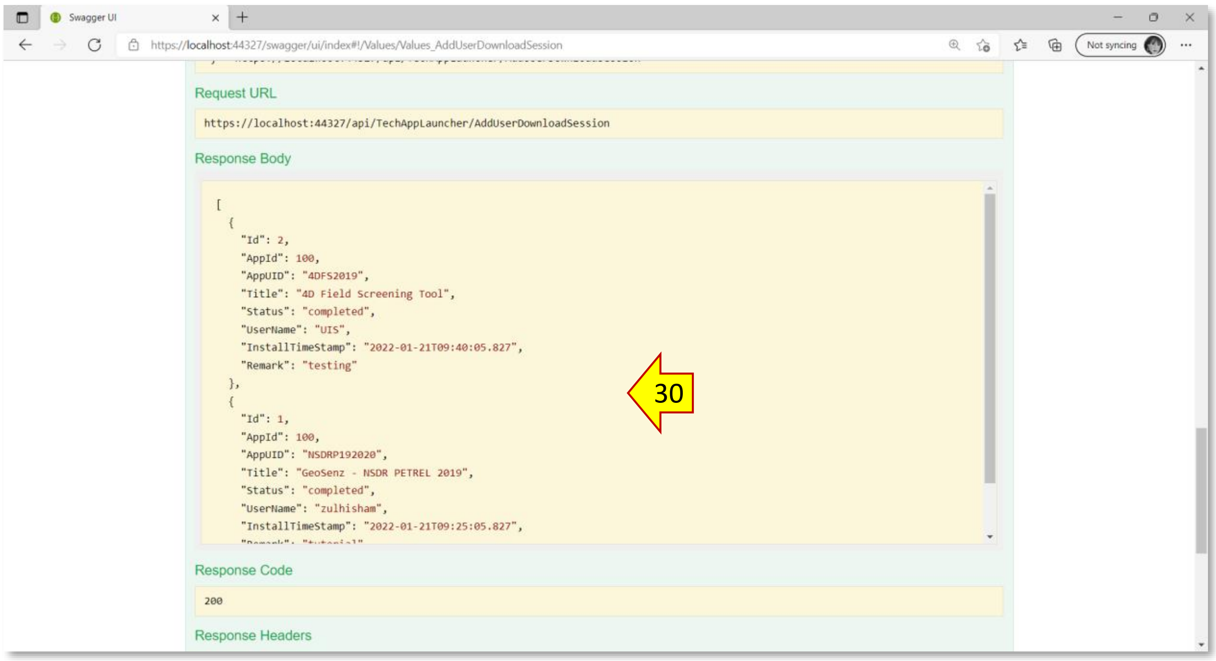
Parameter content type: application/json

```
{
  "Id": 0,
  "AppId": 0,
  "AppUID": "string",
  "Title": "string",
  "Status": "string",
  "UserName": "string",
  "InstallTimeStamp": "2022-01-21T09:36:05.492Z",
  "Remark": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Returns the specific record when success. Otherwise returns null.		
500	Bad Request - Invalid Data!		

Then, click on the '**Try it out**' button. You should be able to see that more and more records being added to the database as it is listed in the response body[30].


Swagger UI
https://localhost:44327/swagger/ui/index#!/Values/Values_AddUserDownloadSession
Not syncing
30

Request URL

```
https://localhost:44327/api/TechAppLauncher/AddUserDownloadSession
```

Response Body

```
[
  {
    "Id": 2,
    "AppId": 100,
    "AppUID": "4DFS2019",
    "Title": "4D Field Screening Tool",
    "Status": "completed",
    "UserName": "UIS",
    "InstallTimeStamp": "2022-01-21T09:40:05.827",
    "Remark": "testing"
  },
  {
    "Id": 1,
    "AppId": 100,
    "AppUID": "NSDRP192020",
    "Title": "GeoSenz - NSDR PETREL 2019",
    "Status": "completed",
    "UserName": "zulhisham",
    "InstallTimeStamp": "2022-01-21T09:25:05.827",
```

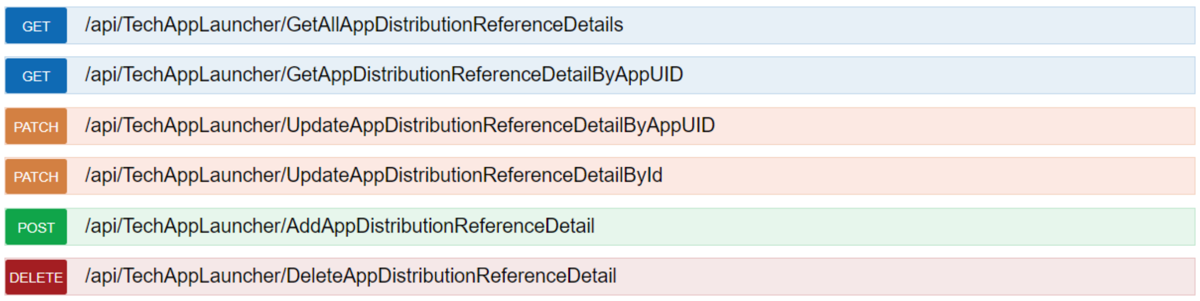
Response Code

```
200
```

Response Headers

DSG – AppDistributionReferenceDetail

For DSG, it engages the same database of **Software Center**. Hence, it uses the following endpoints to manipulate the record in the database.



GET	/api/TechAppLauncher/GetAllAppDistributionReferenceDetails
GET	/api/TechAppLauncher/GetAppDistributionReferenceDetailByAppUID
PATCH	/api/TechAppLauncher/UpdateAppDistributionReferenceDetailByAppUID
PATCH	/api/TechAppLauncher/UpdateAppDistributionReferenceDetailById
POST	/api/TechAppLauncher/AddAppDistributionReferenceDetail
DELETE	/api/TechAppLauncher/DeleteAppDistributionReferenceDetail

You should refer to the payload below as the parameter for these endpoints. The variable highlighted in ***bold italic*** is one that you need to change according to the plugin.

{

"AppUID": "***L1FDDSG2021***",

"LinkID": "-application org.eclipse.equinox.p2.director -noSplash -repository jar:file:/{pluginFilePath}!/ -installIUs {pluginUI}",

"AgentName": "dsg",

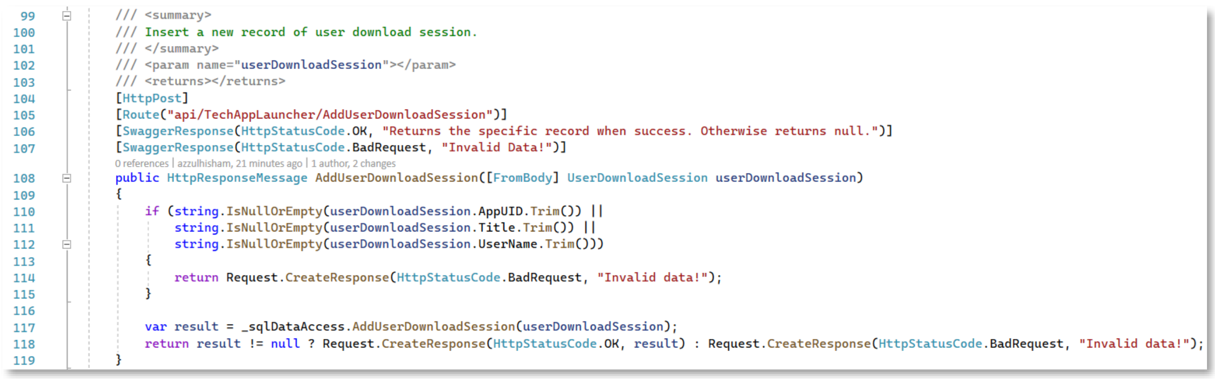
"Description": "***C:\\Users\\zulhisham\\eclipse\\committers-2021-033\\eclipse\\eclipsec.exe***"

}

- **AppUID** : is the ID you should retrieve from TechApp Store portal.
- **LinkID** : is the parameter that the eclipse command will be used.
- **AgentName** : is the name of the software to execute the command.
- **Description** : will be the executable file to execute the command.

Best Practice

It is recommended you shall follow some best practice, refer to the example by comparing the code below with the code above in '**ValuesController.cs**' file. Line 99 to line 103 provides some summary to developer about what the API is all about. It won't appear on the Swagger page. At line 108, instead of returns the data object directly, it is better to return a **HttpResponseMessage** object, so that the user of the API will have more clear information whenever they receive the response from the API. This is because we can create more meaning status code in the response message such as '**OK**', '**Created**', '**Accepted**', '**Bad Gateway**', '**Bad Request**', '**Not Found**' and etc… You can compare the code below at line 114 on how to create a response message with the code above which directly throw an exception.



```
 99  /// <summary>
100  /// Insert a new record of user download session.
101  /// </summary>
102  /// <param name="userDownloadSession"></param>
103  /// <returns></returns>
104  [HttpPost]
105  [Route("api/TechAppLauncher/AddUserDownloadSession")]
106  [SwaggerResponse(HttpStatusCode.OK, "Returns the specific record when success. Otherwise returns null.")]
107  [SwaggerResponse(HttpStatusCode.BadRequest, "Invalid Data!")]
     0 references | azzulhisham, 21 minutes ago | 1 author, 2 changes
108  public HttpResponseMessage AddUserDownloadSession([FromBody] UserDownloadSession userDownloadSession)
109  {
110      if (string.IsNullOrEmpty(userDownloadSession.AppUID.Trim()) ||
111          string.IsNullOrEmpty(userDownloadSession.Title.Trim()) ||
112          string.IsNullOrEmpty(userDownloadSession.UserName.Trim()))
113      {
114          return Request.CreateResponse(HttpStatusCode.BadRequest, "Invalid data!");
115      }
116
117      var result = _sqlDataAccess.AddUserDownloadSession(userDownloadSession);
118      return result != null ? Request.CreateResponse(HttpStatusCode.OK, result) : Request.CreateResponse(HttpStatusCode.BadRequest, "Invalid data!");
119  }
```

By following this best practice, we have a choice to return a status code **400 – Bad Request for 'Http POST'** method as well as return a status code **404 – Not Found for 'Http GET'** method instead of **500 – Internal Server Error** for both of the requests which create confusion to the user of the API. So… Take it as challenge or exercise to change the code in '**ValuesController.cs**' file by follow the best practice.

Well, I hope you've enjoy this short **ASP.NET WebAPI tutorial** | **Zulhisham Tan@2022**.