Web API Token Based Authentication

Ask Question

Gaurav Karroy Date Jan 08, 2021 300.1k 26 22

Files.zip

Introduction

In this article, we will learn how to implement Token Based Authentication in Web APIs to secure the data.

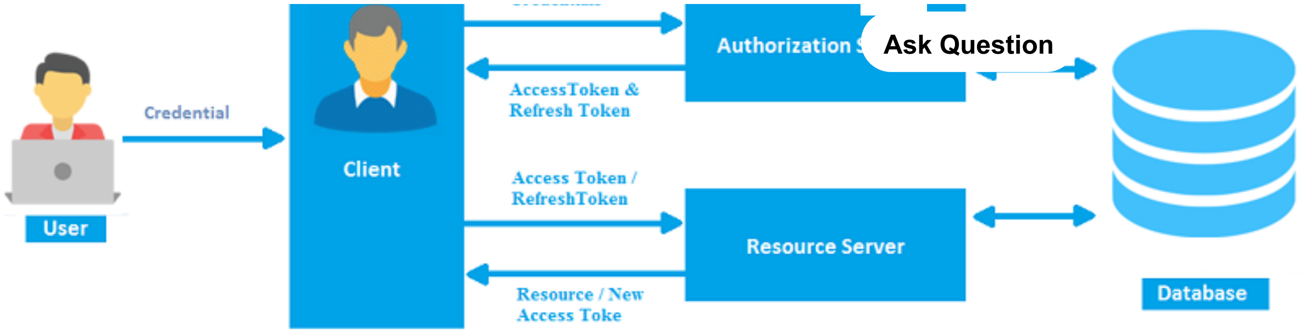
There are 4 common methods of Web API Authentication:

1. HTTP Authentication Schemes (Basic & Bearer)
2. API Keys
3. OAuth (2.0)
4. OpenID Connect

Here we will learn OAuth authentication. OAuth is an open standard for token based authentication and authorization on internet. By using OAuth we can create Token Based Authentication API.

What is Token Based Authentication in Web API?

Token-based authentication is a process where the client application first sends a request to Authentication server with a valid credentials. The Authentication server sends an Access token to the client as a response. This token contains enough data to identify a particular user and it has an expiry time. The client application then uses the token to access the restricted resources in the next requests until the token is valid. If the Access token is expired, then the client application can request for a new access token by using Refresh token.



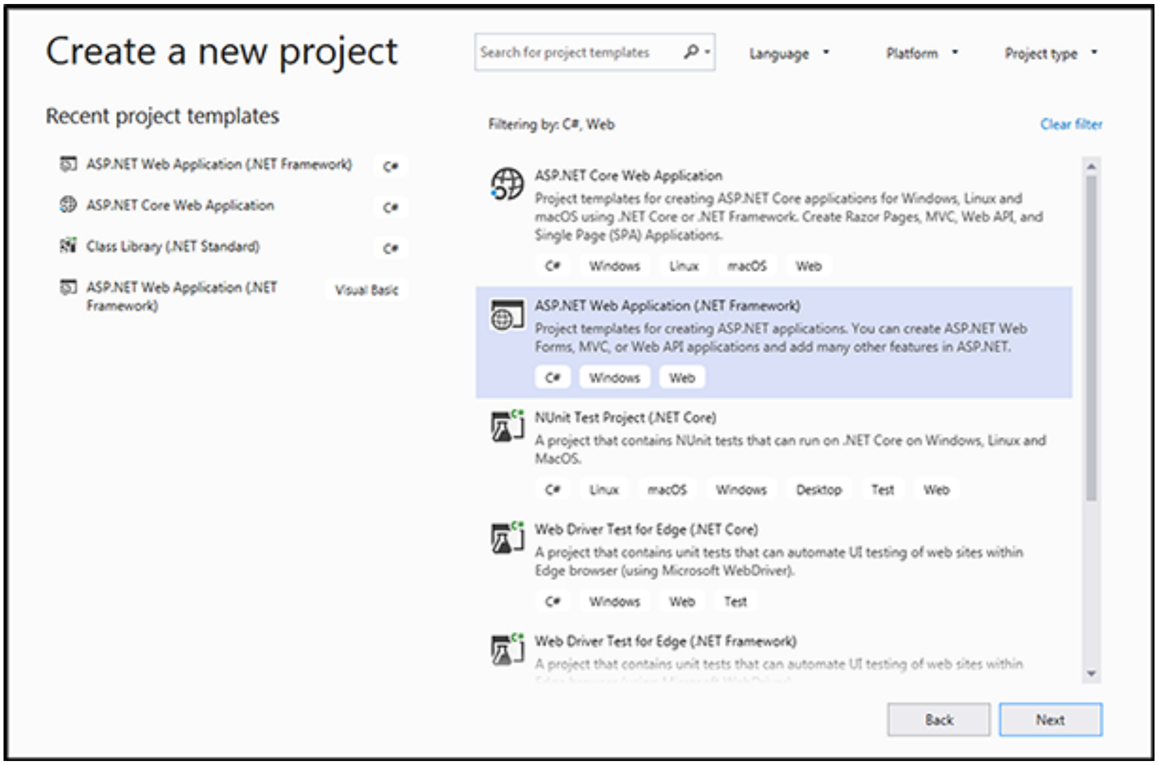
Advantages of Token Based Authentication

- Scalability of Servers
- Loosely Coupling
- Mobile-Friendly

Let’s discuss the step by step procedure to create Token-Based Authentication,

Step 1 - Create ASP.NET Web Project in Visual Studio 2019

We have to create web project in Visual Studio as given in the below image. Choose ASP.Net Web Application from the menu.



Create a new project

Search for project templates

Language ▾ Platform ▾ Project type ▾

Recent project templates

- ASP.NET Web Application (.NET Framework) C#
- ASP.NET Core Web Application C#
- Class Library (.NET Standard) C#
- ASP.NET Web Application (.NET Framework) Visual Basic

Filtering by: C#, Web Clear filter

ASP.NET Core Web Application
Project templates for creating ASP.NET Core applications for Windows, Linux and macOS using .NET Core or .NET Framework. Create Razor Pages, MVC, Web API, and Single Page (SPA) Applications.
C# Windows Linux macOS Web

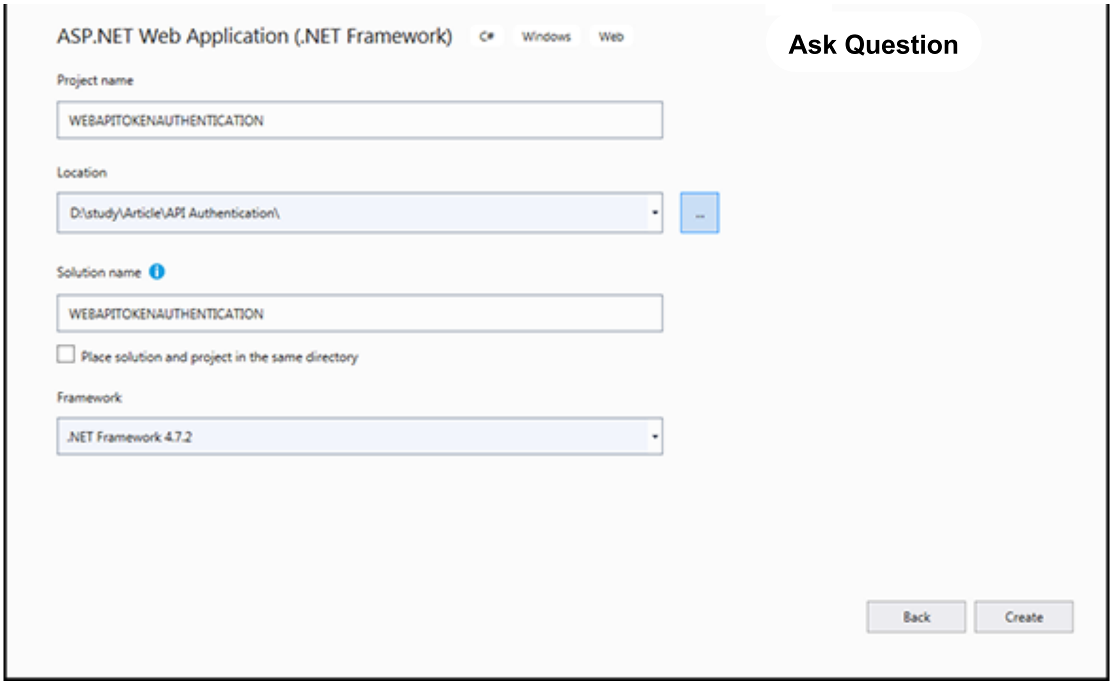
ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Web

NUnit Test Project (.NET Core)
A project that contains NUnit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Desktop Test 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

Give the project name as:WEBAPITOKENAUTHENTICATION.



ASP.NET Web Application (.NET Framework)

Project name: WEBAPITOKENAUTHENTICATION

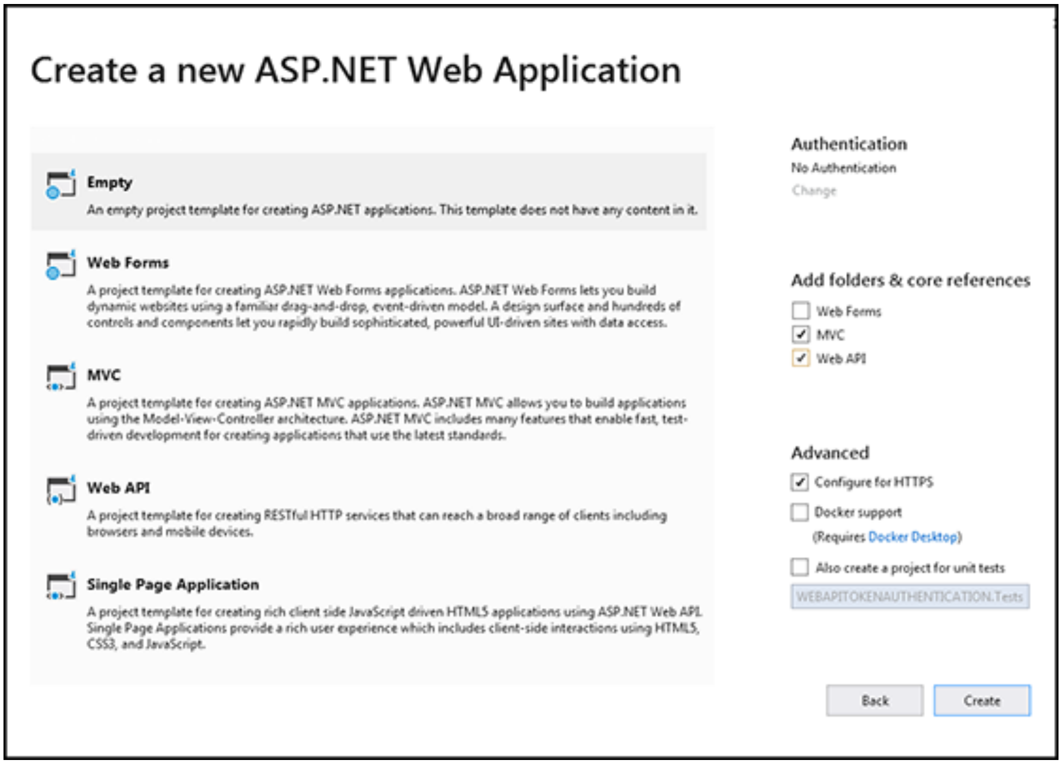
Location: D:\study\Article\API Authentication\

Solution name: WEBAPITOKENAUTHENTICATION

□ Place solution and project in the same directory

Framework: .NET Framework 4.7.2

Now choose the empty template and check the "MVC" and "Web API" on the right hand side.



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.

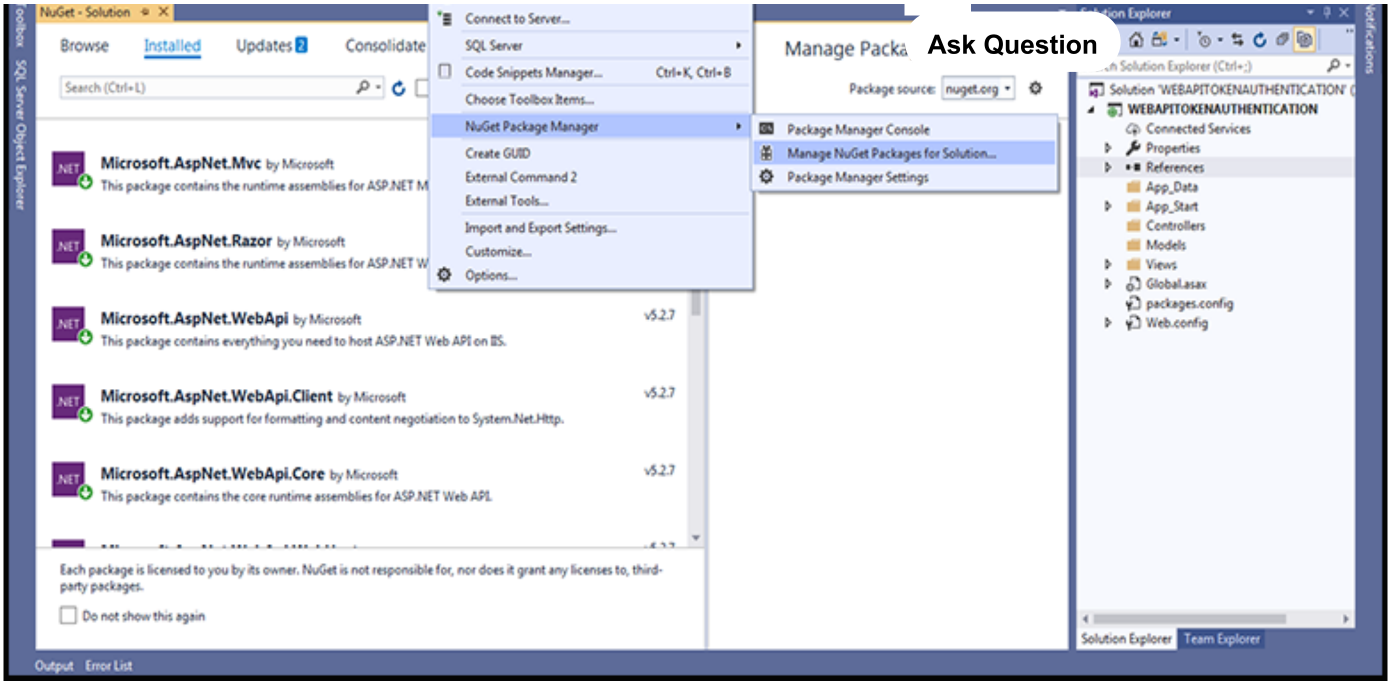
Authentication
No Authentication

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
WEBAPITOKENAUTHENTICATION.Tests

Step 2 - Addition Of References

In this step,we have to add Nuget References like the below image,



Microsoft.AspNet.Mvc by Microsoft
This package contains the runtime assemblies for ASP.NET M

Microsoft.AspNet.Razor by Microsoft
This package contains the runtime assemblies for ASP.NET W

Microsoft.AspNet.WebApi by Microsoft v5.2.7
This package contains everything you need to host ASP.NET Web API on IIS.

Microsoft.AspNet.WebApi.Client by Microsoft v5.2.7
This package adds support for formatting and content negotiation to System.Net.Http.

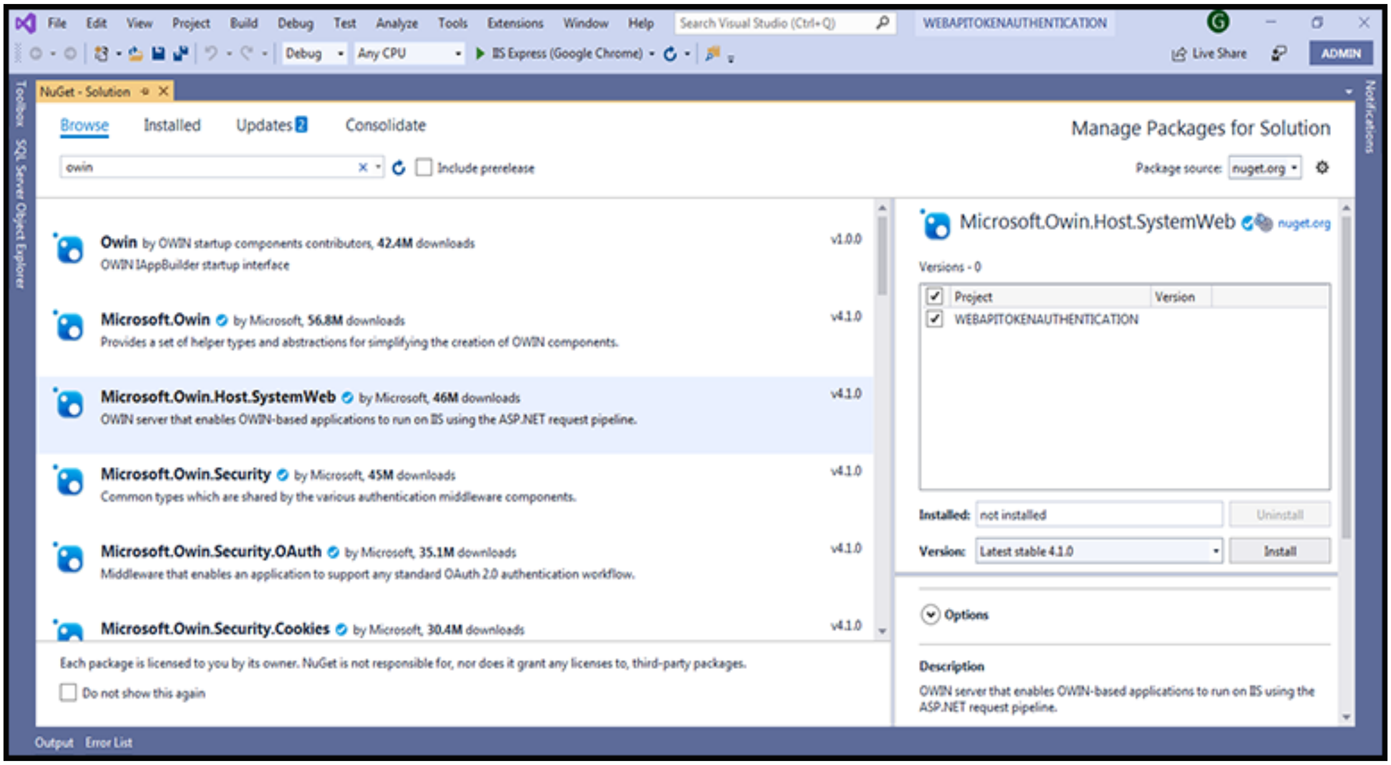
Microsoft.AspNet.WebApi.Core by Microsoft v5.2.7
This package contains the core runtime assemblies for ASP.NET Web API.

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

Here we have to add the following references :

- Microsoft.Owin.Host.SystemWeb
- Microsoft.Owin.Security.OAuth
- Microsoft.Owin.Cors



owin

Owin by OWIN startup components contributors, 42.4M downloads v1.0.0
OWIN IAppBuilder startup interface

Microsoft.Owin by Microsoft, 56.8M downloads v4.1.0
Provides a set of helper types and abstractions for simplifying the creation of OWIN components.

Microsoft.Owin.Host.SystemWeb by Microsoft, 46M downloads v4.1.0
OWIN server that enables OWIN-based applications to run on IIS using the ASP.NET request pipeline.

Microsoft.Owin.Security by Microsoft, 45M downloads v4.1.0
Common types which are shared by the various authentication middleware components.

Microsoft.Owin.Security.OAuth by Microsoft, 35.1M downloads v4.1.0
Middleware that enables an application to support any standard OAuth 2.0 authentication workflow.

Microsoft.Owin.Security.Cookies by Microsoft, 30.4M downloads v4.1.0

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

Microsoft.Owin.Host.SystemWeb nuget.org

Versions - 0

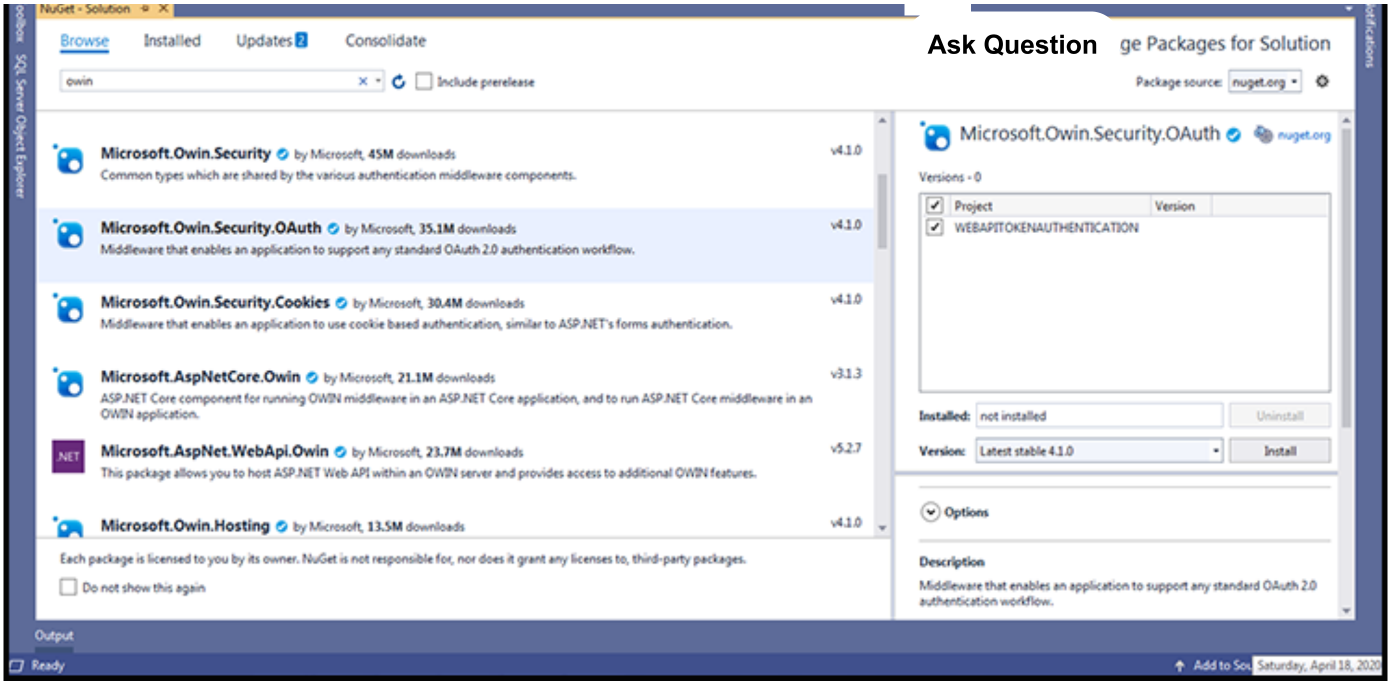
☑ Project	Version
☑ WEBAPITOKENAUTHENTICATION	

Installed: not installed

Version: Latest stable 4.1.0

Description

OWIN server that enables OWIN-based applications to run on IIS using the ASP.NET request pipeline.



Microsoft.Owin.Security by Microsoft, 45M downloads v4.1.0
Common types which are shared by the various authentication middleware components.

Microsoft.Owin.Security.OAuth by Microsoft, 35.1M downloads v4.1.0
Middleware that enables an application to support any standard OAuth 2.0 authentication workflow.

Microsoft.Owin.Security.Cookies by Microsoft, 30.4M downloads v4.1.0
Middleware that enables an application to use cookie based authentication, similar to ASP.NET's forms authentication.

Microsoft.AspNetCore.Owin by Microsoft, 21.1M downloads v3.1.3
ASP.NET Core component for running OWIN middleware in an ASP.NET Core application, and to run ASP.NET Core middleware in an OWIN application.

Microsoft.AspNet.WebApi.Owin by Microsoft, 23.7M downloads v5.2.7
This package allows you to host ASP.NET Web API within an OWIN server and provides access to additional OWIN features.

Microsoft.Owin.Hosting by Microsoft, 13.5M downloads v4.1.0

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

Microsoft.Owin.Security.OAuth nuget.org

Versions - 0

☑ Project	Version
☑ WEBAPITOKENAUTHENTICATION	

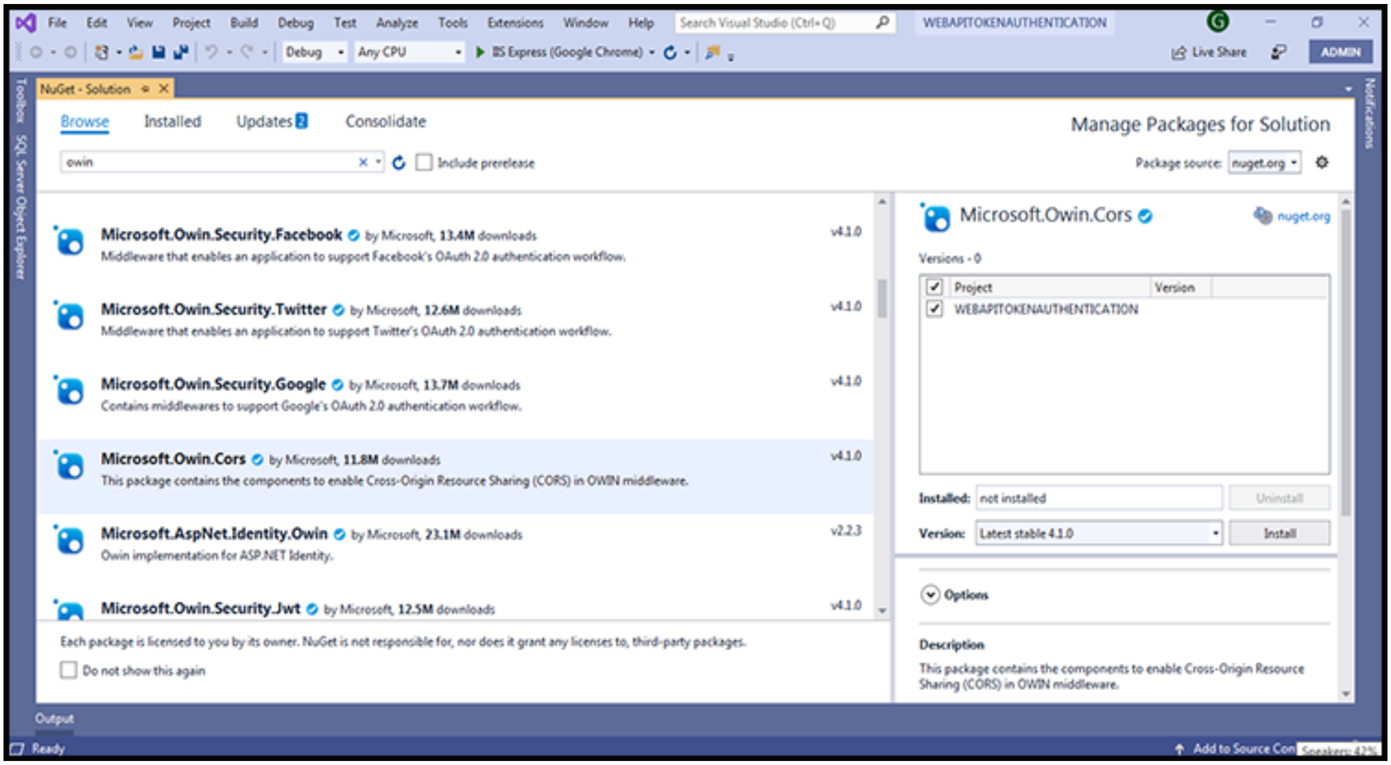
Installed: not installed Uninstall

Version: Latest stable 4.1.0 Install

Options

Description

Middleware that enables an application to support any standard OAuth 2.0 authentication workflow.



Microsoft.Owin.Security.Facebook by Microsoft, 13.4M downloads v4.1.0
Middleware that enables an application to support Facebook's OAuth 2.0 authentication workflow.

Microsoft.Owin.Security.Twitter by Microsoft, 12.6M downloads v4.1.0
Middleware that enables an application to support Twitter's OAuth 2.0 authentication workflow.

Microsoft.Owin.Security.Google by Microsoft, 13.7M downloads v4.1.0
Contains middlewares to support Google's OAuth 2.0 authentication workflow.

Microsoft.Owin.Cors by Microsoft, 11.8M downloads v4.1.0
This package contains the components to enable Cross-Origin Resource Sharing (CORS) in OWIN middleware.

Microsoft.AspNet.Identity.Owin by Microsoft, 23.1M downloads v2.2.3
Owin implementation for ASP.NET Identity.

Microsoft.Owin.Security.Jwt by Microsoft, 12.5M downloads v4.1.0

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

Microsoft.Owin.Cors nuget.org

Versions - 0

☑ Project	Version
☑ WEBAPITOKENAUTHENTICATION	

Installed: not installed Uninstall

Version: Latest stable 4.1.0 Install

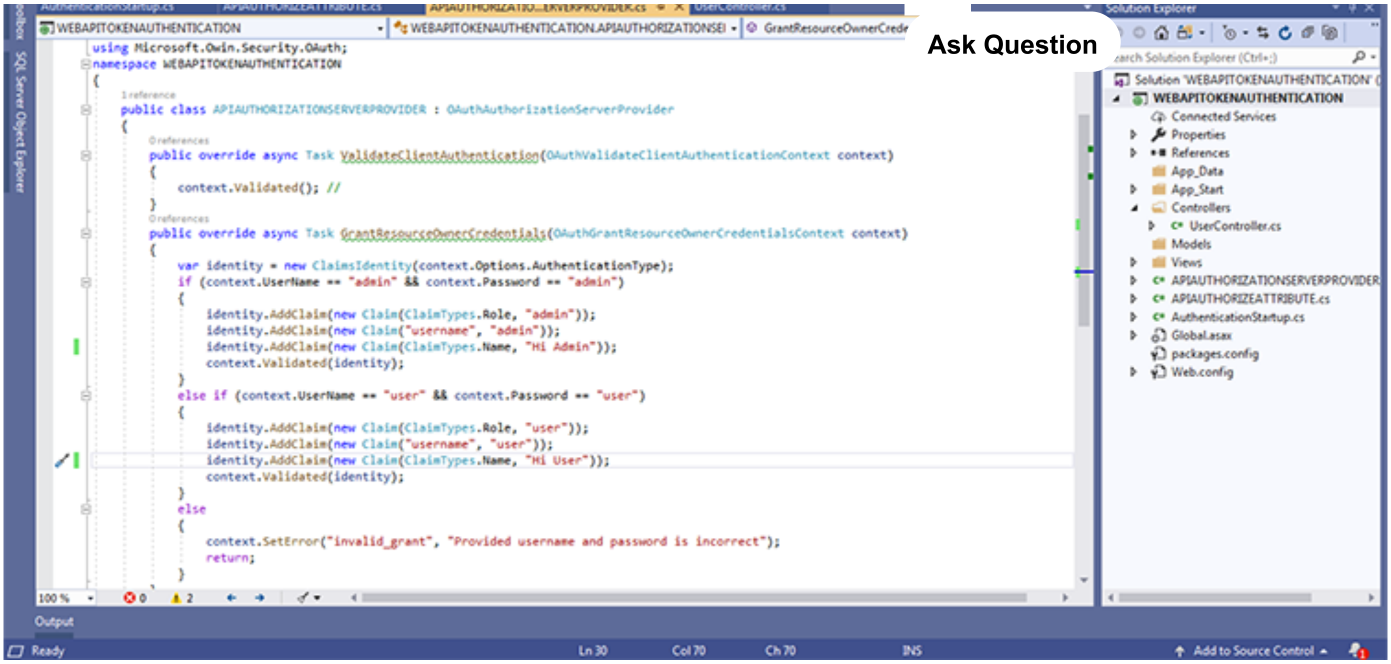
Options

Description

This package contains the components to enable Cross-Origin Resource Sharing (CORS) in OWIN middleware.

Step 3 - Create APIAUTHORIZATIONSERVERPROVIDER.cs Class File

Now, let's create the class file to provide credentials to access data depending on username, password, and roles.



```
using Microsoft.Owin.Security.OAuth;
namespace WEBAPITOKENAUTHENTICATION
{
    1 reference
    public class APIAUTHORIZATIONSERVERPROVIDER : OAuthAuthorizationServerProvider
    {
        0 references
        public override async Task ValidateClientAuthentication(OAuthValidateClientAuthenticationContext context)
        {
            context.Validated(); //
        }
        0 references
        public override async Task GrantResourceOwnerCredentials(OAuthGrantResourceOwnerCredentialsContext context)
        {
            var identity = new ClaimsIdentity(context.Options.AuthenticationType);
            if (context.UserName == "admin" && context.Password == "admin")
            {
                identity.AddClaim(new Claim(ClaimTypes.Role, "admin"));
                identity.AddClaim(new Claim("username", "admin"));
                identity.AddClaim(new Claim(ClaimTypes.Name, "Hi Admin"));
                context.Validated(identity);
            }
            else if (context.UserName == "user" && context.Password == "user")
            {
                identity.AddClaim(new Claim(ClaimTypes.Role, "user"));
                identity.AddClaim(new Claim("username", "user"));
                identity.AddClaim(new Claim(ClaimTypes.Name, "Hi User"));
                context.Validated(identity);
            }
            else
            {
                context.SetError("invalid_grant", "Provided username and password is incorrect");
                return;
            }
```

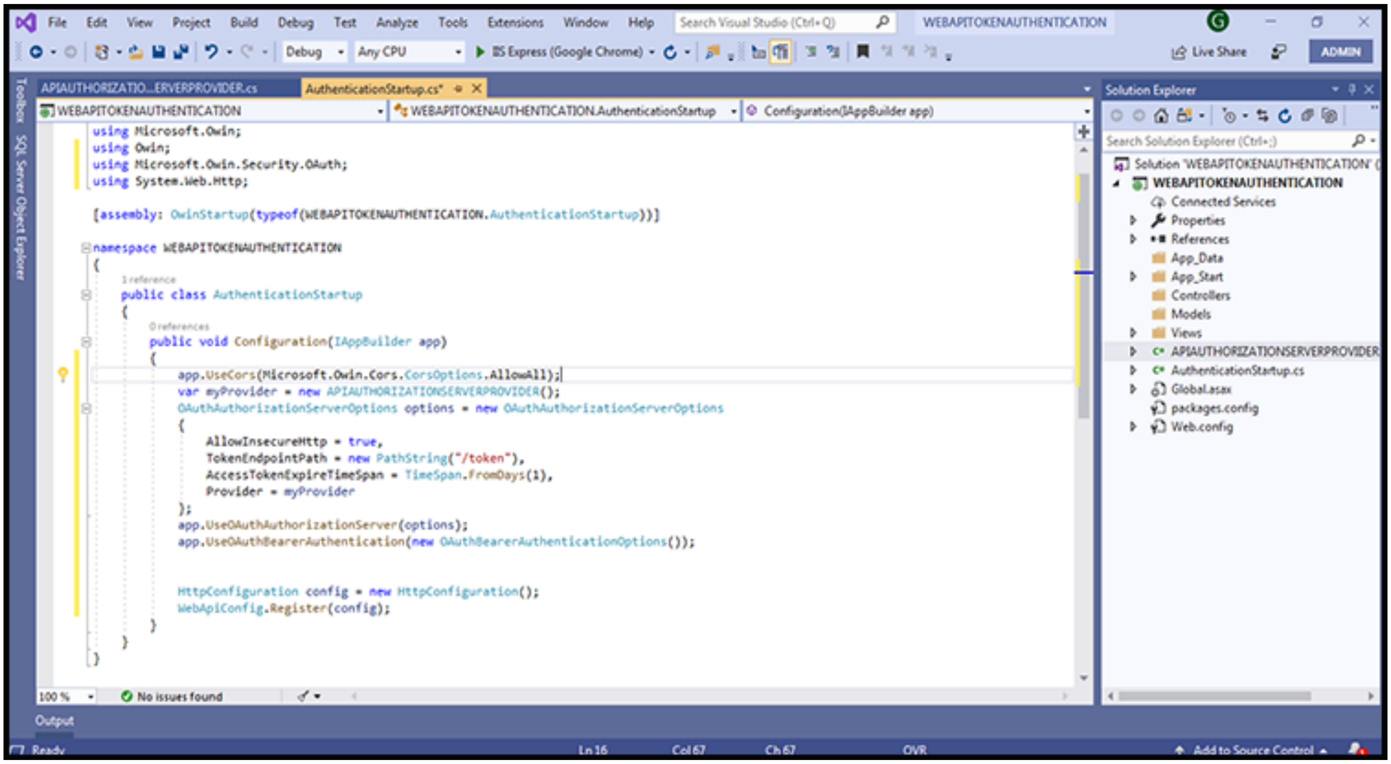
Code is given below:

```
public class APIAUTHORIZATIONSERVERPROVIDER : OAuthAuthorizationServerProvid
{
    public override async Task ValidateClientAuthentication(OAuthValidateCli
    {
        context.Validated(); //
    }
    public override async Task GrantResourceOwnerCredentials(OAuthGrantResou
    {

var identity = new ClaimsIdentity(context.Options.AuthenticationType);
        if (context.UserName == "admin" && context.Password == "admin")
        {
            identity.AddClaim(new Claim(ClaimTypes.Role, "admin"));
            identity.AddClaim(new Claim("username", "admin"));
            identity.AddClaim(new Claim(ClaimTypes.Name, "Hi Admin"));
            context.Validated(identity);
        }
        else if (context.UserName == "user" && context.Password == "user")
        {
            identity.AddClaim(new Claim(ClaimTypes.Role, "user"));
            identity.AddClaim(new Claim("username", "user"));
            identity.AddClaim(new Claim(ClaimTypes.Name, "Hi User"));
            context.Validated(identity);
        }
        else
        {

context.SetError("invalid_grant", "Provided username and password is incorre
            return;
        }
    }
```

Step 4 - Create a AuthenticationStartup.cs Class File

Ask Question

Here, we need to create a new class file to implement the code configuration provider and create an instance of class APIAUTHORIZATIONSERVERPROVIDER.



APIAUTHORIZATIO...ERVERPROVIDER.cs
AuthenticationStartup.cs*
WEBAPITOKENAUTHENTICATION
WEBAPITOKENAUTHENTICATION.AuthenticationStartup
Configuration(IAppBuilder app)
Solution Explorer
Solution 'WEBAPITOKENAUTHENTICATION'
WEBAPITOKENAUTHENTICATION
Connected Services
Properties
References
App_Data
App_Start
Controllers
Models
Views
APIAUTHORIZATIONSERVERPROVIDER
AuthenticationStartup.cs
Global.asax
packages.config
Web.config
No issues found
Ready
Ln 16
Col 67

```
using Microsoft.Owin;
using Owin;
using Microsoft.Owin.Security.OAuth;
using System.Web.Http;

[assembly: OwinStartup(typeof(WEBAPITOKENAUTHENTICATION.AuthenticationStartup))]

namespace WEBAPITOKENAUTHENTICATION
{
    public class AuthenticationStartup
    {
        public void Configuration(IAppBuilder app)
        {
            app.UseCors(Microsoft.Owin.Cors.CorsOptions.AllowAll);
            var myProvider = new APIAUTHORIZATIONSERVERPROVIDER();
            OAuthAuthorizationServerOptions options = new OAuthAuthorizationServerOptions
            {
                AllowInsecureHttp = true,
                TokenEndpointPath = new PathString("/token"),
                AccessTokenExpireTimeSpan = TimeSpan.FromDays(1),
                Provider = myProvider
            };
            app.UseOAuthAuthorizationServer(options);
            app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());


            HttpConfiguration config = new HttpConfiguration();
            WebApiConfig.Register(config);
        }
    }
}
```

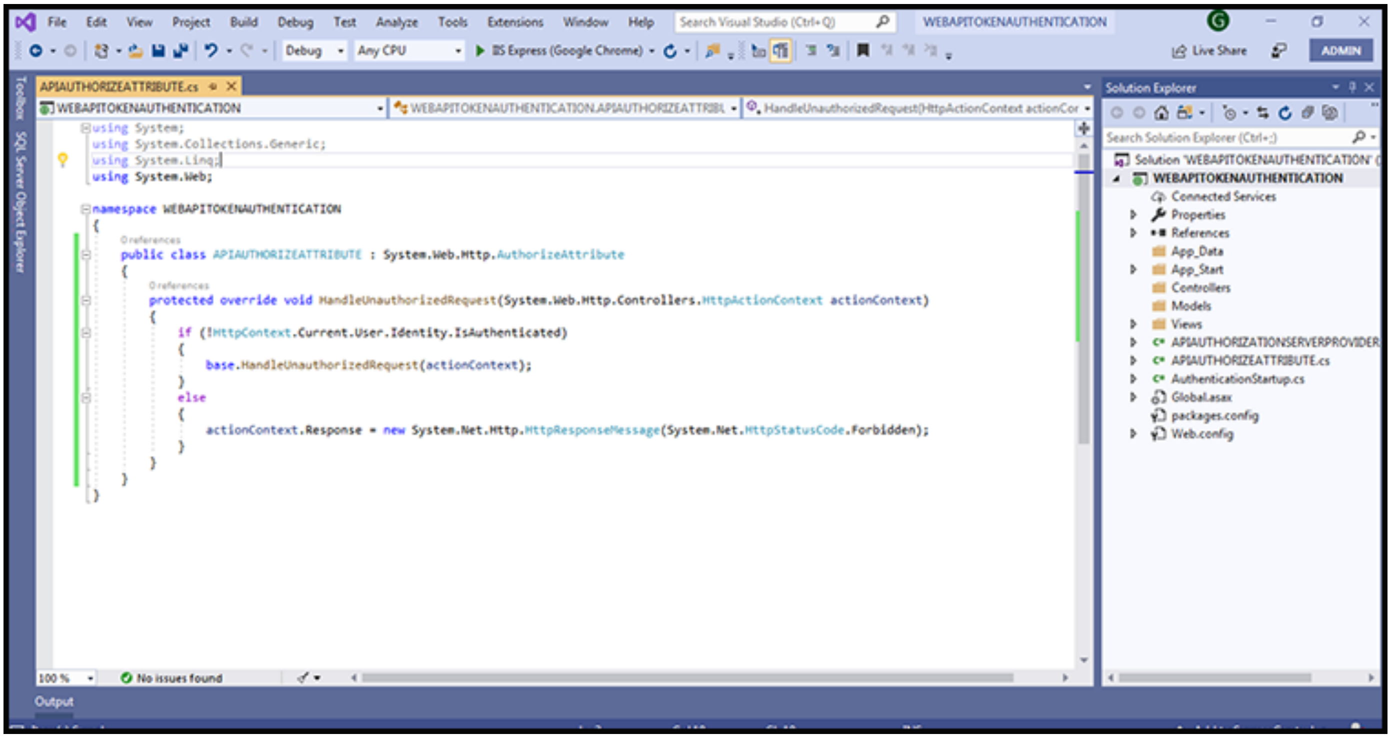
Code is given below:

```
public class AuthenticationStartup
{
    public void Configuration(IAppBuilder app)
    {
        app.UseCors(Microsoft.Owin.Cors.CorsOptions.AllowAll);
        var myProvider = new APIAUTHORIZATIONSERVERPROVIDER();

OAuthAuthorizationServerOptions options = new OAuthAuthorizationServerOption
        {
            AllowInsecureHttp = true,
            TokenEndpointPath = new PathString("/token"),
            AccessTokenExpireTimeSpan = TimeSpan.FromDays(1),
            Provider = myProvider
        };
        app.UseOAuthAuthorizationServer(options);

app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());
        HttpConfiguration config = new HttpConfiguration();
        WebApiConfig.Register(config);
    }
```

Step 5 - Create a APIAUTHORIZEATTRIBUTE.cs Class File

Ask Question

We need to create this class to handle unauthorized access to resources and show the proper message.



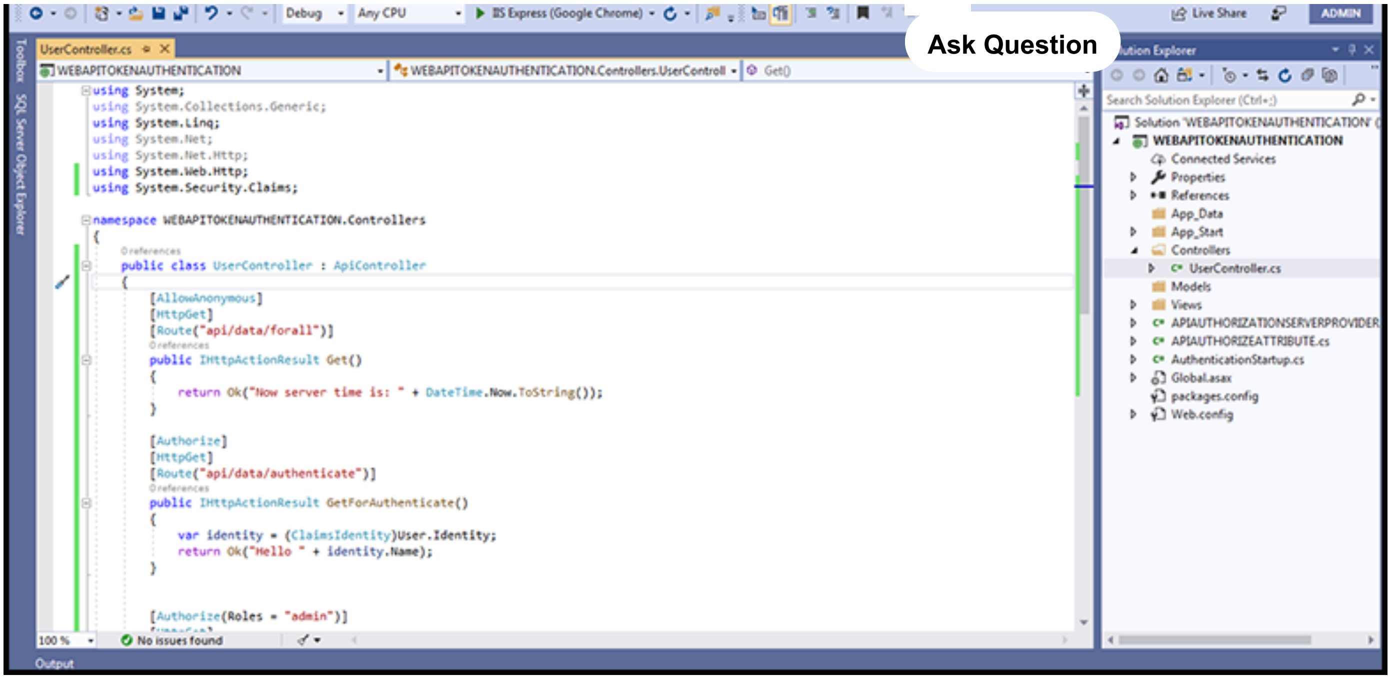
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WEBAPITOKENAUTHENTICATION
{
    0 references
    public class APIAUTHORIZEATTRIBUTE : System.Web.Http.AuthorizeAttribute
    {
        0 references
        protected override void HandleUnauthorizedRequest(System.Web.Http.Controllers.HttpActionContext actionContext)
        {
            if (!HttpContext.Current.User.Identity.IsAuthenticated)
            {
                base.HandleUnauthorizedRequest(actionContext);
            }
            else
            {
                actionContext.Response = new System.Net.Http.HttpResponseMessage(System.Net.HttpStatusCode.Forbidden);
            }
        }
    }
}
```

Code is given below:

```
public class APIAUTHORIZEATTRIBUTE : System.Web.Http.AuthorizeAttribute
{
    protected override void HandleUnauthorizedRequest(System.Web.Http.Contro
    {
        if (!HttpContext.Current.User.Identity.IsAuthenticated)
        {
            base.HandleUnauthorizedRequest(actionContext);
        }
        else
        {

actionContext.Response = new System.Net.Http.HttpResponseMessage(System.Net.
        }
    }
}
```

Step 6 - Create a controller with name UserController

Now we have to create an empty web api controller with name usercontroller. In this controlle we will write the actions with different authorization and roles.



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

namespace WEBAPITOKENAUTHENTICATION.Controllers
{
    0 references
    public class UserController : ApiController
    {
        [AllowAnonymous]
        [HttpGet]
        [Route("api/data/forall")]
        0 references
        public IHttpActionResult Get()
        {
            return Ok("Now server time is: " + DateTime.Now.ToString());
        }

        [Authorize]
        [HttpGet]
        [Route("api/data/authenticate")]
        0 references
        public IHttpActionResult GetForAuthenticate()
        {
            var identity = (ClaimsIdentity)User.Identity;
            return Ok("Hello " + identity.Name);
        }


        [Authorize(Roles = "admin")]
```

The first action "Get" will be available for anonymous users .No authetication or token is needed for this.

```
[AllowAnonymous]
[HttpGet]
[Route("api/data/forall")]
public IHttpActionResult Get()
{
    return Ok("Now server time is: " + DateTime.Now.ToString());
}
```

The second action "GetForAuthenticate" only allows the authorized user to access it.

```
[Authorize]
[HttpGet]
[Route("api/data/authenticate")]
public IHttpActionResult GetForAuthenticate()
{
    var identity = (ClaimsIdentity)User.Identity;
    return Ok("Hello " + identity.Name);
}
```

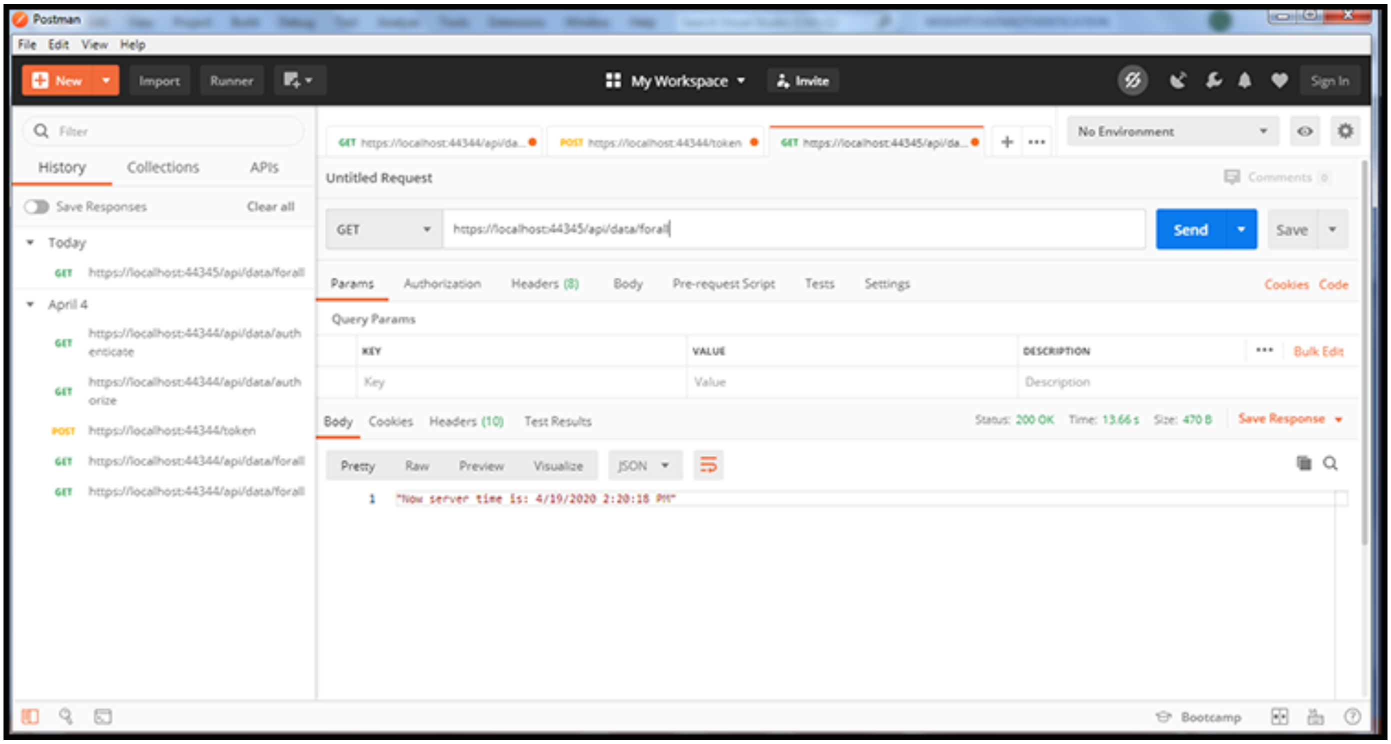
The third action "GetForAdmin" checks authorization and allows only admins to access it.

```
[Authorize(Roles = "admin")]
[HttpGet]
[Route("api/data/authorize")]
public IHttpActionResult GetForAdmin()
{
    var identity = (ClaimsIdentity)User.Identity;
    var roles = identity.Claims
```

```
        return Ok("Hello " + identity.Name + " Role: " - string.Join(",", roles.
}
```

Step 7 - Accessing the controller using Postman

Now we have to access the controller. For that purpose, we are using postman to get data.I n case ofthe first action "Get",we can access the data without generating a token as no authorization is needed for it, by just sending GET request for route "api/data/forall".



Today
GET https://localhost:44345/api/data/forall

April 4
GET https://localhost:44344/api/data/authenticate
GET https://localhost:44344/api/data/authorize
POST https://localhost:44344/token
GET https://localhost:44344/api/data/forall
GET https://localhost:44344/api/data/forall

GET https://localhost:44344/api/da...
POST https://localhost:44344/token
GET https://localhost:44345/api/da...

Untitled Request

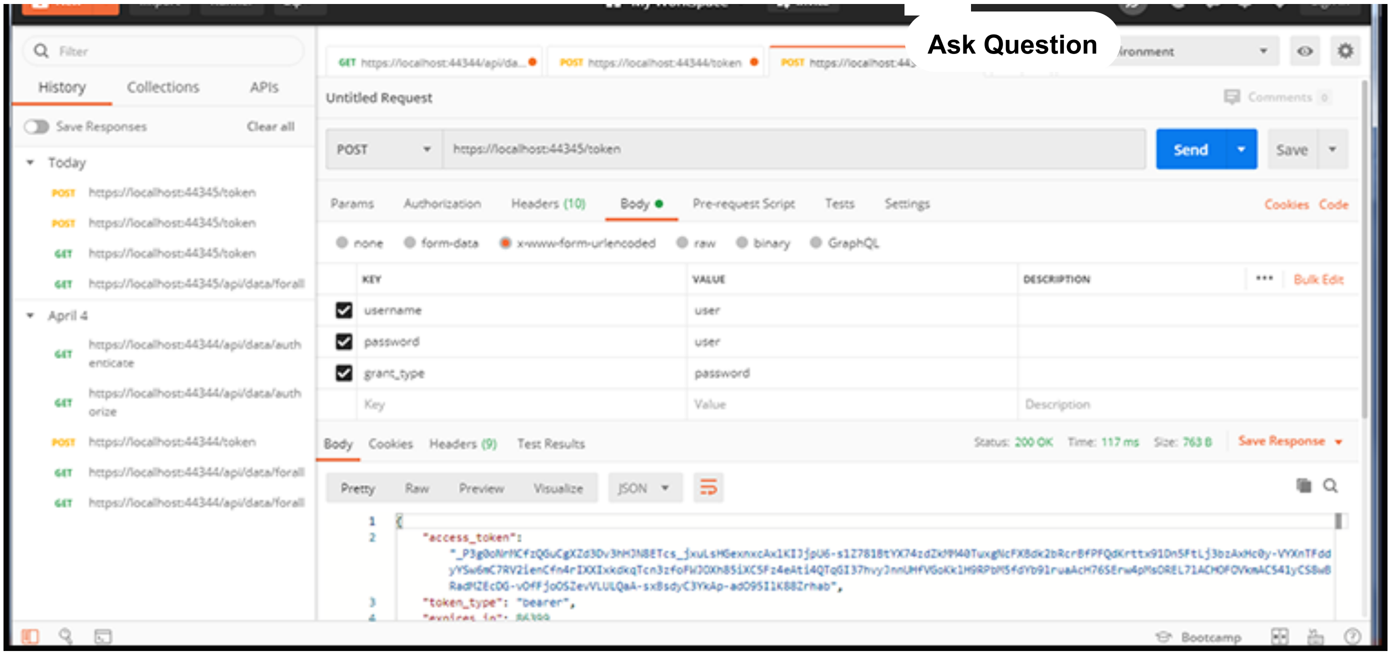
GET https://localhost:44345/api/data/forall

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 13.66 s Size: 470 B

```
1  "Now server time is: 4/19/2020 2:20:18 PM"
```

But for the rest of the actions, we have to generate a token using credentials. So we have to do post request for the token.


Ask Question
History
Collections
APIs
Save Responses
Clear all
Today
POST https://localhost:44345/token
POST https://localhost:44345/token
GET https://localhost:44345/token
GET https://localhost:44345/api/data/forall
April 4
GET https://localhost:44344/api/data/authenticate
GET https://localhost:44344/api/data/authorize
POST https://localhost:44344/token
GET https://localhost:44344/api/data/forall
GET https://localhost:44344/api/data/forall
Untitled Request
Send
Save
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Bulk Edit
Bootcamp

POST https://localhost:44345/token

Body: ○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

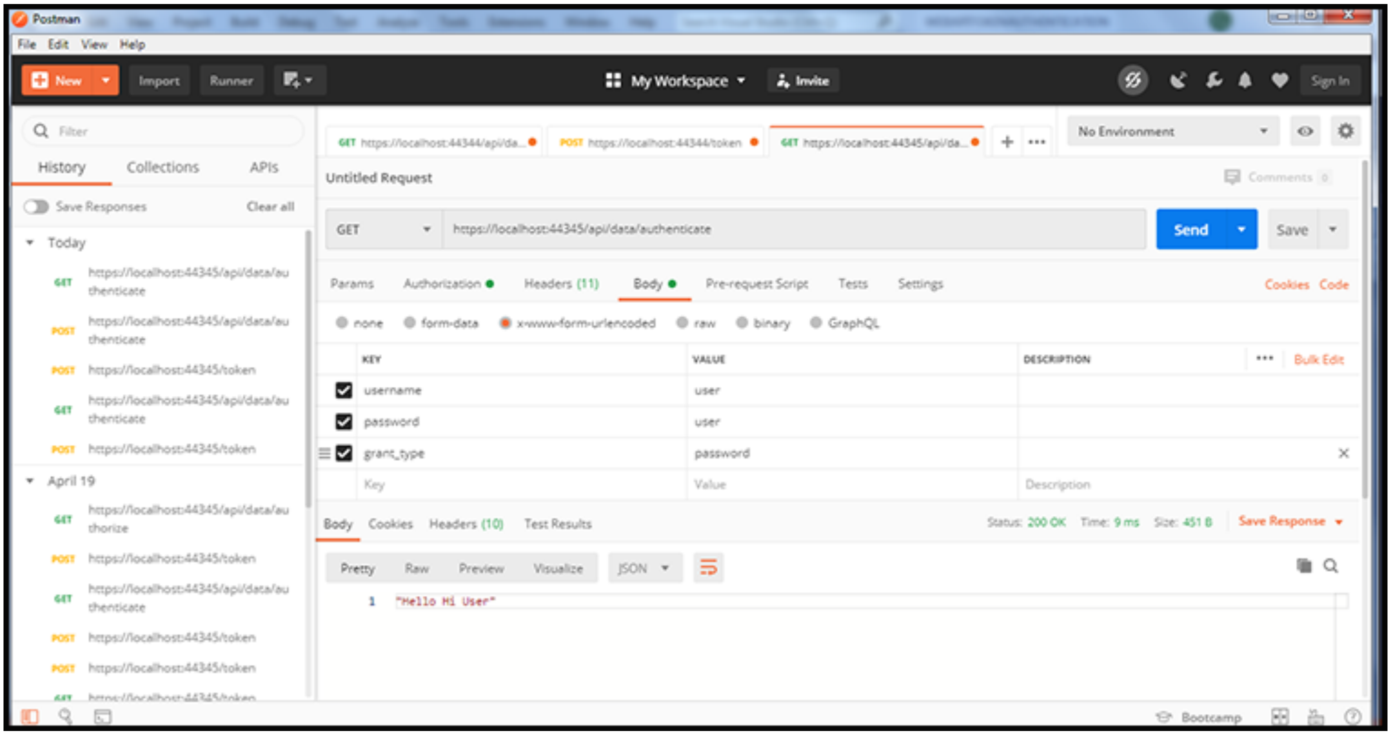
	KEY	VALUE	DESCRIPTION
☑	username	user	
☑	password	user	
☑	grant_type	password	
	Key	Value	Description

Body Cookies Headers (9) Test Results — Status: 200 OK Time: 117 ms Size: 763 B

Pretty Raw Preview Visualize JSON

```
{
    "access_token":
        "_P3g0oNrHCfzQGuCgXZd3Dv3hHJN8ETcs_jxuLsHGexnxcAx1KIJjpU6-s1Z7818tYX74zdZkM940TuxgNcFXBdk2bRcrBfPFQdKrttx91Dn5FtLj3bzAxHc0y-VYXnTFdd
        yYSw6mC7RV2ienCfn4rIXXIxkdkqTcn3zfoFWJOXh85iXC5Fz4eAti4QTqGI37hvyJnnUHfVGoKk1H9RPbM5fdYb91ruaAcH76SErw4pMsOREL71ACHOFOVkmAC541yCS8wB
        RadMZEcDG-vOfFjoOSZevVLULQaA-sxBsdyC3YkAp-ad095I1K88Zrhab",
    "token_type": "bearer",
    "expires_in": 86399
```

Now once the token is generated for the "user" now we can easily access the actions by using the user token.


Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Sign In
History
Collections
APIs
Save Responses
Clear all
Today
GET https://localhost:44345/api/data/authenticate
POST https://localhost:44345/api/data/authenticate
POST https://localhost:44345/token
GET https://localhost:44345/api/data/authenticate
POST https://localhost:44345/token
April 19
GET https://localhost:44345/api/data/authorize
POST https://localhost:44345/token
GET https://localhost:44345/api/data/authenticate
POST https://localhost:44345/token
POST https://localhost:44345/token
No Environment
Untitled Request
Send
Save
Params
Authorization
Headers (11)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Bulk Edit
Bootcamp

GET https://localhost:44345/api/data/authenticate

Body: ○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

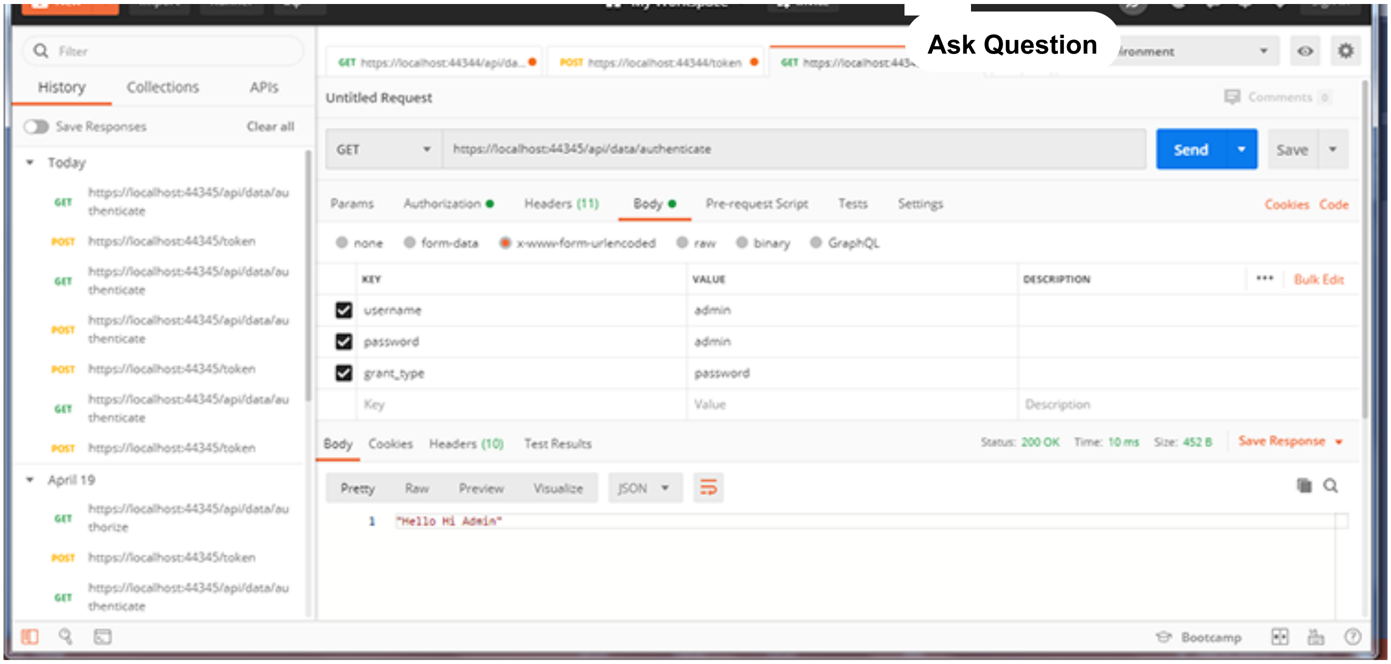
	KEY	VALUE	DESCRIPTION
☑	username	user	
☑	password	user	
☑	grant_type	password	
	Key	Value	Description

Body Cookies Headers (10) Test Results — Status: 200 OK Time: 9 ms Size: 451 B

Pretty Raw Preview Visualize JSON

```
"Hello Hi User"
```

Similarly, we can access the action as "admin" by generating token as admin and then using it.



GET https://localhost:44345/api/data/authenticate

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	username	admin	
☑	password	admin	
☑	grant_type	password	
	Key	Value	Description

Body Cookies Headers (10) Test Results — Status: 200 OK Time: 10 ms Size: 452 B

```
1  "Hello Hi Admin"
```

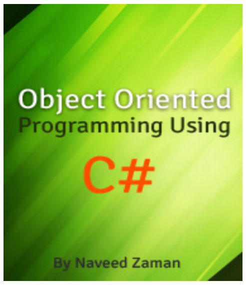
In the same way, the other actions can be accessed by user or admin depending on the way the token is generated.

Authentication Authoriation OAUTH OWIN OWIN OAuth Token

Web Api Token Authentication

OUR BOOKS

Microsoft SharePoint 2010
Getting Started with Managed Metadata in SharePoint 2010
Step-by-Step Tutorial
Anand



Gaurav Karroy
Working on Microsoft Technologies more than 5 years. Skilled in C#.NET, ASP.NET, MVC,Angular, ADO.NET, SQL Server, WebApi,Google Visualization API

2029 300.1k

22 26

Ask Question

Type your comment here and press Enter Key (Minimum 10 characters)

Hi I have this error, Server Error in '/' Application.The following errors occurred while attempting to load the app.- No assembly found containing an OwinStartupAttribute.- No assembly found containing a Startup or [AssemblyName].Startup class. To disable OWIN startup discovery, add the appSetting owin:AutomaticAppStartup with a value of "false" in your web.config. To specify the OWIN startup Assembly, Class, or Method, add the appSetting owin:AppStartup with the fully qualified startup class or configuration method name in your web.config. Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code. Exception Details: System.EntryPointNotFoundException: The following errors occurred while attempting to load the app. - No assembly found containing an OwinStartupAttribute. - No assembly found containing a Startup or [AssemblyName].Startup class. To disable OWIN startup discovery, add the appSetting owin:AutomaticAppStartup with a value of "false" in your web.config. To specify the OWIN startup Assembly, Class, or Method, add the appSetting owin:AppStartup with the fully qualified startup class or configuration method name in your web.config. Source Error: An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Yogesh Shankar Nov 29, 2022
2127 27 0 0 0 Reply

I have the application working adding a sql server look-up in the GrantResourceOwnerCredentials method. How would I go about hosting the application so it can be accessed from the internet?

Scott Thompson Oct 28, 2022
2056 98 0 0 0 Reply

Hai im facing this error config.Register(config); ====>('HttpConfiguration' does not contain a definition for 'Register' and no accessible extension method 'Register' accepting a first argument of type 'HttpConfiguration' could be found (are you missing a using directive or an assembly reference?) WEBAPITOKENAUTHENTICATION)

siva kumar Jan 17, 2022
2075 79 0 0 1 Reply

Install-Package Microsoft.Net.Http

ahmed mustafa Aug 24, 2022
1997 157 0 0

I have add the following lines in webapiconfig register method. Config.SuppressDefaultHostAuthentication(); config.Filters.Add(new HostAuthenticationFilter(OAuthDefaults.AuthenticationType)); but i have getting error message "No OWIN authentication manager is associated with the request". Can you please guide? Thanks

jai seelan Dec 30, 2021
2129 25 0 0 0 Reply

config.Filters.Add(new HostAuthenticationFilter(OAuthDefaults.A ype)); Add following line before namespace in AuthenticationStartup.cs [assembly: OwinStartup(typeof(WEBAPITOKENAUTHENTICATION.AuthenticationStartup))]

Ask Question

Kruti Dave Nov 06, 2021

2086 68 478 0 0 Reply

I am having the same issue....When i use /token in postman return: The resource cannot be found. Token URL not calling for unknown reason, when i can this URL on my local machine: http://localhost:1234/token. Can you please guide?

David Smith Oct 20, 2021

2139 15 0 0 2 Reply

Add the following lines in webapiconfig register method. Config.SuppressDefaultHostAuthentication(); config.Filters.Add(new HostAuthenticationFilter(OAuthDefaults.AuthenticationType)); Add following line before namespace in AuthenticationStartup.cs [assembly: OwinStartup(typeof(WEBAPITOKENAUTHENTICATION.AuthenticationStartup))]

Kruti Dave Nov 06, 2021

2086 68 478 1

Add those lines and still doesn't work. 404 not found error when running in post man

marty kin Jun 02, 2022

2138 16 0 0

When i use /token in postman return: The resource cannot be found. Token URL not calling for unknown reason, when i can this URL on my local machine: http://localhost:3244/token. Can you please guide?

Zarqa Bukhari Oct 20, 2021

2123 31 0 0 1 Reply

For someone struggling still today change the config value to true instead of false as below <add key="owin:AutomaticAppStartup" value="true"></add>

Noorulla Khan Nov 14, 2022

2085 69 0 0

Gaurav Karroy Can we implement okta using the same?

Shashank Sood Oct 02, 2021

2124 30 0 0 0 Reply

Very Good explanation !!!

Ramdas Chavan Aug 05, 2021

2078 76 0 0 0 Reply

Thanks for this, I have one question can we revoke the access token?

VIkram Singh Jun 05, 2021

2152 2 0 0 0 Reply

Ask Question

FEATURED ARTICLES

Static Abstract Interface Members In C# 11 And Curiously Recurring Template Pattern

Introduction To Homomorphic Encryption With Microsoft SEAL

How To Use ClickUp Efficiently For Project Management?

Two-Factor Authentication Using Google Authenticator in Asp.Net MVC

10 Productivity Tips for Software Developers

Ask Question

Learn ASP.NET MVC 5.0

CHALLENGE YOURSELF

Are You Cloud Native?

GET CERTIFIED

Python Developer

Ask Question