More

Dot Net Example

We have many things to learn from others

For any queries/suggestions, you can e-mail me at karan110011@gmail.com. I will try my best to respond to your queries/suggestions.

Search

Sunday, 16 December 2018

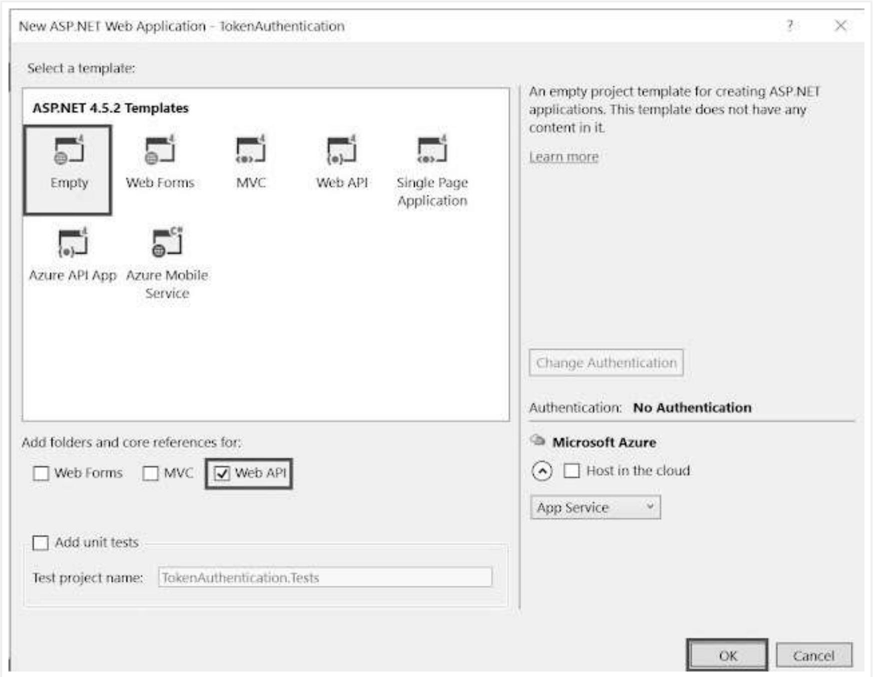
Refresh Token in Web API using OWIN

Step 1: - Open Visual Studio 2015 => Go to File Menu => New => Project...

Step 2: - In the Installed Templates list, select Visual C# => Web

Step 3: - Select **ASP.Net Web Application (.NET Framework)** from the Web list =>
Type **TokenAuthentication** in the Name box and click OK

Step 4: - Select **Empty template** from ASP.NET Templates List and Check **Web API** check box under **Add folders and core references for:**



New ASP.NET Web Application - TokenAuthentication
Select a template:
ASP.NET 4.5.2 Templates
Empty
Web Forms
MVC
Web API
Single Page Application
Azure API App
Azure Mobile Service
An empty project template for creating ASP.NET applications. This template does not have any content in it.
Learn more
Change Authentication
Authentication: No Authentication
Add folders and core references for:
□ Web Forms
□ MVC
☑ Web API
Microsoft Azure
□ Host in the cloud
App Service
□ Add unit tests
Test project name: TokenAuthentication.Tests
OK
Cancel

Step 5: - Right Click on **Models** folder => Add => New Items... => Expand **Visual C#** from Left Pane and Select **Code** => Select **Class** from Middle Pane => Type **Author.cs** in Name box => Click Add

Copy Past following code in **Author.cs**

```
namespace TokenAuthentication.Models
{
    public class Author
    {
        public int Id { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Address { get; set; }
        public string Status { get; set; }

    }
}
```

Labels

- AJAX (9)
- AngularJS (1)
- ASP.NET (96)
- ASP.NET Core (2)
- C Sharp (43)
- C# (93)
- CSS (1)
- Dapper (1)
- DOT NET (18)
- Entity Framework (3)
- Gridview (1)
- Interview Question (10)
- Javascript (21)
- Job interview tips (1)
- jQuery (11)
- MVC (17)
- MYSQL (1)
- OOPS (2)
- Oracle (13)
- Other (1)
- PLSQL (16)
- Regex (2)
- SQL (62)
- Visual Studio (1)
- WCF Service (4)
- WebApi (8)

About Me

Karan

View my complete profile

Blog Archive

- ► 2020 (3)
- ► 2019 (24)
- ▼ 2018 (31)
  - ▼ December (4)
    - Refresh Token in Web API using OWIN
    - Token Based Authentication in Web API using OWIN
    - Copy key values from NameValueCollection to Generi...
    - Adding Swagger to ASP.NET Web API
  - ► November (1)
  - ► October (2)
  - ► September (2)
  - ► August (2)
  - ► July (1)
  - ► June (2)
  - ► April (1)
  - ► March (12)
  - ► February (1)

► January (3)
► 2017 (31)
► 2016 (82)
► 2015 (100)

Step 6: - Right Click on **Controllers** folder => Add => Controller... => Select **Web API Controller - Empty** => Click Add => Type **DefaultController** in Controller Name box => Click Add

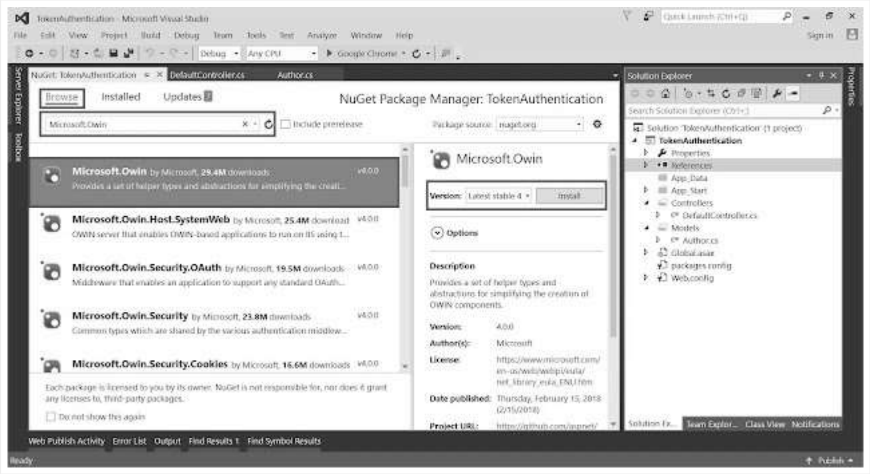
Copy Past following code in **DefaultController**

```
using System.Collections.Generic;
using System.Web.Http;
using TokenAuthentication.Models;

namespace TokenAuthentication.Controllers
{
    public class DefaultController : ApiController
    {
        [HttpGet]
        [Authorize]
        public List<Author> GetAuthor()
        {
            List<Author> author = new List<Author>() {
                new Author() {Id=1,FirstName="Ram",LastName="G",Address="Malad",Status="A" },
                new Author() {Id=2,FirstName="Shyam",LastName="G",Address="Kandivali",Status="A" },
                new Author() {Id=3,FirstName="Ghanshyam",LastName="G",Address="Borivali",Status="A" }
            };
            return author;
        }
    }
}
```

Step 7: - Right click on **References** => Click on **Manage NuGet Packages...** => Click on **Browse** => Search bellow list one by one => Select result => Click on Install

Microsoft.Owin


TokenAuthentication - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Any CPU Google Chrome
NuGet: TokenAuthentication DefaultController.cs Author.cs
Browse Installed Updates
NuGet Package Manager: TokenAuthentication
Microsoft.Owin
Include prerelease
Package source: nuget.org
Microsoft.Owin
Version: Latest stable 4
Install
Options
Solution Explorer
Solution 'TokenAuthentication' (1 project)
TokenAuthentication
Properties
References
App_Data
App_Start
Controllers
DefaultController.cs
Models
Author.cs
Global.asax
packages.config
Web.config
Web Publish Activity Error List Output Find Results 1 Find Symbol Results
Ready
Publish

Microsoft.Owin by Microsoft, 29.4M downloads v4.0.0
Provides a set of helper types and abstractions for simplifying the creati...

Microsoft.Owin.Host.SystemWeb by Microsoft, 25.4M download v4.0.0
OWIN server that enables OWIN-based applications to run on IIS using t...

Microsoft.Owin.Security.OAuth by Microsoft, 19.5M downloads v4.0.0
Middleware that enables an application to support any standard OAuth...

Microsoft.Owin.Security by Microsoft, 23.8M downloads v4.0.0
Common types which are shared by the various authentication middlew...

Microsoft.Owin.Security.Cookies by Microsoft, 16.6M downloads v4.0.0

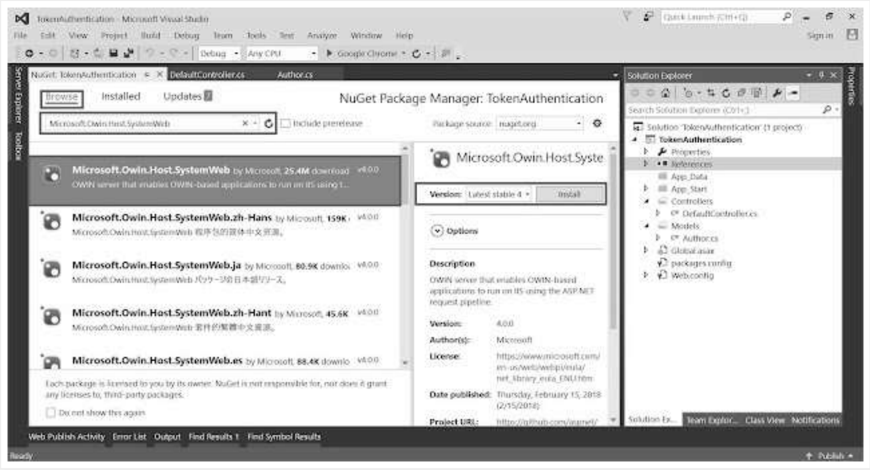
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

□ Do not show this again

Description
Provides a set of helper types and abstractions for simplifying the creation of OWIN components.

Version: 4.0.0
Author(s): Microsoft
License: https://www.microsoft.com/en-us/web/webpi/eula/net_library_eula_ENU.htm
Date published: Thursday, February 15, 2018 (2/15/2018)
Project URL: https://github.com/aspnet/

Microsoft.Owin.Host.SystemWeb


TokenAuthentication - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Any CPU Google Chrome
NuGet: TokenAuthentication DefaultController.cs Author.cs
Browse Installed Updates
NuGet Package Manager: TokenAuthentication
Microsoft.Owin.Host.SystemWeb
Include prerelease
Package source: nuget.org
Microsoft.Owin.Host.Syste
Version: Latest stable 4
Install
Options
Solution Explorer
Solution 'TokenAuthentication' (1 project)
TokenAuthentication
Properties
References
App_Data
App_Start
Controllers
DefaultController.cs
Models
Author.cs
Global.asax
packages.config
Web.config
Web Publish Activity Error List Output Find Results 1 Find Symbol Results
Ready
Publish

Microsoft.Owin.Host.SystemWeb by Microsoft, 25.4M download v4.0.0
OWIN server that enables OWIN-based applications to run on IIS using t...

Microsoft.Owin.Host.SystemWeb.zh-Hans by Microsoft, 159K v4.0.0
Microsoft.Owin.Host.SystemWeb 程序包的简体中文资源。

Microsoft.Owin.Host.SystemWeb.ja by Microsoft, 80.9K downlo v4.0.0
Microsoft.Owin.Host.SystemWeb パッケージの日本語リソース。

Microsoft.Owin.Host.SystemWeb.zh-Hant by Microsoft, 45.6K v4.0.0
Microsoft.Owin.Host.SystemWeb 套件的繁體中文資源。

Microsoft.Owin.Host.SystemWeb.es by Microsoft, 88.4K downlo v4.0.0

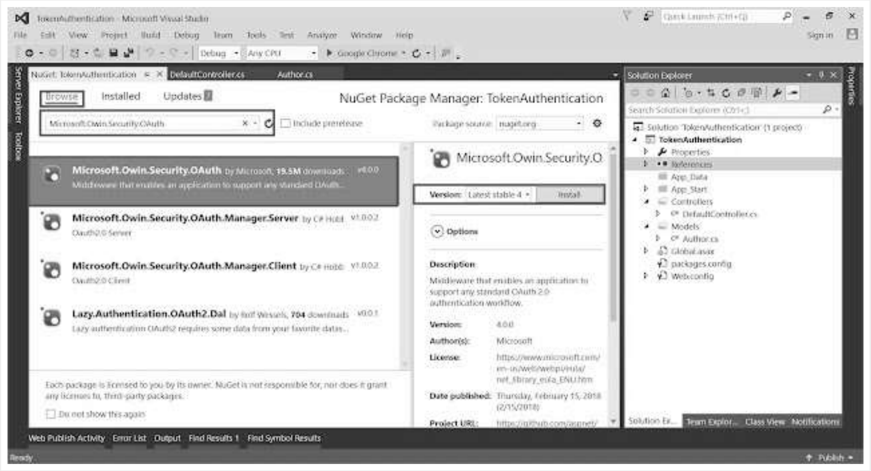
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

□ Do not show this again

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

Version: 4.0.0
Author(s): Microsoft
License: https://www.microsoft.com/en-us/web/webpi/eula/net_library_eula_ENU.htm
Date published: Thursday, February 15, 2018 (2/15/2018)
Project URL: https://github.com/aspnet/

Microsoft.Owin.Security.OAuth
Microsoft.Owin.Security



NuGet Package Manager: TokenAuthentication

Microsoft.Owin.Security.OAuth by Microsoft, 19.5M downloads v4.0.0
Middleware that enables an application to support any standard OAuth...

Microsoft.Owin.Security.OAuth.Manager.Server by C# Hobi v1.0.0.2
Oauth2.0 Server

Microsoft.Owin.Security.OAuth.Manager.Client by C# Hobi v1.0.0.2
Oauth2.0 Client

Lazy.Authentication.OAuth2.Dal by Rolf Wessels, 704 downloads v0.0.1
Lazy authentication OAuth2 requires some data from your favorite datas...

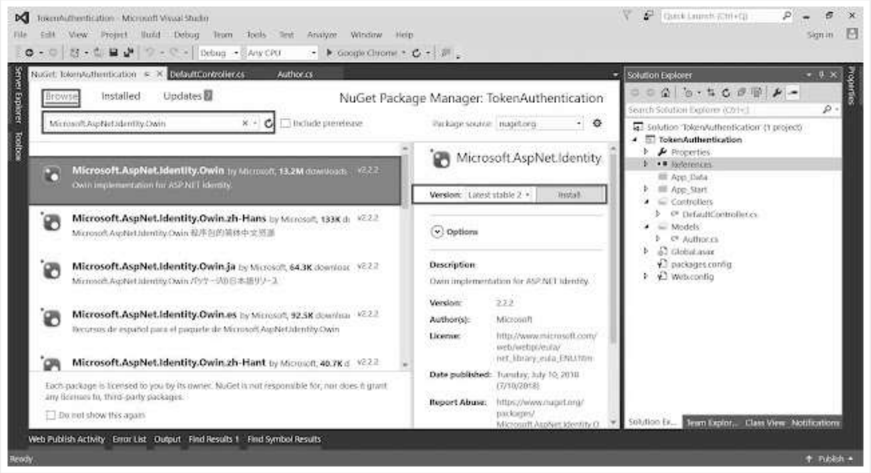
Version: Latest stable 4 Install

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

Version: 4.0.0
Author(s): Microsoft
License: https://www.microsoft.com/en-us/web/webpi/eula/net_library_eula_ENU.htm
Date published: Thursday, February 15, 2018 (2/15/2018)
Project URL: https://github.com/aspnet/

TokenAuthentication - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Any CPU Google Chrome
NuGet: TokenAuthentication DefaultController.cs Author.cs
Browse Installed Updates
Microsoft.Owin.Security.OAuth
Include prerelease
Package source: nuget.org
Microsoft.Owin.Security.O
Options
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
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'TokenAuthentication' (1 project)
TokenAuthentication
Properties
References
App_Data
App_Start
Controllers
DefaultController.cs
Models
Author.cs
Global.asax
packages.config
Web.config
Web Publish Activity Error List Output Find Results 1 Find Symbol Results
Ready
Publish

Microsoft.AspNet.Identity.Owin



NuGet Package Manager: TokenAuthentication

Microsoft.AspNet.Identity.Owin by Microsoft, 13.2M downloads v2.2.2
Owin implementation for ASP.NET Identity.

Microsoft.AspNet.Identity.Owin.zh-Hans by Microsoft, 133K downloads v2.2.2

Microsoft.AspNet.Identity.Owin.ja by Microsoft, 64.3K downloads v2.2.2

Microsoft.AspNet.Identity.Owin.es by Microsoft, 92.5K downloads v2.2.2
Recursos de español para el paquete de Microsoft.AspNet.Identity.Owin

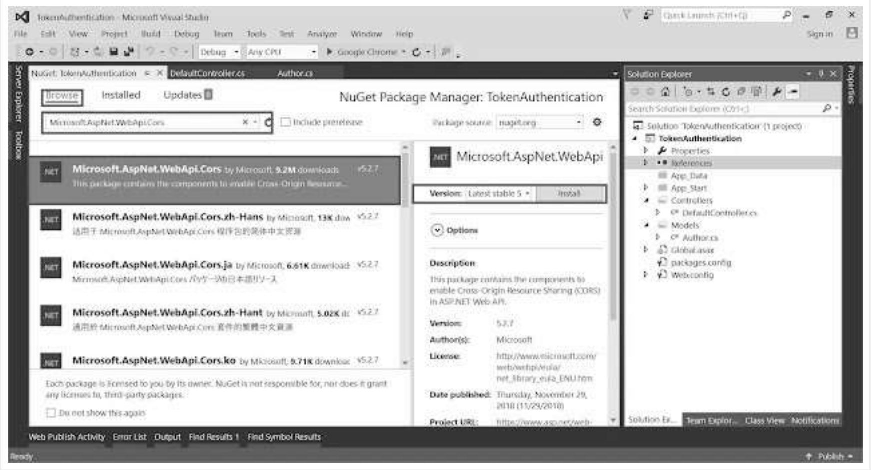
Microsoft.AspNet.Identity.Owin.zh-Hant by Microsoft, 40.7K downloads v2.2.2

Version: Latest stable 2 Install

Description: Owin implementation for ASP.NET Identity.

Version: 2.2.2
Author(s): Microsoft
License: http://www.microsoft.com/web/webpi/eula/net_library_eula_ENU.htm
Date published: Tuesday, July 10, 2018 (7/10/2018)
Report Abuse: https://www.nuget.org/packages/Microsoft.AspNet.Identity.O

Microsoft.AspNet.WebApi.Cors



NuGet Package Manager: TokenAuthentication

Microsoft.AspNet.WebApi.Cors by Microsoft, 9.2M downloads v5.2.7
This package contains the components to enable Cross-Origin Resource...

Microsoft.AspNet.WebApi.Cors.zh-Hans by Microsoft, 13K downloads v5.2.7

Microsoft.AspNet.WebApi.Cors.ja by Microsoft, 6.61K downloads v5.2.7

Microsoft.AspNet.WebApi.Cors.zh-Hant by Microsoft, 5.02K downloads v5.2.7

Microsoft.AspNet.WebApi.Cors.ko by Microsoft, 5.71K downloads v5.2.7

Version: Latest stable 5 Install

Description: This package contains the components to enable Cross-Origin Resource Sharing (CORS) in ASP.NET Web API.

Version: 5.2.7
Author(s): Microsoft
License: http://www.microsoft.com/web/webpi/eula/net_library_eula_ENU.htm
Date published: Thursday, November 29, 2018 (11/29/2018)
Project URL: https://www.asp.net/web-

Step 8: - Right Click on **Project Root** folder => Add => New Folder => Name "**Provider**"

Step 9: - Right Click on **Provider** folder => Add => New Items... => Expand **Visual C#** from Left Pane and Select **Code** => Select **Class** from Middle Pane => Type **AuthorizationServerProvider.cs** in Name box => Click Add

Copy Past following code in **AuthorizationServerProvider.cs**

```
using Microsoft.Owin.Security;
using Microsoft.Owin.Security.OAuth;
using System.Collections.Generic;
using System.Security.Claims;
using System.Threading.Tasks;
using System.Web.Http.Cors;

namespace TokenAuthentication.Provider
{
    [EnableCors(origins: "*", headers: "*", methods: "*")]
    public class AuthorizationServerProvider : OAuthAuthorizationServerProvider
    {
        public override async Task ValidateClientAuthentication(OAuthValidateClientAuthenticationContext
context)
        {
            context.Validated();
        }

        public override async Task
GrantResourceOwnerCredentials(OAuthGrantResourceOwnerCredentialsContext context)
        {
            context.OwinContext.Response.Headers.Add("Access-Control-Allow-Origin", new[] { "*" });

            //Write you DB logic here
            if (!context.UserName.Equals("Ram") && !context.Password.Equals("password@123"))
            {
                context.SetError("invalid_grant", "The user name or password is incorrect.");
            }
            else
            {
                var propertyDictionary = new Dictionary<string, string> {
                {
                    "userName", context.UserName
                }};
                var identity = new ClaimsIdentity(context.Options.AuthenticationType);
                var properties = new AuthenticationProperties(propertyDictionary);
                var ticket = new AuthenticationTicket(identity, properties);
                context.Validated(ticket);
            }
        }
    }
}
```

Step 10: - Right Click on **Provider** folder => Add => New Items... => Expand **Visual C#** from Left Pane and Select **Code** => Select **Class** from Middle Pane => Type **RefreshTokenProvider.cs** in Name box => Click Add

Copy Past following code in **RefreshTokenProvider.cs**

```
using Microsoft.Owin.Security;
using Microsoft.Owin.Security.Infrastructure;
using System;
using System.Collections.Concurrent;
using System.Threading.Tasks;

namespace TokenAuthentication.Provider
{
    public class RefreshTokenProvider : IAuthenticationTokenProvider
    {
        private static ConcurrentDictionary<string, AuthenticationTicket> _refreshTokens = new
ConcurrentDictionary<string, AuthenticationTicket>();

        public void Create(AuthenticationTokenCreateContext context)
        {
            throw new NotImplementedException();
        }

        public async Task CreateAsync(AuthenticationTokenCreateContext context)
        {
            var guid = Guid.NewGuid().ToString();

            // Set lifetime of refresh token
            var refreshTokenProperties = new
AuthenticationProperties(context.Ticket.Properties.Dictionary)
            {
                IssuedUtc = context.Ticket.Properties.IssuedUtc,
                ExpiresUtc = DateTime.UtcNow.AddMinutes(60)
            };

            var refreshTokenTicket = new AuthenticationTicket(context.Ticket.Identity,
refreshTokenProperties);

            _refreshTokens.TryAdd(guid, refreshTokenTicket);

            context.SetToken(guid);
        }

        public void Receive(AuthenticationTokenReceiveContext context)
        {
            throw new NotImplementedException();
        }

        public async Task ReceiveAsync(AuthenticationTokenReceiveContext context)
        {
            AuthenticationTicket ticket;
            string header = context.OwinContext.Request.Headers["Authorization"];

            if (_refreshTokens.TryRemove(context.Token, out ticket))
            {
                context.SetTicket(ticket);
            }
        }
    }
}
```

Step 11: - Right Click on **Project Root** folder => Add => New Items... => Expand **Visual C#** from Left Pane and Select **Code =>** Select **Class** from Middle Pane => Type **Startup.cs** in Name box => Click Add

Copy Past following code in **Startup.cs**

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

namespace TokenAuthentication
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            ConfigureAuth(app);
        }

        public void ConfigureAuth(IAppBuilder app)
        {
            var OAuthOptions = new OAuthAuthorizationServerOptions
            {
                AllowInsecureHttp = true,
                TokenEndpointPath = new PathString("/token"),
                AccessTokenExpireTimeSpan = TimeSpan.FromMinutes(20),
                Provider = new AuthorizationServerProvider(),
              RefreshTokenProvider = new RefreshTokenProvider()
            };

            app.UseOAuthBearerTokens(OAuthOptions);
            app.UseOAuthAuthorizationServer(OAuthOptions);
            app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());

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

Step 12: - Open/Expand **App_Start** folder => Open **WebApiConfig.cs** file => Add following line

```
using System.Web.Http;
using System.Web.Http.Cors;
```

```
namespace TokenAuthentication
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services
            EnableCorsAttribute cors = new EnableCorsAttribute("*", "*", "*");
            config.EnableCors(cors);

            // Web API routes
            config.MapHttpAttributeRoutes();

            config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}/{id}",
                defaults: new { id = RouteParameter.Optional }
            );
        }
    }
}
```

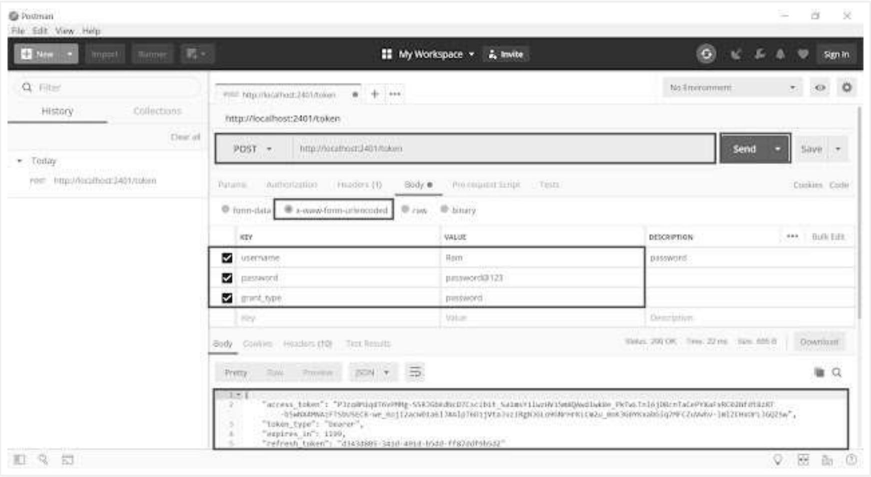
All Done

Run Project

Step 13: - Launch Postman

Step 14: - Copy Past following URL in URI and Value in **Body** tab and Hit **Send** button

http://localhost:2401/token
username - Ram
password - password@123
grant_type - password



POST http://localhost:2401/token

○ form-data ● x-www-form-urlencoded ○ raw ○ binary

	KEY	VALUE	DESCRIPTION
☑	username	Ram	password
☑	password	password@123	
☑	grant_type	password	
	Key	Value	Description

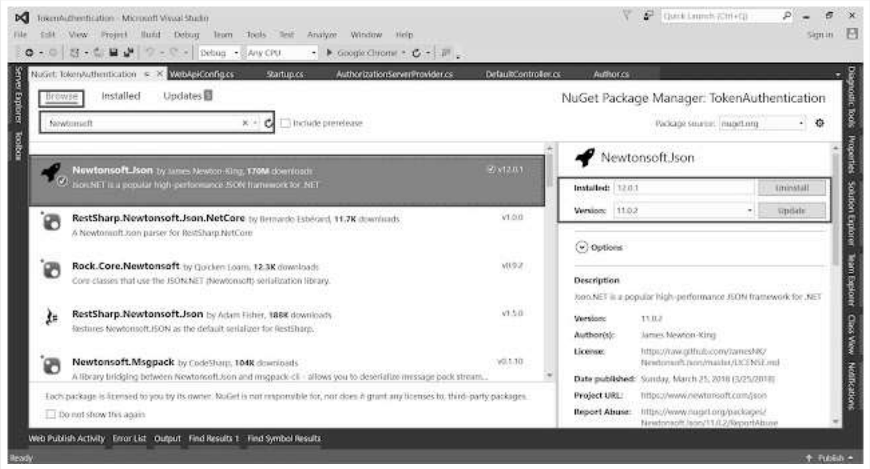
```
{
    "access_token": "[illegible]",
    "token_type": "bearer",
    "expires_in": 1199,
    "refresh_token": "[illegible]"
```

Note: -

If following error occurred open **Manage NuGet Packages...** and Search **Newtonsoft.Json** in **Browse** tab and update it.

Server Error in '/' Application.

Could not load file or assembly 'Newtonsoft.Json, Version=10.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed' or one of its dependencies. The located assembly's manifest definition does not match the assembly reference. (Exception from HRESULT: 0x80131040)

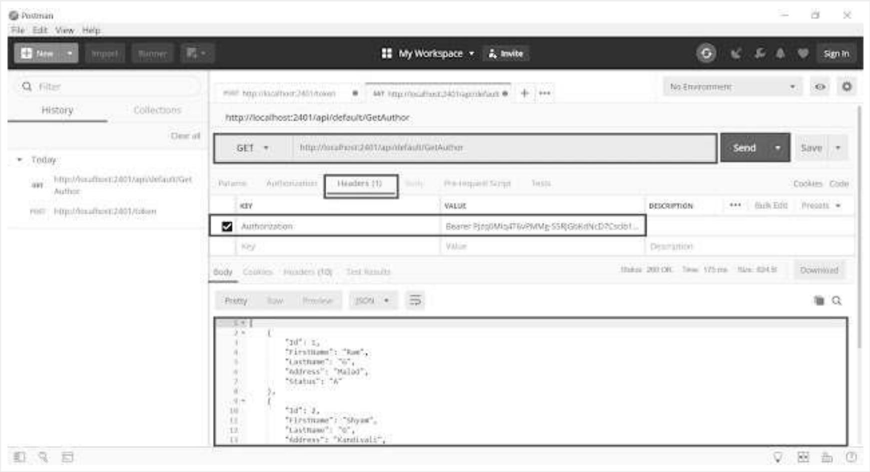

TokenAuthentication - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
NuGet: TokenAuthentication
WebApiConfig.cs
Startup.cs
AuthorizationServerProvider.cs
DefaultController.cs
Author.cs
Browse
Installed
Updates
NuGet Package Manager: TokenAuthentication
Newtonsoft
Include prerelease
Package source: nuget.org
Newtonsoft.Json by James Newton-King, 170M downloads
Json.NET is a popular high-performance JSON framework for .NET
v12.0.1
RestSharp.Newtonsoft.Json.NetCore
Rock.Core.Newtonsoft
RestSharp.Newtonsoft.Json
Newtonsoft.Msgpack
Newtonsoft.Json
Installed: 12.0.1
Uninstall
Version: 11.0.2
Update
Options
Description
Json.NET is a popular high-performance JSON framework for .NET
Version: 11.0.2
Author(s): James Newton-King
Date published: Sunday, March 25, 2018 (3/25/2018)
Project URL: https://www.newtonsoft.com/json
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
Web Publish Activity Error List Output Find Results 1 Find Symbol Results
Ready
Publish

Step 15: - Copy Past following URL in URI and Copy Past

Step 13 **access_token** in **Header** tab and Hit **Send** button

http://localhost:2401/api/default/GetAuthor

Authorization - Bearer PJzq0Miq4T6vPMMg-SSRJGbKdNcD7Cscib1t_Sa1msY1lwzHV1Sm8QAwd1wk8e_PkTwLTnl6jDBcnTaCePYXaFxRC02Nfdt8zRT-b5wNX4MNAzFTSbUSECB-we_6ojI2acW01a6lJXAlpT6O1jVtaJvziRgNJGLo9GNrHrKLCm2u_0oK3G0YKxabGiq7MFCZuVwhv-ImlZCHxOrLJGQZSw



http://localhost:2401/api/default/GetAuthor

GET http://localhost:2401/api/default/GetAuthor

	KEY	VALUE	DESCRIPTION
☑	Authorization	Bearer PJzq0Miq4T6vPMMg-SSRJGbKdNcD7Cscib1...	
	Key	Value	Description

Status: 200 OK Time: 175 ms Size: 824 B

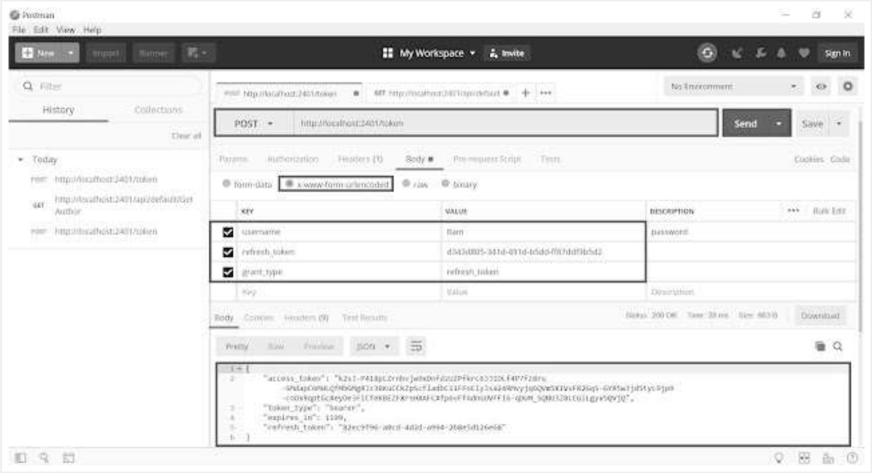
```
[
    {
        "Id": 1,
        "FirstName": "Ram",
        "LastName": "G",
        "Address": "Malad",
        "Status": "A"
    },
    {
        "Id": 2,
        "FirstName": "Shyam",
        "LastName": "G",
        "Address": "Kandivali",
```

Note: -

Token expired in 20 minutes and Refresh Token expired in 60 minutes. After 20 minutes Token will be expired and you need to sign in again. In that case use refresh token to sign in instead of passing username and password again.

Step 16: - Copy Past following URL in URI and Value in **Body** tab and Hit **Send** button

http://localhost:2401/token
username - Ram
refresh_token - d343d805-341d-491d-b5dd-ff87ddf9b5d2
grant_type - refresh_token



POST http://localhost:2401/token

○ form-data ● x-www-form-urlencoded ○ raw ○ binary

	KEY	VALUE	DESCRIPTION
☑	username	Ram	password
☑	refresh_token	d343d805-341d-491d-b5dd-f87ddf9b5d2	
☑	grant_type	refresh_token	
	Key	Value	Description

Status: 200 OK Time: 28 ms Size: 863 B

```
{
    "access_token": "k2sI-P41BpLZrnhvjw8xDnFdzU2PfkrC633IDLf4P7F2dru
        -SMdapCmMbLQfMbGMgRIr3BKoCCkZpScfladbC1IFFoC1y1xa2dRMvyjqGQVmSKIVsFR2Gq5-GYR5wJjdStytPjp9
        -cO0kKgptGcXeyOe3FICTHKBEZFKrnHKAFCXfpbvFfAdnUVFFI6-QDKM_SQNU328tCGiLgyA5QVjQ",
    "token_type": "bearer",
    "expires_in": 1199,
    "refresh_token": "82ec9f96-a0cd-4d2d-a994-268e5d126e68"
}
```

Posted by Karan at 12:14

Labels: WebApi

No comments:

Post a Comment

To leave a comment, click the button below to sign in with Blogger.

SIGN IN WITH BLOGGER

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Karan. Simple theme. Powered by Blogger.