C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
C# Corner 1 KALYANA KUMAR
Post Ask Question

ASP.NET MVC - OAuth 2.0 R Authorization Using Databa irst Approach

Asma Khalid Updated d 21, 2020

353.5k 99 30

Download Free .NET & JAVA Files API
Try Free File Format APIs for Word/Excel/PDF

WebApiOAuth2.zip

REST Web API is a light-weight essential component of web development in order to share the data across multiple client machines or devices e.g. mobile devices, desktop applications or any website. Authorization of REST Web API is an equally important part for sharing data across multiple client machines and devices in order to protect data sensitivity from any outside breaches and to authenticate the utilization of the target REST Web API.

Authorization of REST Web API can be done via a specific username/password with the combination of a secret key, but, for this type of authorization scheme, REST Web API access needs to be authenticated per call to the hosting server. Also, we as the owner of the server have no way to verify who is utilizing our REST Web API, whether it's the clients that we have allowed access to or if some malicious user is also using our API(s) without our knowledge. Finally, since the username/password is packed to a base64 format automatically by the

API calls they can easily decrypt base64 format and could use my REST Web API for malicious activities.

Hmmmmm.....scary stuff! Not to mention that despite the fact that I have authorized my REST Web API, it is still open for malicious users to utilize without even my knowledge. So, what to do? To answer that a new authorization scheme is introduced which can also be utilized in Login flow of any web application as well, but, I will be focusing on it from a REST Web API perspective. So, this new scheme of authorization is OAuth 2.0 which is a token based authorization scheme.

Let's compare OAuth 2.0 authorization scheme to the traditional username/password authorization scheme from REST Web API perspective, i.e.,

	Username/Password REST Web API Authorization	**VS**	**OAuth 2.0 REST Web API Authorization**
1.	Access Authorization to same or different REST		Access Authorization is authenticat

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1 C# Corner KALYANA KUMAR
Post Ask Question

		succ[illegible] authorization, an access tok[illegible] for a specific period of time. REST Web API(s) call can utilize the [illegible]ken which server will not authenticate again and again.
2.	REST Web API(s) call cannot track the user who is consuming the REST Web API(s). Its utilization is based on mutual trust between Producer and consumer.	Only local system users can consume the REST Web API(s), so, REST Web API call can track the user who is consuming the REST Web API.
3.	Username/Password is encrypted in base64 format. So, hackers can easily decrypt the request headers.	Access Token is encrypted in a special format. So, hackers cannot easily decrypt it even with access to request header.
4.	All client machines or devices code needs to be updated in case of the change in username/password for malicious activities.	Access token is activated for a specific time period. Change in system user's credential will not require the change of code in target consumer client machines and devices. Update credential generated a new access token.
5.	Username/Password is fixed.	Access token is generated automatically based on system user's credential.

Today, I shall demonstrate OAuth 2.0 mechanism to authorize a REST Web API which will also give us the benefit of [Authorize] attribute via OWIN security layer.

OAuth 2.0 Web API Authorization

This is web API project for Web API authorization using OAuth 2.0 and Database First Approach.

Copyright © 2018 - Asma's Blog. All rights reserved.

Following are a few prerequisites before you proceed any further,

1. Knowledge of OAuth 2.0.
2. Knowledge of ASP.NET MVC5.
3. Knowledge of C# programming.
4. Knowledge of REST Web API.

You can download the complete source code or you can follow the step by step discussion below. The sample code is developed in Microsoft Visual Studio 2015 Enterprise.

Let's begin now:

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1
C# Corner KALYANA KUMAR
Post Ask
Question

Create a new Web API project and name it "WebApiOaut

Step 2

Install the following NuGet packages into your project, i.e.,

1. Microsoft.Owin.Security.OAuth
2. Microsoft.Owin.Cors
3. Microsoft.AspNet.WebApi.Core
4. Microsoft.AspNet.WebApi.Owin

Step 3

Create "DB_Oauth_API" database into your SQL Server and execute the following script on it, i.e.,

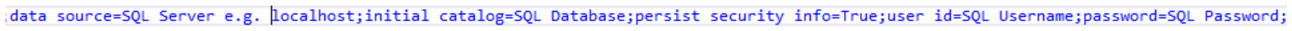
```
USE [DB_Oauth_API]
GO
/****** Object:  StoredProcedure [dbo].
[LoginByUsernamePassword]    Script Date: 5/10/2018 2:37:02 PM ******/
DROP PROCEDURE [dbo].[LoginByUsernamePassword]
GO
/****** Object:  Table [dbo].
[Login]    Script Date: 5/10/2018 2:37:02 PM ******/
DROP TABLE [dbo].[Login]
GO
/****** Object:  Table [dbo].
[Login]    Script Date: 5/10/2018 2:37:02 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

```
GO
CREATE TABLE [dbo].[Login](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [username] [varchar](50) NOT NULL,
    [password] [varchar](50) NOT NULL,
 CONSTRAINT [PK_Login] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
SET IDENTITY_INSERT [dbo].[Login] ON

INSERT [dbo].
[Login] ([id], [username], [password]) VALUES (1, N'admin', N'admin
SET IDENTITY_INSERT [dbo].[Login] OFF
```

C# Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1
C# Corner KALYANA KUMAR
Post Ask
Question

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Asma Khalid>
-- Create date: <Create Date,,15-Mar-2016>
-- Description: <Description,,You are Allow to Distribute this Code>
-- =============================================
CREATE PROCEDURE [dbo].[LoginByUsernamePassword]
    @username varchar(50),
    @password varchar(50)
AS
BEGIN
    SELECT id, username, password
    FROM Login
    WHERE username = @username
    AND password = @password
END

GO
```

In the above script, I have created a simple login table and a stored procedure to retrieve the specific login. I am using entity framework database first approach for database connection for this ASP.NET MVC WebAPI application. Do also update your SQL server connection string in the project "Web.config" file if you have downloaded the project i.e.



```
data source=SQL Server e.g. localhost;initial catalog=SQL Database;persist security info=True;user id=SQL Username;password=SQL Password;
```

Step 4

Rename "Controllers/ValueController.cs" file to "Controllers/WebApiController.cs".

Open, "Controllers/WebApiController.cs" file replace following code,

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

namespace WebApiOauth2.Controllers
{
    [Authorize]
    public class WebApiController : ApiController
    {
        // GET api/WebApi
        public IEnumerable<string> Get()
        {
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1
C# Corner KALYANA KUMAR
Post Ask Question

```
        // GET api/WebApi/5
        public string Get(int id)
        {
            return "Hello Authorized API with ID = " + id;
        }

        // POST api/WebApi
        public void Post([FromBody]string value)
        {
        }

        // PUT api/WebApi/5
        public void Put(int id, [FromBody]string value)
        {
        }

        // DELETE api/WebApi/5
        public void Delete(int id)
        {
        }
    }
}
```

In the above code, I have created very simple and basic REST Web API(s). Notice [Authorize] attribute is already placed at the top of the controller to make the REST Web API(s) access secure.

Step 6

Now open "App_Start/WebApiConfig.cs" file and replace the following code in it i.e.

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

namespace WebApiOauth2
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services
            // Configure Web API to use only bearer token authenticatio
            config.SuppressDefaultHostAuthentication();
            config.Filters.Add(new HostAuthenticationFilter(OAuthDefaul

            // Web API routes
            config.MapHttpAttributeRoutes();

            config.Routes.MapHttpRoute(
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1 KALYANA KUMAR
Post Ask Question

```
                defaults: new { id = Rout                    al }
            );

            // WebAPI when dealing with JSON & JavaScript!
            // Setup json serialization                 lize classes to camel (
            var formatter = GlobalConfiguration.Configuration.Formatter
            formatter.SerializerSettings.ContractResolver = new Newtons

            // Adding JSON type web api formatting.
            config.Formatters.Clear();
            config.Formatters.Add(formatter);
        }
    }
}
```

In the above code the following two lines of code will add authentication filter for Oauth 2.0 authorization scheme and surpass any existing authorization scheme i.e.

```
surpass any existing authorization scheme i.e.
            // Web API configuration and services
            // Configure Web API to use only bearer token authenticatio
            config.SuppressDefaultHostAuthentication();
            config.Filters.Add(new HostAuthenticationFilter(OAuthDefaul
```

Step 7

Now, open "App_Start/Startup.Auth.cs" file and replace following code in it i.e.

```
using System;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
```

```
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.Google;
using Owin;
using WebApiOauth2.Models;
using Microsoft.Owin.Security.OAuth;
using WebApiOauth2.Helper_Code.OAuth2;

namespace WebApiOauth2
{
    public partial class Startup
    {
        #region Public /Protected Properties.

        /// <summary>
        /// OAUTH options property.
        /// </summary>
        public static OAuthAuthorizationServerOptions OAuthOptions { ge

        /// <summary>
        /// Public client ID property.
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1 KALYANA KUMAR
Post Ask Question

```
        public static string PublicClient

        #endregion

        // For more information on confi                uthentication, please v
LinkId=301864
        public void ConfigureAuth(IAppBuilder app)
        {
            // Enable the application to use a cookie to store informat
            // and to use a cookie to temporarily store information abo
            // Configure the sign in cookie
            app.UseCookieAuthentication(new CookieAuthenticationOptions
            {
                AuthenticationType = DefaultAuthenticationTypes.Applica
                LoginPath = new PathString("/Account/Login"),
                LogoutPath = new PathString("/Account/LogOff"),
                ExpireTimeSpan = TimeSpan.FromMinutes(5.0),
            });

            app.UseExternalSignInCookie(DefaultAuthenticationTypes.Exte

            // Configure the application for OAuth based flow
            PublicClientId = "self";
            OAuthOptions = new OAuthAuthorizationServerOptions
            {
                TokenEndpointPath = new PathString("/Token"),
                Provider = new AppOAuthProvider(PublicClientId),
                AuthorizeEndpointPath = new PathString("/Account/Extern
                AccessTokenExpireTimeSpan = TimeSpan.FromHours(4),
                AllowInsecureHttp = true //Don't do this in production
            };

            // Enable the application to use bearer tokens to authentic
            app.UseOAuthBearerTokens(OAuthOptions);


            app.UseTwoFactorSignInCookie(DefaultAuthenticationTypes.Two

            // Enables the application to remember the second login ver
            // Once you check this option, your second step of verifica
            // This is similar to the RememberMe option when you log in
            app.UseTwoFactorRememberBrowserCookie(DefaultAuthentication

            // Uncomment the following lines to enable logging in with
            //app.UseMicrosoftAccountAuthentication(
            //    clientId: "",
            //    clientSecret: "");

            //app.UseTwitterAuthentication(
            //   consumerKey: "",
            //   consumerSecret: "");

            //app.UseFacebookAuthentication(
            //   appId: "",
            //   appSecret: "");
```

```
                //{
                //    ClientId = "",
                //    ClientSecret = ""
                //});
            }
        }
}
```

In the above piece of code, "PublicClientId" is used when "AuthorizeEndpointPath" is utilized for unique instantiation from client-side. Following lines of code will enable the OAuth 2.0 authorization scheme, i.e.

```
from client side. Following lines of code will enable the OAuth 2.0 aut
            // Configure the application for OAuth based flow
            PublicClientId = "self";
            OAuthOptions = new OAuthAuthorizationServerOptions
            {
                TokenEndpointPath = new PathString("/Token"),
                Provider = new AppOAuthProvider(PublicClientId),
                AuthorizeEndpointPath = new PathString("/Account/Extern
                AccessTokenExpireTimeSpan = TimeSpan.FromHours(4),
                AllowInsecureHttp = true //Don't do this in production
            };

            // Enable the application to use bearer tokens to authentic
            app.UseOAuthBearerTokens(OAuthOptions);
```

OAuthAuthorizationOptions are explained as follow i.e.

- *TokenEndpointPath* ->
  This is the path which will be called in order to authorize the user credentials and in

- *Provider* ->
  You need to implement this class (which I have in this tutorial) where you will verify the user credential and create identity claims in order to return the generated access token.

- *AuthorizeEndpointPath* ->
  In this tutorial, I am not using this property as I am not taking consent of external logins. So, if you are using external logins then you can update this path to get user consent then required access token will be generated.

- *AccessTokenExpireTimeSpan* ->
  This is the time period you want your access token to be accessible. The shorter time span is recommended for sensitive API(s).

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1
C# Corner KALYANA KUMAR
Post Ask
Question

More properties can be studied here.

Step 8

Now, create "Helper_Code/OAuth2/AppOAuthProvider.cs" file and replace following code in it i.e.

```
//-------------------------------------------------------------------
---
// <copyright file="AppOAuthProvider.cs" company="None">
//     Copyright (c) Allow to distribute this code.
// </copyright>
// <author>Asma Khalid</author>
//-------------------------------------------------------------------
---

namespace WebApiOauth2.Helper_Code.OAuth2
{
    using Microsoft.Owin.Security;
    using Microsoft.Owin.Security.Cookies;
    using Microsoft.Owin.Security.OAuth;
    using Models;
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Security.Claims;
    using System.Threading.Tasks;
    using System.Web;

        /// <summary>
```

```
    public class AppOAuthProvider : OAuthAuthorizationServerProvider
    {
        #region Private Properties

        /// <summary>
        /// Public client ID property.
        /// </summary>
        private readonly string _publicClientId;

        /// <summary>
        /// Database Store property.
        /// </summary>
        private Oauth_APIEntities databaseManager = new Oauth_APIEntiti

        #endregion

        #region Default Constructor method.

        /// <summary>
```

C# Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
Post Ask Question 1 KALYANA KUMAR

```
        /// <param name="publicClientId">[illegible]parameter</pa
        public AppOAuthProvider(string pu[illegible]ClientId)
        {
            //TODO: Pull from configuratio
            if (publicClientId == null)
            {
                throw new ArgumentNullException(nameof(publicClientId))
            }

            // Settings.
            _publicClientId = publicClientId;
        }

        #endregion

        #region Grant resource owner credentials override method.

        /// <summary>
        /// Grant resource owner credentials overload method.
        /// </summary>
        /// <param name="context">Context parameter</param>
        /// <returns>Returns when task is completed</returns>
        public override async Task GrantResourceOwnerCredentials(OAuthG
        {
            // Initialization.
            string usernameVal = context.UserName;
            string passwordVal = context.Password;
            var user = this.databaseManager.LoginByUsernamePassword(use

            // Verification.
            if (user == null || user.Count() <= 0)
            {
                // Settings.
                context.SetError("invalid_grant", "The user name or pas
```

```
                return;
            }

            // Initialization.
            var claims = new List<Claim>();
            var userInfo = user.FirstOrDefault();

            // Setting
            claims.Add(new Claim(ClaimTypes.Name, userInfo.username));

            // Setting Claim Identities for OAUTH 2 protocol.
            ClaimsIdentity oAuthClaimIdentity = new ClaimsIdentity(clai
            ClaimsIdentity cookiesClaimIdentity = new ClaimsIdentity(cl

            // Setting user authentication.
            AuthenticationProperties properties = CreateProperties(user
            AuthenticationTicket ticket = new AuthenticationTicket(oAut

            // Grant access to authorize user.
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1
C# Corner KALYANA KUMAR
Post Ask Question

```
        }

        #endregion

        #region Token endpoint override 

        /// <summary>
        /// Token endpoint override method
        /// </summary>
        /// <param name="context">Context parameter</param>
        /// <returns>Returns when task is completed</returns>
        public override Task TokenEndpoint(OAuthTokenEndpointContext co
        {
            foreach (KeyValuePair<string, string> property in context.P
            {
                // Adding.
                context.AdditionalResponseParameters.Add(property.Key,
            }

            // Return info.
            return Task.FromResult<object>(null);
        }

        #endregion

        #region Validate Client authntication override method

        /// <summary>
        /// Validate Client authntication override method
        /// </summary>
        /// <param name="context">Contect parameter</param>
        /// <returns>Returns validation of client authentication</retur
        public override Task ValidateClientAuthentication(OAuthValidate
        {
            // Resource owner password credentials does not provide a c


                // Validate Authoorization.
                context.Validated();
            }

            // Return info.
            return Task.FromResult<object>(null);
        }

        #endregion

        #region Validate client redirect URI override method

        /// <summary>
        /// Validate client redirect URI override method
        /// </summary>
        /// <param name="context">Context parmeter</param>
        /// <returns>Returns validation of client redirect URI</returns
        public override Task ValidateClientRedirectUri(OAuthValidate
        {
            {
                // Initialization.
                Uri expectedRootUri = ne          r     .Request.Uri, "/")

                // Verification.
                if (expectedRootUri.Abso         == context.RedirectUri)
                {
                    // Validating.
                    context.Validated();
                }
            }

            // Return info.
            return Task.FromResult<object>(null);
        }

        #endregion

        #region Create Authentication properties method.

        /// <summary>
        /// Create Authentication properties method.
        /// </summary>
        /// <param name="userName">User name parameter</param>
        /// <returns>Returns authenticated properties.</returns>
        public static AuthenticationProperties CreateProperties(string
        {
            // Settings.
            IDictionary<string, string> data = new Dictionary<string, s
                                               {
                                                   { "userName", userNa
                                               };

            // Return info.
            return new AuthenticationProperties(data);


        #endregion
    }
}
```

In the above code, "GrantResourceOwnerCredentials(...)" method is the key method which is called when TokenEndpointPath is called. Notice that "GrantResourceOwnerCredentials(...)" method is used by "grant_type=password" scheme. If you are using "grant_type=client_credentials" scheme then you need to override "GrantClientCredentials(...)" method. Other methods are part of "OAuthAuthorizationServerProvider" class, use them as they are. In "GrantResourceOwnerCredentials(...)" method we are verifying the system login user and then create the require identities claims and then generate the returning access token ticket i.e.

```
#region Grant resource owner credentials override method.
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
1 C#Corner KALYANA KUMAR
Post a Ask Question

```
/// Grant resource owner credentials over
/// </summary>
/// <param name="context">Context parame
/// <returns>Returns when task is completed</returns>
public override async Task GrantResource                dentials(OAuthGrantReso
{
    // Initialization.
    string usernameVal = context.UserName;
    string passwordVal = context.Password;
    var user = this.databaseManager.LoginByUsernamePassword(usernameVal

    // Verification.
    if (user == null || user.Count() <= 0)
    {
        // Settings.
        context.SetError("invalid_grant", "The user name or password is

        // Retuen info.
        return;
    }

    // Initialization.
    var claims = new List<Claim>();
    var userInfo = user.FirstOrDefault();

    // Setting
    claims.Add(new Claim(ClaimTypes.Name, userInfo.username));

    // Setting Claim Identities for OAUTH 2 protocol.
    ClaimsIdentity oAuthClaimIdentity = new ClaimsIdentity(claims, OAut
    ClaimsIdentity cookiesClaimIdentity = new ClaimsIdentity(claims, Co

    // Setting user authentication.
    AuthenticationProperties properties = CreateProperties(userInfo.use
    AuthenticationTicket ticket = new AuthenticationTicket(oAuthClaimId
```

```
    context.Validated(ticket);
    context.Request.Context.Authentication.SignIn(cookiesClaimIdentity)
}

#endregion
```

Step 9

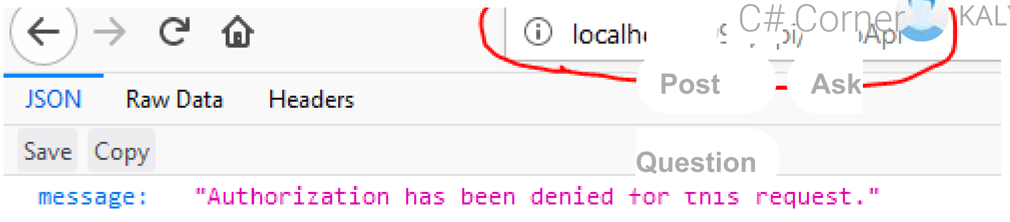
Now, execute the project and use the following link in the browser to see your newly created REST Web API method in action as follows:

```
yourlink:port/api/WebApi
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference

C# Corner KALYANA KUMAR

Post Ask Question

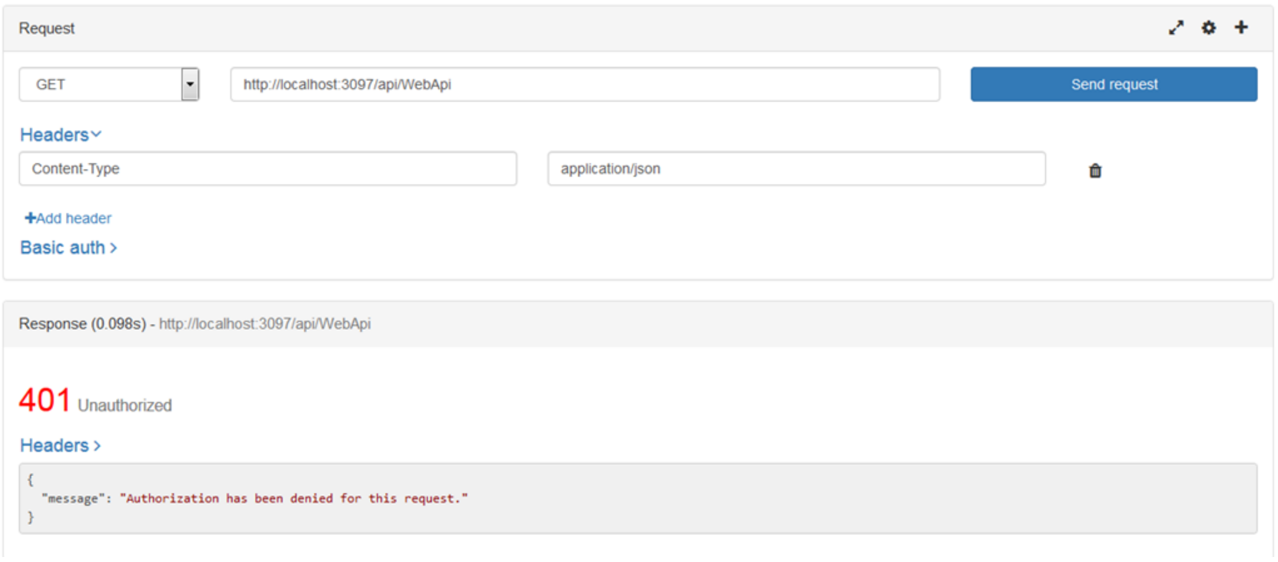


```
message:    "Authorization has been denied for this request."
```

In the above snippet, you will notice that since now our REST Web API has been authorized, therefore, we cannot directly execute the REST Web API URL in the browser.

Step 10

Let's test out REST Web API in REST Web API client. I am using Firefox plugin i.e. "**RESTED**". At, first, I simply try to hit the REST Web API without any authorization details and I will get following response i.e.



GET http://localhost:3097/api/WebApi

Content-Type: application/json

Response (0.098s) - http://localhost:3097/api/WebApi

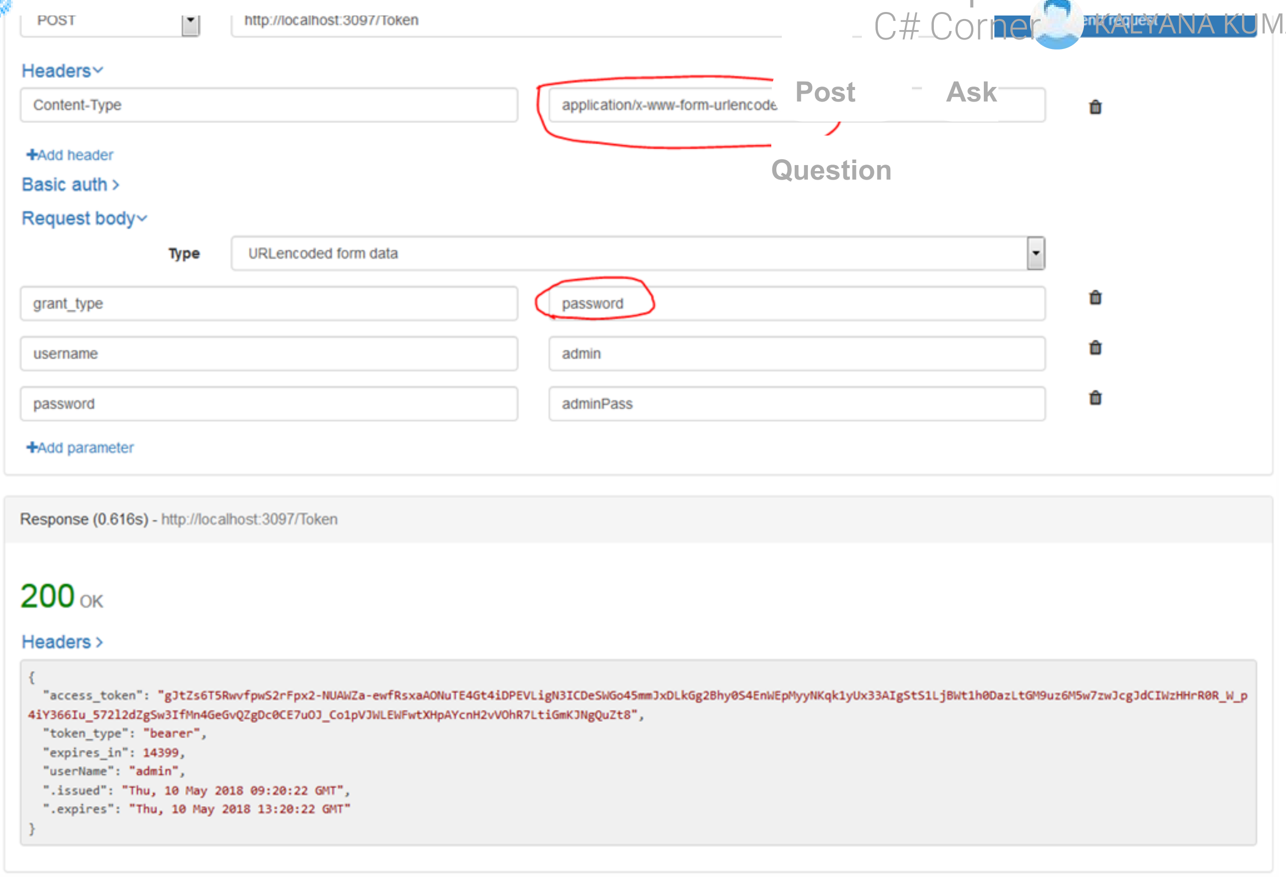
401 Unauthorized

```
{
  "message": "Authorization has been denied for this request."
}
```

Step 11

Now, I will provide the system user authorization to get access token and then use that access token as a header in the REST Web API and try to his the REST Web API which will return the following response, i.e.

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference
C# Corner KALYANA KUMAR
Post Ask Question



POST http://localhost:3097/Token

Headers
Content-Type: application/x-www-form-urlencode

+Add header
Basic auth
Request body
Type: URLencoded form data

grant_type	password
username	admin
password	adminPass

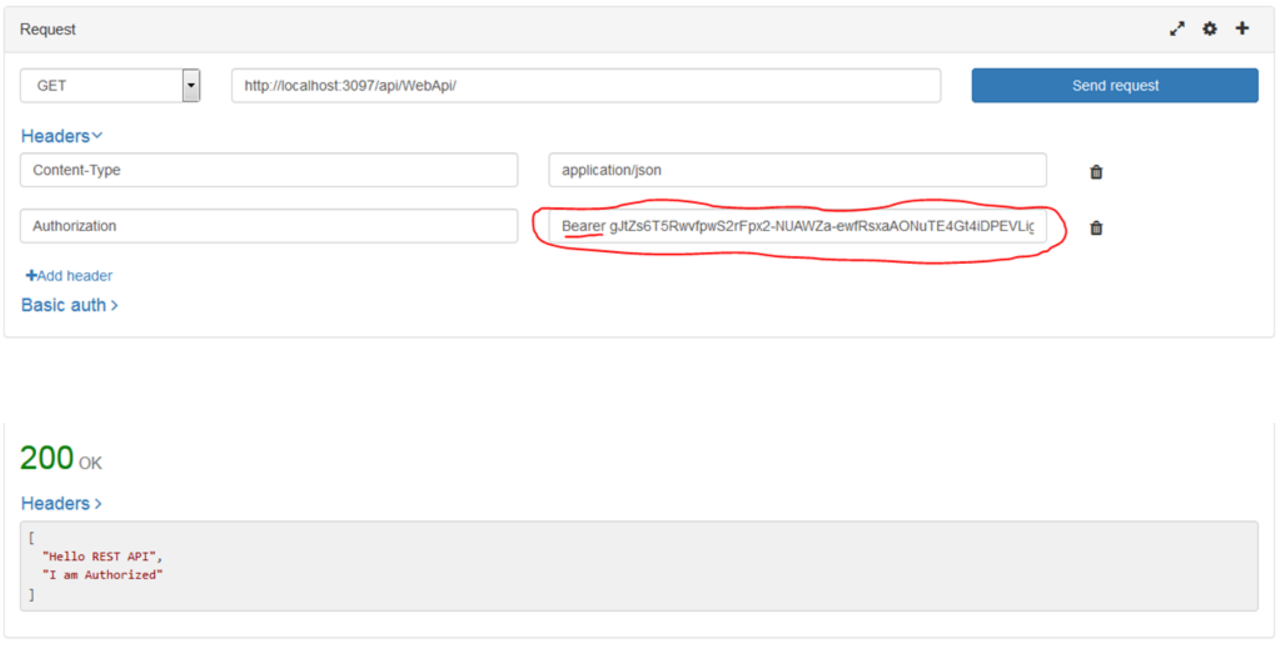
+Add parameter

Response (0.616s) - http://localhost:3097/Token

200 OK

Headers

```
{
  "access_token": "gJtZs6T5RwvfpwS2rFpx2-NUAWZa-ewfRsxaAONuTE4Gt4iDPEVLigN3ICDeSWGo45mmJxDLkGg2Bhy0S4EnWEpMyyNKqk1yUx33AIgStS1LjBWt1h0DazLtGM9uz6M5w7zwJcgJdCIWzHHrR0R_W_p4iY366Iu_572l2dZgSw3IfMn4GeGvQZgDc0CE7uOJ_Co1pVJWLEWFwtXHpAYcnH2vVOhR7LtiGmKJNgQuZt8",
  "token_type": "bearer",
  "expires_in": 14399,
  "userName": "admin",
  ".issued": "Thu, 10 May 2018 09:20:22 GMT",
  ".expires": "Thu, 10 May 2018 13:20:22 GMT"
}
```



Request

GET http://localhost:3097/api/WebApi/ Send request

Headers

Content-Type	application/json
Authorization	Bearer gJtZs6T5RwvfpwS2rFpx2-NUAWZa-ewfRsxaAONuTE4Gt4iDPEVLig

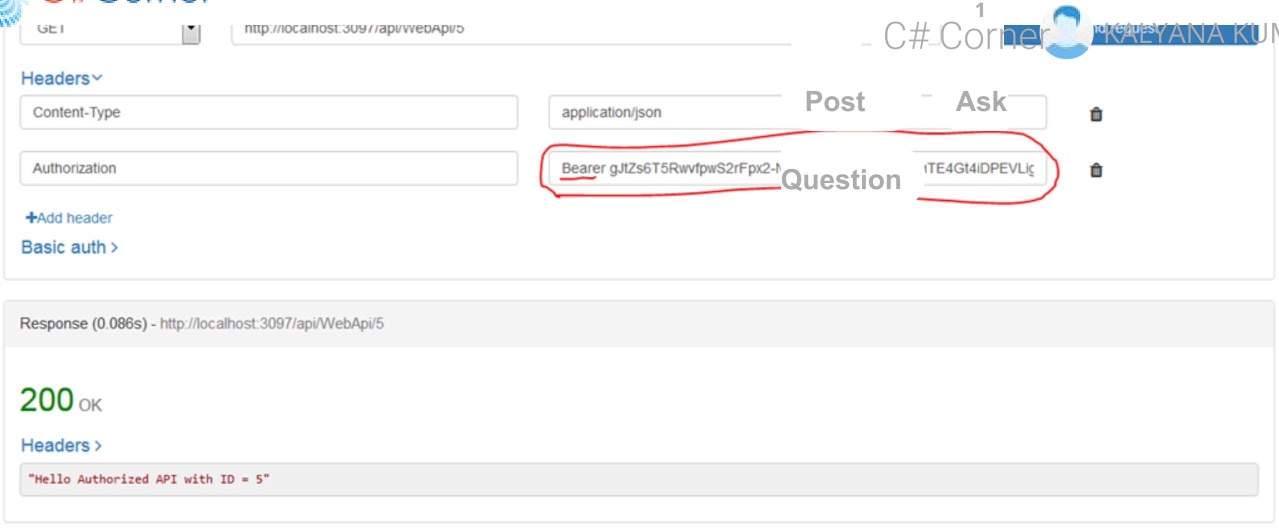
+Add header
Basic auth

200 OK

Headers

```
[
  "Hello REST API",
  "I am Authorized"
]
```

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference


GET http://localhost:3097/api/WebApi/5

Headers

Content-Type application/json

Authorization Bearer gJtZs6T5RwvfpwS2rFpx2-N ... TE4Gt4iDPEVLiç

+Add header

Basic auth

Response (0.086s) - http://localhost:3097/api/WebApi/5

200 OK

Headers

```
"Hello Authorized API with ID = 5"
```

C# Corner KALYANA KUMAR Post Ask Question

Notice in the above snippets that access token is provided as "Authorization" header with "Bearer access_token" scheme in order to call the REST Web API. Also, notice the path when the token is being generated i.e. "{your_site_url}/Token".

Conclusion

In this article, you learned about OAuth 2.0 authorization scheme integration with ASP.NET MVC REST Web API. You also learned about the short comparison between user/password based authorization and OAuth 2.0 token based authorization. You also learned about OAuth 2.0 scheme authentication mechanism for local system users with the Entity Framework database first approach.

Brought to you by: Embed Analytics and Dashboards into yo

Next Recommended Article
Creating ASP.NET Web API REST Service

ASP.NET MVC | OAuth 2.0 | REST Web API

Asma Khalid TOP 500

Computer Programmer by Profession, Computer Scientist by Heart, Fanatic Explorer, Technology Centric. I am a Versatile Computer Science Evangelist. Enjoy doodling with technology.

http://www.asmak9.com/

236 4.5m 4 4

View Previous Comments

30 99

C#Corner Developers: 24 hrs of Non Stop learning, Lightup Virtual Conference

1
C# Corner KALYANA KUMAR

Follow Comments Post Ask Question

Hey Asma, Thank you for such a helpful article. It inde... at help. I have an asp.net website to which i would like to implement oAuth 2.0 authorize endpoint. Please guide.
Shruti Nayak Jun 26, 2020
1917 22 0 0 1 Reply

Hello Shruti - what is your requirement??
Asma Khalid Jun 28, 2020
236 9k 4.5m 0

A great article Asma. This worked fine for me but I have a little different requirement. Instead of grant_type 'password', I want to use 'Client_Credentials' where the scope parameter must be used. Could you please help me in that.
Bhaskar Joardar Jun 12, 2020
1930 9 0 0 1 Reply

Sure, let me look into it first as well.
Asma Khalid Jun 16, 2020
236 9k 4.5m 0

Hi Asma when i am using post method to generate token its shown 404 content not found. your code is everything is fine for me but i click send request showing error like this. Please help me out
jagadeesan vengudusamy May 30, 2020
1925 14 0 0 1 Reply

Make sure that you have configure your SQL server connection string. If you still face the issue I can check it via remote desktop using AnyDesk. You can share your AnyDesk IP via inbox and also let me know your available time according to Pakistan Standard Time
Asma Khalid Jun 01, 2020
236 9k 4.5m 0

What if i want to create an console application and try to consume any OAuth authentication based web service. What will be the code for that
Kamran Rashid May 29, 2020
1936 3 0 0 1 Reply

Kindly look into this https://bit.ly/39NSvnR
Asma Khalid May 31, 2020
236 9k 4.5m 0

Hi, Thanks for the article. Could you please let me know how to do Logout process (and expire/invalidate token)?
C D May 01, 2020
1935 4 0 0 1 Reply

If you want the token itself to be no longer valid, than you would need to mainta

Asma Khalid May 04, 2020
236 9k 4.5m 0

Very good explanation and well working demo. Much ...d.
Burak Köse Mar 09, 2020
1930 9 0 0 0 Reply

I don't want to use the Entity Framework?
mr khan Jan 23, 2020
1919 20 0 0 0 Reply

You are awesome. What a post. executed everything properly in one attempt.
jubula samal Jan 21, 2020
1576 363 0 0 1 Reply

Thank you for your continuous support.
Asma Khalid Jan 22, 2020
236 9k 4.5m 0

Hello, how did you manage to add more properties on the token result? From my result it only shows 3 properties "access_token, token_type, expires_in".
Bryan Gomez Jan 08, 2020
1924 15 0 0 2 Reply

Try executing the provided source code, I am getting these values by default.
Asma Khalid Jan 09, 2020
236 9k 4.5m 0

Also look into "ConfigureAuth" method where most of the settings are made, you might be missing some property.
Asma Khalid Jan 09, 2020
236 9k 4.5m 0

Hello Asama, Thank you very much. very nice article and its complete helpful me to implement oAuth 2 with my web API project. in response i m getting keys "access_token", , "token_type", "expires_in", "refresh_token",. can i add any extra key in response , i want return ID. if yes then how ?
Abhijit Pandya Dec 23, 2019
1915 24 0 0 1 Reply

Thank you for your kind words, You can not add extra tags, I have not seen extra tags by such, but for work around you can attach ID with username separated by comma or underscore as you will get that in response
Asma Khalid Dec 23, 2019
236 9k 4.5m 0

FEATURED ARTICLES

Create Simple COVID-19 Tracker With Angular 10 And Material Design

Learn Everything About REST API

Authentication And Authorization In ASP.NET Core Web API With JSON Web Tokens

Entity Framework DBFirst Approach - Step-By-Step Guide

View All

TRENDING UP

01 An Overview Of Cosmos DB - Start From Scratch

02 Drag And Drop Images From Your Computer To Angular 8 Web Application

03 How To Install Semantic UI In React Application

04 How To Design And Query Data In Cosmos DB

05 COVID 19 Tracker With WPF - Part Two

06 What Is Network-Attached Storage?

07 Storage Concepts In The Data Center

08 What Is Local Disk Storage?

09 Func, Action And Predicate Delegates In C#

10 Delegates In C# Explained

View All

About Us Contact Us Privacy Policy Terms Media Kit Sitemap Report a Bug FAQ Partners

C# Tutorials Common Interview Questions Stories Consultants Ideas Certifications

©2020 C# Corner. All contents are copyright of their authors.