



Quality is our Mantra



Q

X

## o8 Token Based Authentication for Web API's

Aug Deviprasad Kotian | MVC , Web API , ASP.NET , Authentication | Web developers , MVC , Web API , C# , C#.NET | 0 Comments | Return

# Securing ASP.NET Web API using Custom Token Based Authentication

Providing a security to the Web API's is important so that we can restrict the users to access to it. We can provide the security in two different ways:

1. Basic authentication.
2. Token based authentication.

In this blog, we will discuss how we can implement token based authentication. Below diagram shows the control flow of token based authentication.

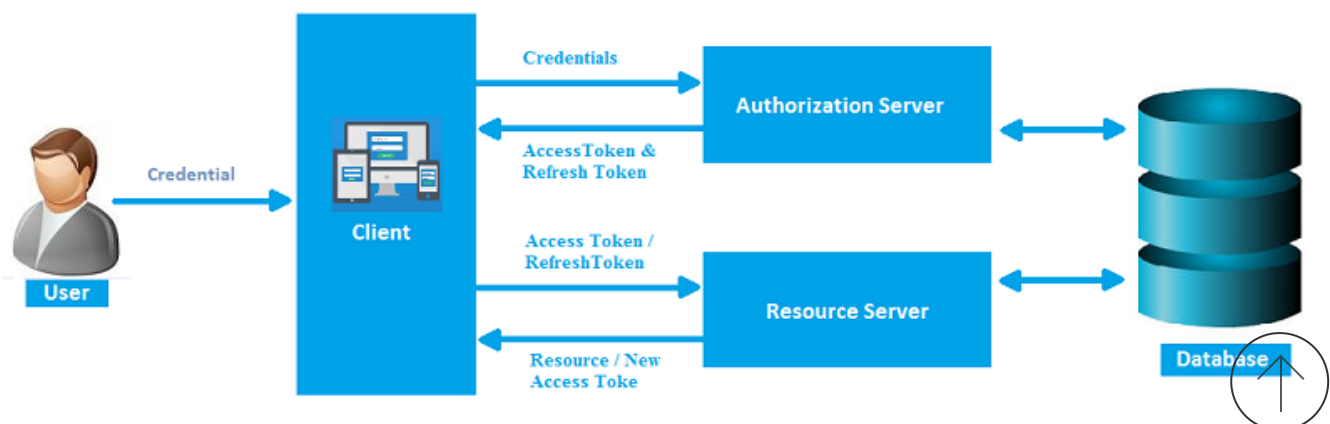


Fig: Token based authentication for Web API's.

## How token based authentication works?

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

Let's see how we can implement the token based authentication for Web Api's:

### Step 1: Create a new project by following the steps below:

Open Visual studio=>File=>New=>Project=>Web=>Enter project name=>Ok=>Select Empty Template=>Check MVC and Web API checkboxes=>Click Ok.

Now the project is created successfully.

### Step 2: Add following NuGet packages:

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

To add NuGet packages right click on References folder of your project then select Manage NuGet Packages and then add above packages.

"Microsoft Owin" is responsible for regenerating and verifying the tokens.

### Step 3: Add 'Startup.cs' inside the 'App\_Start' folder. Then add the following code:

```
[assembly: OwinStartup(typeof(WebApisTokenAuth.App_Start.Startup))]
namespace WebApisTokenAuth.App_Start
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            app.UseCors(Microsoft.Owin.Cors.CorsOptions.AllowAll);
            var myProvider = new AuthorizationServerProvider();
            OAuthAuthorizationServerOptions options = new OAuthAuthorizationServerOptions
            {
                AllowInsecureHttp = true,
                TokenEndpointPath = new PathString("/token"),
                AccessTokenExpireTimeSpan = TimeSpan.FromMinutes(60),
                Provider = myProvider,
                RefreshTokenProvider = new RefreshTokenProvider()
            };
            app.UseOAuthAuthorizationServer(options);
            app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());
        }
    }
}
```



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

Make sure to add [assembly: OwinStartup(typeof(WebApisTokenAuth.App\_Start.Startup))] at the beginning as shown in the above code.

In the above sample, we can see:

1. **AuthorizationServerProvider.**
2. **RefreshTokenProvider**

#### **AuthorizationServerProvider:**

It is the custom class which is inherited from the class OAuthorizationServiceProvider and overrides methods of it. Some of the methods we will discuss here:

- **ValidateClientAuthentication():** Called to validate the client, if context.Validated is not called the request will not proceed further.
- **GrantResourceOwnerCredentials():** This will validate the users credentials.
- **GrantRefreshToken():** It is used to change authentication ticket for refresh token requests.

#### **RefreshTokenProvider:**

This class is inherited from IAuthenticationTokenProvider interface and provides implementation for creating the refresh token and regenerate the new access token, if it is expired.

- **CreateAsync():** This method is responsible for creating the new access token.
- **ReceiveAsync():** This method is responsible for regenerate the new access token by using existing refresh token, if it is expired.

#### **Step 4: Now create api controller and Authorize key word at the top of the Api controller.**

- To provide an authentication/authorization we need to use 'Authorize' key at the top of the action method or the controller.
- If we want to provide authentication/authorization to controller level then we need to add it at the top of the controller.
- If we want to provide it to action method level then we need to add it at the top of the Action method.

#### **Step 5: How to get the access token using user credentials in 'postman client':**

Enter the Url: `http://Hostname:PortNumber /token` in the post man client enter the details as highlighted in the image below:



POST http://localhost:54368/token Params Send Save

Authorization Headers (1) **Body** Pre-request Script Tests Code

form-data **x-www-form-urlencoded** raw binary

Key	Value	Description
<input checked="" type="checkbox"/> username	bush	
<input checked="" type="checkbox"/> password	bush@gmail.com	
<input checked="" type="checkbox"/> grant_type	password	

New key Value Description

Body Cookies Headers (11) Tests **Status: 200 OK** Time: 4685 ms

Pretty Raw Preview JSON

```

1 {
2   "access_token": "G9RMgCJjm87nJdlwng9VHbS1VBwNR3CpHUR9B1D3OxvMV1QuER0YiUbo50bTOyCmU8IXiWgcl3_gCgbGCrAVMUsi8xwg1C1tu2qUbHTRZg0vhe7KdqsYeqSk8ndjc5R46Aj_
3   pk1SU3ggxIDS8n3Ty9Q19QrSD1rQLKGy_IUtqejAvvjgb2godumEA4K-ptLW1ZZE0n23h98mCAa0oJ
4   -pw4mPRn4w14vR5bHuJD5MBjorWiKXFS1ioxhtA105pQ01K6nRKsB2h4nW081WnCZ05Fv2bCS2c4oytx0ZkmxvyrdQngK26DvVGX1-p6F0Eaw",
5   "token_type": "bearer",
6   "expires_in": 59,
7   "refresh_token": "01e81870-4fa9-43f6-94f1-89e109a25d2c"
8 }

```

We need to pass the credentials in the body section.

#### Access Token:

- Allows you to access your Api's without re-entering the user's credentials.
- Each Access token has expiration time and we can set the expiration time in Startup class.
- We can regenerate the access token if it is expired.

#### Refresh token:

- If the current 'Access Token' is expires, then we can get the new access token by using 'Refresh Token'.

#### Expires\_in:

- This indicate the expiration time of access token. We can customize the expiration time according to our requirements.

#### Token type:

- This indicate the type of the token that we need to add in the header.

#### Step 6: How to get the new access token by using existing refresh token in 'postman client':

Enter the Url: `http://Hostname:PortNumber /token` and enter the details as shown in the image below:



POST ▼ http://localhost:54368/token Params Send Save ▼

Authorization Headers (2) **Body** Pre-request Script Tests Code

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> refresh_token	c0d423b1-55d6-4bd8-9a24-953978e60fb7			
<input checked="" type="checkbox"/> grant_type	refresh_token			
New key	Value	Description		

Body Cookies Headers (11) Tests Status: 200 OK Time: 1308 ms

Pretty Raw Preview JSON ≡

```

1 {
2   "access_token": "U44uS70jODN0epfK7mWlzsVH13Dc4d-2jqjnbuT1Kr-vU1yEhGMfhqGtKUGegW0eqv00BU8LULso0wQAS5YhFR0X4n4sUV1bmsEsN3jwh5vBokGh1ap6gE37
3     -Mo_VQXEx9pF5ipJNTf_Axy32uSsgfQL10yLICO3etWvo_kcNuV01K5t7LzhigIdd2bt2iS7L-u56_tio1KzAFZrH
4     -0Zv0299W10BZtn6p3mJNds7tIq00ccUwRLmrCY4tgixdDHd66XJz5GJnr53HjLV23Y0zWbAG0vUTUUhCRksQh-4TEH50gns-xmARVoik3oXx",
5   "token_type": "bearer",
6   "expires_in": 59,
7   "refresh_token": "d756e1e6-67e2-4b6b-a386-dde00bdf804"
8 }

```

### Step 7: How to use the Access token to call the rest api in 'postman client':

Enter your Api URL and then enter the details as highlighted in the image below:

GET ▼ http://localhost:54368/api/Authentication/AuthenticatedUser Params Send Save ▼

Authorization Headers (1) **Body** Pre-request Script Tests Code

Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	bearer eNP01Bqe9luHWqK8pQxUc0sgdt8Leg8Fxf-qhZXd4GX...				
New key	Value	Description			

Body Cookies Headers (10) Tests Status: 200 OK Time: 495 ms

Pretty Raw Preview JSON ≡

```

1 [
2   {
3     "EmployeeName": "Trump",
4     "Designation": "ceo",
5     "Address": "US",
6     "Salary": 250000
7   },
8   {
9     "EmployeeName": "Jhon",
10    "Designation": "manager",
11    "Address": "US",
12    "Salary": 50000
13  },
14  {
15    "EmployeeName": "Thomus",
16    "Designation": "seniorengineer",
17    "Address": "UK",
18    "Salary": 2000
19  }
20 ]

```

**Conclusion:** In this article, we have discussed how we can implement token based authentication to secure the web Api's. The main advantage of token based authentication over basic authentication is that it will avoid the users to provide their credentials multiple times in the application as we are using the access token to authenticate the user. It also provides the security for the web Api's.

## Attachments



1. WebApisTokenAuth.rar 8/8/2017 12:00:43 PM

## Related

ELMAH Integration in ASP.NET MVC Application

Dependency Injection on SignalR

Create custom project templates in Visual Studio

ELMAH(Error Logging Modules And Handlers)What is ELMAH?ELMAH (Error Logging Modules and Handlers) is...

&nbs...

Visual studio installation comes with the various predefined project templates, and we can use one of

[Read More >](#)

[Read More >](#)

[Read More >](#)

## Share

Like 11

Tweet

Share

5

## Post a Comment

Name\*

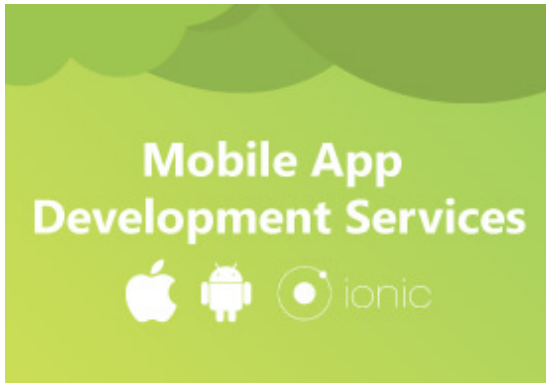
Email\*

WebSite

Comment\*

Submit





Recent

Popular

**How to pass Images from Report Body to Header/Footer in RDLC***Sandeep Suresh Vernekar 2/6/2019 3:35:00 PM*

When using SetData and GetData functions to pass data from report body to report header/footer we can also pass the images

**✂ Angular — Protect your unsaved data***Pranesh Ganapathi 12/27/2018 11:26:00 AM*

using CanDeactivate class route guard to prevent users from navigating with unsaved entries.

**Introducing Azure DevOps Earlier (VSTS)***Nagaraj B Bhairaji 10/22/2018 12:48:00 PM*

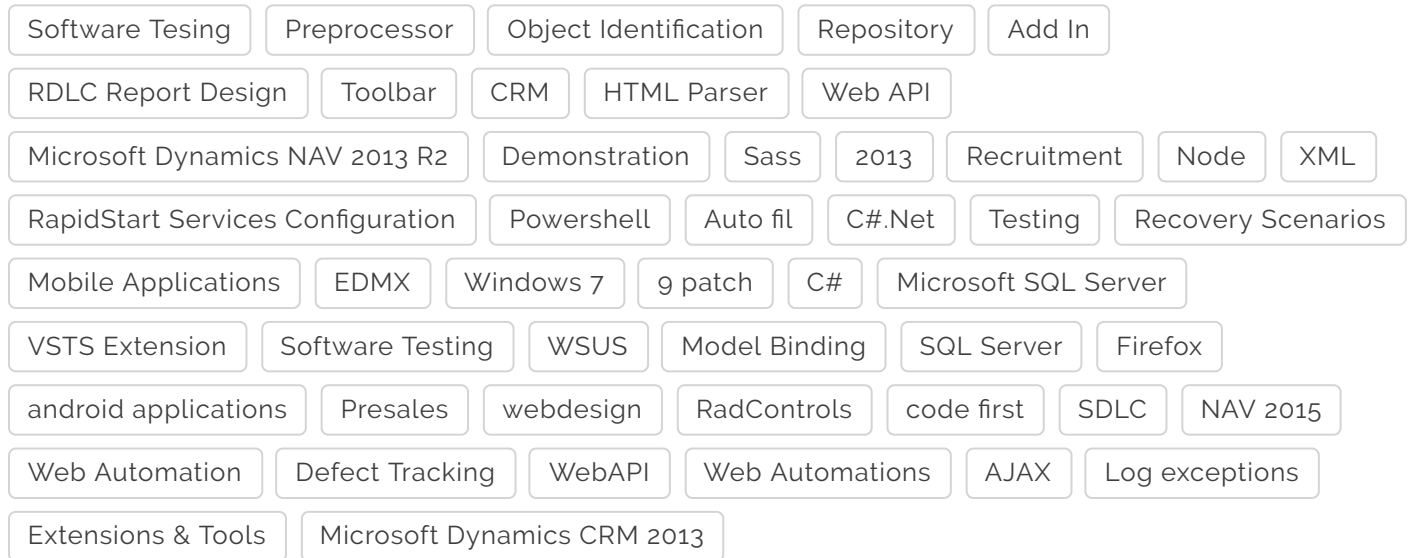
Microsoft recently announced Azure DevOps and an evolution of Visual Studio Team Services (VSTS) which is now called as Azure DevOps Services. All the VSTS users will be able to move or change to new ...

**Create a simple Chatbot App using Botframework & Xamarin.Forms***Darshan S 8/23/2018 10:16:00 AM*

We can integrate Chatbot API with Xamarin.Forms App using two methods, Embed Code with Web View or Native with DirectLine. In this blog, I shall explain how to integrate Chatbot using embed code with W...

Tags





## Monthly Archive

February 2019 (1)

---

December 2018 (1)

---

October 2018 (1)

---

August 2018 (1)

---

## Subscribe

Name

Email\*

Subscribe





# Home

☰ Corporate

---

☰ Services

---

☰ Software Services

---

☰ Mobility Solutions

---

☰ Microsoft Dynamics ERP & CRM

---

## Products

☰ FlowWright

---

☰ e-Conformance

---

☰ Telemetry

---

☰ Mobile Apps

---

☰ Payroll-Addon

---

## Services

☰ Software Development Solutions

---

☰ Staffing Solutions

---

☰ Become a Partner?

---

☰ Feedback

---



☰ Company Brochure

---

## Contact Us

Name:

Name

\*

Email:

Email

\*

Message:

Submit

