Claims And Token Based Authentication (ASP.NET Web API)

khalid_se, 23 Sep 2014

4.88 (13 votes) Rate:

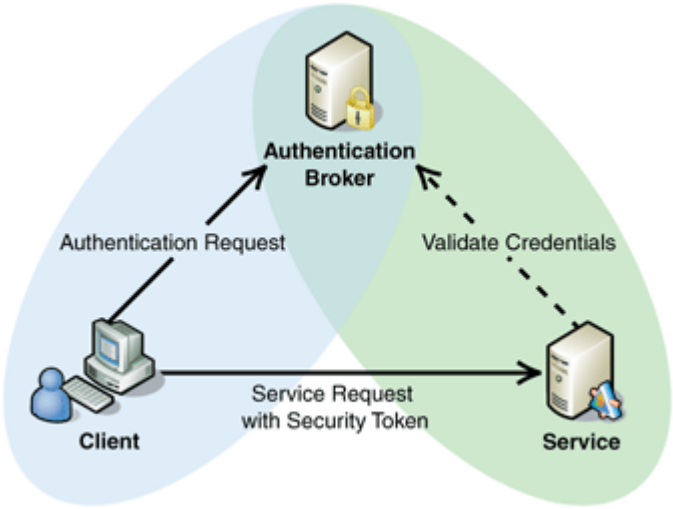
Claims and Token Based Authentication with ASP.NET Web API

Claims Based Authentication

Claims are a set of information stored in a key – value pair form. Claims are used to store information about user like full name, phone number, email address.... and the most important thing is that you can use claims as a replacement of roles, that you can transfer the roles to be a claim for a user.

Claims are part of user identity, so in Web API, you can find your claims in "`User.Identity`".

The most important benefit from claims is that you can let a third party authenticate users, and the third party will retrieve to you if this user is authenticated or not and also what claims are for this user.



Token Based Authentication

Token store a set of data in (local/session storage or cookies), these could be stored in server or client side, the token itself is represented in hash of the cookie or session.

In token based authentication, when a request comes, it should have the token with it, the server first will authenticate the attached token with the request, then it will search for the associated cookie for it and bring the information needed from that cookie.

Sample on Web API

Create an empty web application project (C#) and install the below nuget packages:

- Web API owin
- Owin security cookie
- ASP.NET identity core
- Owin host system web

In the owin start up class, first we will initial web API routes:

Hide Copy Code

```
var configuration = new HttpConfiguration();
            configuration.MapHttpAttributeRoutes();
            configuration.Routes.MapHttpRoute(
            name: "Default",
            routeTemplate: "{controller}/{action}/",
            defaults: new { id = RouteParameter.Optional });
```

Then we will use owin cookie authentication, which will store the cookie and generate the token for us:

Hide Copy Code

```
app.UseCookieAuthentication(new CookieAuthenticationOptions
        {
            AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
            AuthenticationMode = AuthenticationMode.Active
        });
```

and the last line to use web API within owin and register the configuration variable:

Hide Copy Code

```
app.UseWebApi(configuration);
```

Till here, we have a web API application with registered routes and cookie authentication, but we do not have any controller to generate that token, so let's create a new web API controller with login method:

Hide Copy Code

```
[HttpPost]
public HttpResponseMessage Login()
{
  var claims = new List<Claim>() { new Claim(ClaimTypes.Name, "khalid"),
  new Claim(ClaimTypes.NameIdentifier, "1") };
    var identity = new ClaimsIdentity(claims, DefaultAuthenticationTypes.ApplicationCookie);
    var ctx = Request.GetOwinContext();
    var authenticationManager = ctx.Authentication;
    authenticationManager.SignIn(identity);
    return new HttpResponseMessage(HttpStatusCode.OK);
}
```

First, we have created a claim and give that claims a name and id, which may be the user name and the user id.

You can register claims as much as you want. So you could put all user permisions here (as a replacement of roles), or you can put all user information you need like email address, phone... you may use any time in your application, cause claims will be easy to reach and access.

After that we registered our claims list to claims identity, which is the user identity that will store his claims.

In the last three lines, we get the owin context and sign in while passing the claims identity to it. Here owin will store our claims in a cookie and generate a token for that cookie, and the token will be returned in the request body.

At the end, when you request the login method, in the request body, you have something like the below line:

Hide Copy Code

```
Set-Cookie: H32J4J34JH2J#3247987RDHIURWER
```

And this is the token hash.

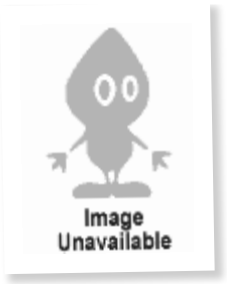
In any request to your web API, now you should send this token in your header to be authenticated in web API.

Thanks for reading this tip.

License

Share

About the Author



khalid_se

No Biography provided

Jordan

Follow this Member

You may also be interested in...

- Securing ASP.NET Web API using Custom Token Based Authentication
- Azure Blob C# Non-Trivial
- RESTful Day #5: Security in Web APIs-Basic Authentication and Token based custom Authorization in Web APIs using Action Filters
- Token Based Authentication for Web API where Legacy User Database is used
- jOOPL: Object-Oriented Programming for JavaScript
- Simple Authentication using Jwt in Asp.Net Web Api 2.2

Comments and Discussions

Add a Comment or Question

Email Alerts

Search Comments

First Prev Next

How does authorization occurs according to said claims?
alex440 2-Oct-14 16:16

How does authorization occurs according to said claims?

Reply · Email · View Thread 5.00/5 (1 vote)

Good job. Need more thorough example
darren_1065 25-Sep-14 1:13

You did a great job explaining this. I think you could create a few more properties to use in your claim object to help give everyone a better idea of what the purpose is. Name and id are so generic. Expand the example to show how you use it customize it and you've got a great article.

Reply · Email · View Thread

Good start...bit more examples
mldisibio 24-Sep-14 21:24

I agree with Taiseer. This is well written so far, but it would be nice to see a bit more complete examples of how to protect a resource by checking for a claim (maybe by an authorization attribute if possible), and also how to harvest the token and send it back with each request from a non-browser client, such as WebClient.

Reply · View Thread

Nice, but...
Taiseer Joudeh 24-Sep-14 17:54

Thanks Khaled for writing this, If you are building an API which will be consumed by different number of clients (Not only Web Browser) then my recommendation is to use tokens instead of cookies.

As well you didn't show us how you can access your protected resource and how you are benefiting from user claims to provide constrained authorization.

Reply · Email · View Thread 5.00/5 (1 vote)

My vote of 5
Humayun Kabir Mamun 24-Sep-14 8:14

Nice...

Reply · Email · View Thread

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin