**Securing API using Bearer Authentication**

January 27, 2023, 8:34AM

1h 24m 54s

 **Aravind Siddoju** 0:03  
Let me know once you start, OK.

 **Bhaskar Verma** started transcription

 **Bhaskar Verma** 0:08  
Yeah, just check it.

 **Aravind Siddoju** 0:10  
Alright, yeah, good afternoon, everyone. As I mentioned, so as part of this session, I'll be covering how we can secure a web application using the bearer authentication. You know, developing an application is quite easy, at least in this current world scenario. But securing an application is bigger and trickiest part.  
That as a developers we face in our day-to-day applications that we're building. Yeah, whenever even like today two years back when I started building this rest APIs. You know, as soon as you create a project things are like pretty straightforward. You just build the controllers, you just build the models and just add domain layers, data layers and everything. Things work fine. But when it comes to authentication, it's always seems like there's some gap in understanding and you always feel.  
OK, there is some magic happening in the background, but we just tend to use that default authentication that our applications are using, but we never really tried. Tried to understand what exactly is happening in the background, so I just want to demystify that as part of this session. So here is the agenda that I have in my mind and feel free to stop me at any place.  
Umm. And ask questions. I just want this session to be interactive and we will have a hands on session on how we can authenticate using the bearer token from a client application. So as part of this session I'll be covering.  
Better authentication scheme and what exactly it is and what exactly is JWT.  
Uh, and then we will be covering how to use Azure Active Directory for securing our web application and then we will do a little hands on. I'll explain the code steps, what we need to perform for securing an API from the application point of view and then we'll be writing a simple client application.  
And we'll try to authenticate and access the Secure API using the client app.  
This would be the agenda for today and we will build on top of this in the coming sessions, so when we want to discuss on robust authentications in the future.  
Umm, so yeah, I'll be using these tools.  
Of for hands on today. So I I'm going to build a .net core web application for the which I need a .net core SDK 3.1 and .net core. Sorry that one runtime as well which comes as part of SDK in major cases. Then I'll be using.  
The Visual Studio and I'll be using an Azure account so.  
Since we all have an enterprise, we just showed that question. So we'll be having a free Azure account.  
Which which has some $13,000 credit.  
I I think $10,000 credit which you can use to just try out new things. So this is what I'll go as my default to do. Any purses and a postman to simply see the endpoint that you're getting accessible or not. It's not mandatory. Even directly hit it on the web page itself.  
So yeah, without wasting time. I just want to cover different types of authentication schemes that we have. So there are a number of authentication schemes that we could have used. I mean, but if you just.  
Put it into four different categories, so we have like these four that are showing up. We have a basic authentication, we have different digest authentication scheme. We have built based authentication scheme and we have NTLM. So basic authentication is nothing but a simple username and password where user you request a user to just provide a username and password to you and that authentication scheme in codes both the credentials in base 64 format and you try to decrypt it.  
When when the data is passed to you as the request header.  
And here like it's up to you. I mean, when, where to stores those details of user and validate them you use some SQL Server or maybe some database just to validate those details and then we have digest based authentication scheme which is just an improvisation on top of basic authentication.  
Where?  
You apply a hash function to the sensitive data that you have. It might be username, password or any any particular information that a user tried to send as part of the header to you.  
And again has decrypted algorithm will be used in the in the server code and we will be decrypting the data like the credentials and we'll validate them.  
Then we have a beared based authentication and this is very particular to.  
Using a token instead of the BURNISHERS to authenticate where anyone is in position of a.  
Uh, I mean, having the right access with the right token to access a particular resource of our Web API can access the resource so.  
Uh, a Web API? A secure web APA associated app registration will be issuing the tokens and we will use that token depending upon what type of roles that you define to request access.  
That's that's pretty much on the pair authentication and then we have NTLM based authentication so.  
So.  
This is nothing but using Windows credentials to authenticate any endpoint or Web API. This authentication is also perfectly fine, so both last two, bearer and NTLM are the modern based of authenticating the web endpoints, but we're not going to discuss NTLM right now, and one major drawback with NTLM based authentication is most of the Windows credentials are proprietary license based, like where your credentials will be defaulted to one particular company and the organization so.  
You might not. You might be restricted to access you resources so so we are going with an token based authentication in the current.  
The way that we're going to build.  
Any questions so far?

 **Deepak Kunwar** 6:30  
So one thing I want to say is that in the instance side we talked about .net 3.2 point one by one person. And since we are creating a new app or anything as since I was checking the list I have posted the link as well. So this is the .net 3.2 already reached its end of like the last month. So if we are going ahead we sort of go with the letters version of .net is one ways.

 **Aravind Siddoju** 6:41  
Umm.  
He yes.  
Yes.

 **Deepak Kunwar** 6:59  
Seeing.

 **Aravind Siddoju** 7:00  
Just be perfectly fine, but I didn't have the setup installed with me, but most of the things will remain same. The APS were not deprecated.

 **Deepak Kunwar** 7:05  
OK.  
Yes.  
So a part of this day this session, but I think it's good, but if you are if you get a chance to update that .NET Framework in your new new things that you are creating then.

 **Aravind Siddoju** 7:10  
Is that OK?

 **Deepak Kunwar** 7:22  
And to keep it in mind for everybody.

 **Aravind Siddoju** 7:22  
Sure.  
Yep.  
Any other questions guys so far?  
Alright, so I just want to cover a little bit on the bearer token and the JWT that we basically use for any JSON payload that we try to pass as part of HTTP header. So as I mentioned, the use of tokens in bearer identification is centralized concept. A token is issued to the requester. In this case Damon client, direct client application and the client who is the bearer of the token then presents it as a secure resource.  
But in order to gain access, so the token is like his resource to access any half hour actual resources. So he just uses the token to validate it from our endpoints and that's how we authenticate the particular client. So I'll show in step by step how token is issued, how token is requested and how we actually validate the token from our.  
Again, what exactly is Chadha PJ ability stands for Jason Web token, which is again just an encoding standard.  
For tokens that contains a Jason payload. So whenever we try to use a Jason payload when we when we try to send the information in HTTP so we we particularly try to use the JWT token we don't have different other tokens as well in within the bearer authentication scheme itself. But we will be using JWT if at all leave under the user JSON will work and in our application current application we would be using a JSON payload. So I'd be going with.  
Get ability bearer authentication.  
Yeah. Again, in short, the period authentication is just an authentication scheme that makes use of a pair is in this case the air is nothing but a token, and generality is a specific implementation of the bearer token, wherein in case of JSON payload.  
So that's that's pretty much you need to know about the theory related to the authentication schemes and the difference between bearer token and in particular what exactly is JWT.  
So here is what we are going to do.  
Ohh, they're going to create a client application which in in my case would be a simple clientconsoleapplication.net core, but it would be proving its identity in Azure and.  
Uh, how that application is related to it and then it will be.  
Requesting an access token to Azure Active Directory and it tries to get access token.  
For accessing the the Secure Web API.  
Linda, once it gets the token, it uses the HTTP header and passes the token as part of the header and tries to access that resource.  
So this is what we're going to build.  
So here is the step by step thing which I will be building today. So I'll be creating a new API in .net core and then I'll be registering our API in Azure Active Directory. Then we will see how to expose that API Azure and how to update app roles permissions. Actually in this case we will talk about it while doing the hands on while creating the app registration and then again we will.  
Add some code in our Web API to use that particular authentication to the changes that we made in Azure and then we create a client application and again we create a client app registration client ID and then we.  
We write code in the client app to authenticate to its app registration. Then the requested token and then access the OR .net secure webapi.  
So yeah, as far as content, yeah, I just start up covering this, this amount of basic theory so that everyone would be aware of what we are building. Are we clear so far?

 **Deepak Kunwar** 11:37  
It's.

 **Aravind Siddoju** 11:38  
Alright.  
Nothing.  
Alright so.  
Just had templates ready so that I don't want to waste much of the time. Let me know once you see the.  
Publication.  
So here is a simple .net core web application.

 **Bhaskar Verma** 12:00  
I don't think. I think, Arvind, you haven't shared your screen.

 **Aravind Siddoju** 12:03  
Ohh sorry one minute.  
Let me know once you see it.

 **Deepak Kunwar** 12:18  
Yes, for us.

 **Aravind Siddoju** 12:19  
And she tried.

 **Bhaskar Verma** 12:20  
Yeah. Yes.

 **Aravind Siddoju** 12:22  
OK, yeah, this is simple.net for web application.  
As soon as you just create the the template, as soon as you use the template of .net core Web API. So this default thing will show up. I mean I haven't had any code yet, so this is the simple template that will show as soon as you.  
Select the Web API NVR studio with .net core framework.  
Just cover a minute. And what are these things for a minute? So this is the basic controller that will come as a default. This is the route for that controller and if you just.  
Go to the startup class. This is the entry point before it reaches to controller where we configure the services where we configure the different other resources that we want to use where we configure the endpoint, route details and everything would be added as part of the startup class.  
And then we have launch settings. So this is a simple page where you can provide the default application URL to open up whenever you click on run and there are few other things that you can set over here and profiles and some other information related to application metadata and.  
What are they missing?

 **Bhaskar Verma** 13:36  
I don't one question I have is that when we are setting up the depending dependency injection during the startup in the startup class, are we loading this launch settings or Jason file?

 **Aravind Siddoju** 13:38  
We'll head.  
So this will be loaded. This will be loaded. By this I mean it doesn't have to do with dependency injection.

 **Bhaskar Verma** 13:55  
Good.

 **Aravind Siddoju** 13:57  
Your dependency injection is to decouple the interfaces and the classes.  
Avitia.

 **Bhaskar Verma** 14:06  
Yeah. Good.

 **Aravind Siddoju** 14:08  
OK, I'll write. So I'm just gonna run this.  
That's tomatoes.  
It's OK.  
Umm.  
One minute.  
Let me run it again.

 **Bhaskar Verma** 14:54  
Ohh just one question I want. Sorry to interrupt her so this is there any specific place where where where it is mentioned that we are loading this settings file or is it part of the framework?

 **Aravind Siddoju** 14:56  
Go ahead.  
Uh, come again? When? When? When you mean Settings page is launch settings or?

 **Bhaskar Verma** 15:11  
This launched India.

 **Aravind Siddoju** 15:13  
Thank you, chatted and startup class. I haven't looked it in particularly.

 **Bhaskar Verma** 15:18  
There might be, I guess, part of the new right.

 **Aravind Siddoju** 15:21  
Yeah.  
Yep.  
So yeah, I mean, since the last setting has a route directly to the weather post test controller, I mean you can add your own controller. I'm I'm not going to spend time on adding controllers, I'll be focusing on authentication for now.  
Here's the get method. That girl called by default and there is the data that is showing up. So let me open that.  
So this got called by default. This get method and we are showing up the data. What do you want to return from here? It's like a date range and it's like a weather forecast, an array of whether details showing up with date, temperature and summary some random data.  
Alright, now this endpoint can be accessible by anyone.  
Considering now there isn't any authentication scheme, we added nothing, so this this as long as you post it somewhere and you expose and you call this controller and this will be returned by default so.  
Now we will see how we can use Azure Active Directory to secure this application.  
I'm gonna start this application for now.  
So before before I even jump into app registration. Uh, I mean, why we are using app registration and what what exactly the need for that so?  
In Azure Active Directory, if you want to, you know define your application or if you just want to give a definition kind of thing to the application that you're building. So I have to registration is away. So every instance of app registration will actually represent an application.  
It it, it doesn't have to be. You know that the direct application that's that's been running but.  
It it is finished and it it is like 1 to one map to the application that you're building and why we are using Azure app registration so.  
We are using Microsoft as considering here I mean don't consider it as a third party, but here to my application and to the client. I'm putting the Microsoft Azure Active Directory in between to authenticate that client. So it's like a middle layer where I'm telling OK, I'm trusting Microsoft to prove the identity of the client. For that I will be creating an app registrations in Azure and I'll be granting permissions to the client. Whoever want to register.  
So that's how.  
Be bring the Microsoft to prove identity of the clients while accessing the secure resources.  
The second thing that you want to note here is like so we are trusting Microsoft for authentication, but it's not the converse like Microsoft is is not supposed to trust that application that you are building.  
To prove the identity like so if if I say if I bind 1 app registration to my secure application it simply means that I believe that Microsoft will validate authentication and it will.  
Provide the right token for authentication. It's not, it's not the opposite way where.  
You just simply rely on Microsoft telling OK my my I.  
The application is trusted by the Microsoft and.  
It's not converse. That's what I was trying to tell. Like you have to configure it properly. It's not the other way of.  
So I'll, I'll and I'm going to create a new app registration over here, let's call it.  
Server which is very.  
Umm particular to the?  
New application that I just created a secure server application. So far nothing is like related between the app registration that I've created and the.  
The application that I have on my dev machine. So as soon as you create an app registration so it will have its own identity. I mean consider it as an application name on the Azure organization. The directory which way the application is put up.  
In, in a very it behaves as an individual.  
A client within the arc itself, so it has its own ID, has its own resource ID URLs, and there's a way to access and prove its identity, prove the identity on the Azure portal.  
Once, once you have an app registration created. Ohh here.  
What? What? Further, we are going to do. So first, first thing we are going to leverage the app registration that I have created for my secure server to you know the code or enable the JWT based authentication on my application. So here is what I'm going to do.  
So in the startup class I'll be configuring this application to.  
Use the right.  
Azure app registration for generating the tokens, issuing the tokens and decode the token that comes up whenever a user tries to hit the secure Web API and.  
For that I would need JWT token nugget.  
I'm going to install that.  
Indication.  
Not sure why it's taking longer time.  
Let me try the full name. I noted it already.  
To.  
If some time, so I'll be using that.  
Change had no sound.  
Yeah.  
Test. Can you hear me?

 **Deepak Kunwar** 22:32  
Yes, yes.

 **Bhaskar Verma** 22:33  
Uh, yes, seven. You can hear it.

 **Aravind Siddoju** 22:34  
OK, I'm I'm not sure why it's not showing up.

 **Bhaskar Verma** 22:37  
Under uh, in the meantime, I win the float out, so check that you were showing in the charting.

 **Aravind Siddoju** 22:40  
OK.  
OK.

 **Bhaskar Verma** 22:43  
Ohh which which step are we on?

 **Aravind Siddoju** 22:43  
But let it come up.

 **Bhaskar Verma** 22:46  
Are you on the exposing part?

 **Aravind Siddoju** 22:46  
So yeah, I'll I'll come back to that.  
So we haven't came till exposing our API. I just created an A an API in Azure app registration. So let's let's do the remaining party medium. Actually I want to show this part first but it's taking some time anyway.  
So.  
Any question?  
OK so here here is the same app that we created in Microsoft Azure, so I'll I'll add in a bit once this nugget. OK, this nugget is, but let me install in the meantime.  
And this is not supported for .net Core 3, so I'm going to use the supported version.  
OK, let me accept that.  
OK. In the meantime, UH, Bhaskar, regarding your question. So we are at Step 2 right now once. So we just registered a new app API in Azure Active Directory and we are yet to map the.

 **Bhaskar Verma** 23:57  
Didn't.

 **Aravind Siddoju** 24:00  
We had to.

 **Bhaskar Verma** 24:02  
100.  
Only bond with them.  
It's working.  
People working with.

 **Aravind Siddoju** 24:08  
Are you hearing any noise?

 **Deepak Kunwar** 24:10  
And you can mute by not speaking Hindi.

 **Bhaskar Verma** 24:12  
I think your voice, but it broke in between.

 **Yashasvini Rathore** 24:16  
Bhaskar, there is some boy noise from your end. Can you mute?

 **Bhaskar Verma** 24:19  
Ohh.

 **Aravind Siddoju** 24:23  
Alright, so yeah, back to the procedure. So first we create an API that we created secure server.  
Then we created an app registration in Azure Active Directory, so that simply means we although we haven't like made the relation between the the Secure Application network option that I have built and I have registration but this is going to be the entity that's going to represent my application in the Azure portal. So this will consider this as an application in at Azure Portal level it Azure Active Directory level. So as soon as you have.  
It will be having an application ID as I just mentioned and the tenant ID points to the organization or the directory where it will be pointing to. So and.  
As soon as you have an API or an application created an app registration. So other thing that you need to do is expose an API. If you don't expose the API it won't be available, it won't be visible for any other applications which are trying to access this API in the Azure portal. So I'm going to expose the app registration that I've just created. So for that first you need an application ID Uri. This will behave as a default Uri to identify this.  
Doors Images, Active Directory. So I'm going with the default name that Azure provides me which is nothing but an API colon forward slash followed by the application ID which is just showing you. So I'm gonna save that.  
Is it done?  
So now I just created an app. I exposed the app to the outside world, which means like anyone within that.  
Yeah.  
Active Directory are in the directory organization with the right tenant details they can access my application with the right set of authentication right set of validations which I'll show in how they will do that.  
So this is done and then I'm going to add roles to my application, so roles are nothing but.  
A kind of you know.  
Uh, definite set of rules on who can access your application and what type of.  
A client can access your application, so if I just try creating an app for let me.  
It's really to you.  
Sensory tests.  
Ohh some you know application or whatever it is.  
So this is just a role name and here you you specify what type of.  
Applications are clients can access your API using this particular role. So role is the kind of.  
Uh, you know, if a default it's not a default one, but rather a permission. Whoever has this permission to this particular app role in their app registration can access it. I'll show that also in a bit, but consider it as a way to expose our API with the right set of permissions. Here I'm just creating a new app role, naming IT application role, or maybe call it.  
Uh, or a client or whatever it is, and I'm going to specify what type of members can access my API using this app work. So there are three different ways, right? So users or clients can directly hit your Web API rather an application. A different application can directly talk to your Web API and then we have different role where both users, groups and applications can all talk to your Web API. It depends on the role that you specify over here a token will be generated based on that.  
So I'm going to.  
Select applications for now, since I'm going to interact with the client application but not with user credentials. So what exactly and to provide here? It should be anything like.  
Some.  
That's the plan.  
It doesn't matter what name you you provide, but so this value will be used while generating the token. I'll show you in a bit how that will be and.  
Test.  
Turn off.  
That's it.  
So I created a new role for my server application in Azure Active Directory and this role will be exposed since I already exposed my API, this role will be exposed to a different app registration, so they want to see they want to hit.  
So this particular application, so this is all I want to do as part of my app registration that I've created for the server. So I just created a new app registration. It will have its own application identity showing up. It will have its directory in which the application is residing and I just exposed my application in Azure Active Directory and there is a.  
The a unique Uri to identify my resource in Azure Active Directory. Then I added a new app role to let the applications not the users to access my.  
Application using this particular role.  
So yeah, this is all I did and this this role will be used to generate the token.  
When they when they request first so.  
Uh, so before I I even jump into the code so just want to make one thing clear. So Azure Active Directory is the one who will be issuing the token. So the directory will be again recognized with the tenant. So any application which is residing in particular directory. So this directory is responsibility to issue the token for accessing this particular application within its domain. So it's all.  
The directory are in in. In our case you can just simply call it Azure Active Directory.  
Uh is the one who would be seeing tokens to access your application, so here is all I did, but I still haven't related our correlated my application that I have just built with the app registration that I have created on the Azure portal. So now I'm going to configure that.  
So.  
I'm going to.  
Add authentication now.  
And the scheme that I will be using as I just mentioned it will be JWT authentication so.  
JWT.  
Better.  
Fault.  
So every all the libraries related to generally bear authentications will fall under.  
Oh oh, Microsoft authentication.  
JWT beta nugget so you will use this nugget to configure your services to acquire and to be serialize the tokens based on JWT authentication scheme.  
I'll be using the retention scheme.  
And I'll be adding the code to.  
Creative error which?  
And kind of and. So we should have using options.  
Feel free to stop me if you don't understand anything.  
I'm going to set few options for JWT based authentication in my service. So what all I'm going to configure?  
So.  
I need couple of things, one being the audience. So in this case the audience is our application, are the secure web application.  
Which I'm going to do.  
But before that.  
I'm going to add a the values in my app setting.  
Uh, different values that I need, so I just need a couple of values just to identify my own app registration on the portal. I just need the application ID URL which is again simple to the same application ID just preceded with the API. So I'm going to use that.  
And then we have an authority. Authority is again nothing but. So the one who has the authority to issue the token. So as I mentioned, in our case, who is the one? So it's it's the directory, it's the Azure Active Directory who would be issuing tokens. So this guy is the one who will be issuing tokens for my application, which is the setting in in this particular directory. And the library that we are using is msal Microsoft Authentication Library and now.  
It we should provide a paid to it for identifying my particular directory in Azure. So I'm just going to show you.  
Simple information that you that would be useful for you. So you can just search for this authority is a route or endpoint that will be used by the msal to identify the particular Azure Active Directory.  
I know I do portal, so this will be by default, uh Uri followed by the directory so that that would identify your particular Azure Active Directory. Issue the token. So I'm going to use the same in my application.  
So this will be the default and I'm going to add my.  
Directory ID value or there?  
Right.  
So.  
So my startup class I'm going to you just let's you those values from the configuration. So configuration builder I configuration is the one which generates the config.  
A dictionary out of the app setting Jason file that we have. I'll show in client application how we can write our own configuration builder using app setting. But bear with me for now so I'm going to use.  
30 society.  
The colon provides a a hierarchical way to access the resources within App Register. Sorry app setting structure.  
And we have options that.  
I need to specify the authority as well. The one who is issuing the tokens.  
Configuration.  
Aid.  
It's this value.  
Alright, so I just set up only couple of details over here. I just can't figure it authentication for my service. The scheme used is stability where authentication scheme and I provided a couple of options which are required for the bare authentication. This is where my application will know where these are all my details and these things will be used.  
To secure my application these this guy will take care.  
Uh to issue tokens and to deserialize them when someone hits my service?  
The audience I just mentioned, it's free our application, so it will. It will be again pointed to the Just app registration, client ID and authority is simply the one who has authority to issue tokens. I'll just mentioned it's Active Directory, so just provide the directory name which is simpler tenant ID. Remember like if if multiple arcs are trying to access applications in different areas so you'll be having multiple tenant IDs and you have to provide those details particularly in the client.  
And server to access authentication, but the server settings will remain same, nothing will change. So these are all you need to configure.  
Uh, I, I added. I can't figure the service, but I never asked my application to use the authentication, so as soon as you add nothing will happen. But you need to enable authentication over here.  
Probably do.  
Tell my app use authentication.  
Alright.  
Let's try to hit it.  
OK, it's still showing up. I'm gonna pass here for a minute to let does anyone of you know why it's still showing up? Any guesses?

 **Mohit Gupta** 37:55  
It must be silent here.

 **Aravind Siddoju** 37:55  
So we enabled.  
Enabled authentication. We provided JWT bearer authentication but it's still the resource is still publicly accessible.  
Go ahead with your things sometime.

 **Mohit Gupta** 38:08  
I can think of two things. One, it may be silently authenticating or the API has to be marked for authentication.

 **Aravind Siddoju** 38:16  
Correct, correct.  
So as soon as you add.  
I I as soon as you configure the services are telling to use authentication to use JWT bearer authentication. It doesn't mean that everything is like now.  
Uh, like, secured, like nothing is accessible. Everything is just so it's not the case, so it's just.  
We we just provided or we just added that ability to our application, but we have to use that for that API to access that. I mean every controller kind of exposes its own different requirements. So necessarily you don't want to restrict everything. So there comes a different.  
Decorator, so this is provided by default by Microsoft .net Core SDK itself. So authorized is the one which actually.  
Provides a way to use the authentication that we have put over in the startup dot CS and to verify it whenever someone tries to access this endpoint. So this is the real world scenario. I mean whenever you create a Web API, you necessarily don't want to restrict the people to access the main page or the login page itself. So once you once you let the user login, that's when you restrict based on controller. So it depends on like.  
No, what you want to restrict so you have to use authorize over there and this authorize is going to do the magic for us.  
I'm going to put the same thing in first man so that.  
You'll have some idea on.  
What?  
This bunch we're getting.

 **Mohit Gupta** joined the meeting

 **Aravind Siddoju** 40:22  
401 unauthorized. It's not 404, it's not.  
Uh, success code, but particularly unauthorized code, which means our application is trying to use some authentication scheme and we are trying to hit it without any you know.  
Ohh authorization.  
So that's that's when uh application will written as a four hour.  
So yeah, now we finished one part we we secured our API and how we secured our API we we generated the kind of a replica in our app register Azure App registration.  
For our particular application and we we just binded these details to my application.  
So every time my application is running, it tries to use the details related to its own application to check whether the tokens are issued for this particular.  
Uh audience, which means this particular application ID with the authority or the directory being the the directory that is showing up. Here I will try to see whether the token is valid or not. And there are different ways. Actually you can restrict in deeper. I'm going to cover that in our future cases where you can, you know specify what type of people within authentication itself so.  
A home or or what particular type of?  
Of people can access your resource.  
Uh can authenticate your resource, but cannot access your.  
Are API so there are there are different parameters that you can specify in Azure app configuration like policies, schemes and different things and different depending upon that. Again you can make further restrictions to your own Oba. Sorry I forgot to tell one more important thing so so there is a difference between authentication and authorization. Although in this case.  
Uh generality better itself is doing both for us. But if you see in the configure we have two different things, one being the authentication and one being the authorization differently, so authentication is.  
Letting an application access.  
Your.  
It's not even letting it's like letting the application to prove its identity. Telling it OK it is valid client to even touch my application or to even hit my application. Whereas authorization provides an ability for that client to access a particular resource. So and both will happen using the same decorator that we used over here the authorized. But if you want to restrict for the particular user, you can add.  
Umm, the principles schemes in our app registration which I'm going to.  
Uh, not going to cover much for now. That will eat up most most of my time. I have to create a client app as well, but so you can you can specify the roles administrator over here and these configurations will not come up for my default subscription and depending upon that again you can add code over here to bot people can access and you can go and specify the level of accessibility for a particular user in authorization. So that's where authentication and authorization are different. Authentication is basically.  
Just telling a client. OK, you are a valid user. You can hit my API but you can hit my application but again for eating an API if you don't have a private access I'm going to allow it. So that's where authentication and authorization will be different. Any questions so far? So we are done with creating a server application and resting it using an app registration and we are looking for only clients who will provide me the right token and only I'll authorize them.  
Uh.  
Are we clear so far?

 **Deepak Kunwar** 44:29  
Yes.

 **Aravind Siddoju** 44:30  
Alright.  
So I'm done with server, so let's see a different application who wants to access.  
Uh, the secure API that we just built.  
So I for this case I just I haven't used a new independent. Again that's waste of time, right? I just need a way where an application is trying to access a different application. So I thought console would be a simple way to demonstrate that.  
So now this is a different application and this guy wants to actually access my technique and files now. Now we will follow the same procedure.  
What I'm going to do is I'm going to create a new app registration for my client as well.  
And you can stop me if I do any wrong. Now you know some part of app registration or how they do it.  
I'll call it client.  
App maybe?  
I'm going to register it and again these are the settings that we can.  
Not used to restrict this app registration accessibility to the people within single tenant multitenant and this won't be applicable for this. But if in our.  
Microsoft account so this this will play bigger roles and every organization will have its own admin for accessing the access request access.  
I'm going to register a new app registration.  
So here I created a new app again for my client, my console, whatever it's trying to fit the secure API. So this guy will also have its own application ID. This guy will have its own directory and in this case, since this is my default.  
Ohh subscription so everything will fall under one directory default directory for my private Microsoft account that I am using so that I have to remain the same but most of the cases is directly might not remain same.  
Again, again the application ID will reside in the particular directory where you are creating the the app and so here is what we're going to do. Now imagine this application wants to access.  
My secure application or secure Web API that I just created so there are a few two things that you need to do. One, we already exposed our app registration, the server app registration, and we also defined an app role for that server app registration telling. OK, anyone, any application, any client application? Not the users, not users and groups, but we defined a role for only applications to access. So when I mean application the application is.  
Again, nothing but a simple app registration defined in.  
An Azure Active Directory, so just don't confuse yourself with application. So whenever we we say an application it's it's a different app registration we are referring to in Azure Active Directory. So everything every application, every app registration is an application to.  
Uh, active direction?  
So it will read it as a user, it will feed it as an application.  
So I'm going to do one thing we already exposed, so here is what we're going to do. So I am going to click on API permissions for my client app.  
And I'm going to add permission for my client app.  
To access the server so.  
OK, I'm going to come back and I'm going to show one more time. So. So I think one thing that you need to understand in app registrations is so here is here is a client app that actually should request for permission not the other way around. I mean it's still goes other way around but.  
What I'm trying to tell you is like, so we we exposed an API server API and we just made it available to the world telling with this client role anyone can access and in my client app I'm just going to.  
Enable that. I mean telling. OK, hi. I'm granting permissions to access this particular app registration from my application, so that's what we're trying to do here. I'm just adding a permission to access my secure Web application from my client. So here is the server approach that we created.  
So if you just click on that, so this is the role that I particularly created to access my server application. So I'm going to select that particular role.  
And I'm going to add permissions, so you might be confusing how come a client is.  
A being authoritative over here telling OK it will access this particular application, so this is not the case in the real world. Considering that so you'll be having an admin consent, you'll be having a grant access consent and everything coming up in.  
Uh, a siddoju Microsoft or the arc wide account? Whatever we use so there admins are different.  
Directories will have to provide permissions before you even access this slide, so you see over here, right grant admin consent to default directory. So in this case for both client and server I'm the one.  
Uh and the admin, I'm just accessing granting the access, but in case of the real world or in case of the the new API that you're going to build in in our.  
But our team or whatever, so whoever is part of the server.  
Uh directory admin so they have to grant permission to even access the client to to even access is have people off the server on the client side, so that's where like we still have the server server being the authoritative guy to grant consent or admin consent to access the directories to issue the tokens.  
So I'm going to go into plant it for now.  
OK, so here's what we did.  
If you wanted a server application, we just expose the API. We create an app role to access my Secure API to issue tokens. So in the client app I added API permissions to access that Web API using this client rule that I created and I got admin consent to access that. So that's that's all we need to do.  
Now.  
I'm going to build a client application.  
So I just added this comment so that I just forget the steps so.  
Alright.  
So you might be knowing by now I need the details of my clients.  
Administration and Mike, sorry my client application ID and my client tenant ID, so I'm going to create.  
A simple file, so added a shortcut, so I'm going to use Shift tab too.  
I'm going to create an app settings start just on file.  
Alright, so I just put up a few things over here, so I'm going to use these details.  
I'm going to fill up these details. I'll I'll show you in a bit where they are getting used and you might be already aware around client ID and authority by now, so this will be finding to my application ID of my client.  
Look at what's the other for now and again the authority being login microsoft.com followed by tenant ID. So I'm going to just copy some information from my.  
Previous settings.  
Loginmicrosoftonline.com.  
My client application.  
Followed by the directory ID. So this would be again the same ID. Since this is my private.  
No subscription account.  
And then you know the client secret. So why I need a client secret?  
I am actually creating a console application and how would this this console application would be identified in Microsoft Azure? Although just I created a client just like I minded my details in server for the event console to prove its identity you need a way to.  
Through the identity so.  
It's like so the identity will be proven by different ways where you can pass the certificate, secrets or other Federated credentials. This I'm not aware though. Pardon me. So secret is a way. It's it's nothing but a password to the app registration. So as long as you have a a secret with you so you are just simply used to simply add a code telling OK, this is a secret that I have for this.  
Application ID in this particular tenant so.  
This ticket will be validated for this particular application ID and tenant and it will.  
But OK, the console application is the right client, so that's how like Azure will know Active Directory will know OK console is the right the console application that I'm building is the right client who is trying to access the registration. So I'm just going to create a client again. This is no more new for anyone team, so we use Certs. We use clients client secrets to access our applications.  
So I'm just going to create a client.  
Whatever correct, doesn't matter. That's client.  
So I just created a secret so and you just need to copy the secret as soon as you create it. So once you refresh it it it won't. You will not be able to copy it. So that's some security issue thing that Microsoft added. So I'm just going to.  
In the secret. Oh yeah.  
Ideally, you should not waste any secrets in in your application. Rather you should use keywords and you should put your secret in within keyword. But yeah, I'm not going to use that for now and then we have server resource ID so.  
Hope you hope you have got some idea by now. So I want to hit the server application ID using these details. That's what I'm trying to do. That all details that I'm trying to gather.  
So.  
This is my server app registration. I'm going to go to the application ID, URL, resource ID.  
I'm going to start.  
And.  
I'll tell you in a bit why I'm adding the default over here.  
So this this is the route that M cell uses if there is any hierarchy to identify that particular resource ID or the target resource ID in the in the Active Directory. And I'll show you what error it will show up if we don't use that. All right, I created an app setting Jason.  
And I'm going to use these configurations so.  
But I do that there is one more thing you need to do if you don't.  
If you don't select copy always, it won't be copied to the bin directory and you will not be able to use it. So just keep in mind too, always copy the app settings file whenever you create one.  
So you have one, uh, upsetting to your son. So I'm just trying to trying to write a code.  
To.  
Ohh.  
And table uh secure authentication.  
It stopped me. If you don't understand anything while I'm writing this quote, I'll I'll walk through each line of the code that I'm writing.  
OK so.  
I'm going to 1st create a.  
Of configuration.  
So as I mentioned, so there is a configuration builder you might have seen already in my server application which comes by default one minute.  
That's.  
Which gets generated by default as part of the Web API directory. So this guy holds the the configuration settings from appsettings that Jason and this is directly binded and I want to show you how you can create your own ice configuration which you can just bind it to any app Jason file or any XML file to extract information in the real world. So that's what I'm going to build my client since my client is not a Web API so it doesn't have that in build.  
Today.  
OK, so for that I just need a couple of new gates.  
OK, I'm going to use configuration.  
All.  
And I need a configuration binder to bind the data from a particular resource.  
And I do need a Jason to to extract information from a Jason file which helps in deserializing the data and helps in reading the data from Jason.  
Did you go?  
Only two got added.  
Which is son, let me just, son.  
Perfect.  
So I added the configuration packages in my.  
Source code for client application, so I'm just going to create a little helper. You'll put it in. Basically help us, but I'm just going to write it over here public.  
My configuration.  
You start.  
Each other. You don't need it. I'm just trying to show you. You can write it in a real world case to build a configuration.  
So I'm going to write a configuration builder which will build the configurations from my app setting Jason files.  
Ohh.  
On big fella.  
It's going to have.  
Build up.  
Dutch.  
I'm going to set up base, but to find the resource.  
So this would be like showing up by default.  
Directory get current directory will go to my bin during the runtime bin folder.  
And from where you want to read the configuration. So I'm going to provide the Jason file path, but in this case it would be apps settings.  
Dot.  
Get on.  
Alright, just provider configuration builder. All the details to build the configuration, so I'm going to build it iteration builder dot build.  
Ohh so this will be the one that will returning I configuration.  
But as soon as you build it, it will return the the. I can't variation object which which basically behaved the same way that you can just provide this configuration followed by brackets and just access the resources.  
So that's how you you can build a configuration uh, in your applications if you want.  
Alright, have a configuration with me. So I'm going to use that.  
My contribution is equal to.  
It contradiction.  
Alright, now I'm going to build a client.  
Using an insult .net library, which is basically.  
You know that Azure Active Directory form of the application that I have on my local for the client, so this would be I.  
Confidential.  
Like OK before I even talk about this. So all the all the.  
All the libraries related to the the client accessing the.  
Tokens the client trying to prove its identity are provided in Microsoft identity client Nugget Library, so I'm going to install that.

 **Mohit Gupta** joined the meeting

 **Aravind Siddoju** 1:01:48  
With the search for identity.  
So here is the new get. So this is uh ansara Garnet library which stands for Microsoft Authentication Library. Earlier we used to have Adal which is stands for Azure Active Directory.  
Uh, like Active Directory authentication libraries?  
No at all. It's actually stands for Active Directory Authentication library which Microsoft recently deprecated it and they provided a new libraries to authenticate.  
That you are validate the client and so we are we're going to use that in in this case.  
So I'm going to use. I'm saying that net I didn't declined for proving my clients identity to Microsoft Azure Active Directory. So I have that.  
Start hopefully.  
Alright, so this cutting start, so I'm going to use its library. It's API to build my.  
Uh client application.  
I can't fetch your client application.  
I'm going to use the builder again. It's a new builder to build rate the client, so I'm going to create the the the client for what which application.  
For my client application, so my client application will be identified using its client ID, yeah, and this information is stored in.  
In the app settings Jason file which I'm going to access the simple way.  
Cool.  
If my client ID.  
Then how I'm going to validate my client application so I will be validating with.  
With the client secret.  
Then secret as I mentioned is it's kind of a application password, so you are holding an application password that just telling Active Directory. Hey this is the password I have so I'm the valid client. Hello. Let me. So that's how it works.  
So I'm going to use the parameter that I had for client secret.  
And with what authority you are going to?  
Access the client so as soon as you provide a client ID to Mstar library, so it's not going to recognize your resource. So it would be tougher so different directories, different AIDS will have might have the same app ID. So you just need to provide the authority as well to the client library with.  
Authority, which basically should basically providing the.  
The details of the tenant or the arc where my application is deciding.  
So this would be the authority.  
Alright, these are the details that I need to prove my identity, so I'm just telling the the MC library confidential client application builder that hey, I have an application in Azure Active Directory with application ID. This authority is this. I mean it should not allow or let you in by default so you are giving it a secret telling they already have a password. Just validate it and authenticate me. So that's what we did.  
And I'm going to build it.  
So this will build a client. It will generate a proper client which basically represents its accessibility on the Active Directory or Azure level client with me. Now I need to access our acquire token for accessing the server.  
So for that I am just going to.  
Pardon me, have you been uh, vap?  
Still time.  
So.  
I'm just creating a list of.  
Target applications for which you are trying to access the token. It will be passed as a list, although in our case we just need it only for one server application.  
So I'm just putting it here.  
So what I'm going to do is.  
Syndication result is the return type.  
While acquiring the token, so I'm going to put.  
My acquiring token that try catch block.  
All right, so.  
So I'm I'm going to tell my client.  
Ohh to acquire token.  
So I'm just telling the I'm sad library which created a client object for my application to acquire token for what for this particular server or target application ID that I'm passing as an input?  
So I want to execute it asynchronously.  
Ohh.  
You just think.  
Going to use async.  
And I'm going to use a waiter.  
So.  
It's. I'm sorry.  
Some format. Pardon me for that. So I I've just used the ML library and just called it API to acquire token from my client application and I just passed the target resource ID target application ID.  
As an input to it and I'm just not going to do anything over here.  
We'll see if it fails to access. Still think we'll try to fail it once and see if it works. Alright so. So here is what happens in the background. So as soon as you tell your client to access token. So during this phase you are just proving your identity. It's here. The server never came into picture. You're just proving your identity to your own app registration using the details I mean the credentials that you just used and just.  
It just proved identity, didn't it? Anything. Then in this case.  
Of acquiring token for client. So this guy is going to go to app permissions. It's going to see whether this guy has a permissions to access this particular resource ID. So this is going to call the Active Directory and it tells hey I have permissions for this particular role for this particular.  
Ohh, a application target application and I just want to acquire tokens to access it and the the target Applications directory validate that and it's going to issue a token to the client that would be returned.  
Uh. As part of this?  
Please.  
Uh, any any questions so far?  
I guess no, that's are you with me.  
Hello.

 **Yashasvini Rathore** 1:10:03  
Listen.

 **Deepak Kunwar** 1:10:04  
Yes.

 **Aravind Siddoju** 1:10:04  
OK.  
Hope I'm not making it very boring.  
So I'm just going to create a simple HTTP client now.  
I'm going to going to.  
Do this operation only when.  
My access token is returned or not so.  
It is simply authentication result. Uh, which happened asynchronously. So the token will be available as part of access token attribute.  
So.  
So I'm just trying to create an HTTP header and I'm going so here's what I'm going to do now. I have a token with me and just going to bind the token as part of my HTTP header and I'm going to call, hit that target Uri and see if I have an access.  
So yeah, just bear with me.  
But I said the request headers.  
Sorry, typo.  
But this should give me the default requested us. I'm just trying to see whether she, as I mentioned, the JWT will be used for the Jason payload so just need to make sure the media type is application slashes and if not the point of using disability at all. So I'm just trying to check and see if the header media type is JSON or not.  
That's what I'm going to do.  
Pull request header.  
That app.  
Equals to null.  
If there is no default request, either or.  
If.  
Request headers dot accepted media type.  
At any.  
It's plenty. Use a land now.  
Your type.  
Then just.  
Just adding a condition to add application that Jason media type if if it if it is it doesn't exist by default so that's what I did.  
It should be explained if it doesn't exist, I'm going to add.  
Except.  
At winter add.  
You've media type it quality header, so it's a is showing up me by default. I'm going to miss that. So yeah, I did nothing as part of these two steps I checked whether the media type is application that JSON and if it doesn't exist as part of my default request address I'm adding one.  
That's it.  
So.  
Do you need to do so? I create an HTTP client. Now I need to provide that access token as part of the header.  
I'm going to do it and.  
Uh, that isation?  
You've authentication header value.  
So there are two things over here.  
So you just need to you just. There's a way you can provide the scheme and the value, so this this provides an ability to the HTTPS telling what type of authentication scheme we are using. So in my case it's a Baylor based authentication.  
And my value would be resolved dot.  
Access token that I got for the for the target resource ID.  
That's it. I just set up my HTTP client.  
Oh no. I'm going to check my response.  
Going to use HTTP clients get a sync API.  
When to pass the URA?  
But I just used in postman.  
Putting.  
And use verbatim operator so that I will not miss any.  
Specific parameters you want to use. Once you have it.  
Is that?  
I'm going to use funny last step.  
Rather, I would use this stuff conditional check to see.  
On got content cut.  
I see nothing that.  
That's it.  
Uh, I can't figure my client.  
So just a quick background. What I just did so I.  
I created a A a similar client app registration in Azure Active Directory. So again as soon as I create again it will not have any binding in this case. Now I just need to prove my identity for even just to tell you that it's my client app registration that I'm using. So for that you you just use the secret or the application password to prove your identity for your own client under the authority.  
That you have, once you have that, you just need to acquire a token for the target application for which you already have a permission and admin granted you the permission. So and you just use the API for quite token for client. So this guy will basically go through the resource IDs that you pass and try to the Active Directory is the one who is going to validate the resources that you pass and it will grant the token with if at all you have the right permissions to access it.  
But you have that token. I'm just creating a new HTTP client since particularly we're using JWD bearer authentication. I'm just making sure whether I have media type of application JSON as part of requested or if not, there's no point of using J ability. It won't work for other type of payload. So just create a default request header if it doesn't have with application media type as JSON. So then I am using appending the authorization as the access token.  
That I got with scheme as beta, it in the header value and then just trying to hit the client. So let's see if this works or not.  
Any questions before we run the application?

 **Yashasvini Rathore** 1:17:36  
I don't know. I have one question. So when we are trying to set the applications or JSON as is the default request header and we trying to tell the client that you will be using the GWT.  
Kind of a authorization.

 **Aravind Siddoju** 1:17:48  
So. So in this case I'm the client, I mean we.

 **Yashasvini Rathore** 1:17:52  
Yeah. Yeah. So you're telling yourself that we'll be?

 **Aravind Siddoju** 1:17:54  
Yeah, I'm telling my application that we'll be using application touches.  
Application forward slashes.

 **Yashasvini Rathore** 1:18:01  
OK. Yeah. Because anyway we are adding the authorization header.  
After, right, right, right. That is a different thing. So application slash Jason is something else.

 **Aravind Siddoju** 1:18:06  
So this is authorization that's the media type.  
Yeah.  
Let me let me simplify it for you in postman. So you see authorization at different.

 **Yashasvini Rathore** 1:18:19  
Yeah.

 **Aravind Siddoju** 1:18:19  
Attribute and you see a body.  
To the format right?

 **Yashasvini Rathore** 1:18:25  
Right.

 **Aravind Siddoju** 1:18:26  
So if you just select and rock, you can select a media type over here. This is what I set up now.  
As part of a media type.  
Application Jason.

 **Yashasvini Rathore** 1:18:39  
OK.

 **Aravind Siddoju** 1:18:39  
You still need to give authorization, which is a different thing, and here you need to check that as well. You need to set what type of authentication scheme, so here I'm telling bearer authentication and I need to provide the value in the client.

 **Yashasvini Rathore** 1:18:41  
Umm.  
Right. OK.

 **Aravind Siddoju** 1:18:56  
Any other questions, Sir?  
Alright, let's run the application.  
He's yesterday's hanging for get reasons. I'm sorry.  
That it open.  
It's not just that time in the meantime, let it let it.  
Excited to postman again.  
Let's not use any authentication here for now.  
You see photo one another.  
Let me run the application client application. I'm just going to put a break point, just want to show you.  
Ohh.  
If at all we get.  
OK, hopefully we will.  
Right.  
Can't be here.  
So we got a result.  
And I got it token.  
Going to show you.  
Yes.  
It's save it. I'm going to use this in postman as well.  
Umm.  
What I can do here is.  
Just a better authentication. Uh, you just don't need to have any party since it's the gate.  
I'm still telling application Jason.  
I'm sorry I haven't passed the right value yet.  
See it right so 200.  
The light bearer token that I got, I'm going to run my client application further.  
Just to see that my client is also returning the same as of my client application is working.  
And relaxing Lesley without my intervention where postman.  
So.  
I'm sorry I did wrong check here.  
My bad.  
I'm going to write one more time.  
Alright.  
That each day here.  
I'm going to pass the access token that I got. I'm calling HTTP client.  
So I got 200 status success status code. I'm just going to retrieve the data.  
Perfect. So this is how an application is talking to an application without any intervention and it's using JWT token to authenticate using app registration using Active Directory as the middle layer for authentication.  
So yeah, this is how the real world application start to each other using the bearer token.  
Ohh with right permissions write access and the API gets secured using its own app registration defined in it directory.  
Yeah, that's that's pretty much I want to show you guys any questions.  
Did you get any clarity on how authentication happens at the bigger lever?  
I still still troubling to understand.

 **Deepak Kunwar** 1:23:01  
Yes. And actually I've been. I was having a question that how we can hit by the using postman and by getting the bearer token but meanwhile you also explained that means how you copied the bearer token and used in the postman. So I was thinking to ask that thing but you have already covered so thanks for that also.

 **Aravind Siddoju** 1:23:15  
So. Umm.  
Alright.  
That's what I just wanted to show you. So as long as you hold a token write access, it doesn't matter how you hit it.  
Umm so HTTP client via C# code? Or maybe some Java code or using postman UI it doesn't matter.

 **Deepak Kunwar** 1:23:38  
Yes, got it.

 **Aravind Siddoju** 1:23:38  
He hit his thumb code that we are writing to adjust replicate what I am just doing over here to use token to use body as and application just hunt.  
So, but this is a real world case, you will not be using postman if an application is starting to an application.

 **Deepak Kunwar** 1:23:56  
And admin for test so.

 **Mohit Gupta** 1:23:56  
Should be understanding and.

 **Aravind Siddoju** 1:23:59  
Uh, yeah, sure, right.

 **Deepak Kunwar** 1:23:59  
OK.  
And Arvind, for testing purpose, we save this bearer token. We will be saving this bearer token in some file and then while testing we will be using picking that and hitting using the postman.

 **Aravind Siddoju** 1:24:16  
But that's up to you on how you want to test it. Whether you want to test it manually or you can automate that as well. In the test, use the token, create an HTTP client and write a test telling response code is 404 or 201. It doesn't matter, right?

 **Deepak Kunwar** 1:24:23  
OK.  
OK.

 **Aravind Siddoju** 1:24:31  
So the main main blocker or the the main magic is happening.  
Uh at accessing the token? Not at like hitting up creating a client.  
HTTP client creation is like simple or three line code.  
How we are accessing the token is the right the the main?