ASP.NET Interview Questions & Answers- Part 6

(JWT Token, Principal, Claim, Roles, Refresh Token, OpenId, OAuth)

Contents

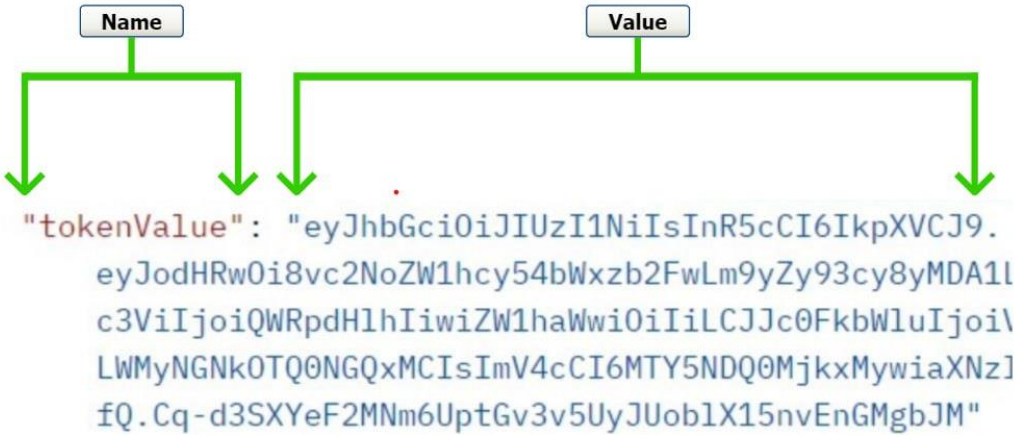
How does Token based Authentication works?

Token based Auth is a two-step process: -

1. Client sends credentials to the server.
2. Server respond backs with a token.
3. Later to access the resource only this token is needed.

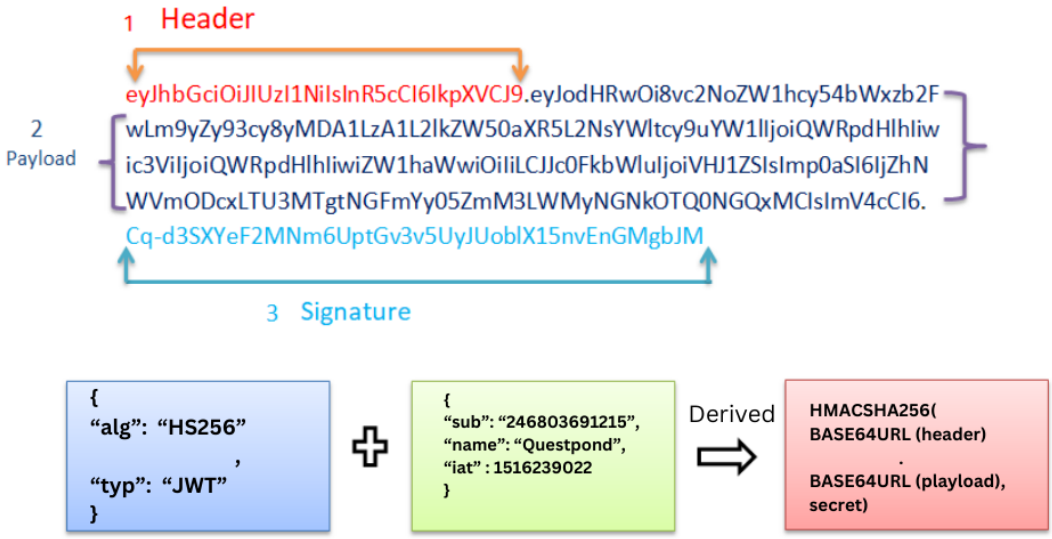
Why is it called JWT Token?

JWT stands for (JSON Web based Token). JSON stands for JavaScript object notation. JSON is a data format with name and value as shown in the below figure. Because we receive the token in JSON format so it's termed as JWT Token.



Explain the 3 sections of JWT Token?

1. Header: - This section has Algorithm and type of token.
2. Payload: - This has the identity and claims information.
3. Signature: - This section is created using the first two sections (Header, Payload) and Secret key.



What are Identity and claims?

Identity identifies the user or entity uniquely. Claims talks about what roles / rights the user has with respect to the system.

Differentiate between Authentication VS Authorization?

Authentication ensures that the user exists in the system. Authorization talks about the roles and rights of the users. Authentication talks about WHO the user is and Authorization talks about WHAT the user can do with your system.

Claims vs Roles?

Roles	**Claims**
Role classifies user types (Admin, User, Superuser) and to these user types roles are assigned.	Claims can have permissions but it talks more about the Subject.
Claims can have roles but not all Claims can be roles. Some claims which can not be roles are :- Browser =Chrome , Senior Citizen=true, Lang Known=English.	
Roles are assigned to a user type.	Claims are assigned to the actual USER.

Principal vs Identity

Identity represents a user + roles + claims. Principal encapsulates identity object and can be assigned to a code / thread context.

Can we put critical information in JWT Token?

No. It's just BASE64 encoded and can be decoded very easily.

How do you create JWT Token in MVC ?

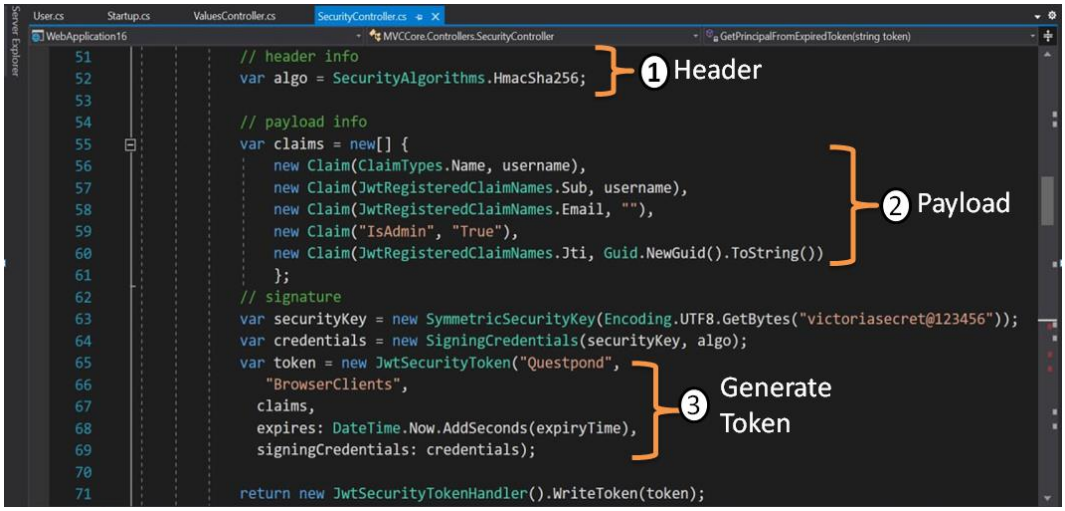
Step 1: - Import "Microsoft.AspNetCore.Authentication.JwtBearer" package from Nuget.

Step 2: - Use the package for generating the token.

Below is the sample code for generating token. Now remember that this is a interview so its difficult to explain code in detail and that also verbally. So the best would be map the code with the three sections of the token structure and explain the same.

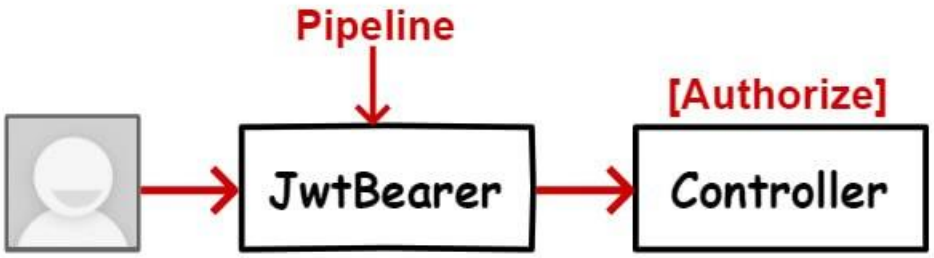
This will make the interviewers life easy in understanding what you are speaking.

- First step is to select the algorithm. Interviewer can ask you name of the algorithm. Does not hurt to remember HMACSHA256.
- Next step is creating the claims collection. Remember there are standard claims and you can add your own.
- Last step is to use the Algorithm, use claims and generate token.



Where is Token Checked in ASP.NET MVC ?

Token authenticity is checked in the pipeline. In the startup we need to add this “JwtBearer” check in the pipeline.



Below is how to add the same in pipeline.

```
services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
            .AddJwtBearer(options =>
            {
                options.TokenValidationParameters = new TokenValidationParameters
                {
                    ValidateIssuer = true,
                    ValidateAudience = true,
                    ValidateLifetime = true,
                    ValidateIssuerSigningKey = true,
                    ValidIssuer = "Questpond",
                    ValidAudience = "BrowserClients",
                    ClockSkew = TimeSpan.Zero,
                    IssuerSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes("victoriasecret@123456"))
```

```
            };
        });
```

You also need to call “UseAuthentication” and “UseAthorization” in same sequence first Authentication and then Authorization as shown in the below code.

```
app.UseAuthentication();
app.UseAuthorization();
```

What is use of Authorize Attribute?

JWT Token authentication is only applied to controllers who have Authorize Attribute decorated.

```
[Authorize]
public class ValuesController : ControllerBase
{}
```

How did you implement JWT token security ?

Interviewer can ask you in general to talk about the steps as well. Summarizing what we discussed in the last 3 questions:-

1. Import the “Microsoft.AspNetCore.Authentication.JwtBearer” JWT package.
2. Create the three sections of the token.
3. Add the Bearer Check in the pipeline.
4. Apply the [Authorize] attribute to controllers which needs to be protected.

What HTTP status code do you send for unauthorized access ?

HTTP 401 is the status code which represents UnAuthorized access. Many developers answer 500 , please note 500 stands for internal server error which represents validation or technical issues. And yes 200 is when everything is ok.

How do we send tokens from Client Side?

Token is sent in the REQUEST HEADER in standard format defined by W3C as shown below.

```
"Authorization: Bearer XXTokenXX"
```

From Javascript,Jquery,Angular etc , How is token passed ?

Which ever technology you use the most important thing is you will need follow the W3C structure i.e.

```
"Authorization: Bearer XXTokenXX"
```

Below is code using simple FETCH API

```
fetch('http://localhost:8080/resourceserver/protected-no-scope', {
    method: 'GET',
    headers: new Headers({
        'Authorization': 'Bearer <token>',
        'Content-Type': 'application/x-www-form-urlencoded'
    })
});
```

JQuery Code again same you will need to stick to the structure.

```
$.ajax({
  url: 'http://localhost:8080/resourceserver/protected-no-scope',
  type: 'GET',
  contentType: 'application/json'
  headers: {
    'Authorization': 'Bearer <token>'
  },
  success: function (result) {
     // CallBack(result);
  },
  error: function (error) {

  }
});
```

Angular code looks something like this.

```
var headers_object = new HttpHeaders();

headers_object.append('Content-Type', 'application/json');
```

```
headers_object.append("Authorization", "Bearer " + tokenvar);
const httpOptions = {headers: headers_object};
 this.http.post(
                'http://localhost:8000/api/role/Post', httpOptions
               ).subscribe(resp => {
                this.roles = console.log(resp)
                }
              );
```

So whatever framework you are working you can talk about that code but make sure you stick to structure of JWT token.

Increase UX experience in Mobile apps to avoid relogin ?

Use a refresh token.

What is a refresh tokens?

Refresh token is a separate token by which you can get a new JWT token.

Differentiate between Access tokens and Refresh tokens?

Access tokens (JWT Token) are tokens which are checked when you access a secured resource while Refresh tokens

Differentiate between Access tokens and Refresh tokens?

Access tokens (JWT Token) are tokens which helps you to access a secured resource while Refresh tokens helps to get new JWT token.

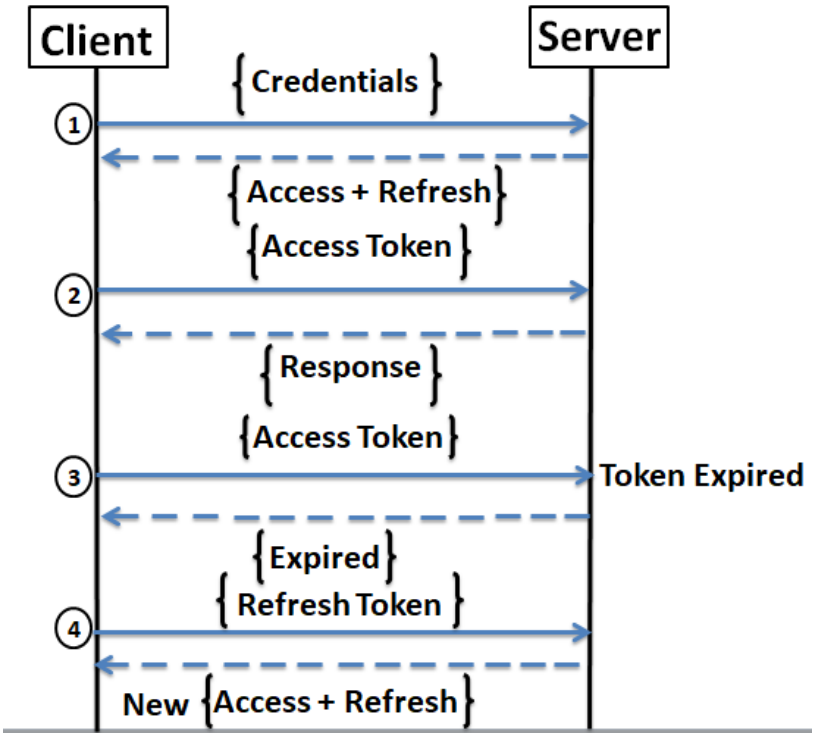
How does Refresh token work ?

Step 1: - Credentials sent access token + refresh token is generated.

Step 2: - Using access token clients can now access resource.

Step 3 :- Access token expires.

Step 4:- Client uses refresh tokens and generates new access tokens plus refresh tokens and continues with step 2.



Whose expiry time is more Access tokens or Refresh tokens ?

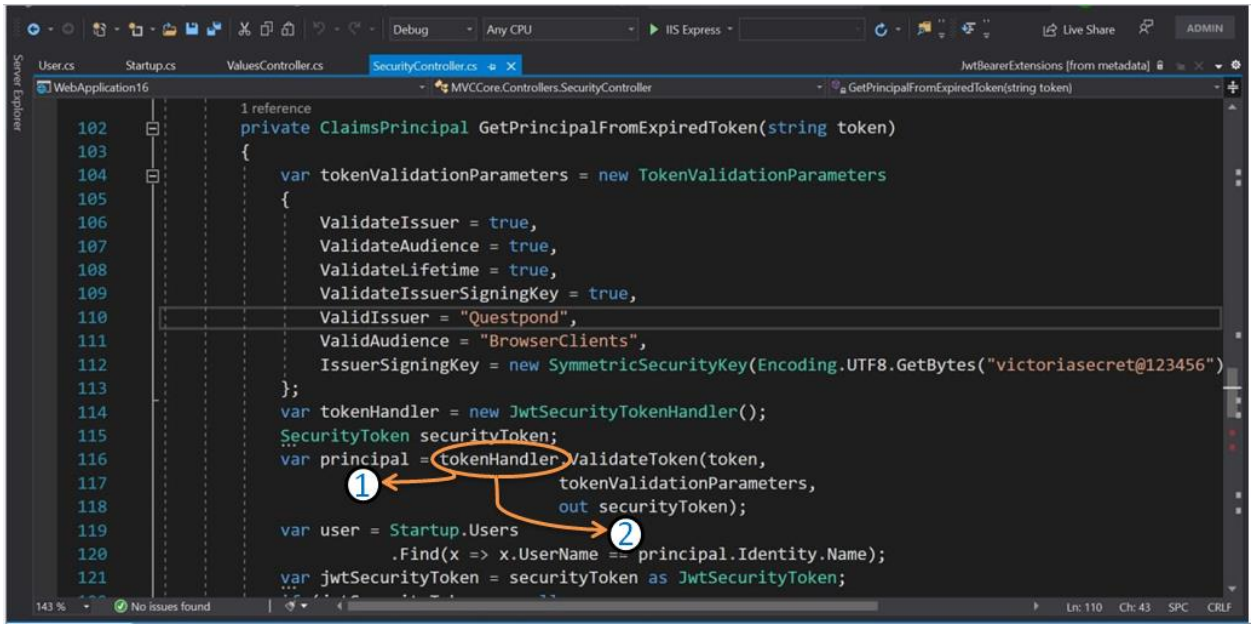
Expiry time of Refresh token is more than Access token. Access token expiry time should be less so that if some one gets access of Access token he has a very less window to make attack.

Explain revocation of Refresh token?

Revocation of Refresh token means expiring / invalidating the token for creating any more JWT token. User needs to freshly authenticate to get new Refresh token.

How to extract Principal from a Token ?

You can extract “Principal” and “Token” by using “TokenHandler” class.



What is the best practice to store tokens at client side?

In-memory:- Using the application variable we can store the token when the application is running.

Cookie :- If you still want to store cookie is a good place.

IndexedDB, Session storage and local storage are prone to XSS attacks.

If we store JWT in cookie how to save from XX attacks?

Create cookie using HTTP only. By doing so this cookie can not be read using JavaScript “document.cookie”. In other words cookie is safe from XSS attacks.

OAUTH vs OpenID vs OpenIdConnect vs JWTToken ?

OpenId,OAuth and OpenIdConnect are protocols while JWTToken is just a token.

1. OpenId is all about authentication.
2. OAUTH is all about authorization.
3. OpenId connect has both of them.

When should we use what?

- OpenId (Who ?): - You want to just know if the user exists in the system in or not. Like shopping sites , mobile applications , single sign on etc.
- OAUTH (What ?) : - When third party application tries to access resources.
- OpenIDConnect: - When you want to authenticate and authorize as well like intranet websites.

What is Identity Server ?

Identity server is a free opensource server which helps to implement openId connect and OAuth.

How to implement Single Sign on ?

For single sign on the authentication / authorization server should be separate. The websites who want to belong to a common federation gets in to trust with this centralized Authentication / Authorization server.

Many developers use IdentityServer for single sign on.

What is a scope in IdentityServer ?

Scope is nothing but roles.