Securing microservices

(Demo prep)

- Remove all tokens from Postman
- Clear Authorization values in Postman

Securing the Catalog microservice

It's time to start securing our microservices to make sure that only authorized clients can get access to them. Let's start with our Catalog microservice.

In Catalog Microservice

1. Remove these obsolete entries from **Play.Catalog.Service.csproj**:

```
<PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="5.0.1" NoWarn="NU1605" />
<PackageReference Include="Microsoft.AspNetCore.Authentication.OpenIdConnect" Version="5.0.1" NoWarn="NU1605" />
```

2. Add NuGet package:

dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer

3. Update Startup.cs:

```
public void ConfigureServices(IServiceCollection services)
{
  serviceSettings = Configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();

  services.AddMongo()
        .AddMongoRepository<Item>("items")
        .AddMassTransitWithRabbitMq();

  services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
    .AddJwtBearer(options =>
    {
      options.Authority = "https://localhost:5003";
      options.Audience = serviceSettings.ServiceName;
    });

    …
}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
  …
  app.UseRouting();
```

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

4. Update **ItemsController**:

```
[ApiController]
[Route("items")]
[Authorize]
public class ItemsController : ControllerBase
{
  …
}
```

5. Update **appsettings.Development.json**:

```
{
 "Logging": {
  "LogLevel": {
   "Default": "Debug",
   "Microsoft": "Warning",
   "Microsoft.AspNetCore.Authorization": "Information",
   "Microsoft.Hosting.Lifetime": "Information"
  }
 }
}
```

6. **dotnet run** in Identity and Catalog microservices

7. Try a GET https://localhost:5001/items (401 Unauthorized)

8. Request an access token as before.

   Explore the generated **Access Token** in https://jwt.ms or https://jwt.io

9. Try another GET https://localhost:5001/items with the token (401 Unauthorized)

In Identity Microservice

10. Add to **appsettings.json**:

```
 "IdentityServerSettings": {
  "ApiScopes": [
   {
```

```
      "Name": "catalog.fullaccess"
    }
  ],
  "ApiResources": [
    {
      "Name": "Catalog",
      "Scopes": [
        "catalog.fullaccess"
      ]
    }
  ]
 },
```

11. Update **appsettings.Development.json**:

```
 "IdentityServerSettings": {
  "Clients": [
    {
      "ClientId": "postman",
      "AllowedGrantTypes": [
        "authorization_code"
      ],
      "RequireClientSecret": false,
      "RedirectUris": [
        "urn:ietf:wg:oauth:2.0:oob"
      ],
      "AllowedScopes": [
        "openid",
        "profile",
        "catalog.fullaccess"
      ],
      "AlwaysIncludeUserClaimsInIdToken": true
    }
  ]
 }
```

12. Update **IdentityServerSettings.cs**:

```
namespace Play.Identity.Service.Settings
{
   public class IdentityServerSettings
   {
      public IReadOnlyCollection<ApiScope> ApiScopes { get; init; }
      public IReadOnlyCollection<ApiResource> ApiResources { get; init; }
```

```
    public IReadOnlyCollection<Client> Clients { get; init; }
    public IReadOnlyCollection<IdentityResource> IdentityResources =>
      new IdentityResource[]
      {
        new IdentityResources.OpenId(),
        new IdentityResources.Profile()
      };
  }
}
```

13. Update **Startup.cs**:

```
public void ConfigureServices(IServiceCollection services)
{
  …
  services.AddIdentityServer()
      .AddAspNetIdentity<ApplicationUser>()
      .AddInMemoryApiScopes(identityServerSettings.ApiScopes)
      .AddInMemoryApiResources(identityServerSettings.ApiResources)
      .AddInMemoryClients(identityServerSettings.Clients)
      .AddInMemoryIdentityResources(identityServerSettings.IdentityResources)
      .AddDeveloperSigningCredential();
  …
}
```

14. Start Identity again

15. Check the OpenID Configuration doc (new scope available)

16. Request an access token with **catalog.fullaccess** scope.

17. Decode the token and notice:

    a. The new **aud** claim: **Catalog**

    b. The new **catalog.fullaccess** scope.

18. Try another GET https://localhost:5001/items with the new token (200 OK)

In the next lesson we will generalize how we enable authentication in our microservices so that it is easier to enable in any future microservice.