

Securing microservices

(Demo prep)

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

Generalizing how to secure a microservice

Let's see how we can generalize and simplify the authentication configuration for Catalog, Inventory and any future microservices.

In Common repo

1. dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer

2. Update **ServiceSettings.cs**:

```
namespace Play.Common.Settings
{
    public class ServiceSettings
    {
        public string ServiceName { get; init; }
        public string Authority { get; init; }
    }
}
```

3. Add the **Identity** directory

4. Create **ConfigureJwtBearerOptions.cs** in Identity directory:

```
namespace Play.Common.Identity
{
    public class ConfigureJwtBearerOptions : IConfigurationNamedOptions<JwtBearerOptions>
    {
        private readonly IConfiguration configuration;

        public ConfigureJwtBearerOptions(IConfiguration configuration)
        {
            this.configuration = configuration;
        }

        public void Configure(string name, JwtBearerOptions options)
        {
            if (name == JwtBearerDefaults.AuthenticationScheme)
            {
                var serviceSettings = configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();
```

```

        options.Authority = serviceSettings.Authority;
        options.Audience = serviceSettings.ServiceName;
        options.MapInboundClaims = false;
        options.TokenValidationParameters = new TokenValidationParameters
        {
            NameClaimType = "name",
            RoleClaimType = "role"
        };
    }
}

public void Configure(JwtBearerOptions options)
{
    Configure(Options.DefaultName, options);
}
}
}

```

5. Add **Extensions.cs** under Identity directory:

```

namespace Play.Common.Identity
{
    public static class Extensions
    {
        public static AuthenticationBuilder AddJwtBearerAuthentication(this IServiceCollection services)
        {
            return services.ConfigureOptions<ConfigureJwtBearerOptions>()
                .AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
                .AddJwtBearer();
        }
    }
}

```

6. Generate new NuGet package:

```
dotnet pack -p:PackageVersion=1.0.3 -o ../../..\packages
```

In the next lesson we will use our updated NuGet package to simplify how we configure authentication in our Catalog and Inventory microservices.