

## Configuring SignalR Authentication

### Script start

When using any of the protocols supported by SignalR, like WebSockets, the access tokens presented by the client will not be read by the authentication middleware by default, which will cause authentication issues between client and server.

Let's fix this by updating our authentication configuration logic in our common library.

### In Common repo

1. Update ConfigureJwtBearerOptions.cs:

```
public class ConfigureJwtBearerOptions : IConfigureNamedOptions<JwtBearerOptions>
{
    private const string AccessTokenParameter = "access_token";
    private const string MessageHubPath = "/messageHub";

    private readonly IConfiguration configuration;

    ...

    public void Configure(string name, JwtBearerOptions options)
    {
        if (name == JwtBearerDefaults.AuthenticationScheme)
        {
            ...
            options.TokenValidationParameters = new TokenValidationParameters
            {
                ...
            };

            options.Events = new JwtBearerEvents
            {
                OnMessageReceived = context =>
                {
                    var accessToken = context.Request.Query[AccessTokenParameter];
                    var path = context.HttpContext.Request.Path;

                    if (!string.IsNullOrEmpty(accessToken) &&
                        (path.StartsWithSegments(MessageHubPath)))
                    {
                        context.Token = accessToken;
                    }
                }
            }
        }
    }
}
```

```

        }

        return Task.CompletedTask;
    }
};
}
}
...
}

```

2. Create new NuGet package:

```
dotnet pack -p:PackageVersion=1.0.5 -o ..\..\packages
```

In the next lesson you will use this new NuGet package and the SignalR APIs to enable real time notifications in the purchase experience.