

身份认证工作原理

2018年12月13日 10:13

原始版本的 HttpContext 提供 AuthenticationManager 管理器，但已弃用。

[System.Security.Claims]

[ClaimsPrincipal.cs](#)

[Microsoft.AspNetCore.Authentication.Abstractions]

核心对象: [AuthenticationScheme.cs](#)、[AuthenticationTicket.cs](#)、[AuthenticationProperties.cs](#)

[AuthenticationHttpContextExtensions.cs](#)

SignInAsync 用户登录成功后颁发一个证书（加密的用户凭证），用来标识用户的身份。

SignOutAsync 退出登录，如清除Cookie等。

AuthenticateAsync 验证在 SignInAsync 中颁发的证书，并返回一个 AuthenticateResult 对象，表示用户的身份。

ChallengeAsync 返回一个需要认证的标识来提示用户登录，通常会返回一个 401 状态码。

ForbidAsync 禁止访问，表示用户权限不足，通常会返回一个 403 状态码。

GetTokenAsync 用来获取 AuthenticationProperties 中保存的额外信息。

以上扩展方法调用依赖注入容器中 [IAuthenticationService.cs](#) 同名的方法。

```
context.RequestServices.GetRequiredService<IAuthenticationService>().SignInAsync(context, scheme, principal,
properties);
context.RequestServices.GetRequiredService<IAuthenticationService>().SignOutAsync(context, scheme,
properties);
context.RequestServices.GetRequiredService<IAuthenticationService>().AuthenticateAsync(context, scheme);
context.RequestServices.GetRequiredService<IAuthenticationService>().ChallengeAsync(context, scheme,
properties);
context.RequestServices.GetRequiredService<IAuthenticationService>().ForbidAsync(context, scheme,
properties);
context.RequestServices.GetRequiredService<IAuthenticationService>().GetTokenAsync(context, scheme,
tokenName);
```

[Microsoft.AspNetCore.Authentication.Core]

```
public class AuthenticationService : IAuthenticationService
```

来自 <<https://github.com/aspnet/AspNetCore/blob/master/src/Http/Authentication.Core/src/AuthenticationService.cs>>

[IAuthenticationSchemeProvider.cs](#) 和 [IAuthenticationHandlerProvider.cs](#)

[IClaimsTransformation.cs](#)

微软自己实现的常用认证方案（可参考）

[\[Microsoft.AspNetCore.Authentication.Cookies\]](#)

[\[Microsoft.AspNetCore.Authentication.MicrosoftAccount\]](#)

[\[QQ和微信认证\]](#)

既然我们都理解了整个架构，那么接下来就是用这些方案或者编写自己的方案啦。

```
install-package IdentityManager2.AspNetIdentity -pre  
install-package IdentityManager2 -pre
```

```
services.AddIdentityManager().AddIdentityMangerService<AspNetCoreIdentityManagerService<ApplicationUser,  
string,ApplicationRole, string>>();  
app.UseIdentityManager();
```

双因素认证（2FA）教程

来自 <<http://www.ruanyifeng.com/blog/2017/11/2fa-tutorial.html>>

二维码生成前端库

<https://davidshimjs.github.io/qrcodejs/>