

nova  
net

Identity og Auth - Gjør det enkelt  
med Asp.Net Identity og OpenIdDict

# Duende



nova  
net



**BIKE LOCK AND REGULAR LOCK.  
DIFFICULT TO BREAK**

nova  
net



nova  
net

**Keep it simple**

# Asp.Net Identity

- Cookie based Auth
- Trygt nok for å beskytte WebApp
- Masse som kommer gratis
- Egentlig ikke noe Identity Provider

<http://letmegooglethat.com/?q=asp.net+identity>

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity>

<https://github.com/dotnet/AspNetCore.Docs/tree/master/aspnetcore/security/authentication/identity/sample/src/ASPNETCore-IdentityDemoComplete/IdentityDemo>

# Code + Demo



nova  
net

**API?**

# Code + Demo

# External token provider

- Selvutviklet
- IdentityServer
- OpenIdDict

# Konfigurere en klient

```
private async Task<AuthResponse> RetrieveTokenAsync(string login, string password)
{
    var postData = new List<KeyValuePair<string, string>>();
    postData.Add(new KeyValuePair<string, string>("username", login));
    postData.Add(new KeyValuePair<string, string>("password", password));
    postData.Add(new KeyValuePair<string, string>("grant_type", "password"));
    postData.Add(new KeyValuePair<string, string>("client_id", _clientId));
    postData.Add(new KeyValuePair<string, string>("client_secret", _clientSecret));

    HttpContent content = new FormUrlEncodedContent(postData);

    var response = await _identityClient.PostAsync("connect/token", content);
    if (!response.IsSuccessStatusCode)
        return null;

    var responseObject = await response.Content.ReadFromJsonAsync<AuthResponse>();
    return responseObject;
}
```

```
services.AddAuthentication().AddJwtBearer(options => { options.Authority = "http://localhost:58795/"; options.Audience = "resource_server"; options.RequireHttpsMetadata = false; options.TokenValidationParameters = new TokenValidationParameters { NameClaimType = OpenIdConnectConstants.Claims.Subject, RoleClaimType = OpenIdConnectConstants.Claims.Role }; });
```

nova  
net

# Konfigurere et API

```
services.AddAuthentication()  
    .AddJwtBearer(options =>  
    {  
        options.Authority = "http://localhost:58795/";  
        options.Audience = "resource_server";  
        options.RequireHttpsMetadata = false;  
        options.TokenValidationParameters = new TokenValidationParameters  
        {  
            NameClaimType = OpenIdConnectConstants.Claims.Subject,  
            RoleClaimType = OpenIdConnectConstants.Claims.Role  
        };  
    });
```

eller

```
services.AddAuthentication()  
    .AddOAuthIntrospection(options =>  
    {  
        options.Authority = new Uri("http://localhost:58795/");  
        options.Audiences.Add("resource_server");  
        options.ClientId = "resource_server";  
        options.ClientSecret = "875sqd4s5d748z78z7ds1ff8zz8814ff88ed8ea4z4zzd";  
        options.RequireHttpsMetadata = false;  
    });
```

## Bra

- Gjør-det-selv
- God måte å forstå magikken bak
- «Gratis»
- Enkelt
- Kan gjenbruke eksisterende DB
- Separation of concern
- Trenges ikke noe eksternt for å få Google, FB, Spotify osv

## Ikke fullt så bra

- OpenSource fra privat for token generering
- Vanskelig å vokse fra MVP og legge til tryggere flyt

nova  
net

# Spørsmål?