**HRM CODE TUTORIAL COLLECTION**

**Collection By:** R&D Team

Index

[IdentityServer4 (Latest) 2](#_Toc23858531)

[Features 2](#_Toc23858532)

[Update Note 2](#_Toc23858533)

[How to use 3](#_Toc23858536)

# CONFIGURE SWAGGER API

## How to use - Localhost

### Server Side

#### Identity Serve

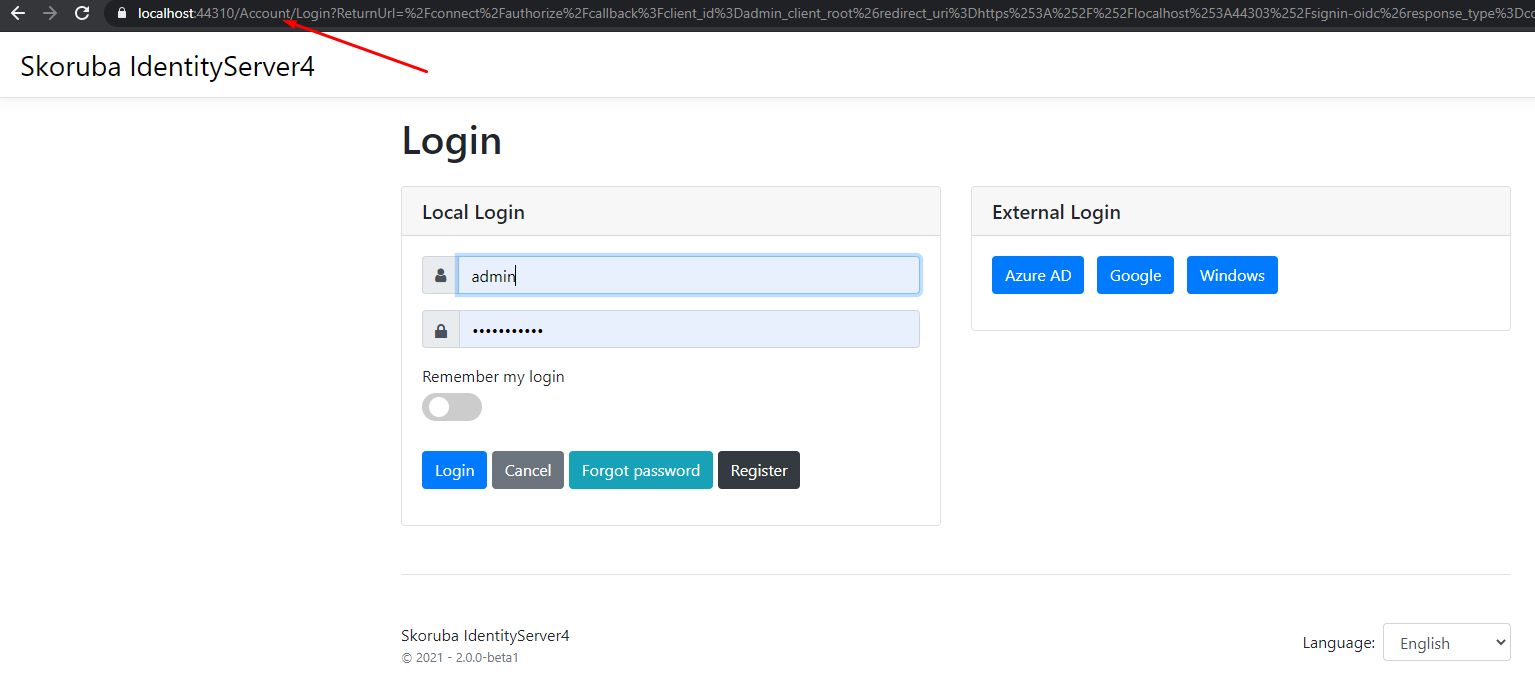
URL: <https://localhost:44310>

#### Identity Admin UI

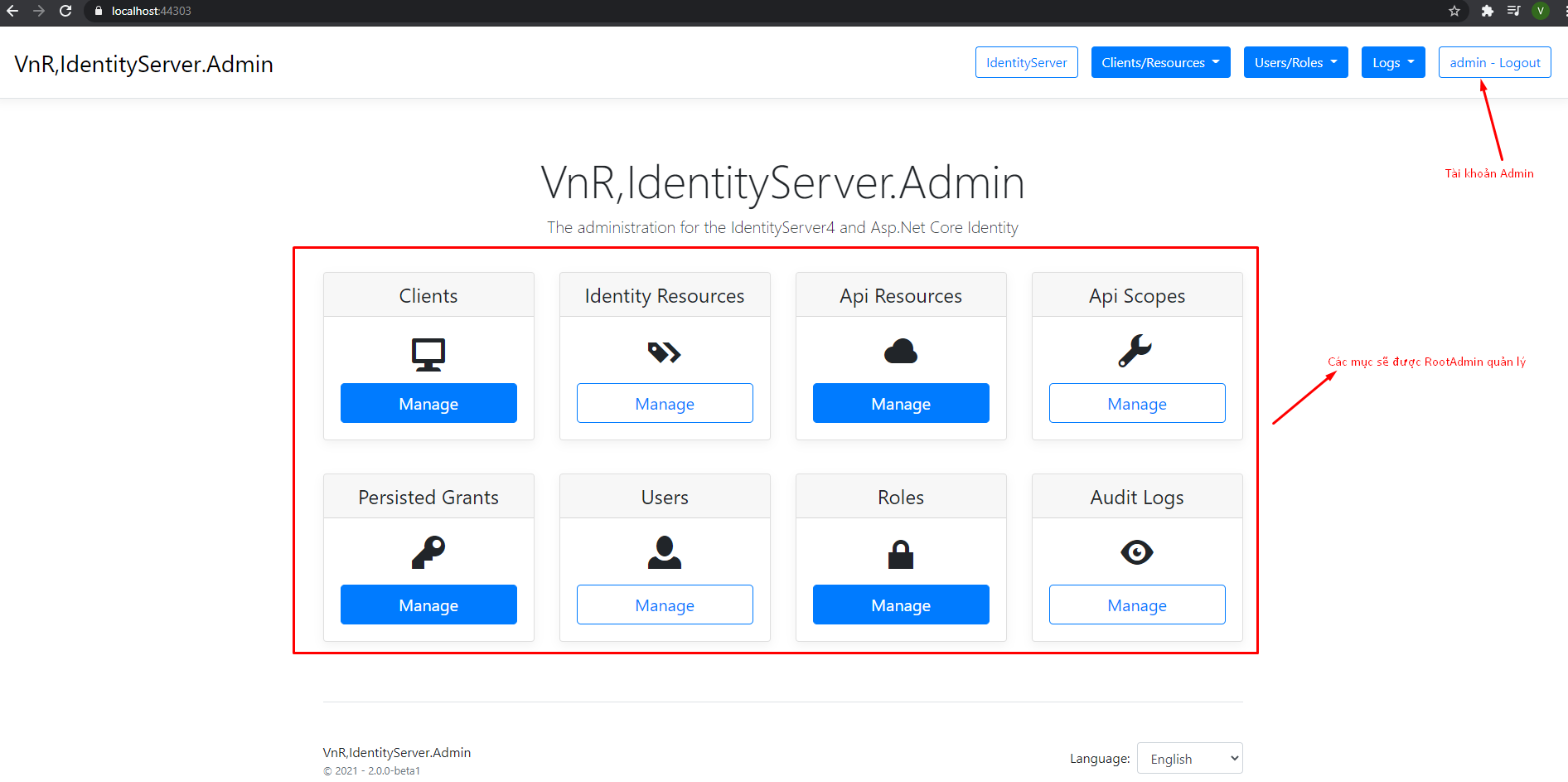
URL: <https://localhost:44303>

##### Đăng nhập vào Admin Page

Truy cập <https://localhost:44303>, khi này chúng ta sẽ bị Redirect sang IdentityServer (<https://localhost:44310>) để thực hiện việc đăng nhập



Chỉ có các tài khoản có Role là “RootAdmin” mới có thể truy cập vào Identity Admin Page, ở trong ảnh trên tài khoản là admin và có Role này nên có thể truy cập

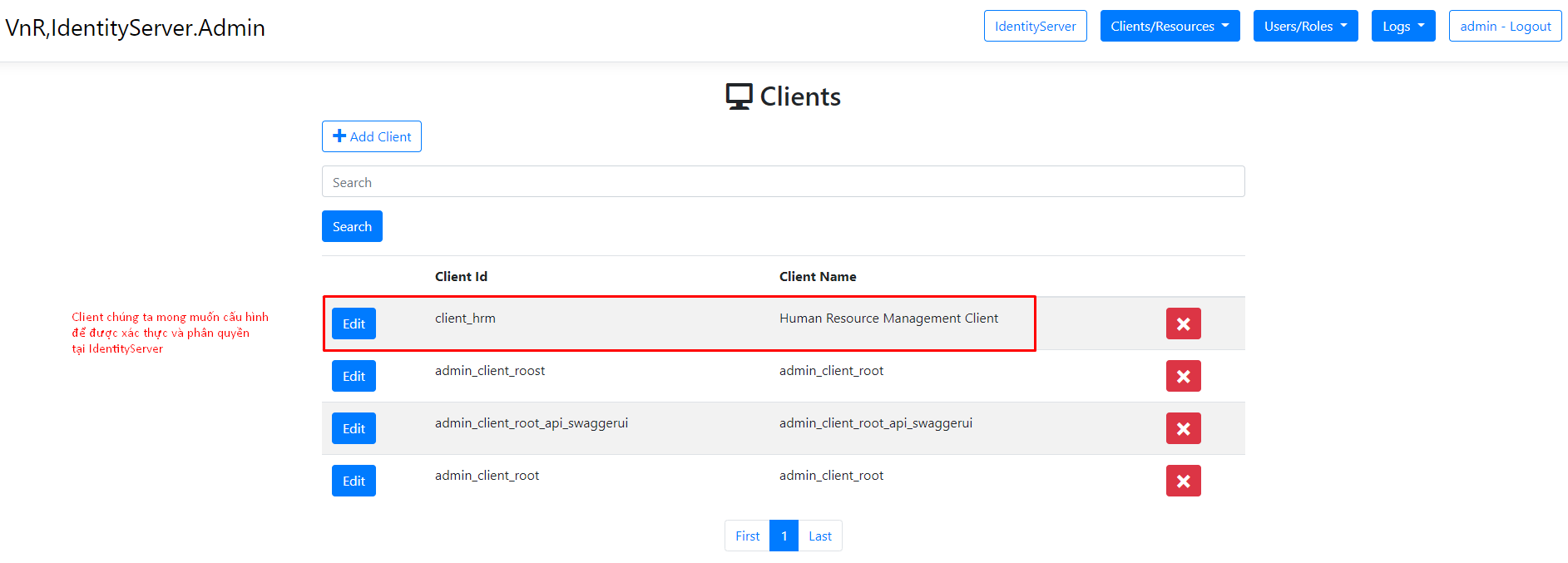


Đăng nhập thành công chúng ta sẽ được Redirect lại về Admin Page (port 44303) và có thể thực hiện việc cấu hình các mục như Client, Api Scopes, Api Resources …

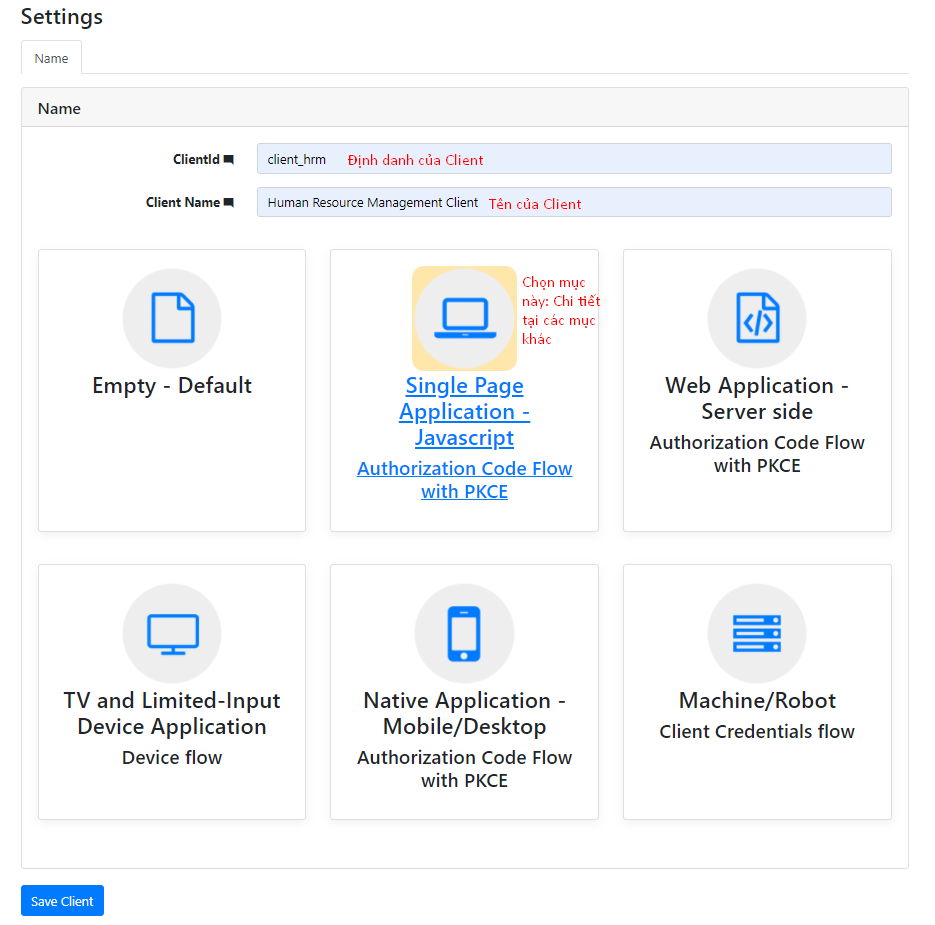
##### Clients

Clients là mục sẽ quản lý những Client nào được truy cập tới một Protected Resource nào đó cụ thể thông qua việc xác thực và phân quyền thông qua Identity Server.

###### Tạo Client



Chúng ta vào mục Clients -> Add Client để tạo mới một Client, ở đây client\_hrm đã được tạo với các thông tin cơ bản ban đầu giống như hình ở dưới



ClientId là định danh của Client, không được đặt trùng với các Client khác. Ví dụ nếu chúng ta có các HRM Client, CRM Client, EDU Client thì ClientId có thể lần lượt là client\_hrm, client\_crm, client\_edu.

ClientName thì chúng ta có thể tùy ý đặt, nhưng nên là một mô tả cho ClientId

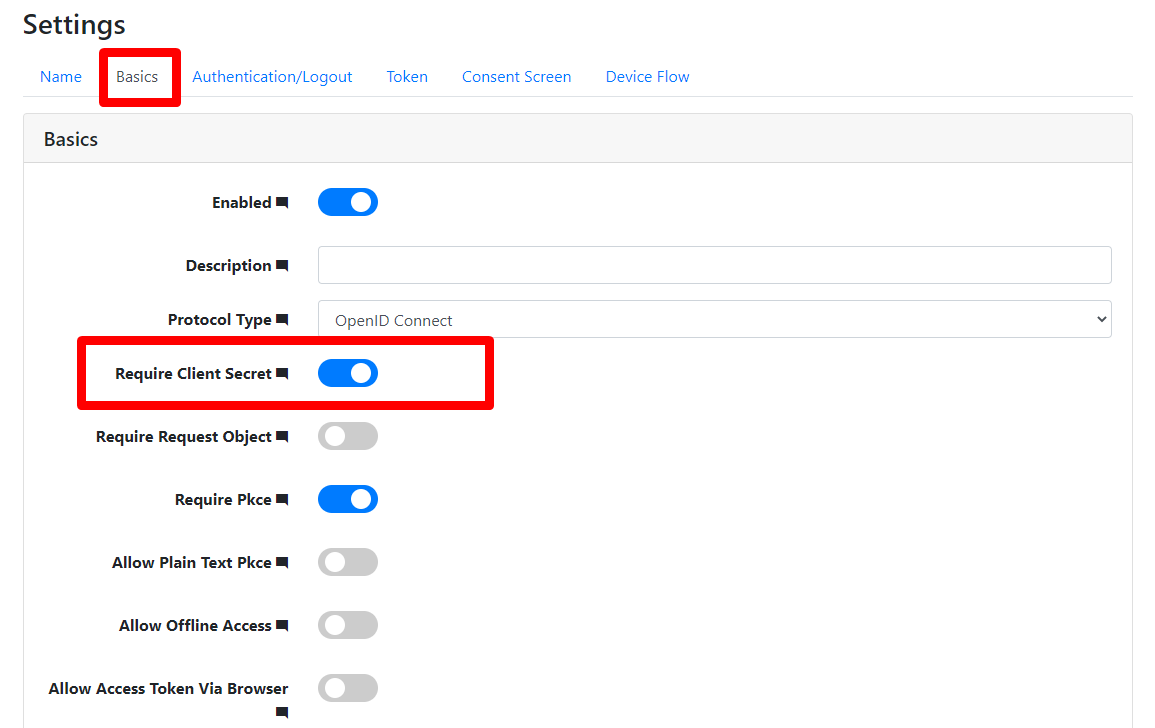
Cuối cùng chúng ta chọn Single Page Application – Javascript Authorization Code Flow with PKCE

Save Client để lưu Client vừa tạo mới

###### Edit Client

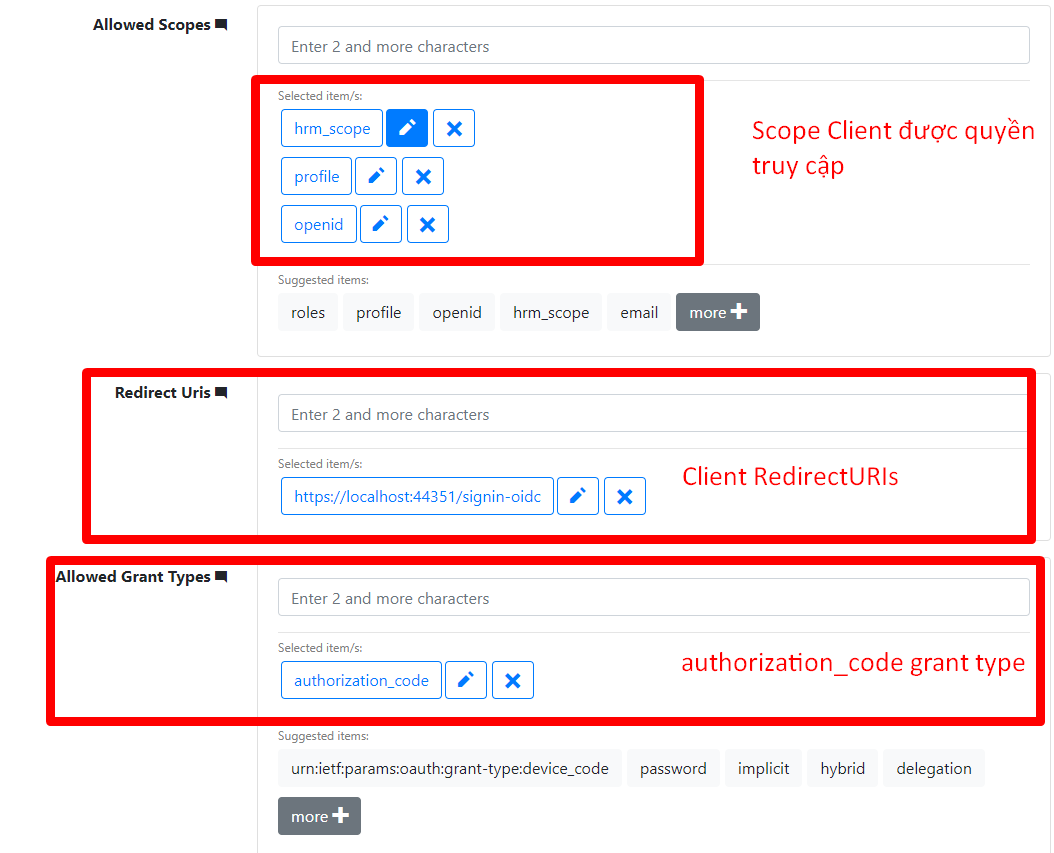
Chúng ta cần cấu hình lại hai mục chính là Basics và Authentication/Logout

Basics



Giao diện cơ bản của Basics Settings.

Mục đầu tiên cần chọn trong Basics Settings là Require Client Secret, nếu mục này chưa được chọn chúng ta cần chọn lại, việc này nói cho Identity Server biết mỗi khi hrm\_client cần được xác thực thì phải gửi kèm một Client Secret (được mã hóa)

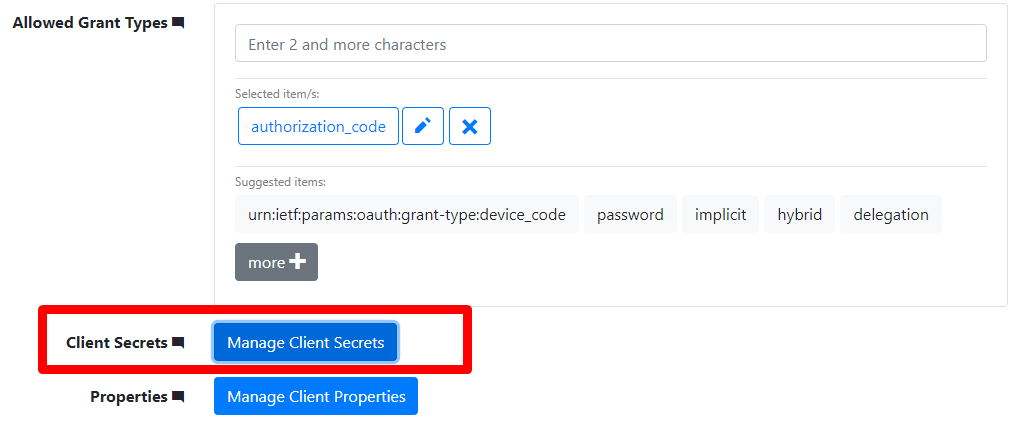


Các mục tiếp theo cần cập nhật là Allowed Scopes, các Scopes như profile, openid là những Scopes mặc định, còn hrm\_scope là một Scope được quản lý tại API Scopes (chi tiết …).

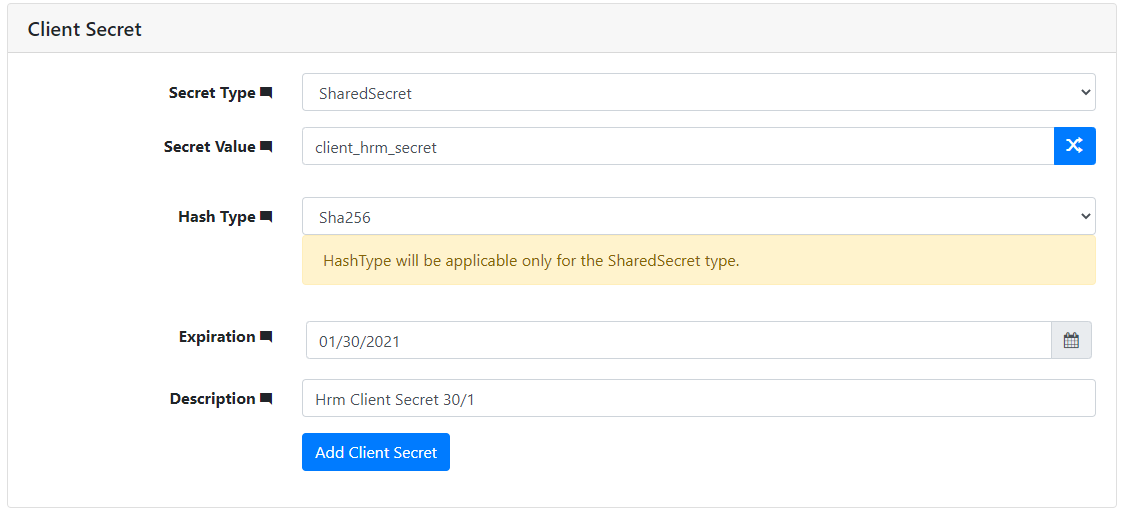
Hiện tại Client chúng ta đang cấu hình là client\_hrm sẽ được truy cập vào hrm\_scope

Redirect Uris, tại đây địa chỉ chúng ta đang điền vào là <https://localhost:44351/signin-oidc>, địa chỉ <https://localhost:44351> là địa chỉ Client\_HRM chúng ta sẽ sử dụng để truy cập tới Identity Server sau này

Allowed Grant Types: Mặc định sẽ là authorization\_code (mục tạo Client)

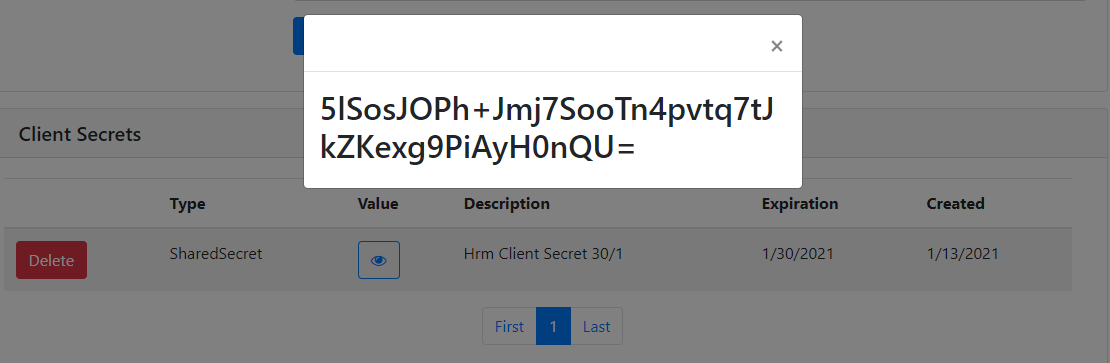


Mục cuối cùng cần cấu hình tại Basic Settings là Client Secret, vì như phía trên chúng ta đã chọn Require Client Secret …



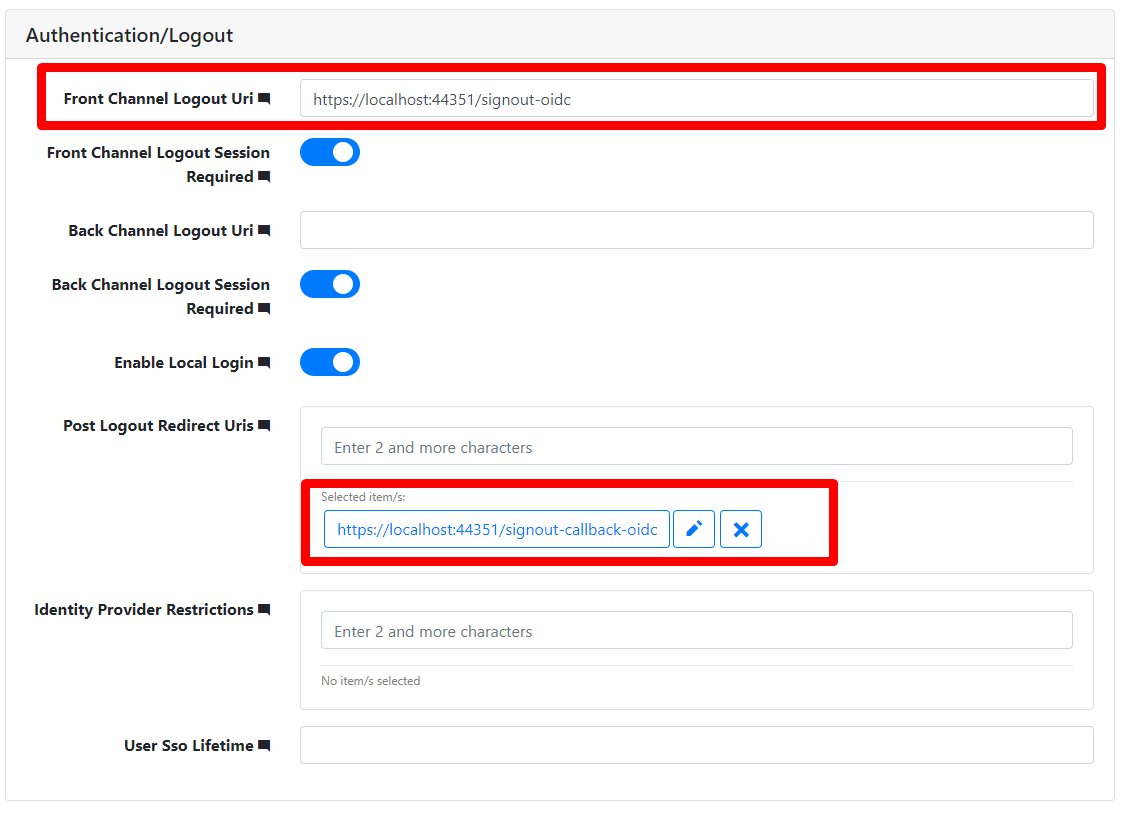
Tạo một Secret với Secret Type là SharedSecret, Value = client\_hrm\_secret, Hash Type = Sha256

Các mục Expiration và Description có thể bỏ qua



Secret sau khi tạo, bây giờ chúng ta sẽ chỉ có thể thấy Value là một chuỗi ký tự đã được mã hóa (hrm\_client\_secret)

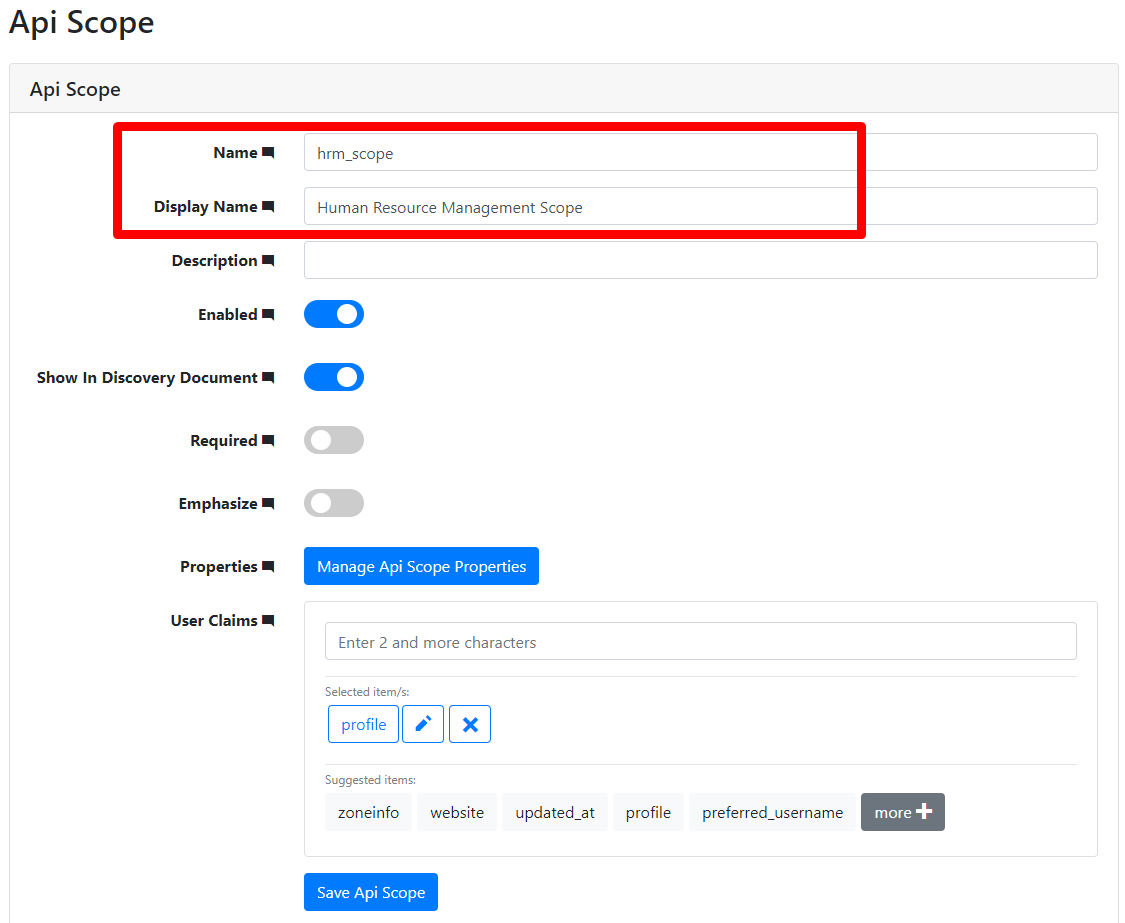
Authentication/Logout



Cập nhật lại Front Channel Logout Uri và Post Logout Redirect Uris như trong hình (44351 là port của client\_hrm khi chạy trên localhost)

##### API Scopes

###### Tạo Api Scope



Tạo một API Scope là hrm\_scope

### Client Side

##### HRM Client ASP.NET Core Project

Cấu hình ConfigurationServices tại Startup

services.AddAuthentication(options =>

{

options.DefaultScheme = CookieAuthenticationDefaults.AuthenticationScheme;

options.DefaultChallengeScheme = OpenIdConnectDefaults.AuthenticationScheme;

options.DefaultSignOutScheme = OpenIdConnectDefaults.AuthenticationScheme;

})

.AddCookie()

.AddOpenIdConnect(options =>

{

options.SignInScheme = "Cookies";

options.SignOutScheme = "OpenIdConnect";

options.Authority = "https://localhost:44310";

options.RequireHttpsMetadata = true;

options.ClientId = "client\_hrm";

options.ClientSecret = "client\_hrm\_secret";

options.ResponseType = "code";

options.Scope.Add("profile");

options.Scope.Add("hrm\_scope");

options.GetClaimsFromUserInfoEndpoint = true;

options.ClaimActions.MapAll();

options.SaveTokens = true;

// Set the correct name claim type

options.TokenValidationParameters = new TokenValidationParameters

{

NameClaimType = "name"

};

});

Authority: Identity Server

ClientId, ClientSecret, Scope là những thông tin được cấu hình tại Clients và Scopes ở Server Side

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

}

Thêm app.UseAuthentication() và app.UseAuthorization tại Configure method

namespace Client.Mvc.Controllers

{

[Authorize]

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

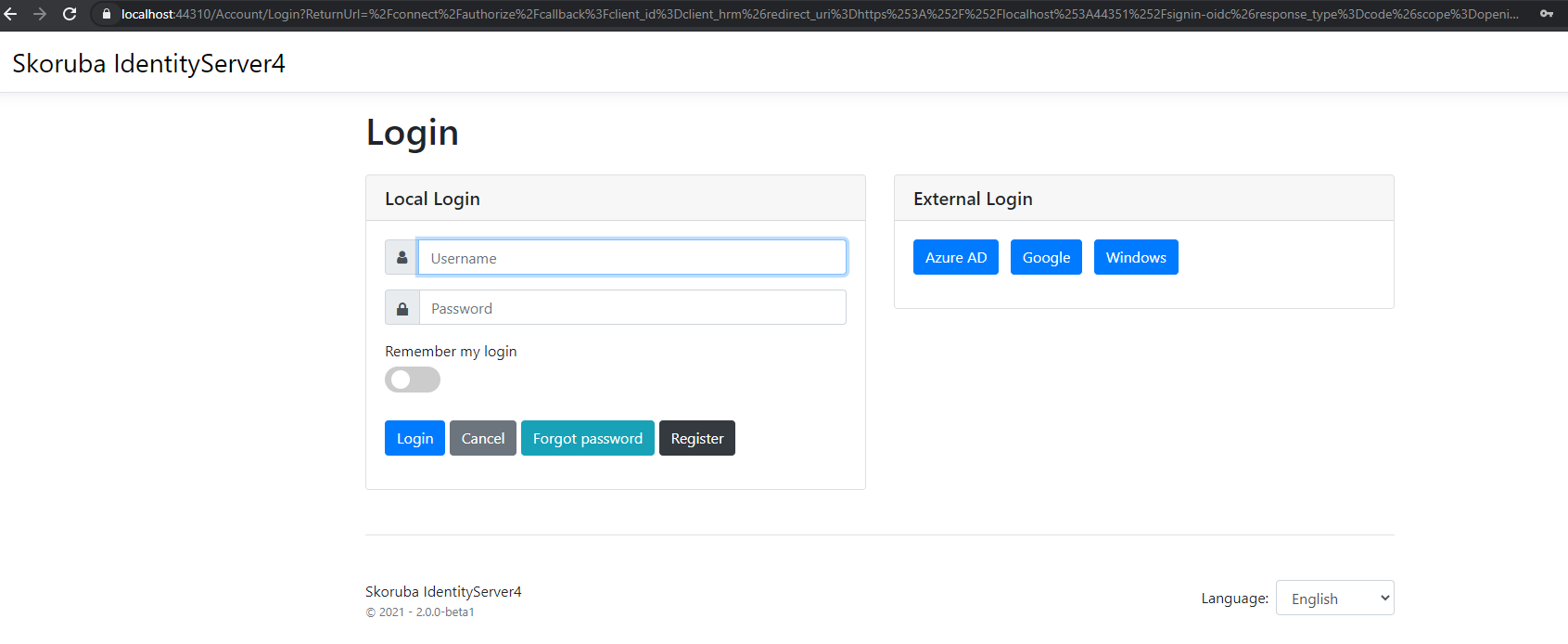
}

}

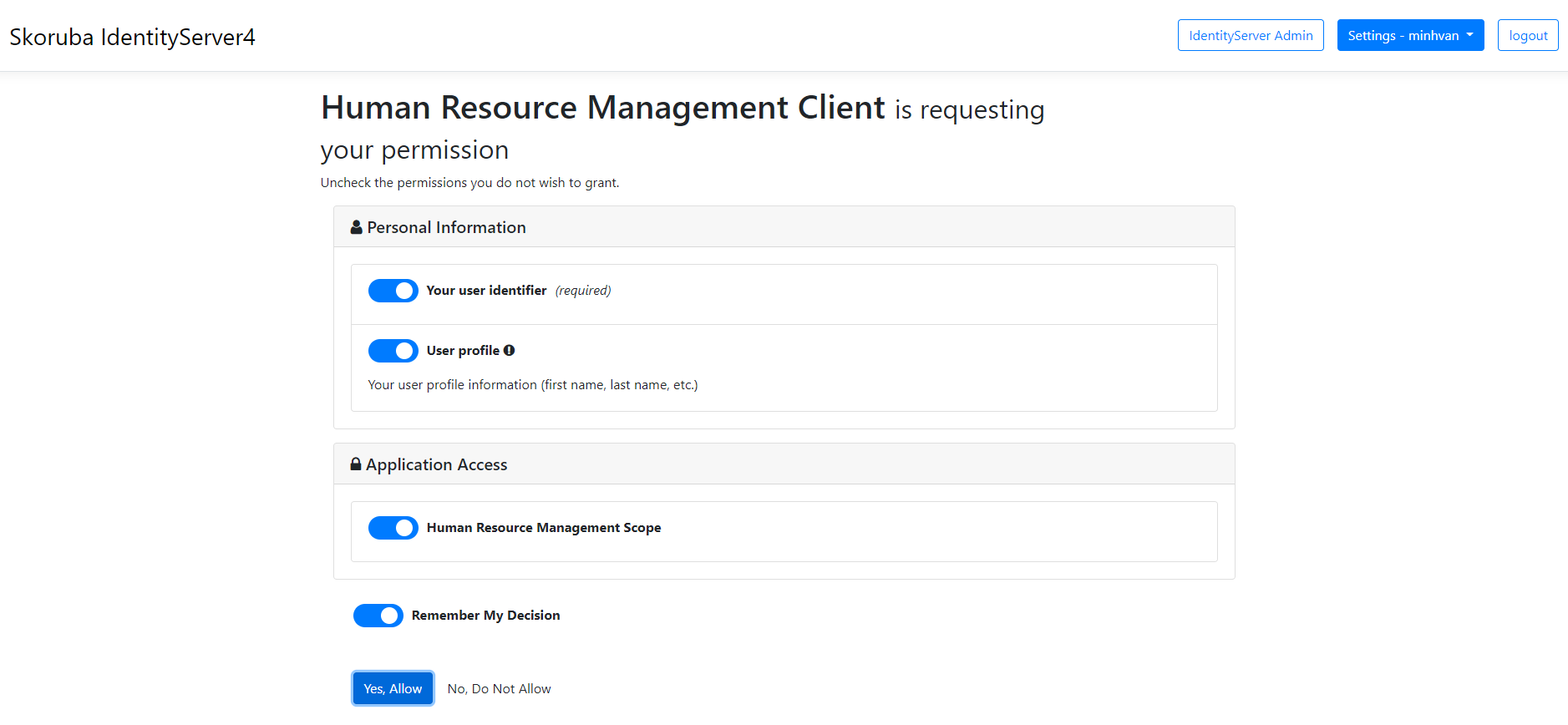
}

Tại HomeController thêm AuthorizeAttribute, việc này để kiểm tra khi muốn truy cập vào trang chủ của Client HRM cần được phân quyền

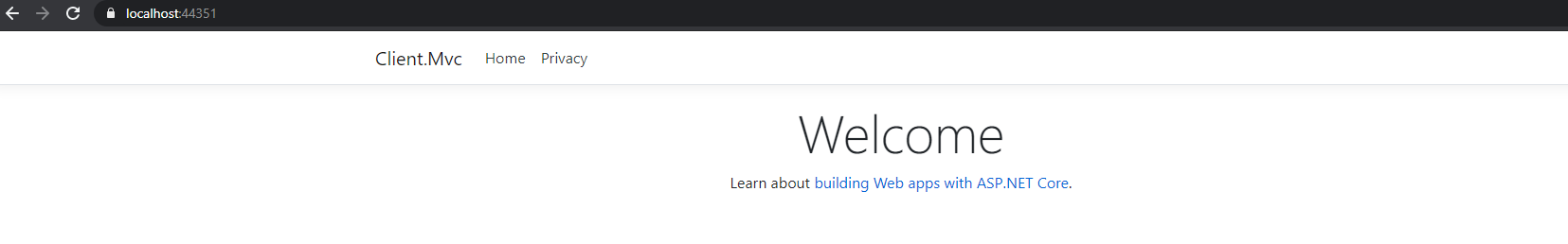
###### Chạy HRM Client App



Chúng ta được Redirect sang trang Login của Identity Server (<https://localhost:44310/>...)



Login bằng một tài khoản hợp lệ, chúng ta sẽ được thông tin các Resource có thể truy cập, chọn Yes, Allow



Redirect tới HRM Client của chúng ta.

# Client Configuration

## Net Core MVC Client

* ClientId: Identifier của Client
* ClientName: Tên Client
* ClientUri: Địa chỉ Client - <https://vnr-identityserver-admin.com>
* AllowedGrantTypes: “authorization\_code”
* RequirePkce: true
* ClientSecrets: Client Secrets
* RedirectUris: <https://vnr-identityserver-admin.com/signin-oidc>
* FrontChannelLogoutUri: <https://vnr-identityserver-admin.com/signout-oidc>
* PostLogoutRedirectUris: <https://vnr-identityserver-admin.com/signout-callback-oidc>
* AllowedCorsOrigins: <https://vnr-identityserver-admin.com>
* AllowedScopes: openid, profile, email, roles

### Note

RedirectUris là địa chỉ nơi Identity và/hoặc Access Token lấy từ Authorization Server sẽ gửi tới Client.

FrontChannelLogoutUri