

Atualização .NET 6

Atualização de suporte ao Identity no ASP.NET 6

No ASP.NET 6 a classe `Startup.cs` foi substituída pela `Program.cs`, a configuração para o Identity utilizando ASP.NET 6 não mudou em nada basicamente, é apenas uma questão de onde colocar as devidas configurações.

Considere a classe `Program.cs` abaixo como o resultado final das aulas deste módulo (atenção aos comentários no código):

```
using.AspNetCore.Identity.Areas.Identity.Data;
using.AspNetCore.Identity.Extensions;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

builder.Configuration
    .SetBasePath(builder.Environment.ContentRootPath)
    .AddJsonFile("appsettings.json", true, true)
    .AddJsonFile($"appsettings.{builder.Environment.EnvironmentName}.json",
true, true)
    .AddEnvironmentVariables();

// *** Configurando serviços no container ***

// Política de cookies (LGPD)
builder.Services.Configure<CookiePolicyOptions>(options =>
{
    options.CheckConsentNeeded = context => true;
    options.MinimumSameSitePolicy = SameSiteMode.None;
});

// Adicionando suporte ao contexto do Identity via EF
builder.Services.AddDbContext<AspNetCoreIdentityContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("AspNetCoreIdenti
tyContextConnection")));

// Adicionando configuração padrão do Identity
builder.Services.AddDefaultIdentity<IdentityUser>(options =>
options.SignIn.RequireConfirmedAccount = true)
    .AddEntityFrameworkStores<AspNetCoreIdentityContext>();

// Adicionando Autorizações personalizadas por policies
builder.Services.AddAuthorization(options =>
{
```

```

        options.AddPolicy("PodeExcluir", policy =>
policy.RequireClaim("PodeExcluir"));

        options.AddPolicy("PodeLer", policy => policy.Requirements.Add(new
PermissaoNecessaria("PodeLer")));
        options.AddPolicy("PodeEscrever", policy => policy.Requirements.Add(new
PermissaoNecessaria("PodeEscrever")));
    });

    // Resolvendo DI para o Handler de Authorization
builder.Services.AddSingleton<IAuthorizationHandler,
PermissaoNecessariaHandler>();

    // Adicionando MVC no pipeline
builder.Services.AddControllersWithViews();

    // Adicionando suporte a componentes Razor (ex. Telas do Identity)
builder.Services.AddRazorPages();

var app = builder.Build();

    // *** Configurando o request dos serviços no pipeline ***

if (app.Environment.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
    app.UseMigrationsEndPoint();
}
else
{
    app.UseExceptionHandler("/erro/500");
    app.UseStatusCodePagesWithRedirects("/erro/{0}");
    app.UseHsts();
}

    // Redirecionamento para HTTPS
app.UseHttpsRedirection();

    // Uso de arquivos estáticos (ex. CSS, JS)
app.UseStaticFiles();

    // Adicionando suporte a rota
app.UseRouting();

    // Autenticacao e autorização (Identity)
app.UseAuthentication();
app.UseAuthorization();

    // Rota padrão
app.MapControllerRoute("default", "{controller=Home}/{action=Index}/{id?}");

    // Mapeando componentes Razor Pages (ex: Identity)
app.MapRazorPages();

app.Run();

```