

第23期-权限管理系统

2020年8月10日 13:01

理解概念

权限提供者名：用户、角色、客户端和其它扩展。

权限提供主键：能够在某维度唯一标识提供者的主键

权限策略名称：权限名称

权限控制列表：编号为xx的xx提供者授予xx权限，编号为xx的XX提供者撤销XX权限。

权限管理：权限控制列表的 CRUD 增删改查。

安装权限管理模块

```
Install-package Volo.Abp.PermissionManagement.Web
Install-package Volo.Abp.PermissionManagement.Application
Install-package Volo.Abp.PermissionManagement.EntityFrameworkCore
Install-package Volo.Abp.EntityFrameworkCore.SqlServer
```

```
[DependsOn(typeof(AbpPermissionManagementWebModule),
typeof(AbpPermissionManagementApplicationModule),
typeof(AbpPermissionManagementEntityFrameworkCoreModule), typeof(AbpEntityFrameworkCoreSqlServerModule))]
```

```
Configure<PermissionManagementOptions>(options =>
{
    options.ProviderPolicies.Add(UserPermissionValueProvider.ProviderName,
"Cat_Management");
    options.ProviderPolicies.Add(RolePermissionValueProvider.ProviderName,
"Cat_Management");
});
```

添加 DbContext 和 迁移程序

```
"ConnectionStrings": {
    "Default":
"Server=(LocalDb)\\MSSQLLocalDB;Database=HelloWorld;Trusted_Connection=True"
}
```

```
[ConnectionStringName("Default")]
public class HelloWorldDbContext : AbpDbContext<HelloWorldDbContext>
{
    public HelloWorldDbContext(DbContextOptions<HelloWorldDbContext> options) :
base(options)
```

```

    {
    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        base.OnModelCreating(modelBuilder);

        modelBuilder.ConfigurePermissionManagement();
    }
}

public class MyMigrationsDbContextFactory :
    IDesignTimeDbContextFactory<HelloWorldDbContext>
{
    public HelloWorldDbContext CreateDbContext(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("appsettings.json", optional: false).Build();

        var builder = new DbContextOptionsBuilder<HelloWorldDbContext>()
            .UseSqlServer(configuration.GetConnectionString("Default"));

        return new HelloWorldDbContext(builder.Options);
    }
}

Configure<AbpDbContextOptions>(options =>
{
    options.UseSqlServer();
});

context.Services.AddAbpDbContext<HelloWorldDbContext>(options =>
{
    options.AddDefaultRepositories(includeAllEntities: true);
});

```

Install-package Microsoft.EntityFrameworkCore.Tools

演示迁移

```
context.ServiceProvider.GetRequiredService<IDataSeeder>().SeedAsync().Wait();
```

使用代码管理权限

通常权限管理是管理员用户使用权限管理模态框进行授权，如果你想要通过代码管理权限，可以注入使用 `IPermissionManager` 接口，**该接口在权限管理模块定**

义。Volo.Abp.PermissionManagement.sln 解决方案中提供源码。

```
public class MyPermissionExample : ITransientDependency
{
    private readonly IPermissionManager _permissionManager;

    public MyPermissionExample(IPermissionManager permissionManager)
    {
        _permissionManager = permissionManager;
    }

    public async Task GrantPermissionForUserAsync(string userId, string
permissionName)
    {
        await _permissionManager.SetAsync(permissionName, "U", userId, true);
    }

    public async Task ProhibitPermissionForUserAsync(string userId, string
permissionName)
    {
        await _permissionManager.SetAsync(permissionName, "U", userId, false);
    }
}
```

使用 WebAPI 管理权限

GET /api/permission-management/permissions?providerName=R&providerKey=admin
PUT /api/permission-management/permissions?providerName=U&providerKey=userid1

内部处理机制

PermissionsController->IPermissionAppService->
IPermissionDefinitionManager+IPermissionManager

IPermissionManager->*IPermissionManagementProvider->IPermissionGrantRepository

上节课权限判断->PermissionStore->
IPermissionDefinitionManager+IPermissionGrantRepository+IDistributedCache

权限管理提供者

```
public class MyPermissionManagementProvider : PermissionManagementProvider
{
    public MyPermissionManagementProvider(
        IPermissionGrantRepository permissionGrantRepository,
        IGuidGenerator guidGenerator, ICurrentTenant currentTenant)
        : base(permissionGrantRepository, guidGenerator, currentTenant)
```

```
{  
  
}  
  
    public override string Name => RolePermissionValueProvider.ProviderName;  
}  
  
options.ManagementProviders.Add<MyPermissionManagementProvider>();
```