### 角色权限

如果我们**授予**角色权限，则具有此角色的所有用户都有权获得该权限（除非特定用户明确禁止）。

我们使用**RoleManager**来更改角色的权限。例如， **SetGrantedPermissionsAsync**可用于在一个方法调用中更改角色的所有权限：

public class RoleAppService : IRoleAppService

{

private readonly RoleManager \_roleManager;

private readonly IPermissionManager \_permissionManager;

public RoleAppService(RoleManager roleManager, IPermissionManager permissionManager)

{

\_roleManager = roleManager;

\_permissionManager = permissionManager;

}

public async Task UpdateRolePermissions(UpdateRolePermissionsInput input)

{

// 获取要修改的角色

var role = await \_roleManager.GetRoleByIdAsync(input.RoleId);

// 获取权限列表

var grantedPermissions = \_permissionManager

.GetAllPermissions()

.Where(p => input.GrantedPermissionNames.Contains(p.Name))

.ToList();

// 修改角色权限

await \_roleManager.SetGrantedPermissionsAsync(role, grantedPermissions);

}

}

### 用户权限

public class UserAppService : IUserAppService

{

private readonly UserManager \_userManager;

private readonly IPermissionManager \_permissionManager;

public UserAppService(UserManager userManager, IPermissionManager permissionManager)

{

\_userManager = userManager;

\_permissionManager = permissionManager;

}

public async Task ProhibitPermission(ProhibitPermissionInput input)

{

// 获取要修改的用户

var user = await \_userManager.GetUserByIdAsync(input.UserId);

// 获取权限列表

var permission = \_permissionManager.GetPermission(input.PermissionName);

// 修改权限

await \_userManager.ProhibitPermissionAsync(user, permission);

}

}

UserManager有许多方法来控制用户的权限。使用UserManager的ProhibitPermissionAsync方法禁止用户的权限。

配置好权限或可进行 授权 使用，详见 授权文档 ，