### 角色实体

Role实体表示**应用程序**的**角色**。它应该来自**AbpRole**类，如下所示：

public class Role : AbpRole<Tenant, User>

{

//add your own role properties here

}

AbpRole定义了一些属性。最重要的是：

* **名称**：租户中角色的唯一名称。
* **DisplayName**：角色的显示名称。
* **IsDefault**：默认情况下是否将此角色分配给新用户？
* **IsStatic**：这个角色是静态的吗？（在预构建期间设置，不能删除）。

角色用于对**权限**进行**分组**。当用户拥有角色时，他/她将拥有该角色的所有权限。用户可以拥有 **多个**角色。此用户的权限将合并所有已分配角色的所有权限。

#### 动态与静态角色

* **静态角色**：静态角色具有已知**名称**（如“admin”），无法更改（我们可以更改**显示名称**）。它存在于系统启动时，无法删除。
* **动态（非静态）角色**：我们可以在部署后创建动态角色。然后我们可以为该角色授予权限，我们可以将角色分配给某些用户，我们可以将其删除。

使用**IsStatic**属性为角色设置它。我们还必须 在模块的[PreInitialize](https://aspnetboilerplate.com/Pages/Documents/Module-System)方法中**注册**静态角色。假设我们对租户有一个“管理员”静态角色：

Configuration.Modules.Zero().RoleManagement.StaticRoles.Add(new StaticRoleDefinition("Admin", MultiTenancySides.Tenant));

#### 默认角色

可以将一个或多个角色设置为**默认值**。使用**IsDefault** 属性进行设置。

### 角色管理

**RoleManager**是一种为角色执行**域逻辑**的服务：

public class RoleManager : AbpRoleManager<Tenant, Role, User>

{

//...

}

您可以[注入](https://aspnetboilerplate.com/Pages/Documents/Dependency-Injection)并使用RoleManager来创建，删除，更新角色，为角色授予权限等等。