租户管理

关于多租户

我们强烈建议阅读这个文件前阅读[多租户文档](http://www.aspnetboilerplate.com/Pages/Documents/Multi-Tenancy)。

启用多租户

ASP.NET Boilerplate和module-zero可以运行多租户或单租户模式。多租户默认为禁用。我们可以在我们的模块PreInitialize方法启用它，使之如下所示：

[DependsOn(typeof(AbpZeroCoreModule))]

public class MyCoreModule : AbpModule

{

public override void PreInitialize()

{

**Configuration.MultiTenancy.IsEnabled = true;**

}

...

}

需要注意的是：即使我们的应用程序是不是多租户，我们必须定义一个默认的租户（见本文件的默认租户部分）。

当我们创建一个基于ASP.NET Boilerplate和module-zero的工程模板时，我们有一个租户的实体和TenantManager领域服务。

租户实体

租户实体代表应用程序的一个租户。

public class Tenant : AbpTenant<Tenant, User>

{

}

它源自泛型类AbpTenant。租户实体被存储在数据库的AbpTenants表中。您可以添加自定义属性到租户类。

AbpTenant类定义一些基本的属性，最importants是：

* TenancyName：这是一个租户在应用程序中的唯一名称。它通常不应该被改变。它可用于分配的子域给租户如“mytenant.mydomain.com”。 Tenant.TenancyNameRegex常量定义命名规则。
* Name：租户的一个任意的，人性化可读的，长名称。
* IsActive：True，这个租户可以使用该应用程序。如果是false，这个租户没有用户可以登录到系统。

AbpTenant类是从FullAuditedEntity继承。这意味着它具有的创建，修改和删除审计属性。这也是软删除。所以，当我们删除租客，它不是从数据库中删除，只是标记为删除。

最后，AbpTenant的Id被定义为int。

租户管理器

租客经理器是租户执行域逻辑的服务：

public class TenantManager : AbpTenantManager<Tenant, Role, User>

{

public TenantManager(IRepository<Tenant> tenantRepository)

: base(tenantRepository)

{

}

}

你可以在这里添加你自己的方法。你也可以覆盖AbpTenantManager基类的任何方法针对自己的需要。

默认租户

ASP.NET Boilerplate和module-zero假设有一个预先定义的租户，它的TenancyName是“Default”和Id是1。在一个单租户应用，这被用作为单租户。在一个多租户应用程序，你可以删除它或使其无效。