

# 第33期-多租户存储与源码解析

2020年9月27日 9:14

## 租户存储器

ITenantStore 用于从数据源获取租户配置，默认实现 DefaultTenantStore 从配置文件中读取租户配置。您可以配置AbpDefaultTenantStoreOptions 选项或在appsettings.json文件中进行设置：

```
Configure<AbpDefaultTenantStoreOptions>(options =>
{
    options.Tenants = new[]
    {
        new TenantConfiguration(
            Guid.Parse("b1f14a2a-9539-6f03-5db5-0898ed7ae901"),
            "tenant1"
        ),
        new TenantConfiguration(
            Guid.Parse("b1f14a2a-9539-6f03-5db5-0898ed7ae902"),
            "tenant2"
        )
    }
    {
        ConnectionStrings =
        {
            {ConnectionStrings.DefaultConnectionStringName,
"..."}
        }
    }
});

});

"Tenants": [
{
    "Id": "446a5211-3d72-4339-9adc-845151f8ada0",
    "Name": "tenant1"
},
{
    "Id": "25388015-ef1c-4355-9c18-f6b6ddbaf89d",
    "Name": "tenant2",
    "ConnectionStrings": {
        "Default": "...tenant2's db connection string here..."
    }
}
]
```

## 租户机制实现原理

模块: Volo.Abp.MultiTenancy -> Volo.Abp.AspNetCore.MultiTenancy

MultiTenancyMiddleware -> ITenantConfigurationProvider -> ITenantResolver + ITenantStore +  
ICurrentTenant

ITenantResolver -> \* ITenantResolveContributor

ITenantStore -> DefaultTenantStore -> AbpDefaultTenantStoreOptions -> IConfiguration

ICurrentTenant -> ICurrentTenantAccessor -> SET or GET BasicTenantInfo

数据: AbpDbContext -> HasQueryFilter