

## 第32期-多租户使用独立数据库部署

2021年5月8日 22:16

### 租户信息与连接字符串配置

```
[Serializable]
public class TenantConfiguration
{
    public Guid TenantId { get; set; }

    public string TenantName { get; set; } = string.Empty;

    public Dictionary<string, string>? ConnectionStrings { get; set; }
}
```

### 连接字符串提供者接口

```
public interface IConnectionStringProvider
{
    Task<string> GetAsync(string? connectionStringName = null);
}
```

### 默认连接字符串提供者实现

```
public class DefaultConnectionStringProvider : IConnectionStringProvider
{
    protected readonly IConfiguration _configuration;

    public DefaultConnectionStringProvider(IConfiguration configuration)
    {
        _configuration = configuration;
    }

    public virtual Task<string> GetAsync(string? connectionStringName = null)
    {
        connectionStringName ??= DbConstants.DefaultConnectionStringName;
        return
            Task.FromResult(_configuration.GetConnectionString(connectionStringName));
    }
}
```

### 多租户连接字符串提供者实现

```
public class TenantConnectionStringProvider : DefaultConnectionStringProvider
```

## 从远程租户配置系统读取连接字符串

将租户存储在 IdentityServer 服务器上，使用公开的 WebApi 请求不同租户的连接字符串，可自行实现。

## 使用数据库连接拦截器动态变更连接字符串

```
public class TenantDbConnectionInterceptor : DbConnectionInterceptor
```

```
var connectionStringProvider =  
serviceProvider.GetRequiredService<IConnectionStringProvider>();  
optionsBuilder.AddInterceptors(new  
TenantDbConnectionInterceptor(connectionStringProvider));
```

## 多租户独立数据库部署方案

```
Update-Database -Context DeviceCenterDbContext -Project  
ZeroStack.DeviceCenter.Infrastructure -StartupProject ZeroStack.DeviceCenter.Infrastructure  
-Connection your_connection_string
```