

第11期-扩展实体对象属性

2020年7月31日 9:39

自定义扩展属性

扩展属性是一种存储实体的一些额外数据但不用更改实体的方式，实体应该实现 IHasExtraProperties 接口，所有预构建模块定义的聚合根实体都实现了 IHasExtraProperties 接口，所以你可以在这些实体中存储额外的属性。

```
[Fact]
public async Task Should_Set_Extra_Properties()
{
    var book1 = await _bookRepository.GetAsync(bk => bk.Name == "ASP.NET Core");
    book1.SetProperty("Title", "This is .NET Core book.");
    await _bookRepository.UpdateAsync(book1);

    var book2= await _bookRepository.GetAsync(bk => bk.Name == "ASP.NET Core");
    string title = book2.GetProperty<string>("Title");
    title.ShouldBe("This is .NET Core book.");
}
```

设置实体隐藏属性

```
ObjectExtensionManager.Instance.MapEfCoreProperty<Book, string>("Remark", b =>
b.HasMaxLength(30));
```

Add-Migration Add_Book_Entity_Remark_Field

Update-Database

第06期-隐藏属性与关系术语约定

```
[Fact]
public async Task Should_Set_Extra_Properties()
{
    var book1 = await _bookRepository.GetAsync(bk => bk.Name == "ASP.NET Core");
    book1.SetProperty("Remark", "This is book.");
    await _bookRepository.UpdateAsync(book1);

    var book2= await _bookRepository.GetAsync(bk => bk.Name == "ASP.NET Core");
    string title = book2.GetProperty<string>("Remark");
    title.ShouldBe("This is book.");
}
```

多个实体映射到同一个数据库表

你需要处理 EF Core 的数据库迁移架构，你的应用程序数据库和模块数据库将是同一个物理数据库，后续无法分离，耦合度较高，数据强一致性。

```
builder.Entity<AppUser>(b =>
{
    //Sharing the same table "AbpUsers" with the IdentityUser
    b.ToTable(AbpIdentityDbProperties.DbTablePrefix + "Users");
    b.ConfigureByConvention();
    b.ConfigureAbpUser();
});
```

创建自己的同步数据表

单体应用：在进程内使用本地事件总线监听实体更改进行数据同步。

分布式微服务：使用分布式事件总线订阅实体的更改事件并同步。

```
Add-Migration Add_Book2_Entity
```