

第19期-多模型切换与空间数据模型

2019年5月15日 22:16

多个模型之间切换

```
public class BloggingContext : DbContext
{
    public bool? IgnoreXmlUrlProperty { get; set; } = true;

    public BloggingContext(DbContextOptions<BloggingContext> options) :
base(options)
    {

    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        if (IgnoreXmlUrlProperty.HasValue)
        {
            if (IgnoreXmlUrlProperty.Value)
            {
                modelBuilder.Entity<Blog>().Ignore(e => e.XmlUrl);
            }
        }

        base.OnModelCreating(modelBuilder);
    }

    public DbSet<Blog> Blogs { get; set; }
}

public class Blog
{
    public int Id { get; set; }

    public string Url { get; set; }

    public string XmlUrl { get; set; }
}

public class CustomModelCacheKeyFactory : IModelCacheKeyFactory
{
    public object Create(DbContext context)
    {
        if (context is BloggingContext bloggingContext)
        {
            return (context.GetType(),
```

```

bloggingContext.IgnoreXmlUrlProperty());
    }

    return context.GetType();
}
}

options.UseSqlServer(configuration.GetConnectionString("test"))
.ReplaceService<IModelCacheKeyFactory, CustomModelCacheKeyFactory>()

```

GIS空间数据模型

空间数据是表示的物理位置和形状的对象，很多数据库都支持空间数据类型。EF Core 支持 NetTopologySuite 空间库，允许开发者建立数据库类型与CLR类型之间的空间数据映射。NetTopologySuite 是一个开源库。

若要使用 EF Core 使用空间数据，您需要安装相应的支持 NuGet 包，点线面映射。

EF Core 提供程序	空间 NuGet 包
Microsoft.EntityFrameworkCore.SqlServer	Microsoft.EntityFrameworkCore.SqlServer.NetTopologySuite
Microsoft.EntityFrameworkCore.Sqlite	Microsoft.EntityFrameworkCore.Sqlite.NetTopologySuite
Microsoft.EntityFrameworkCore.InMemory	NetTopologySuite
Npgsql.EntityFrameworkCore.PostgreSQL	Npgsql.EntityFrameworkCore.PostgreSQL.NetTopologySuite

这些包还支持反向工程，支持 Scaffold-DbContext 或 dotnet ef dbcontext scaffold。

```

Install-Package Microsoft.EntityFrameworkCore.SqlServer.NetTopologySuite -Version 2.2.3

```

```

optionsBuilder.UseSqlServer(@"", x => x.UseNetTopologySuite());

var geometryFactory = NtsGeometryServices.Instance.CreateGeometryFactory(4326);
var currentLocation = geometryFactory.CreatePoint(new Coordinate(-122.121512, 47.6739882));
_context.Blogs.Add(new Blog() { Url = "www.xcode.me", Location = currentLocation });
_context.SaveChanges();

Blog blog = _context.Blogs.Find(1);
Console.WriteLine(blog.Location);

var nearestCity = db.Cities
    .OrderBy(c => c.Location.Distance(currentLocation))

```

```
.FirstOrDefault();  
  
var currentCountry = db.Countries  
    .FirstOrDefault(c => c.Border.Contains(currentLocation));
```