

第53期-查询实体的元数据

2019年8月23日 10:56

什么是元数据？

[「百科」元数据被定义为：描述数据的数据，对数据及信息资源的描述性信息。](#)

[「知乎」什么是元数据？为何需要元数据？](#)

EF Core 底层元数据信息

[Microsoft.EntityFrameworkCore.Metadata Namespace](#)

若要获取上述底层元数据，可使用 IModel 接口中提供的两个最核心的方法。

```
IEnumerable<IEntityType> GetEntityTypes();  
IEntityType FindEntityType(string name);
```

通过两种方式获取 IModel 接口：DbContext.Model 或者 ModelBuilder.Model。

写几个常用的示例

查询表的元数据

```
var entityType = _context.Model.FindEntityType(typeof(Blog).FullName);  
string tableName = entityType.Relational().TableName;
```

动态获取属性元数据

```
var entityType = _context.Model.FindEntityType(typeof(Blog).FullName);  
IEnumerable<IProperty> properties = entityType.GetProperties();
```

```
foreach (var property in properties)  
{  
    Console.WriteLine(property.Name);  
}
```

动态获取列元数据

```
var entityType = _context.Model.FindEntityType(typeof(Blog).FullName);
```

```
IProperty property = entityType.FindProperty(nameof(Blog.Url));  
string columnName = property.Relational().ColumnName;
```

Lambda 表达式扩展方法

```
public static class DbContextMetadataExtensions  
{
```

```

        public static Iproperty FindProperty<TEntity, TProperty>(this DbContext dbContext, Expression<Func<TEntity,
TProperty>> expression)
        {
            var memberExpression = (MemberExpression)expression.Body;
            PropertyInfo propertyInfo = typeof(TEntity).GetProperty(memberExpression.Member.Name);
            var entityType = dbContext.Model.FindEntityType(typeof(TEntity).FullName);

            IProperty property = entityType.FindProperty(propertyInfo.Name);

            return property;
        }
    }

    IProperty property = _context.GetProperties<Blog, string>(m => m.Url);
    string columnType = property.Relational().ColumnType;

```