

第39期-集成Dapper数据访问

2020年9月27日 9:14

Dapper 介绍

Dapper 是一个轻量级 ORM 开源库，在速度方面被称为轻量级 ORM 之王，实际上与使用原始 ADO.NET 数据读取器一样快，ORM 是对象关系映射器，它负责数据库记录和编程语言对象之间的映射。

Dapper 通过为 IDbConnection 提供有用的扩展方法来对数据库进行 CRUD 操作。

<https://github.com/StackExchange/Dapper>

ABP 与 Dapper 集成

ABP 没有为 Dapper 封装许多功能，ABP Dapper 库在 ABP EntityFrameworkCore 的基础上提供了 DapperRepository<TDbContext> 基类，在其中提供了 Dapper 需要的 IDbConnection 和 IDbTransaction 属性，这两个属性可以和工作单元很好的配合。

Install-Package Volo.Abp.Dapper

[DependsOn(typeof(AbpDapperModule))]

实现 Dapper 仓储

```
public interface IDogDapperRepository
{
    Task<int> Create(Dog dog);

    Task<int> Delete(int id);

    Task<int> Update(int id, string name, int age);

    Task<IEnumerable<Dog>> GetAll();
}
```

```
public class DogDapperRepository : DapperRepository<DogStoreDbContext>,
IDogDapperRepository, ITransientDependency
{
    public DogDapperRepository(IDbContextProvider<DogStoreDbContext>
dbContextProvider) : base(dbContextProvider)
```

```

    {
    }

    public virtual async Task<int> Create(Dog dog)
    {
        string sql = "INSERT INTO AbpDogs (Name, Age) VALUES (@Name, @Age)";
        return await DbConnection.ExecuteAsync(sql, dog, DbTransaction);
    }

    public virtual async Task<int> Delete(int id)
    {
        string sql = "DELETE FROM AbpDogs WHERE ID=@ID";
        return await DbConnection.ExecuteAsync(sql, new { id },
        DbTransaction);
    }

    public virtual async Task<int> Update(int id, string name, int age)
    {
        string sql = "UPDATE AbpDogs SET Name=@Name, Age=@Age WHERE
        ID=@ID";
        return await DbConnection.ExecuteAsync(sql, new { id, name, age },
        DbTransaction);
    }

    public virtual async Task<IEnumerable<Dog>> GetAll()
    {
        string sql = "SELECT ID, Name, Age FROM AbpDogs";
        return await DbConnection.QueryAsync<Dog>(sql, DbTransaction);
    }
}

public class DogAppService : ApplicationService
{
    private readonly IDogDapperRepository _dogRepository;

    public DogAppService(IDogDapperRepository dogRepository)
    {
        _dogRepository = dogRepository;
    }

    public async Task<object> GetListAsync()
    {
        await _dogRepository.Create(new Dog { Name = "Dog3", Age = 33 });

        await _dogRepository.Delete(6);

        await _dogRepository.Update(1006, "Dog55", 55);

        var dogs = await _dogRepository.GetAll();
    }
}

```

```
        return new { dogs };  
    }  
}
```

Dapper.Contrib 开源库

Dapper 是介于 EF Core 与 ADO.NET 的折中选择，既满足手写查询的高性能需求，又简化了数据库对象映射为内存对象的繁杂工作。Dapper.Contrib 是对 Dapper 的进一步封装，使对象的基本增删改查等操作进一步简化。

<https://www.nuget.org/packages/Dapper.Contrib>

<https://github.com/StackExchange/Dapper/tree/main/Dapper.Contrib>

Dapper Transaction 开源库

<https://www.nuget.org/packages/Dapper.Transaction>

<https://github.com/zzzprojects/Dapper.Transaction>

<https://github.com/zzzprojects/Dapper.Transaction/blob/master/README.md>

关于 Dapper Plus 商业库

<https://www.nuget.org/packages/Z.Dapper.Plus>

<http://dapper-plus.net>