1 自定义上下文

public class ACConfigurationDbContext : DbContext, IConfigurationDbContext, ITransientDependency

{

public ACConfigurationDbContext(DbContextOptions<ACConfigurationDbContext> options) : base(options)

{

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

base.OnModelCreating(modelBuilder);

}

public Task<int> SaveChangesAsync()

{

return base.SaveChangesAsync();

}

}

2 自定义上下文工厂（用于迁移）

public class ACConfigurationDbContextFactory : IDesignTimeDbContextFactory<ACConfigurationDbContext>

{

public ACConfigurationDbContext CreateDbContext(string[] args)

{

var builder = new DbContextOptionsBuilder<ACConfigurationDbContext>();

var configuration = AppConfigurations.Get(WebContentDirectoryFinder.CalculateContentRootFolder());

builder.UseSqlServer(configuration.GetConnectionString("ACConfigurationDbContext"));

return new ACConfigurationDbContext(builder.Options);

}

}

3 自定义仓库

public class IdentityServerRepository<T>: IIdentityServerRepository<T> where T : class

{

private IDbContextProvider<ACConfigurationDbContext> \_dbContextProvider { get; set; }

public virtual DbContext Context => \_dbContextProvider.GetDbContext();

// private DbContext \_dbContext { get; set; }

private DbSet<T> \_dbSet => Context.Set<T>();

public IdentityServerRepository(

IDbContextProvider<ACConfigurationDbContext> dbContextProvider

)

{

\_dbContextProvider = dbContextProvider;

}

public T FirstOrDefault(object id)

{

return \_dbSet.Find(id);

}

public T FirstOrDefault(Expression<Func<T, bool>> expression)

{

return \_dbSet.FirstOrDefault(expression);

}

public async Task<T> FirstOrDefaultAsync(Expression<Func<T, bool>> expression)

{

return await \_dbSet.FirstOrDefaultAsync(expression);

}

public void Insert(T entity)

{

\_dbSet.Add(entity);

}

public void Update(T entity)

{

\_dbSet.Update(entity);

}

public void Remove(T entity)

{

\_dbSet.Remove(entity);

}

public IQueryable<T> GetAll()

{

return \_dbSet;

}

public IQueryable<T> GetAllIncludes(Expression<Func<T, object>>[] expressions)

{

IQueryable<T> dataList = \_dbSet;

foreach (var item in expressions) {

dataList = dataList.Include(item);

}

return dataList;

}

}

4 配置上下文选项

services.AddAbpDbContext<ACConfigurationDbContext>(options =>

{

var configuration = AppConfigurations.Get(WebContentDirectoryFinder.CalculateContentRootFolder());

options.DbContextOptions.UseSqlServer(configuration.GetConnectionString("ACConfigurationDbContext"));

});