后端开发说明文档

修订：2017-4-26

目录

[一、 项目架构 1](#_Toc481671480)

[二、 工具代码类 2](#_Toc481671481)

[三、 配置数据获取 2](#_Toc481671482)

[四、 实体数据生成 2](#_Toc481671483)

[五、 数据上下文配置 3](#_Toc481671484)

[六、 基础控制器BaseController 3](#_Toc481671485)

[七、 应用API 3](#_Toc481671486)

[八、 区域开发扩展 4](#_Toc481671487)

[九、 具体各模块的接口 4](#_Toc481671488)

[1 Api接口 (返回都是Json结构) 4](#_Toc481671489)

[1.1 Syshis 获取基本配置数据 4](#_Toc481671490)

[1.1.1 获取工作站的树形数据 4](#_Toc481671491)

[2 系统设置 4](#_Toc481671492)

[2.1 菜单设置 4](#_Toc481671493)

[2.1.1 Jqgrid查询系统菜单接口 4](#_Toc481671494)

[2.1.2 每个功能的新增修改删除操作 4](#_Toc481671495)

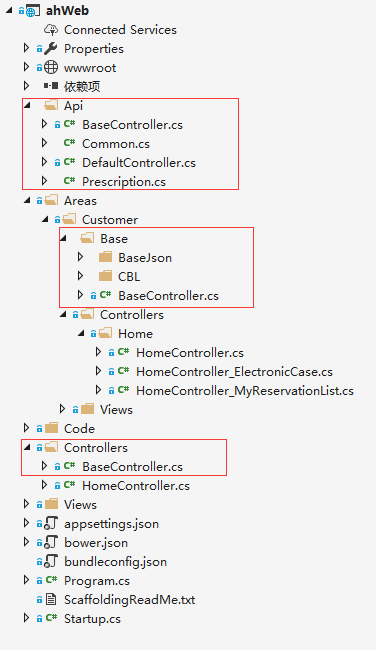
[2.1.3 为树形Select提供数据，选择父级功能。 4](#_Toc481671496)

[2.1.4 获取当前级别的菜单的个数 4](#_Toc481671497)

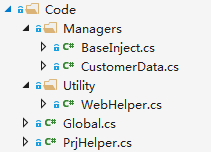
[2.2 工作站角色配置 4](#_Toc481671498)

[2.2.1 保存工作站选择的角色 4](#_Toc481671499)

1. 项目架构

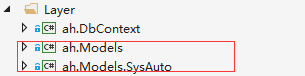


1. 工具代码类

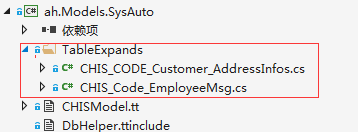


* 1. Managers管理器类
     1. 此部分主要是采用依赖注入方式来独立获取相关文件
  2. Utitlity应用类
     1. 此部分是一些与项目紧密相关的应用类，如果是公用函数，请写入到Ass基础类里
  3. PrjHelper
     1. 此类用于处理基本数据的转译操作等

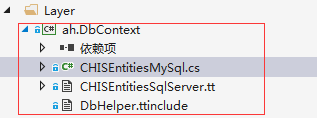
1. 配置数据获取
   1. 配置数据采用独立获取方式，不采用依赖注入方式获取
   2. 直接获取配置数据，可使用Global.Config获取  
      比如获取数据库连接字符串  
      
2. 实体数据生成



* 1. SysAuto是自动数据生成，里面有T4模板负责从数据库生成实体模型
  2. 实体模型的扩展



1. 数据上下文配置
   1. 基础的上下文



* 1. 实际应用的上下文，根据连接字符串和选择数据库，在BaseController配置

1. 通用增删该查
   1. 简单实体的增删改查方法说明
   2. 简单实体的增加

public virtual EntityEntry<TEntity> Add<TEntity>([NotNullAttribute] TEntity entity) where TEntity : class;

* 1. 简单实体的删除

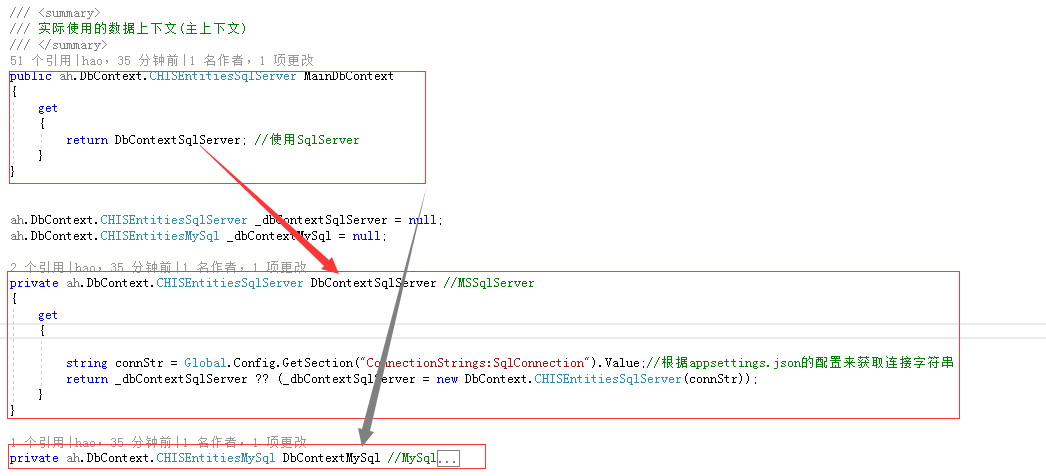
public virtual void RemoveRange([NotNullAttribute] IEnumerable<object> entities);

* 1. 简单实体的更新

public virtual EntityEntry<TEntity> Update<TEntity>([NotNullAttribute] TEntity entity) where TEntity : class;

* 1. 简单实体的查询
  2. 复杂对象的增删该查

1. 基础控制器BaseController
   1. 用于进行基础数据和工具的设置
   2. 配置和使用上下文，连接到不同的数据库

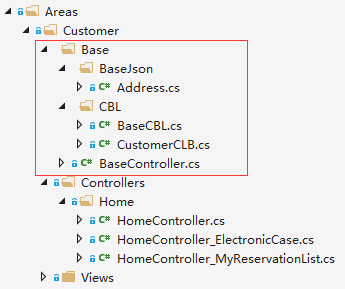


1. 应用API



* 1. 应用api将请求后返回对应的Json数据回传。
  2. 基础应用API BaseController
     1. 带DB的Controller —— BaseDBController
     2. 不带DB的Controller —— BaseController （因为在不同的命名空间，注意和基础的BaseController区别）

1. 区域开发扩展



* 1. 区域开发是把集中的功能都集成到域下开发，注意配置路由映射
  2. Base用于存取基础数据
     1. BaseJSON用于具体到此域下获取json的接口。
     2. CBL 是ControllerBissnessLayer 是用于处理复杂数据库逻辑用的层
     3. Aarea下的Controller使用自身的BaseController

1. 具体各模块的接口
2. Api接口 (返回都是Json结构)
   1. Syshis 获取基本配置数据
      1. 获取工作站的树形数据
         * StationsOfMy(int mode = 0)
3. 系统设置（Sys）
   1. 菜单设置
      1. Jqgrid查询系统菜单接口
         * GetGridJson\_Function(string keyWord)
      2. 每个功能的新增修改删除操作
         * public IActionResult CHIS\_SYS\_Function\_Edit(string op,CHIS.Models.CHIS\_SYS\_Function model, int recID = 0)
      3. 为树形Select提供数据，选择父级功能。
         * GetTreeSelectJson
      4. 获取当前级别的菜单的个数
         * Json\_FunctionCount(int parentId = 0)
   2. 工作站角色配置
      1. 保存工作站选择的角色
         * SaveStationRoles(int stationId, List<int> roles)
4. 基本设置（Basic）

3.1 数据字典（）