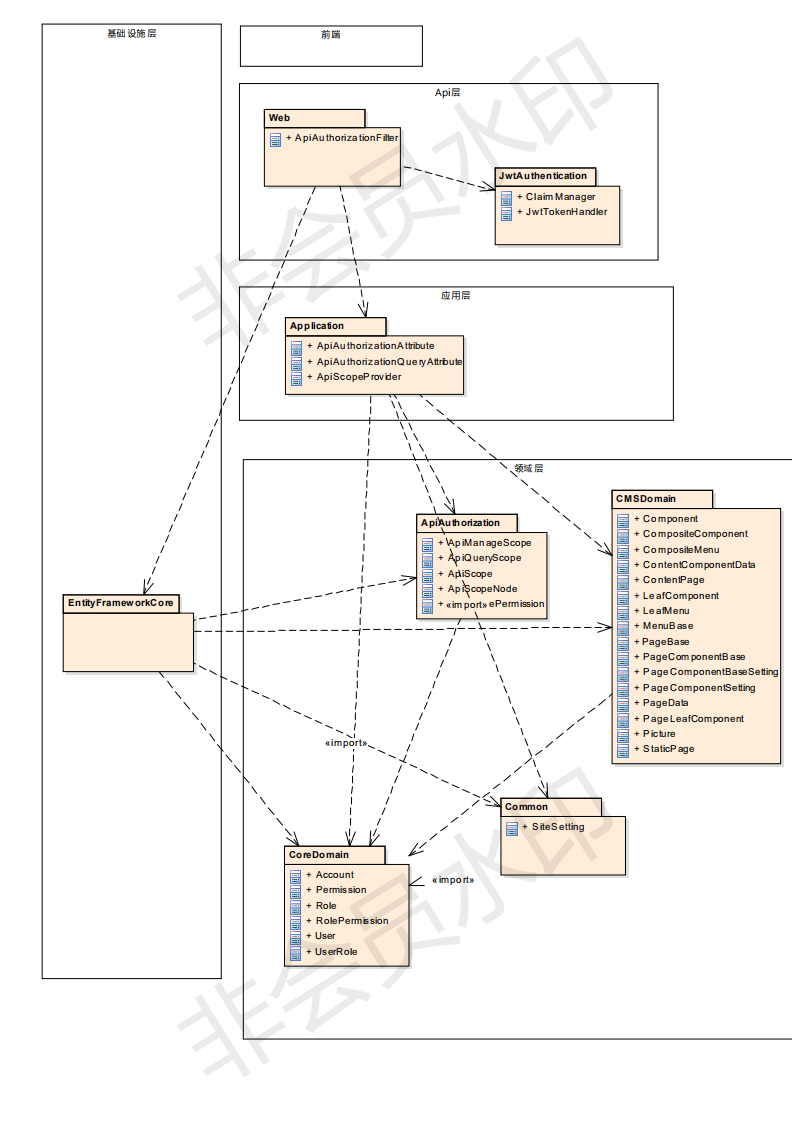
**后段使用技术**

Asp.net core Abp

**Cms整体架构设计**



**模块开发**

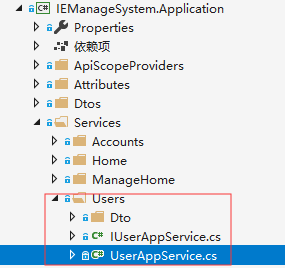
后端模块开放请参考Abp文档，地址：https://aspnetboilerplate.com/Pages/Documents

**Api接口开发**

接口开发同样遵循Abp文档，示例：

一般只实现业务逻辑的接口，只需在Application中添加即可，以UserAppService为例

推荐：在Services下添加



编写IUserAppService为接口

public interface IUserAppService : IApplicationService

{

    Task<GetUserInfoOutput> GetUserInfo (GetUserInfoInput input);

Task<SetUserInfoOutput> SetUserInfo (SetUserInfoInput input);

    Task<SetSafetyProblemOutput> SetSafetyProblem (SetSafetyProblemInput input);

}

实现IUserAppService

public class UserAppService : IEManageSystemAppServiceBase, IUserAppService {

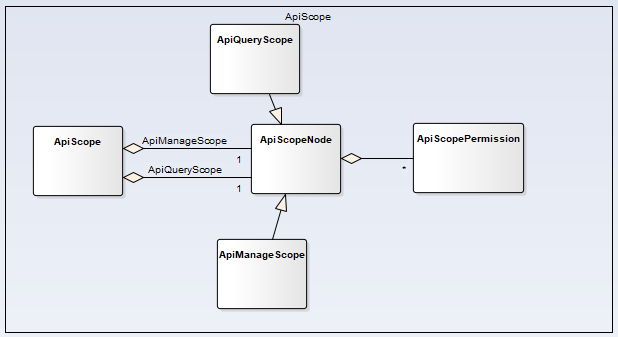
    ...

}

默认UserAppService 下Public的方法将被公布为WebApi

**Api域**

Api域描述一组webapi



Api域包含查询域节点和管理域节点，节点包含Api域权限，用户拥有其中的一个权限即可访问该节点

添加Api域：

Api域有ApiScopeProvider提供，如下添加一个User Api域

public class ApiScopeProvider

{

    // 指定一个域名称

    public const string User = "Personal.User";

    protected void ApiScopeProvide () {

        // 在 ApiScopeGroupDescribes 添加

        // 这里有一个ApiScopeGroupDescribe对象，可以不用搭理

        ApiScopeGroupDescribes.Add (

            new ApiScopeGroupDescribe () {

                Name = "Personal",

                ApiScopeDescribes = new List<ApiScopeDescribe> () {

                    new ApiScopeDescribe () {

                        Name = User,

                        DisplayName = "用户信息"

                    }

                }

            });

    }

}

**接口访问权限**

接口访问权限使用Api域进行控制

// 指定该类公布的 Webapi 为 User Api域

[ApiAuthorization (ApiScopeProvider.User)]

public class UserAppService : IEManageSystemAppServiceBase, IUserAppService

{

    // 该 webapi 为管理节点Api

    public async Task<SetUserInfoOutput> SetUserInfo (SetUserInfoInput input) {

        ...

    }

    // 该 webapi 为查询节点api

    [ApiAuthorizationQuery]

    public async Task<GetUserInfoOutput> GetUserInfo (GetUserInfoInput input) {

        ...

    }

}