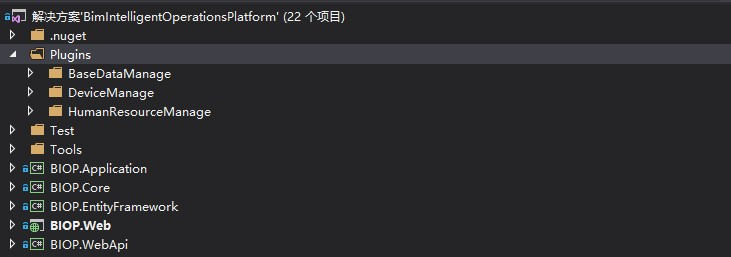
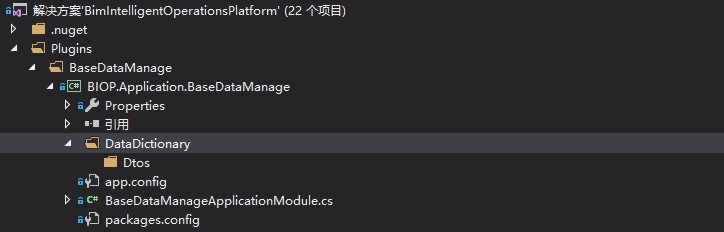
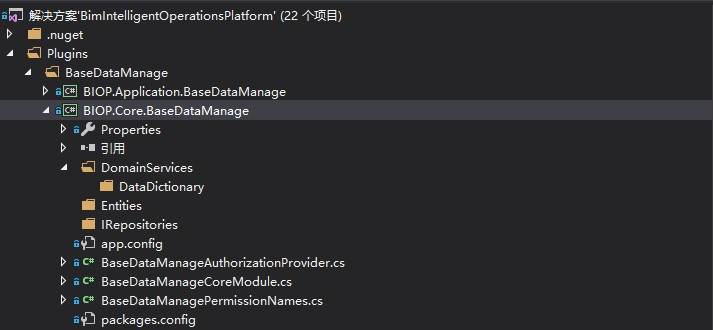
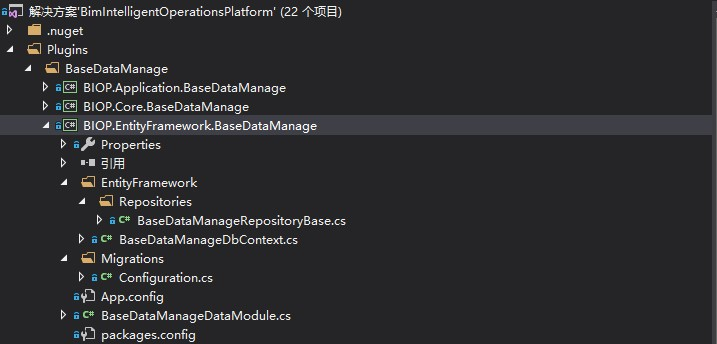
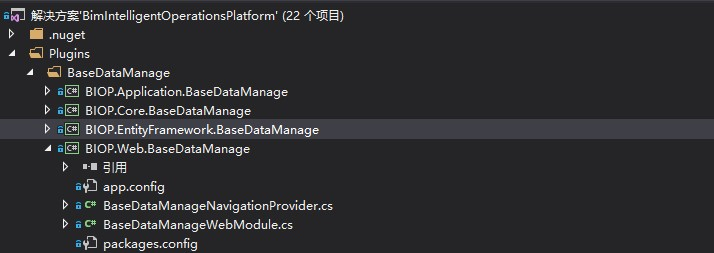
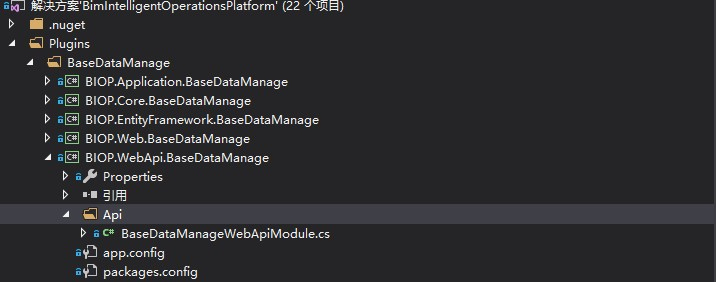
BIOP系统代码结构说明

1. 项目整体结构  
     
   如上图：  
   1>、项目主程序依次分为Application（应用层）、Core（领域层）、Entityframework（基础设施层）、Web（web页面层）、WebApi（Api层）；主程序模块中实现系统管理相关功能，web层包含所有业务模块的页面实现。  
   2>、业务组件模块统一放置于Plugins节点下，每个业务组件模块创建自身命名的节点
2. 业务组件模块结构  
     
   每个业务组件包含Application（应用层）、Core（领域层）、Entityframework（基础设施层）、Web（web页面层）、WebApi（Api层），命名规则依次为BIOP. Application.业务名称、BIOP. Core.业务名称、BIOP. Entityframework.业务名称、BIOP. Web.业务名称、BIOP. WebApi.业务名称。
3. 业务组件模块内部结构  
   1>、Application  
     
   如上图：模块内部包含一个AbpModule子类作为模块注入标识，命名规则为模块名称+ApplicationModule。每个功能点创建自身节点文件夹，文件夹中放置自身的Dto对象、应用层服务接口及实现类；应用接口类命名规则为：I+功能名称+AppService，实现类命名功能名称+AppService；Dto对象命名规则：名称+Dto/InputDto/OutputDto。  
   2>、Core  
     
   如上图：Core模块下分为DomainService、Entities、IRepository三个子目录；  
   Domain Service中按功能名称创建子目录节点，子节点放置领域服务接口及服务实现，领域服务命名规则为：名称+DomainService；  
   Entities目录下放置领域实体类，实体命名规则：名称+Entity，实体类内部做数据库表名称映射，表名称去掉Entity字符；  
   IRepository目录下放置仓储接口类,命名规则：名称+Repository；  
   Core模块包含系统权限定义信息，XXXPermissionNames中定义权限名称（字符串）、XXXAuthorizationProvider中负责权限信息的初始化操作；在权限定义中各业务模块根据需要对相应功能页面、页面中的操作定义相应的操作权限，用于在页面显示以及后台操作接口中进行权限验证。  
   3>、Entityframework  
      
   Entityframework模块负责与数据库交互功能的实现。XXXDbContext类用于初始化数据库上下文；XXXRepositoryBase作为当前模块仓储实现类的基类，继承自Abp框架中的仓储基类实现基本的数据库操作。业务仓储的实现类放在EntityFramework/Repositories目录下，根据业务定义仓储类名称，命名与Core层定义的仓储接口名称相匹配。  
   4>、Web  
     
   业务组件模块中的Web模块目前只负责定义导航菜单信息，在XXXNavigationProvider类中初始化导航菜单信息。  
   5>、WebApi  
     
   业务组件模块中的WebApi模块目前负责对当前业务模块的Api接口的注入实现，在Api目录下的XXXWebApiModule中实现。
4. Web主模块系统结构  
     
   Web主模块中负责所有业务功能模块的Web层实现，各个业务模块页面统一放置在App/Main/Views目录下，在该目录下按照业务模块名称创建子目录节点，子节点下按照功能名称创建目录，功能页面按照功能分类放置在相应的功能目录节点。Fonts目录下存放字体图标文件；css下放置样式文件；Scripts目录下放置Js文件。  
   