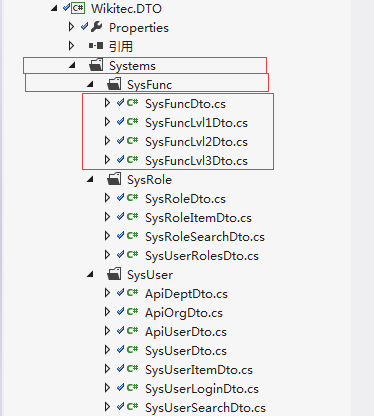
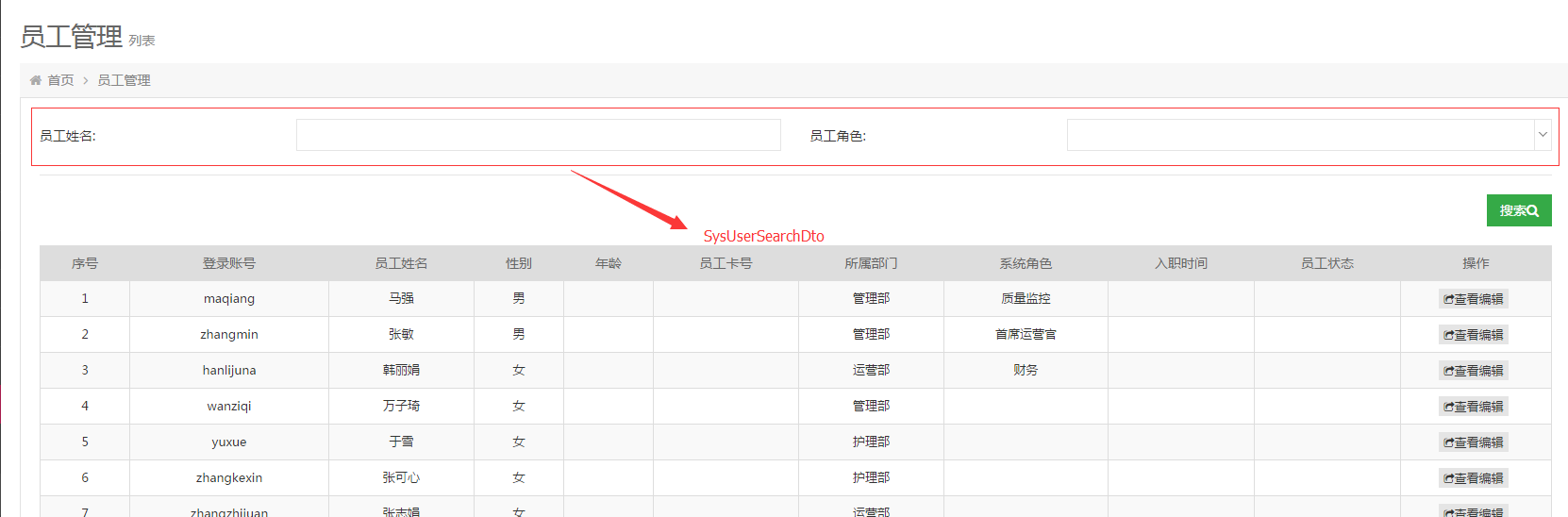
1. DTO定义：全称DataTransferObject, DTO的主要作用是用于Application和Repository层的数据传输。隔离Domain Model，让Domain Model的更改不会影响UI的展示，也不会暴漏业务逻辑。
   1. 好处如下
      1. DTO可以封装实体模型，减少远程调用。
      2. 隐藏内部逻辑
      3. …
2. DTO 目录结构
   1. 一级目录与Domain层保持一致。
   2. 由于一个功能可能涉及到多个DTO，所以在一级文件夹下建功能对应的文件夹



1. DTO命名规则
   1. 所有DTO文件以Dto为结尾。
   2. 列表使用的DTO命名规则为：实体名+Item+Dto 。如SysRoleItemDto，用于角色列表的展示。XXItemDto中之定义列表中使用的字段。这样可以减少数据库的查询字段。

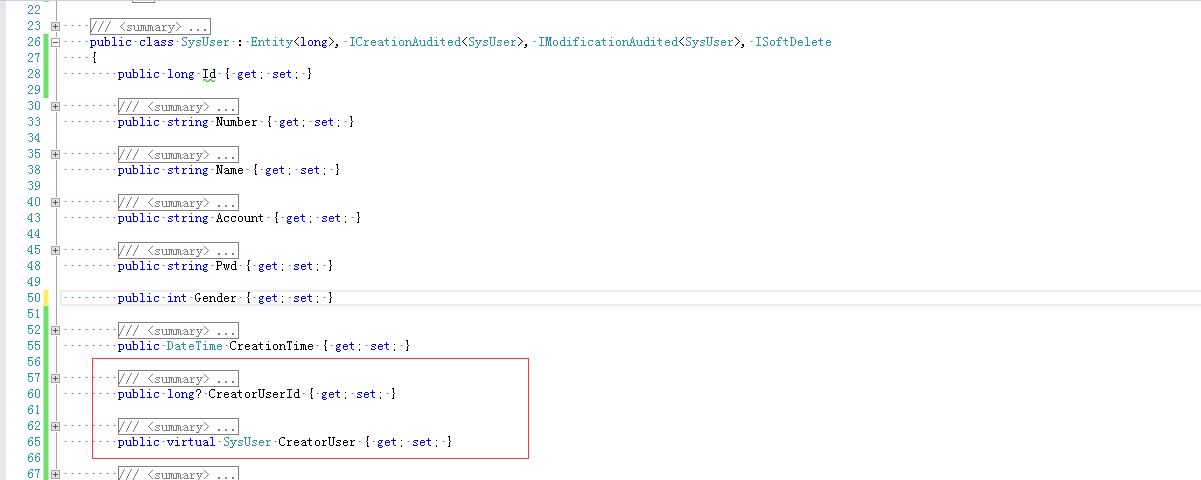


* 1. 查询使用的DTO命名规则：实体名+Search+Dto，如



* 1. 其它DTO命名规则：除Search和Item以外的DTO命名规则：实体名+用途+Dto 例如：SysUserLoginDto，这个DTO的用途是用于登陆后返回的保存在Cookie或缓存内的信息。

1. 普通DTO的说明
   1. DTO和实体类之间是通过AutoMapper做自动转换的。所以普通变量的名字是需要一致对应的。
   2. AutoMapper分为三种形式
      1. AutoMap：双向的，DTO可以转换为实体类，实体类也可以转换为DTO
      2. AutoMapFrom：只能由实体类转换为DTO
      3. AutoMapTo：只能由DTO转换为实体类
2. DTO嵌套或互相引用的说明
   1. DTO是实体类的一种封装，有时候实体类和实体类之间存在互相依赖的关系。比如每个用户都有一个创建人，在实体类中是通过外键和属性关联的。DTO和实体类之间是通过AutoMapper自动转换的。所以如果想在列表的DTO中展示创建人名字，只要在列表的DTO中定义一个CreatorUserName字段，AutoMapper会自动转换。



1. 关于DTO的后台验证：目前系统的验证分为前台验证和后台验证，前台验证主要就是脚本的验证，但是脚本的验证是不安全的。所以我们通过Unity的AOP做了切面验证。如果想要对DTO进行后台验证，首先要调用DTO的Application继承InterceptiveObject接口。然后DTO继承IValidate接口，并在字段上通过添加属性生命来确定验证逻辑。

