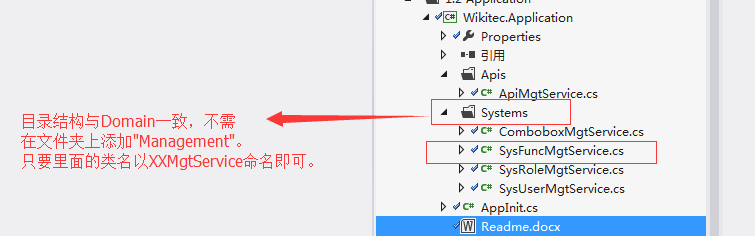
# Application层说明

Application层是我们主要业务逻辑的封装，引入Infrastructure之后，几乎不需要再ASPNET MVC的controller层中写任何逻辑。Controller只负责接收并将数据传递至MgtService层中。

# 目录结构



# 业务逻辑类说明

1. 类文件的命名：类的命名采用XXMgtService的形式。
2. 字段和属性说明



1. 内部方法的命名
   1. 基本的方法包括

void AddOrUpdate(SysRoleDto sysRoleDto)

PagedResult<SysRoleItemDto> GetSysRoles(SysRoleSearchDto sysRoleSearchDto) 例如GetSysRoleList可以直接写成GetSysRoles，不需要加list的后缀。

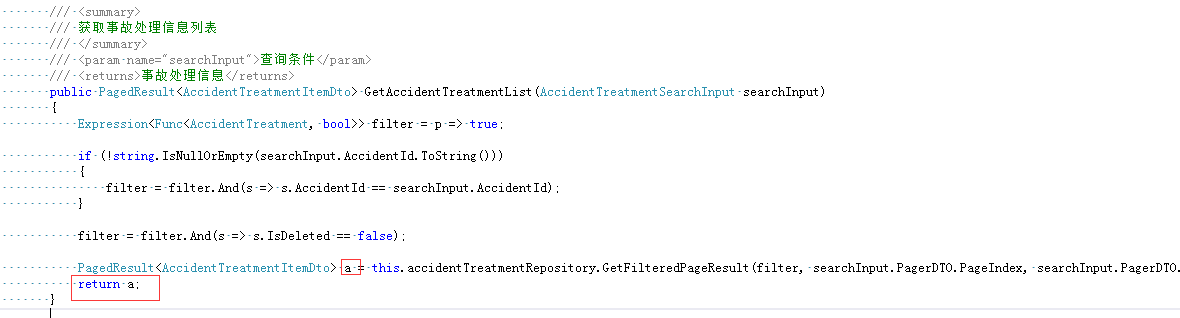
SysRoleDto GetSysRole(long? id)

void Delete(int id)

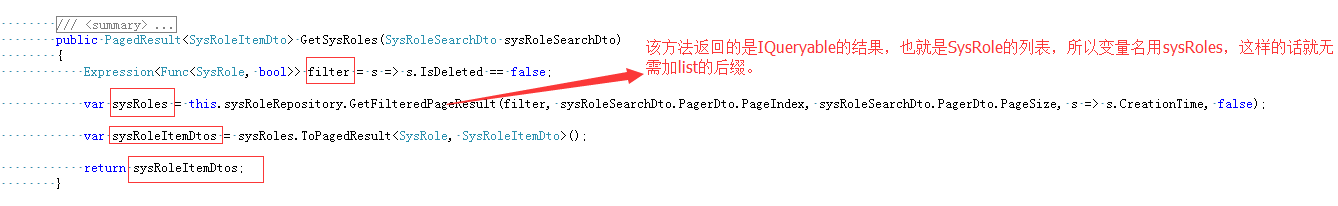
* 方法名以动词开头，明确的说明方法要做的动作
* 对于自定义类型的参数，参数名就是类名，只不过第一个字母小写SysRoleDto sysRoleDto。 方法中的参数尽量不使用缩写或简写。要有明确的含义。
  1. 其它方法的命名
* 方法名以动词开头，明确的说明方法要做的动作
* 对于自定义类型的参数，参数名就是类名，只不过第一个字母小写SysRoleDto sysRoleDto。 方法中的参数尽量不使用缩写或简写。要有明确的含义。

1. 方法内的一些规则

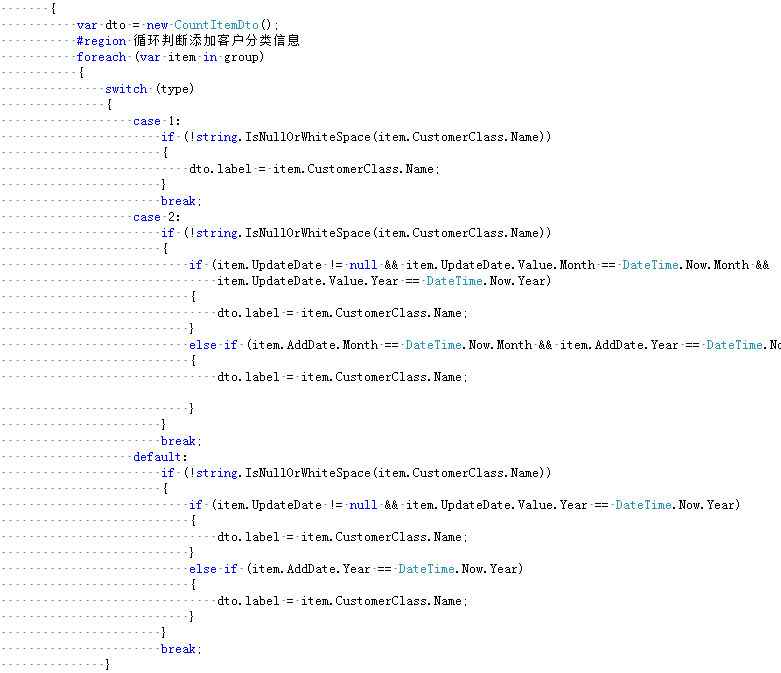
* 局部变量要有明确的说明，一定要杜绝a、b 这种变量名



* 可以参照下面的说明，每一个局部变量都要有意义。



* 对于foreach的内循环使用的变量，不建议命名为item，最好是明确的内容
* Swich case的操作尽量使用枚举



1. 其它说明
   1. 业务逻辑类中的代码无特殊情况不需要try catch处理，对于友好的异常只要抛出一个UserFriendlyException的异常。
   2. 对于事物的处理使用 this.unitOfWork.Commit();提交。