模板地址： <https://github.com/shoted/Template.git>

例子地址： [/社区运营/活动中心/Lexun.CelebrationAct](http://113.108.233.86:8080/svn/社区运营/活动中心/Lexun.CelebrationAct)

1.环境配置 （很重要）

开发环境 预生产环境 生产环境

Webconfig => appsettings 添加

开发环境 <add key="AspNetEnvironment" value="Development"/> 对应本地调式

预生产环境 <add key="AspNetEnvironment" value="Staging"/> 对应f8

生产环境 <add key="AspNetEnvironment" value="Production"/> 对应线上环境

配置环境的作用是

例如在本地调试，用户头像那些取不到，如果有取头像的代码，会一直等待，所以在开发环境就不需要取头像，所以在Ucommon里面有如下代码：

public static string GetUserNick(int userid)

{

return IsDevelopment ? "乐讯\_" + userid % 1000 : CUser.GetUserNick(userid);

}

当是开发环境的时候，就不会取头像，上线的时候，改下环境就行了

再如，在f8发内信可能只需要发送给测试id，这时候就判断是不是预生产环境，执行对应代码，上线的时候直接改环境就行，默认环境是生产环境，这样可以防止上线的时候忘记把代码改回来

2.页面

页面名字应该处理对应逻辑，所有页面继承TemplatePage ，继承后会强制重写get和post方法，对应http请求方式的get和post，http get请求执行get方法，post请求执行post方法。

Get方法只处理数据请求，不做业务逻辑

Post方法只处理业务逻辑，不做数据请求

例如，一个聊天功能，那么页面名字大概应该是ChatMessage.aspx 它的get方法应该是获取聊天记录，他的post方法应该是发送聊天

区分get和post的作用，当活动停止时，可能页面还需要展示数据，但是活动无法参与的时候只需要让base.Post方法返回活动结束，那么所有页面post方法都不会执行 base.IsEndActivity

是否需要登录重写base.NeedLogin()

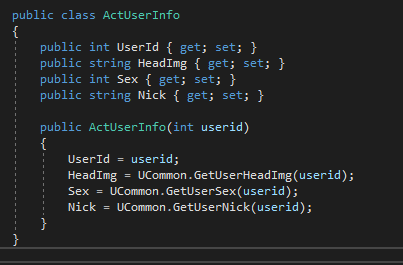
所有页面返回http状码 200系列表示成功 400表示客户端校验失败 500表示服务端出错

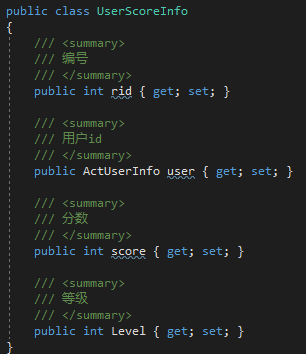
返回状态码代码 WJson.SetValue(true, HttpStatusCode.OK, "获取数据成功");

返回数据 WJson.AddDataItem("question", question);

3.对象

前端需要用户的昵称，头像数据，因此，所有的userid改成ActUserInfo，调用构造函数会生成用户信息对象，代码

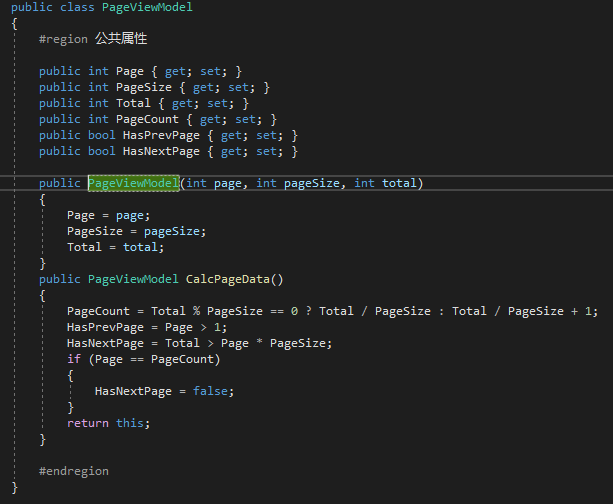




分页数据，后端帮前端算好，例如

PageViewModel pageViewModel = new PageViewModel(page, pageSize, total).CalcPageData();

WJson.AddDataItem("pagination", pageViewModel);



4.Ucommon

Ucommon中提供一些基本帮助方法，例如取用户头像，用户昵称，最好是取用户的基本信息用ucommon中的形式，区分好是开发环境还是生产环境，这样本地调试的时候会更方便，也可以自己灵活的添加。

SaftExcute（） 安全执行，catch异常

TaskExcute（） 线程执行，调用任务工厂执行。

判断环境方法，Ucommon. IsDevelopment

5.跨域

修改 base.ProcessAllowOrigin()下的allowDomain改为当前站点