GC新前台使用规范

2017/12/25 wujun

前言

随着公司业务的发展，GC老前台（http://wap.xiaotui.so）的业务模块俞显臃肿，且使用的框架还是ThinkPHP 3.1，如果再不断往里面新增模块，会对后期业务的维护工作增加成倍的难度。为减缓这一问题，现已将一些新接的模块写入到了GC新前台（http://newwap.greencampus.cc）,但同时由于GC新前台使用的框架版本不一样（ThinkPHP 3.2），容易在开发过程中出现各种问题。故特此编写这篇关于GC新前台使用规范的文章，着重介绍需要注意的地方。如有不对，欢迎及时斧正之。

目录

[GC新前台使用规范 1](#_Toc502061778)

[前言 1](#_Toc502061779)

[1， 关于GC新前台的说明： 2](#_Toc502061780)

[1.1） 测试服网址： 2](#_Toc502061781)

[1.2） 正式服网址： 2](#_Toc502061782)

[1.3） 使用场景： 2](#_Toc502061783)

[1.4） 目录说明： 3](#_Toc502061784)

[1.5） 前端目录说明： 3](#_Toc502061785)

[2， 关于新模块的建立： 4](#_Toc502061786)

[2.1）建立新模块： 4](#_Toc502061787)

[2.2）建立说明： 5](#_Toc502061788)

[3， 关于Controller层的编写： 5](#_Toc502061789)

[3.1）视图类： 5](#_Toc502061790)

[3.2）接口类： 6](#_Toc502061791)

[3.3）统一验证码发送： 7](#_Toc502061792)

[3.4）统一图片上传：（阿里OSS） 7](#_Toc502061793)

[3.5）统一获取微信配置：（微信JSSDK） 8](#_Toc502061794)

[4， 关于Service层的编写： 8](#_Toc502061795)

[4.1）模型对象定义： 9](#_Toc502061796)

[4.2）统一数据查询： 9](#_Toc502061797)

[4.3)统一发送微信消息： 10](#_Toc502061798)

[5， 关于平台的跳转和登陆： 11](#_Toc502061799)

[5.1)平台跳转：（老GC前台） 11](#_Toc502061800)

[5.2)平台登陆：（新GC前台） 12](#_Toc502061801)

1. 关于GC新前台的说明：
   1. 测试服网址：

http://alpha.newwap.greencampus.cc

位于228服务器（测试服服务器）

代码目录：/mnt/lost+found/www/new\_wap\_greencampus

* 1. 正式服网址：

http://newwap.greencampus.cc

位于10服务器（正式服服务器）

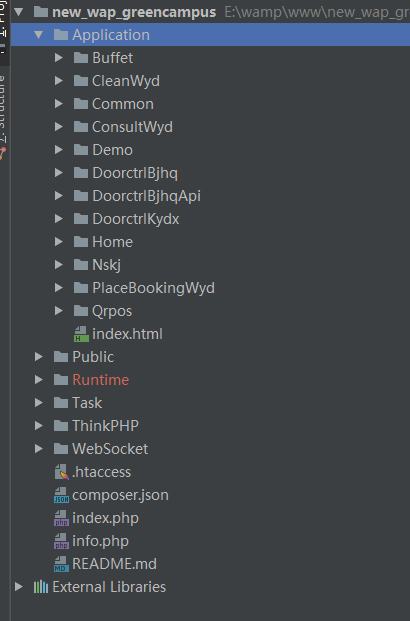
代码目录：/mnt/lost+found/www/new\_wap\_greencampus

* 1. 使用场景：

普通的管理类应用，暂时不涉及到用户的登陆和注册（后续可以完善），目前支持微信授权拿取微信用户信息，所以后续应用更倾向于基于微信类的管理应用模块开发。

* 1. 目录说明：

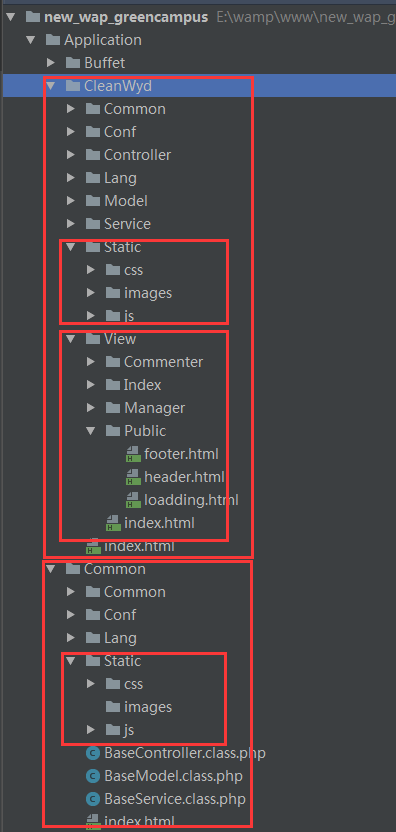
使用的是ThinkPHP 3.2框架（各个模块相互独立）



所有应用模块放在Application目录下，其中Application/Common文件夹是公共文件夹，供其他项目去继承一些基础的应用支持；Application/Demo文件夹是示例文件夹，定义了项目文件夹的规范（供创建新项目时使用）；Public文件夹是公开资源文件夹，主要包括文件上传、通用前端静态资源和框架；Runtime文件夹是缓存文件夹（框架运行自动生成）；Task文件夹是定时任务文件夹，用来编写定时任务，通过命令行去执行；ThinkPHP是框架核心文件夹，后续添加的插件可以放在ThinkPHP/Library/Vendor文件夹下；WebSocket文件夹是workman的应用文件夹。

* 1. 前端目录说明：

每个模块的静态资源和视图都是分开的：



如CleanWyd为某一模块文件夹，Common为项目公共文件夹。其中静态资源统一放在模块文件夹下的Static文件夹下；视图文件统一放在模块文件夹下的View文件夹下，具体的里面还可以再细分；在引用模块的静态资源时，使用 \_\_JS\_\_、\_\_CSS\_\_、\_\_IMG\_\_等常量定位到模块静态资源文件夹下；如要引用公共的静态资源，使用\_\_COMMON\_JS\_\_、\_\_ COMMON\_CSS\_\_、\_\_ COMMON\_IMG\_\_等常量定位到公共静态资源文件夹下。

1. 关于新模块的建立：

2.1）建立新模块：

在建立新模块时，务必复制Application文件夹下的Demo文件夹，然后重命名文件夹变成所需的项目文件夹。打开Controller/IndexController.class.php文件，修改命名空间：



把namespace Demo\Controller改改自己项目的命名空间即可（尤为重要，很容易忘记，导致后面网页访问不了）。

2.2）建立说明：

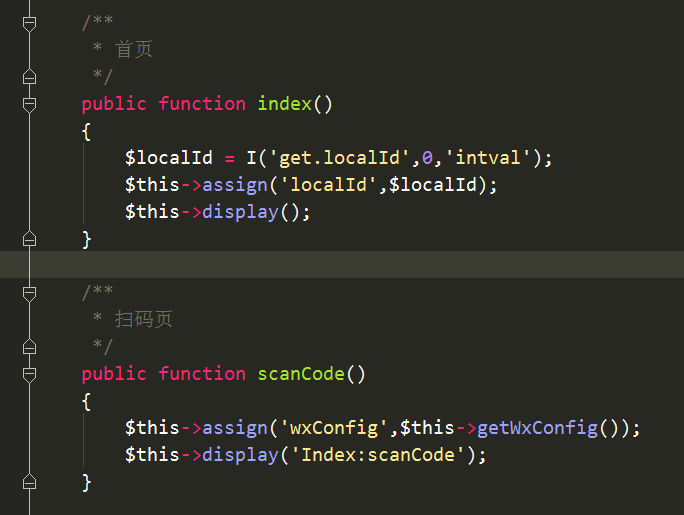
通过复制文件夹建立的项目，公用配置会统一加载，同时也保留本模块的单独配置功能，公用前端静态资源和视图也一并建立；所以强烈建议不要自己单独去新建项目文件夹。

1. 关于Controller层的编写：

目前Controller可分为两大类：视图类和接口类，统一继承BaseController。

3.1）视图类：

只做简单的视图展示及一些必要参数的赋值，如涉及到复杂的业务逻辑，统一放入到Service层：



3.2）接口类：

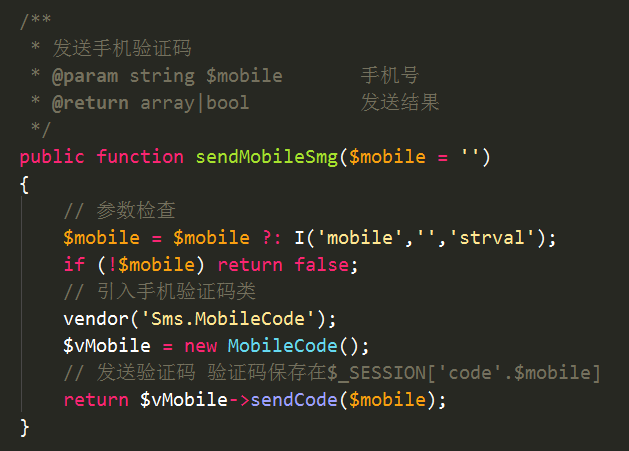
一般为HTTP POST请求，统一用： 模块名+ApiController.class.php命名，主要用来提供接口给外部调用，可以加载Service文件夹下的SelfApiService类（需要自己根据业务需要去建立），里面定义了接口的验证、返回及错误捕获等常用功能。



接口一般分三步，第一步参数的检查，即判断参数是否合理；第二步查询获取数据，根据业务去获取数据，但不做复杂的数据查询（放到Service去执行）；第三步接口返回，把结果返回给调用者。

## 3.3）统一验证码发送：

继承BaseController后，可直接调用sendMobileSmg()方法发送手机验证码：



## 3.4）统一图片上传：（阿里OSS）

继承BaseController后，可以基于微信直接调用阿里OSS服务进行图片上传：



## 3.5）统一获取微信配置：（微信JSSDK）

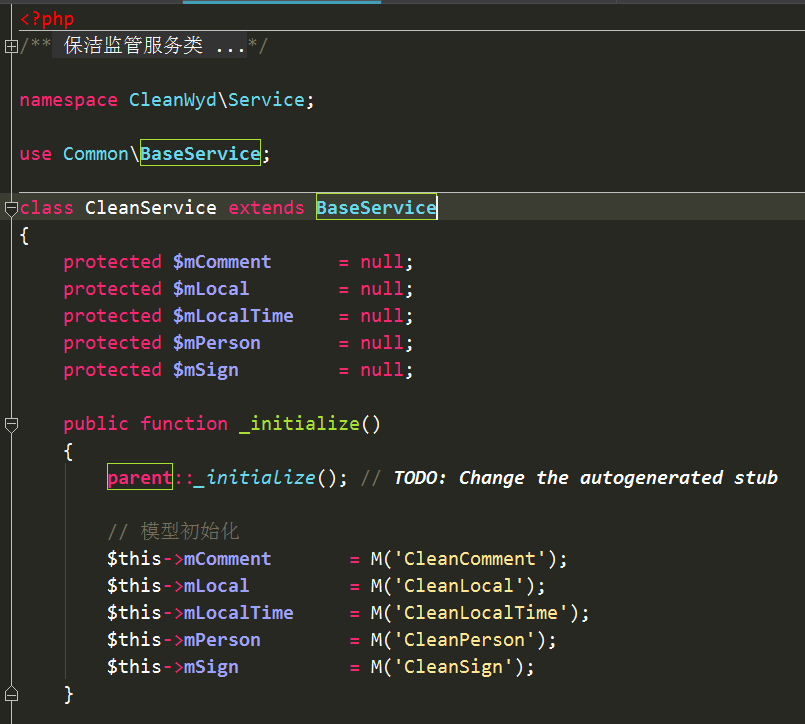
继承BaseController后，可以直接使用getWxConfig()方法获取微信配置：



1. 关于Service层的编写：

Service层一般用来处理复杂的业务逻辑，统一继承BaseService。

4.1）模型对象定义：



如果不是特别复杂的业务，可以不单独建立Model，直接调用M()方法申明即可；如果是复杂的，需要单独建立Model，然后调用D()方法申明。

强烈建议申明所有该模块需要用到的表，这样方便做统一的数据查询处理，也方便后面对模块表的维护。

4.2）统一数据查询：



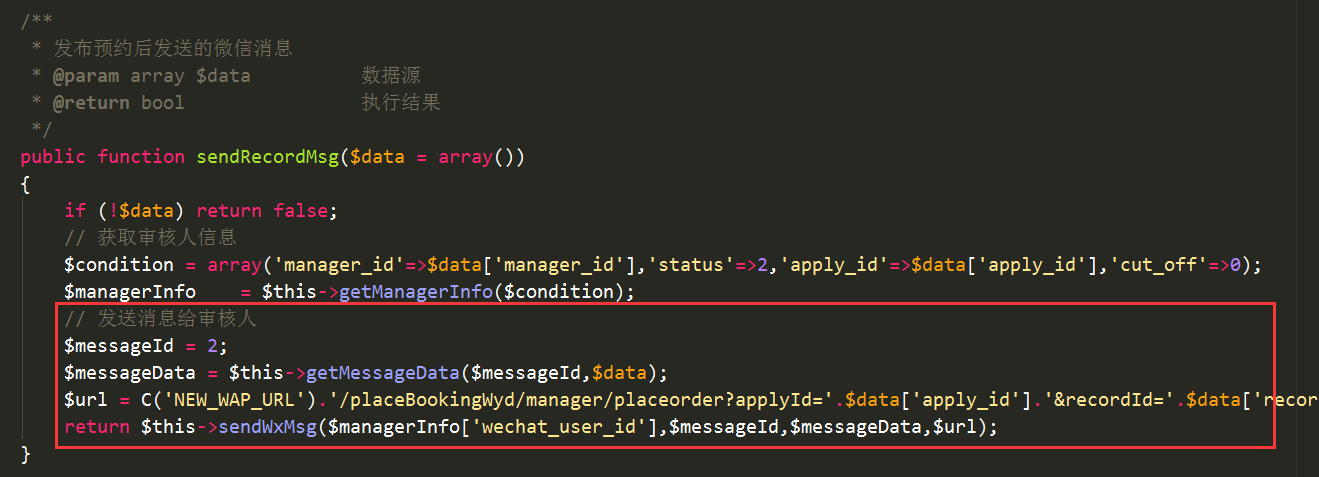
统一数据数据查询位于BaseService里，基于调用的方法名去检索对象模型（需要事先在子类中定义好），形如getXXXList、getXXXInfo的方法，会自动如找对应的模型，然后通过TP自带的数据查询器获取数据。

调用方式：（在Service中或引用Service的地方都可用）

获取列表：getXXXList($condition,$field,$limit,$order,$group)

获取详细信息：getXXXInfo($condition,$field)

4.3)统一发送微信消息：

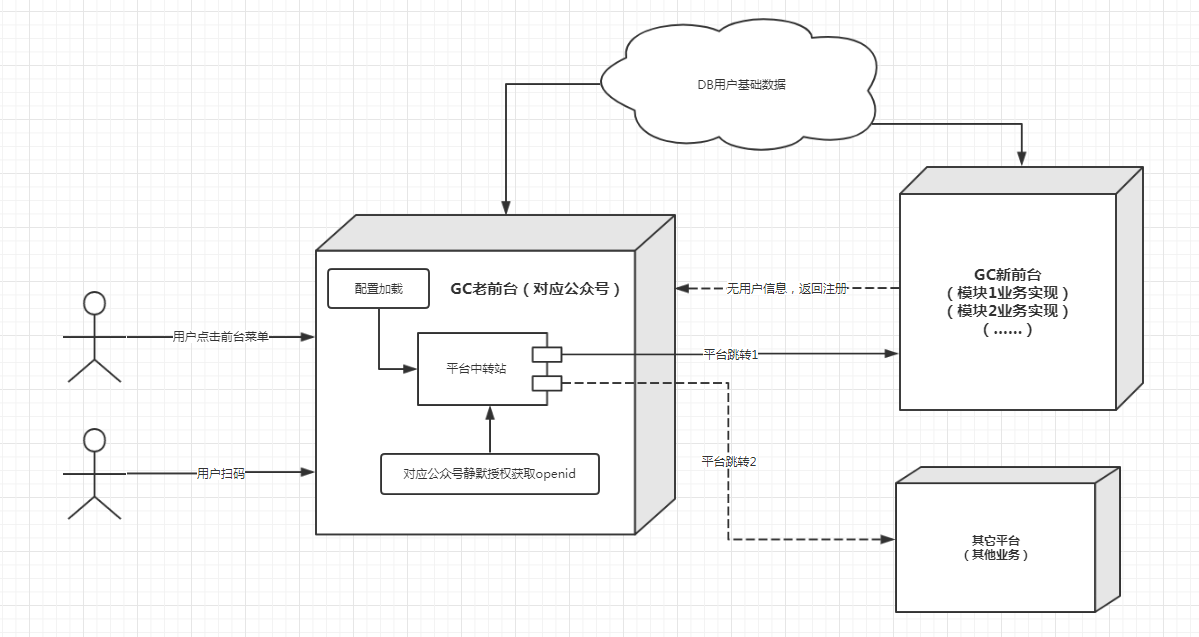


messageId指定用哪一套消息模板，messageData是需要发送的数据，url是发送消息后，点击消息的跳转链接，这三个参数定义好后就可以直接调用BaseService里面的sendWxMsg方法了，若返回status为S，代表发送成功。

1. 关于平台的跳转和登陆：

新GC前台并不直接对应公众号，所以关于授权部分还是在老GC前台中完成。

5.1)平台跳转：（老GC前台）



用户先通过点击菜单或扫二维码进入老GC前台中，如涉及到新GC前台的业务，需要先到平台中转中先进行特定的授权和跳转处理:

代码位置：BEESCRM/Lib/Action/Base/DoorTransferAction.class.php



5.2)平台登陆：（新GC前台）

从老GC前台跳转过来后，如要获取微信用户信息，需进行平台跳转验证：

代码位置：Application/Common/BaseController.class.php

