MSVC模式开发

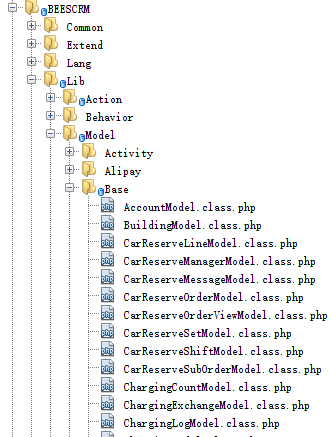
在遇到比较复杂的业务逻辑的情况下，将业务逻辑写在C层或者M层都不合适，随着业务越来越多，以后代码量太长不方便来维护，还有重要的方面是代码逻辑的重用性问题。如果在多个不同的Action来实现类似的业务逻辑，这个时候这个在这个时候就要去分层了。在介于Model层和Controll层之间再加一个Service层。加Service层的好处就是可以抽象出业务上一些公用的东西，以及完善代码的健壮性，以及完善代码的可维护性。下面分别来介绍：

1. **Model层**

Model来实现最基础的数据处理（增删查改）以及数据验证,自动完成功能看图1-0，1-1，1-2

主要实现自动完成，关联模型，表设置增操作，我们编写的Model类都去继承BaseModel

Model层的代码放在/BEESCRM/Lib/Model下



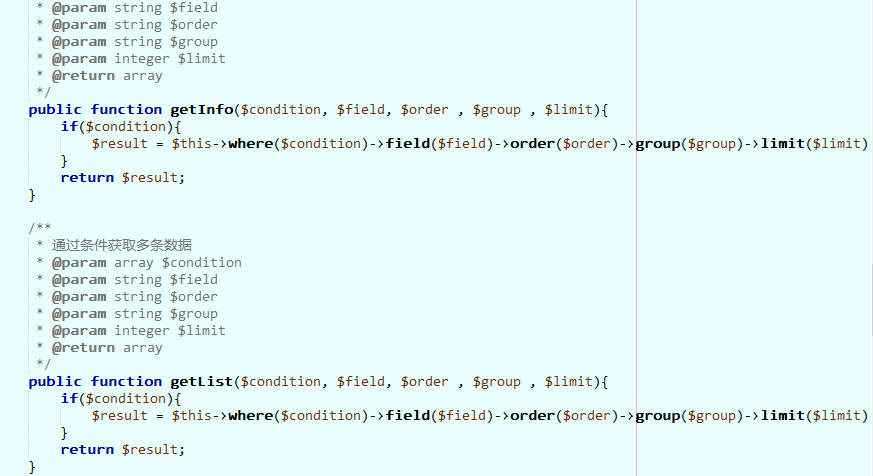
1-0

如下是一个Mode类的具体实现



1-1

这是BaseModel下的获得数据的方法，子类去继承即可

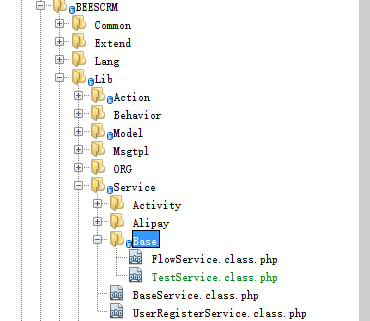


1-2

1. **Service层**

Service层是来实现具体的业务逻辑处理，其实是Model层的分支。像获得数据，操作数据等操作到Model层来调用，调用第三方接口获得数据，逻辑的处理等操作，最后返回结果给Controller层，Service层是采用面向对象的单一职责，一个方法只做一件事情。比如订单提交就是一个方法，订单获取也是一个方法。注意：Service层不能被Model层调用，只能被Controller层或者Service层调用

Service层的代码放在BEESCRM/Lib/Service下如图2-1



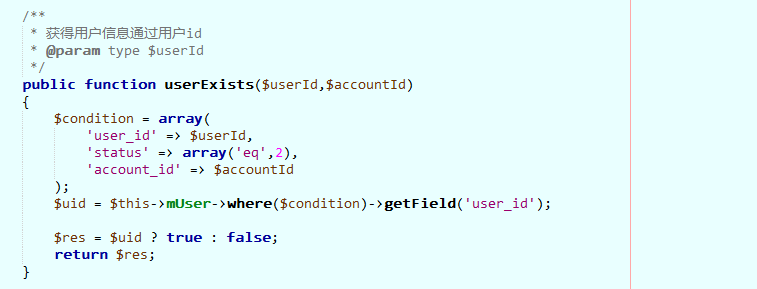
2-1

下面是一个Service类的具体实现，这个Save方法是用来实现保存用户信息，最后返回处理结果图2-2



2-2

下面是判断用户是否存在的判断如图2-3：



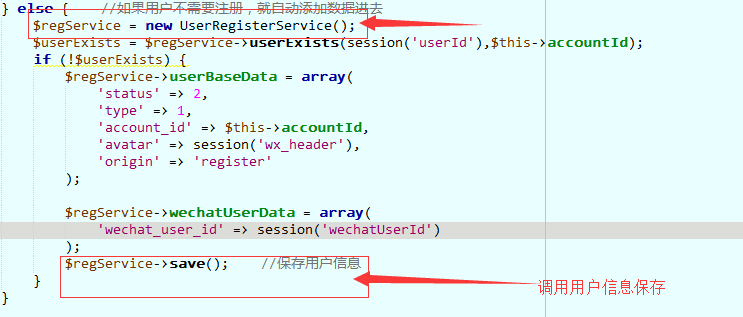
2-3

Controller层或者其他Service层在调用时可以用两种方式来调用：

第一种是针对于在同一个分组模块下的，比如Controller层也是在Base模块，Service层也是在Base模块下:

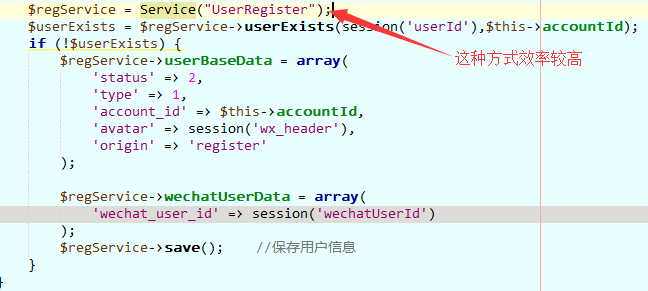
第一种：直接申明:

直接new一个对象



第二种方式：

对内存控制效率较高的方式



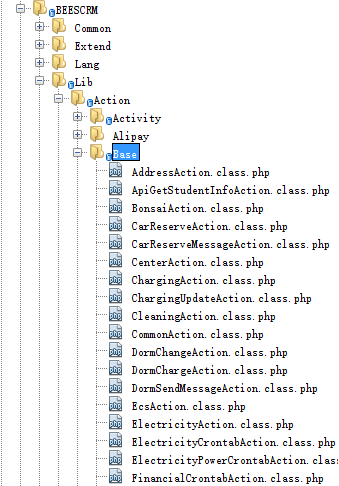
如果是跨模块调用：



1. **Controller层**

Controller来调用Service层的方法，Controller层将客户端的数据作为参数的形式，或者对象的属性传给Service层，Service来处理业务逻辑，得到处理结果反馈给Controller层。这样子Controller层就变得比较简单了，只负责客户端数据的获取传输以及数据结果的返回.

Controller层的代码目录是放在/BEESCRM/Lib/Action下，目录结构如下：



3-1

Controller层的代码示例：



1. **View层**

**View层是来渲染controller层过来的数据，代码目录结构放在/themes下:**