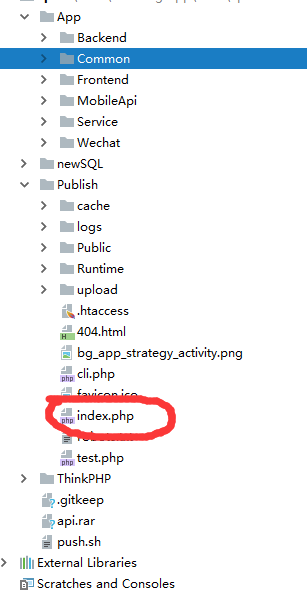
项目的整体目录结构:



其中ThinkPHP为框架的源码实现 开发者无需关心，

newSql 是新建的一个sql语句储存 只是记录下而已 无任何作用

Publish是向外开放的一个模块,HTTP的所有请求都通其中的index.PHP 进行分发处理

APP是应用的意思代表一个总的程序 其中分成以下几个模块

Backend ---- 后台管理模块

Common --- 整个APP对应的公共模块

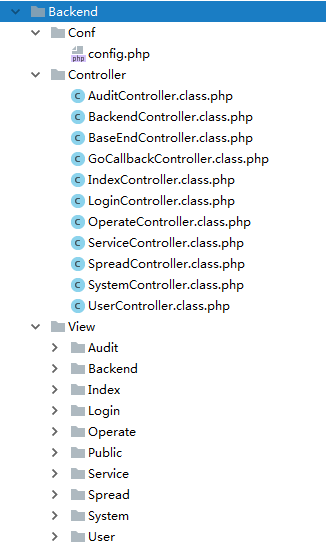
Frontend --- 网站前台模块

MobileApi --- APP的API接口模块

Service --- 为所有模块提供底层模块的 服务模块

Wechat --- 微信模块

其中Backend的目录结构如下



Conf 为对应小模块的配置文件但一般不会用

Controller 为控制器

View 为HTML文件

了解整体的运行需通过请求的URL来进行分析，在本应用中存在这两种运行的URL分别为

A类

http://47.106.21.73/Backend/Operate/chargeList

这样的URL 在index.php中将会分配到对应的Operacte控制器类中然后实例化对应的方法运行

http://47.106.21.73/Backend/

当URL残缺时 index.php将会将连接补全为http://47.106.21.73/Backend/Index/index 这是默认的对应控制器的类和方法

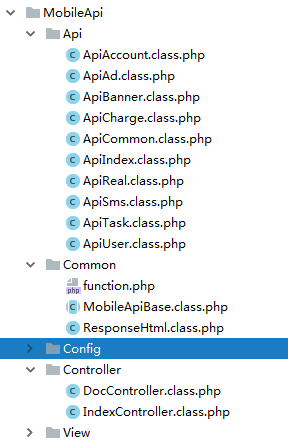
例如: http://47.106.21.73 就会默认补全http://47.106.21.73/Frontend/Index/index 其中Frontend也是默认值

B类型

http://47.106.21.73/MobileApi?service=User.register

这种类型的链接一般是APP的接口链接

看对应的MobileApi目录

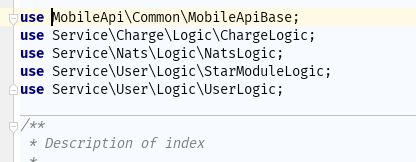


此时的API就是对应的控制器 User控制器中执行Register 方法

PHP调用类类似于java

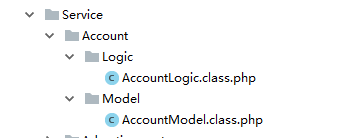
但java是以包的形式 PHP是以命名空间的形式

例如



对应命名空间可以根据对应的文件夹结构找到对应的类和方法这里就不细说了

Service文件夹的结构



大部分的业务逻辑代码会在这个目录下面

其中Loigc一般都是处理业务的 而Model一般都是和数据库相关的操作

所有大部分的情况都是在Logic里面调用Model这样，当然可以直接在Logic调用数据库

数据库的调用方式有3种

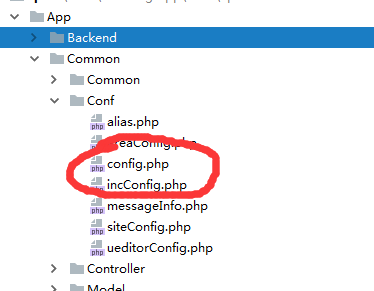
一种就是M（）方法 对应的是调用每张表

第二种是 直接实例 $a = new Model() 直接实例化数据库类

第三种是 在Model文件夹中的Model类直接继承 ThinkPHP的Model 然后就可以直接使用了类似JAVA的一个表对应着一个类这样的操作

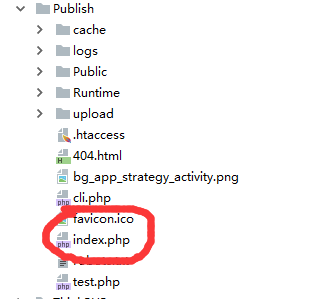
配置文件

主要有三个地方分别对应着



这个文件主要配置 数据库 和redis的

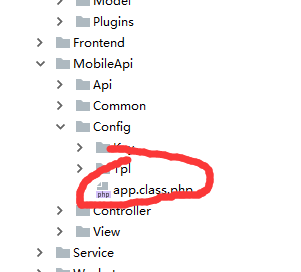
还有第二文件



这个是全局参数配置

也是应用程序的唯一主入口 里面会定义这一些路由地址 包括GRPC的地址 NAts推送的地址 上传图片后的全局域名

这个是针对APP接口另外封装的一些验证的规则



包括是否开启签名检测 是否开启接口访问的时间戳检测 是否验证版本号之类的 require

这个数组建值改为true 代表开启 false 代表关闭

应用自带接口文档

域名根据部署的情况自己替换

http://192.168.11.17/mobileApi/doc/apilist

GPRC篇

这里集成的GRPC客户端 需要在linux 上去手动生成对应的PHP客户端代码文件

protoc --proto\_path=./../protos --php\_out=./ --grpc\_out=./ --plugin=protoc-gen-grpc=./../../bins/opt/grpc\_php\_plugin ./../protos/helloworld.proto

需要在Linux上装protoc 应用, 然后执行上面的命令就好了

protoc 下载地址

https://github.com/google/protobuf/releases



下载完后，将二进制代码然后命令集合就可

下面解析下生成命令的个个参数

protoc 是命令的前缀 就像ls那样

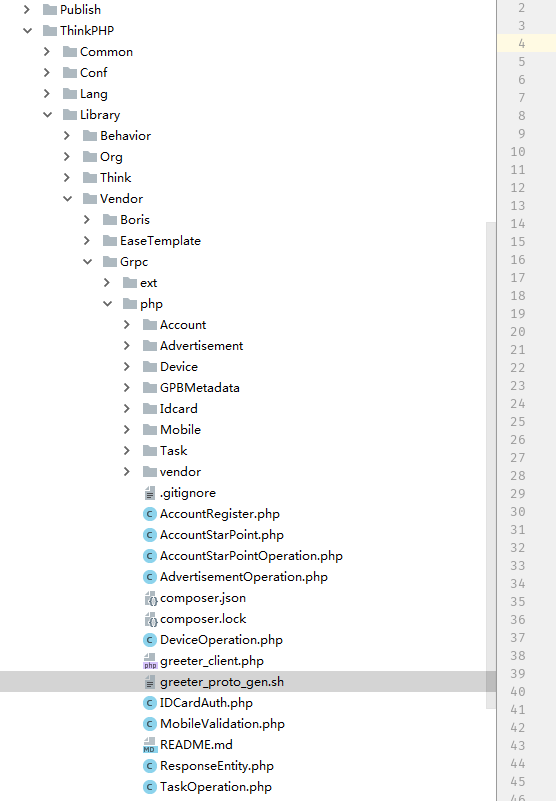
--proto\_path= 代表你protos文件放的地方

--php\_out= 代表你生成的PHP文件存放的位置

--grpc\_out= 和上面是一致的

--plugin=protoc-gen-grpc= 对应使用的客户端插件 一定要用这个不用这个生成的文件不完整

最后就结尾的 就是你想对那个protoc文件生成的命令



生成后放置改目录下 需要写对应的调用类



然后在系统中



直接引用 就可以调最近的类