Yaf 请求 响应

对应的http的请求和响应

对应的



封装的都是请求的方法和附带的属性

一些请求的路径 主机的地址

响应的时候 设置响应头 重定向

做就完成这一部分工作

说一些常用的方法和细小的差别

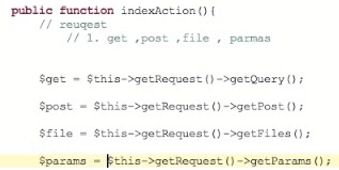
从Http Request说起



获取这个对象 拿到客户端传递的参数

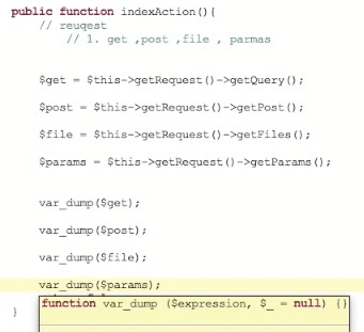
这里面request这个分成了get参数 post参数 和 file 还有 params --- params是路由中的map参数

搞清楚yaf如何接受这些数据

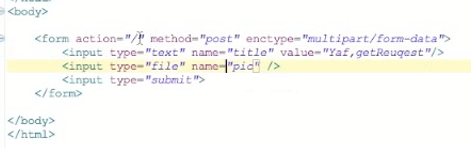


可以获取到客户端的请求数据

打印看看



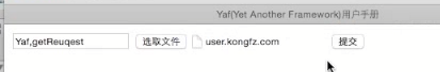




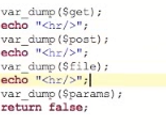


那么这个form就融合了3种类型的数据

Get post file

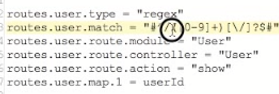


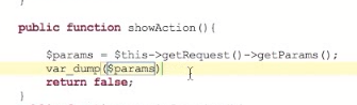


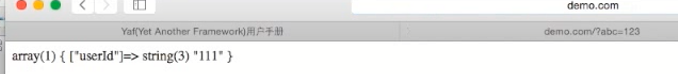




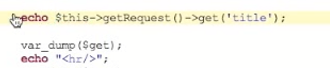




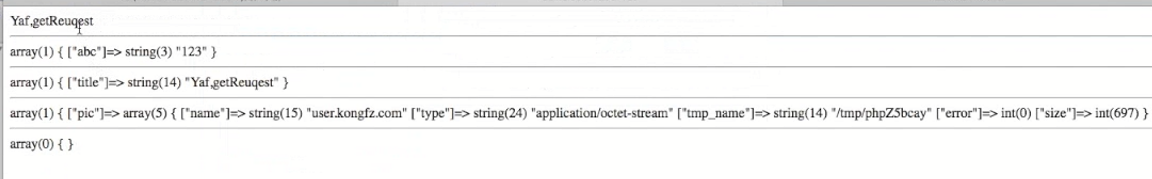




这个是路由的时候 控制器的参数











这个get()方法就是获取参数信息的



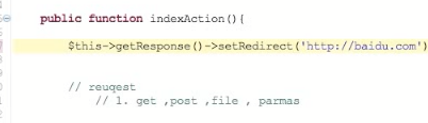
http请求有很多个方法



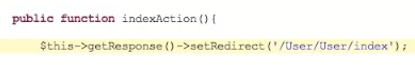
Response中的方法

访问一个控制器 做一个跳转

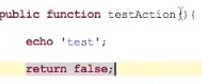
重定向



这个重定向是发301重定向 会多形成一次http请求



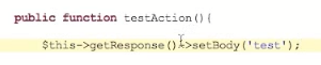
Yaf还有一个setBody和response



访问这个test

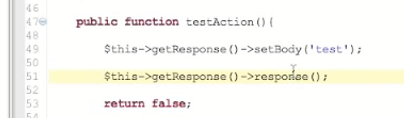


现在这里面有一个setBody



会同样得到上面的结果

这是因为是Yaf开启了自动响应

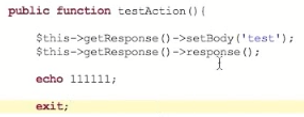
但是 这样就会显示两次



Response()输出了一次

然后 还会自动输出一次

如果不想让他自动输出 那么就把return false; 更改成exit



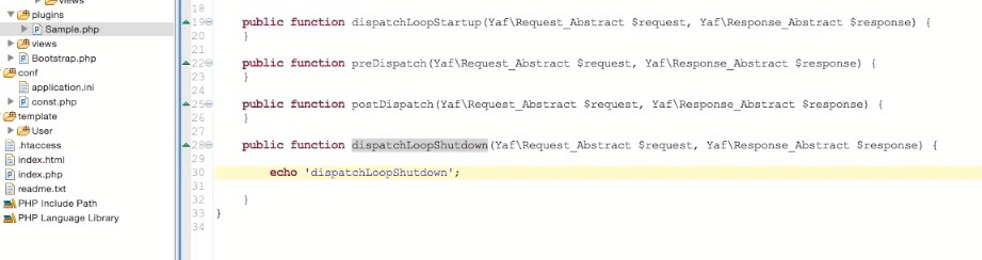


禁用自动响应

通过dispatcher进行禁用

Yaf中有一个插件机制 如果你在controller中exit

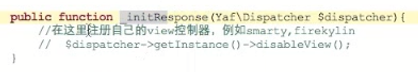
插件就会没有完成

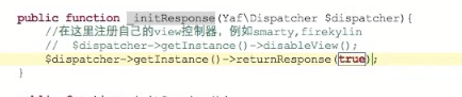
刷新没有看到任何内容

但是return false 会打印出hook中的内容



所以 注册了插件之后 就不要使用exit







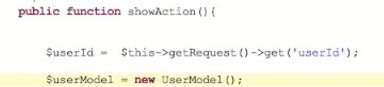
这样test同样没有输出

这个test是hook走完才输出 就是最开始的

方法都很简单

======== model

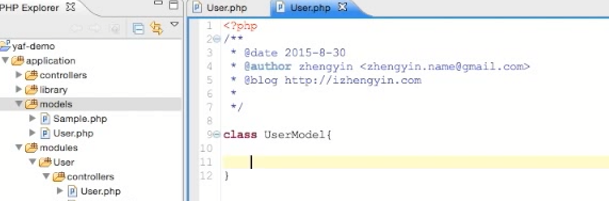
提供一个类 外部调用 完成整个模型的业务逻辑处理



Yaf中 如果你去new一个不存在的类 Yaf会自动到models下面找这个类文件

找到之后 就省略了Model这个后缀 所以 在models这个文件夹下面 直接定义User.php即可

但是定义完成的类名

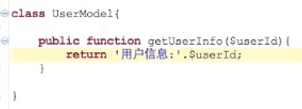
【注意到 文件名是User.php 但是里面定义的类名就是class UserModel】

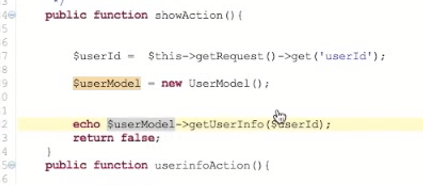
那我的数据库如何去操作？

Yaf没有提供数据库的方法 没有ORM那一套

后面课程会说

Yaf会根据载入规则 载入









使用模型就这么简单 剩下都是依靠自己实现

现在获取用户信息

用户信息放到哪里？

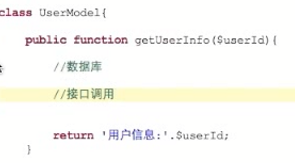
可以放到数据库中

如果是多个站点 用户信息是分布在不同的地方中

目前只提供了用户的用户名 密码 QQ 邮箱等等

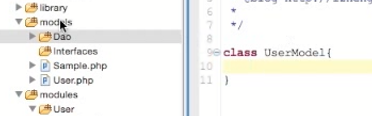
但是 还有论坛 使用的discuz 这个要对接用户 用到用户的等级

这里面涉及到接口的调用



经验：会把接口调用和数据库 抽象成一层 抽象出Dao

对其他系统会抽象出interface

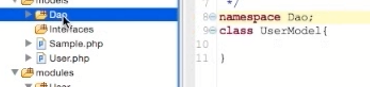


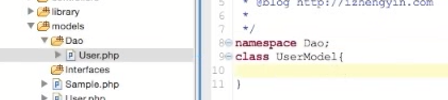
这个都放到models/Dao下面

注意 model的类名必须以Model结尾

才能找到这个类

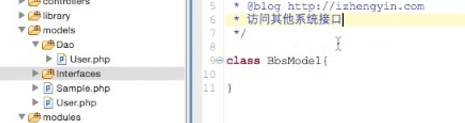
因为多了一个目录 所以要添加一个命名空间



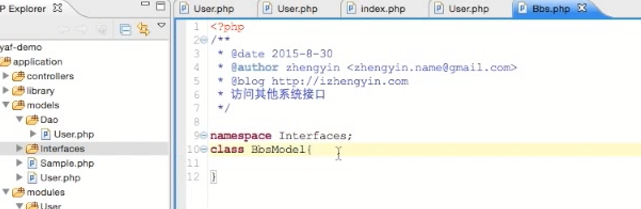


这个就是访问数据库的入口

然后在interface中定义接口

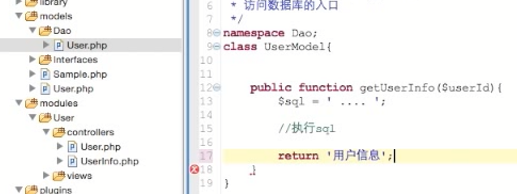


定义 命名空间

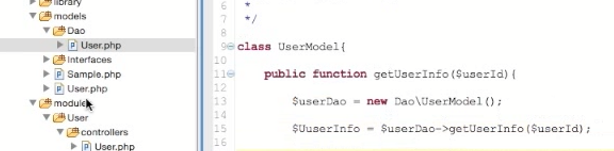


这样 在外部的User.php中





获取用户信息的时候

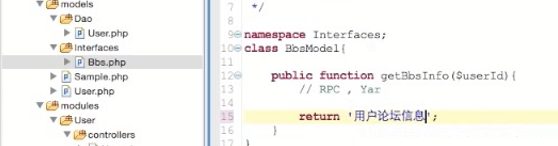


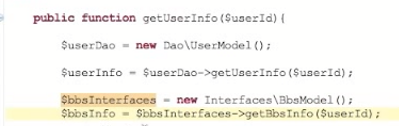
再去获取论坛的信息

因为这些接口和数据库的逻辑 百年都不会变

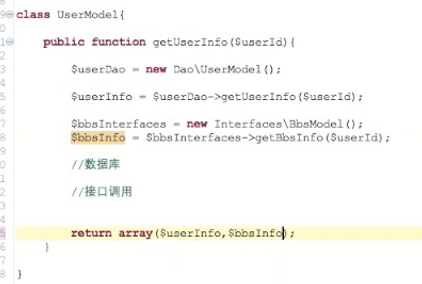
但是 对于外部的用户逻辑 可能会修改

【这里面的models相当于java ssh中的service】





返回的时候



这个直接返回到控制器





这个就是model的一个操作

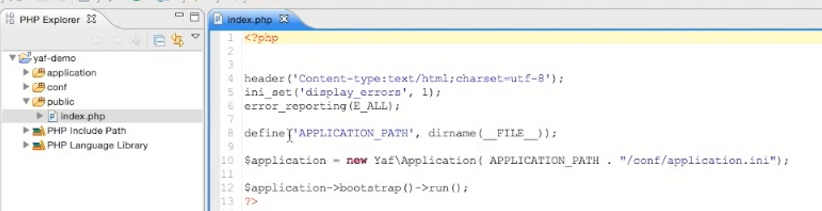
================ Yaf的工程结构 ==================

使用的Yaf的方法

用一个什么样的目录结构来管理项目

在做的时候

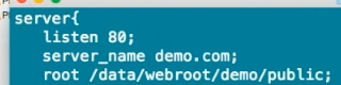
会把index.php放到public目录下



修改一下application path

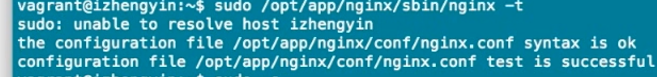


修改一下nginx

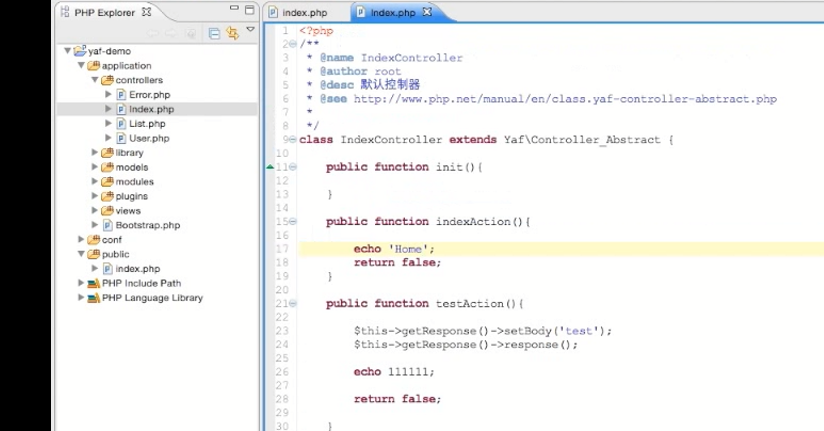


把这个root从/data/webroot/demo/修改为demo下面的子目录 /public

重启nginx

完成之后

访问项目名



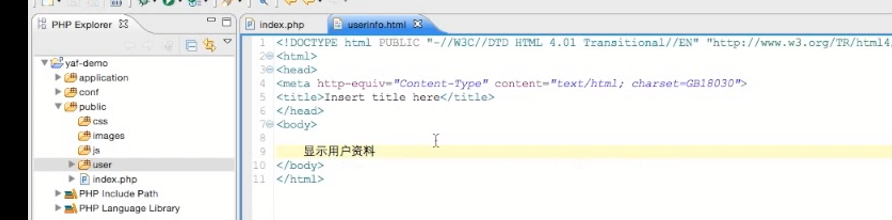
把这个index.php这个入口移到public的原因

就是网站有一些静态文件 在public中放置一些图片

Img css js

这样做一个管理

实现前后端开发



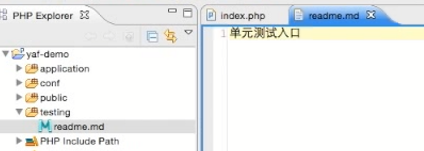
使用js框架进行开发

这个是public放置的东西 存在就是给前端使用

还有一个

 做单元测试入口

写一个readme.md

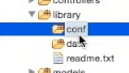


在这里面写一些文档

走到application里面

在library中放一些数据文件 比如一些关键字 或者一些和数据配合的数据文件 还有放一些conf

Conf是网站具体业务员的配置文件



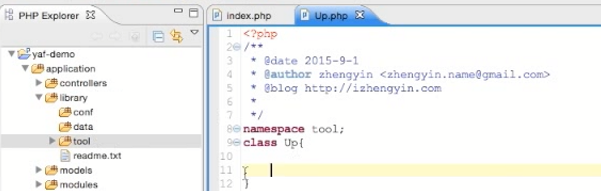
还会放置一些tool

比如分页类 图像处理类 上传类



这里面写的时候 就要根据命名空间

现在定义一个文件上传类



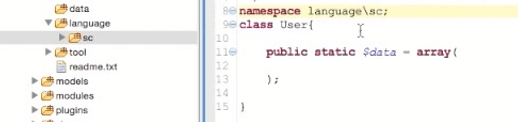
前面对这个讲过类库加载

在library还放置一个目录



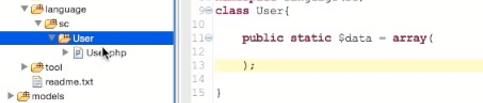
继续分

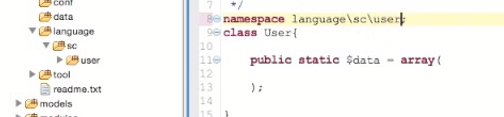
用户管理这个业务场景



这里面定义了一个语言包

通常还会分一个





这里面可能定义一些显示给前台用户的提示信息

除此之外

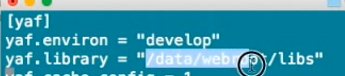
models中没有提供ORM

链接数据库操作 都会使用他

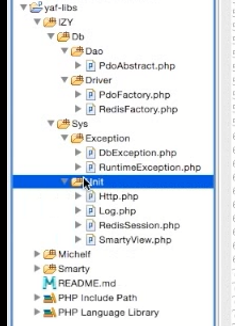
可能会使用到数据库的工厂类

可能会把这个放到第三方类库 直接使用就可以了

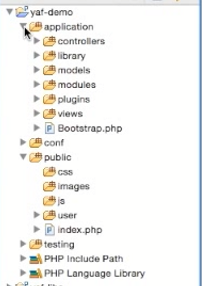
在php.ini



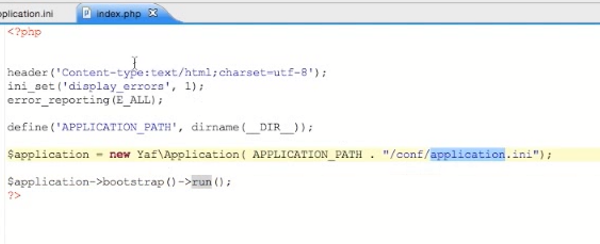
这个类库要自己实现

 ---- 这个是老师做的工具类

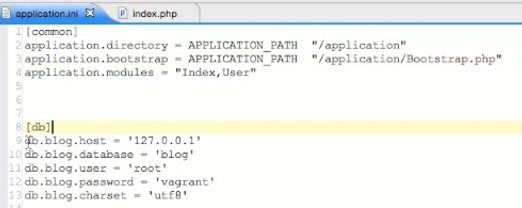
===== 统一Yaf使用过程中的开发模式 ==============



在入口文件index.php中[我们的项目中是使用www文件夹下面的index.php来载入application.ini]

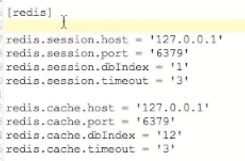


在这里面引入application.ini

  
[common] ---- 定义基础的模块

[db] ---定义的数据库

可能操作的是多个库

这个是redis的配置

这里面就是redis有很多种用途

根据后台的redis驱动 去挂钩的

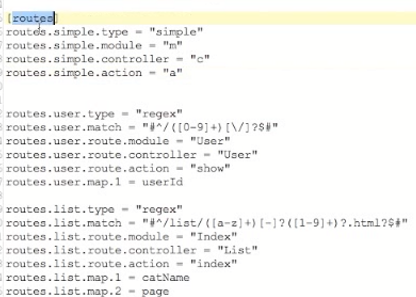


只是单站点的网站 这个没必要 否则 就很有必要

输出配置



Yaf有一个定义视图的接口





这个是本地的配置

放到线上怎么办？

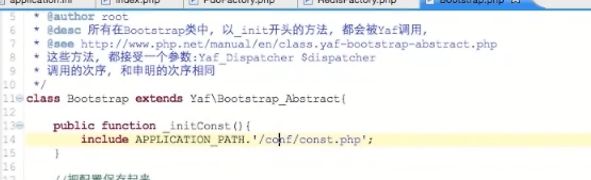
我们copy一份配置项



通过代码处理吧applicationOnline替换成application.ini

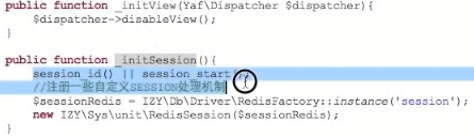
下面就是BootStrap

一般会载入一个常量

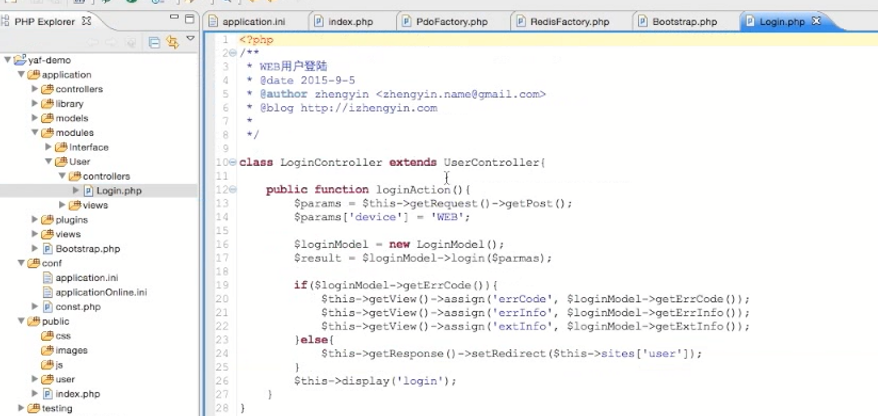




可以通过插件进行通用的验证工作



这样就可以具体看一些模块



这个是用户登录

写这个登录

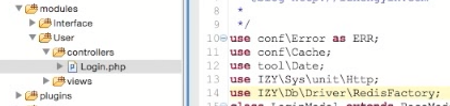
我们的网站 后台程序面对的不止是一个终端

那我们的前台youweb端

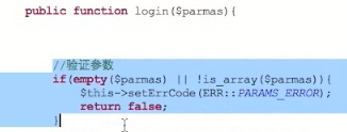
还有移动 app

你的m站和网站单独的站点 和 传统的pc站点不一样 但是 后台的登录逻辑是一样的

根据不同的终端 会有产品上的差别 但是 基本的逻辑是可以复用的



开开放给外面就是login

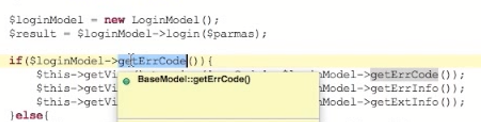


验证参数 验证完成之后

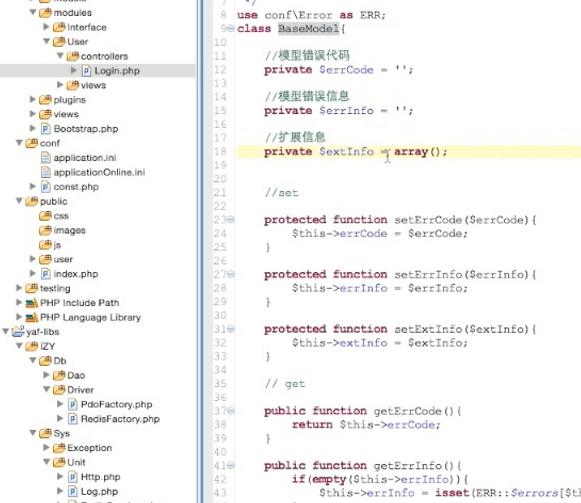
客户端拿到这个错误以后



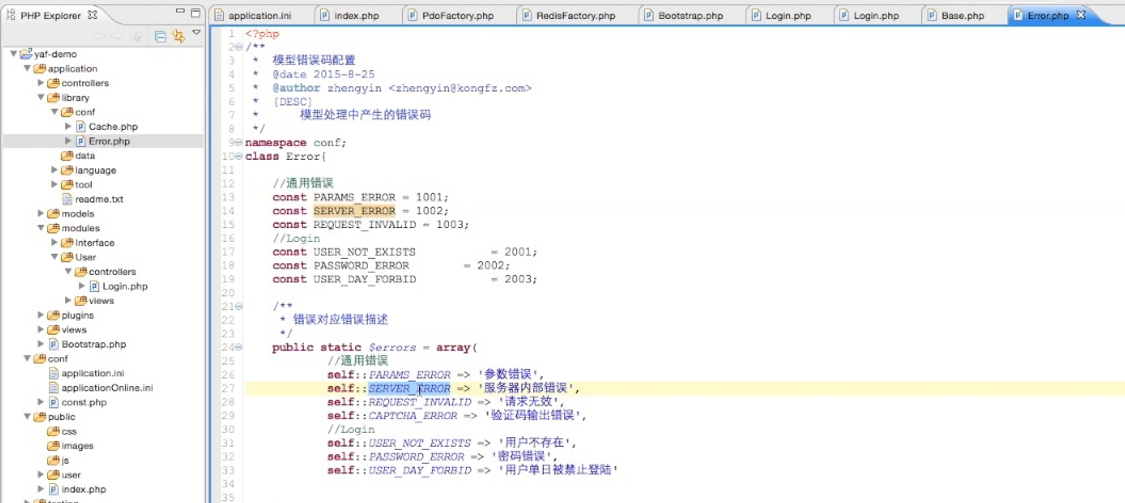
还可以使用getError是不是有错误



有一个父类

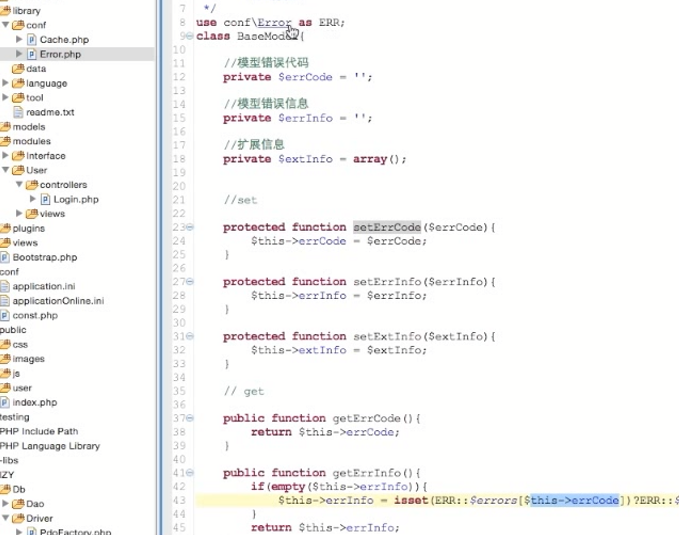


在conf中定义常量 是错误码的配置

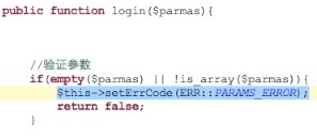


这样的话 走一个配置定义错误码

还有一个设置错误码



在这里面写的时候 就很轻松了



需要验证ip登录失败的次数

【老师讲的比较乱 这一讲没继续看】