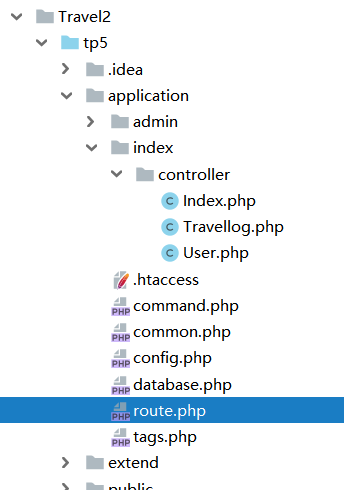
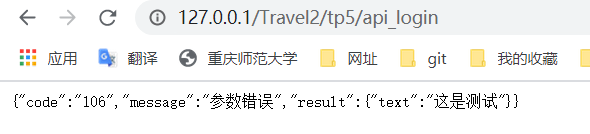
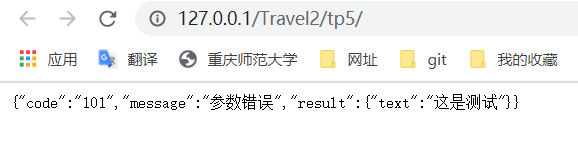
# 服务器端使用手册

1. xampp+tp5开发环境，这两个官网都有开发手册，xampp在官网下载即可，解压后，运行xampp-control.exe即可，具体配置请自己查找资料，tp5下载我的项目后，在看相关网上资料，调整文件位置之类的。
2. 上传到这一阶段的项目文档，都是关于客户端这边辅助的开发，现阶段的主要功能是登录、注册、退出登录、上传旅游日志到数据库、显示个人旅行日志、显示首页旅行日志等功能。
3. 关于路由的配置，一开始没有配置的时候，客户端访问路径很长，还容易造成数据泄露。

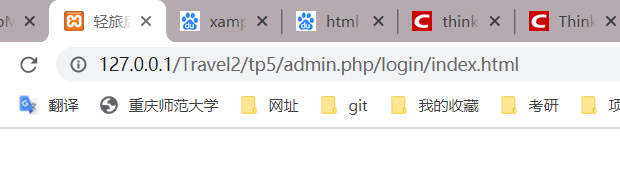
配置路由的第一不，如果入口文件index.php在public下的话，移到根目录，相关路径记得跟着一起改，第一步完成后，先去验证一下地址对不对；第二步，如图所示：



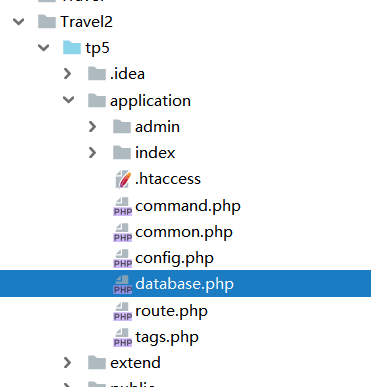
Travel2是我的项目文件夹，找到route.php文件后，分别为模块名+控制器名+方法名**admin/login/index，** 左边为你选择代替的命名，所以如图，服务器访问我的接口时很简单又不失神秘。



后台访问路由配置同上。当然，还是有一点不同的，首先复制index.php文件，重命名为admin.php，作为后台的入口文件。

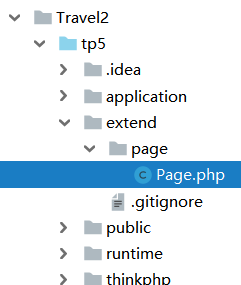
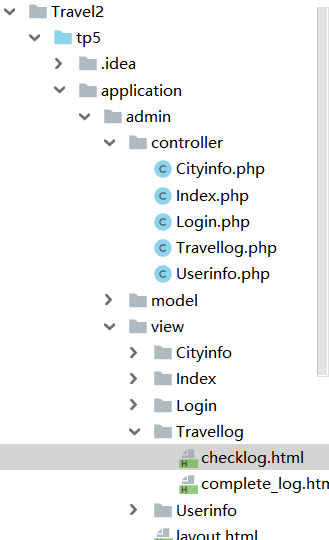


1. api接口部分的代码设计思路：首先，登录成功后，服务器端会给每个用户分配专属的token，标识独一无二的身份，如同网页的Session值一样。具体生成它的函数放在公共函数文件里common.php。其次，api接口与客户端规范了统一格式，其函数也在common.php中。而关于返回给客户端的旅行日志，则有两种处理方法，用户自己的旅行日志则是在个人页面时，仅自己可见，并且按照时间排序，最新的在最前面，而首页的旅行日志则是,排除无内容（只有背景图片+标题）日志，同样也是最新的在前面。同登录一样，退出时，客户端退还token，服务器清除。
2. 最重要的一点忘记说了，连接数据库，数据库密码就根据自己的情况来。



1. 接下来是后台部分，后台要理解MVC开发，api部分几乎没有MVC，因为，视图本身就不存在，M部分的话，可能是代码不够精简，有数据冗余的存在，造成内存不够，最后还是找不到解决的方法，只能放弃MVC模式了，但是，后台就很容易运用，姜还是老的辣。后台完成的功能有管理员登录、管理员修改昵称、管理员退出登录、管理员查看用户信息、管理员多用户进行冻结、解冻、查看旅行日志（一部分是在旅行日志模块，查看数据库所有的旅行日志，另一部分是在用户信息这边查看，一个用户所写的所有旅行日志）、对旅行日志进行状态删除（之所以是状态删除，而不是真正意义上的删除，是我认为，这是一种对数据的保护）等功能。
2. 贯穿整个后台，最重要的可能就算分页功能了，分页的库文件存放位置如下，其中用户也很简单，但也有局限。举个项目中的例子，如图中代码，



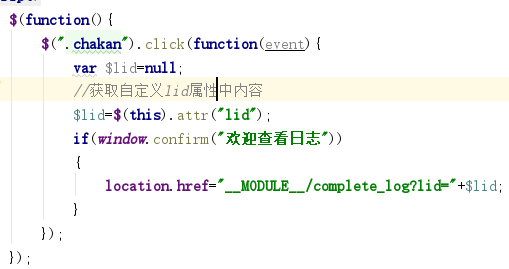


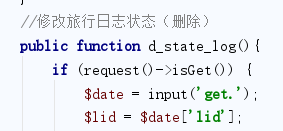


遍历后，记得加{$page},就能达到分页的效果。

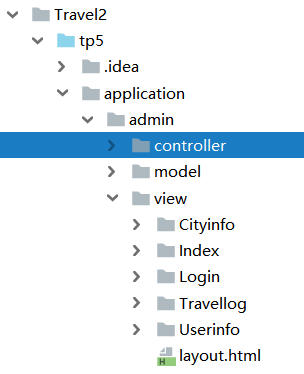
1. MVC模式中，控制器名和视图的文件名一样，方法名和HTML名一样。
2. 而在修改、查看用户信息类似的功能，用到了Ajax技术，button提交时，.d\_state类被触发，Ajax获取信息后，再提交到控制器，控制器接收参数。







10、layout布局，首先，要config.php模板设置开启模式，在视图根文件夹下放置布局HTML文件，配置公共路径。在layout.html文件中，配置好页面的基本风格和导航，在可以改动的地方放置{\_\_CONTENT\_\_}参数，这样视图中所有的页面默认会被layout渲染，{\_\_NOLAYOUT\_\_} *<!--特殊页面不需要布局-->*像登录页面就不需要同意布局，则不需要，放置这个参数即可。



11、数据库表在public文件夹下。

总结：这些都是开发时踩过的坑，有些可能因为记忆问题，会有错误，或者不完善，下次争取完善它。