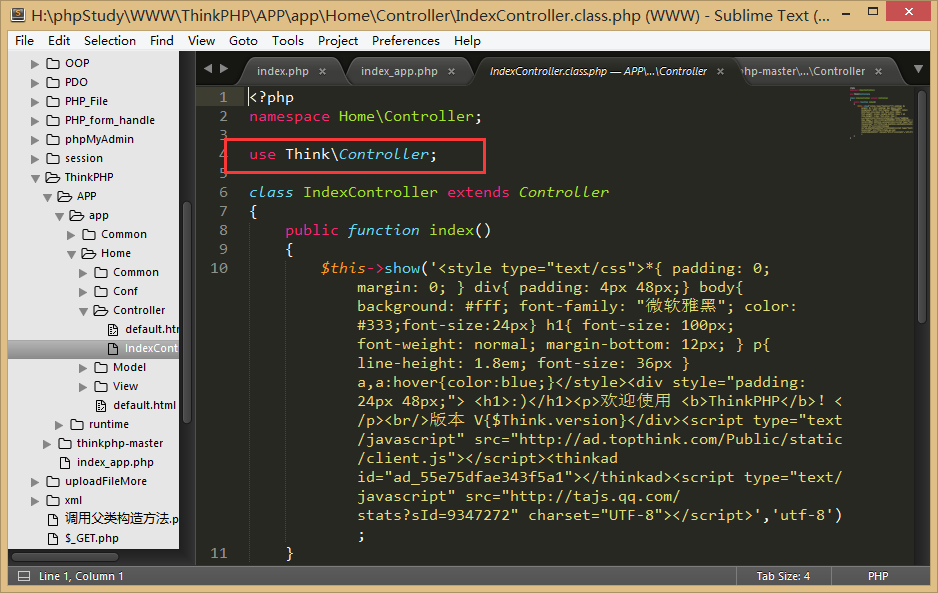
函数是独立存在的，引入后即可调用；

方法是依赖类存在的，需要实例化后调取类方法。

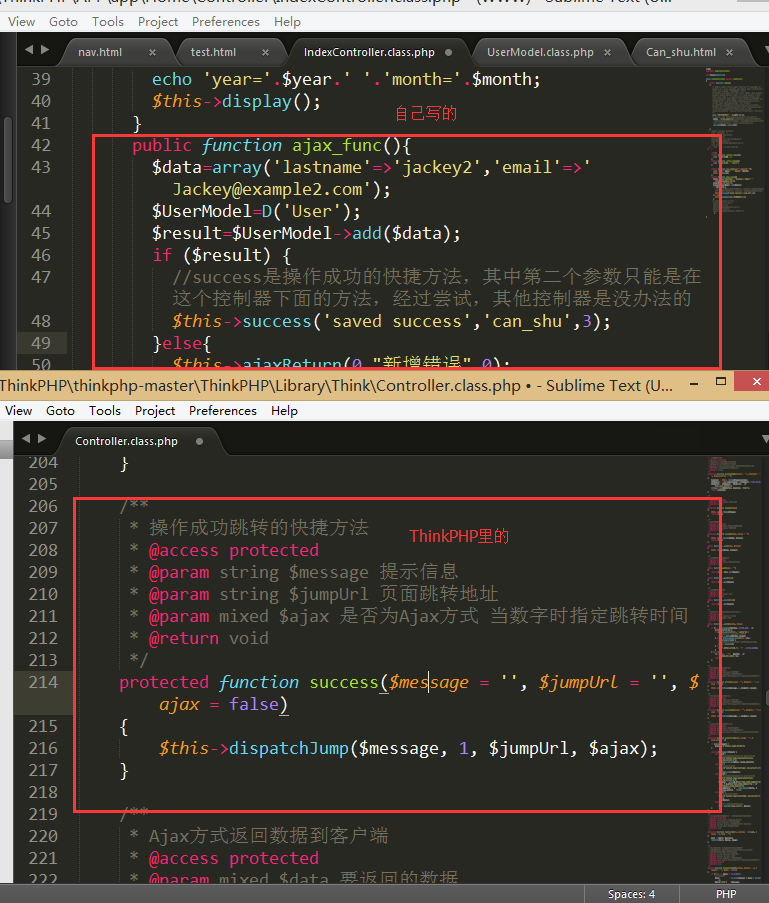
1. TP 中在定义’APP\_PATH’,’RUNTIME\_PATH’、等都是应该在后面带个 ‘/’ ,然后TP中的IndexControler 中的 use是引用类：



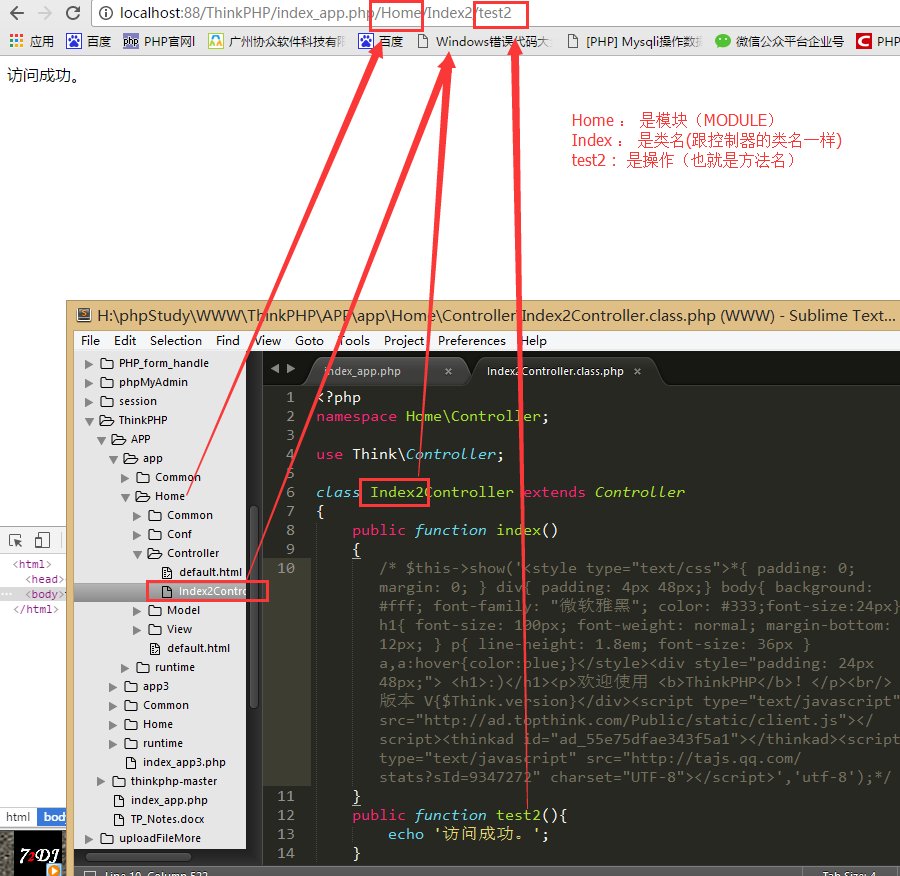
1. 注意C(‘参数的名称’,’新的值’)方法中修改参数的值是”只对当前的请求有影响，对以后是没有影响的”;

3、URL模式有四种，下面这种是第2种: pathinfo , 系统默认的是第二种的URL模式访问的，并会忽略后面的，也就是到 index\_app.php 这里结束了

1. 这里要特别的留意tp里面的ajax操作成功后的跳转的第二个参数，只能具体到本控制器下的方法
2. create()方法的创建一般是基于post回来的数据，然后进行一些防SQL语句的注入，然后再通过add()方法来加入到数据库中，如果在create()方法的时候里面添加了参数，那add()方法不要在添加参数了，不然会覆盖create()方法里面的参数的



5



6、如果你在define(\_\_NAMESPACE\_\_.'\name',value)的时候使用了\_\_NAMESPACE\_\_,那就等于全局默认使用了首选 name,如下面的值为Jackey 的MESSAGE ，而不会用222 的MESSAGE，如果要用当前的，那就会增

加一个根号

namespace test;

define(\_\_NAMESPACE\_\_.'\MESSAGE', 'Jackey');

define('MESSAGE', '222');

echo MESSAGE; \\Jackey

echo \MESSAGE.PHP\_EOL; \\222

echo \test\MESSAGE; [\\Jackey](file:///\\\\Jackey)

1. TP中查询数据库的方法有 select(),find(),getField()，初始化缓存的时候有 S($name,$value,$time,array())方法，来初始化缓存和读取缓存，如果不要缓存的话，用

S($name,null) 或者 unset(S($name))); 保存缓存的方法有 cache($key,$expire),其他的可以默认，然后默认缓存是保存在 runtime 的Temp目录下。

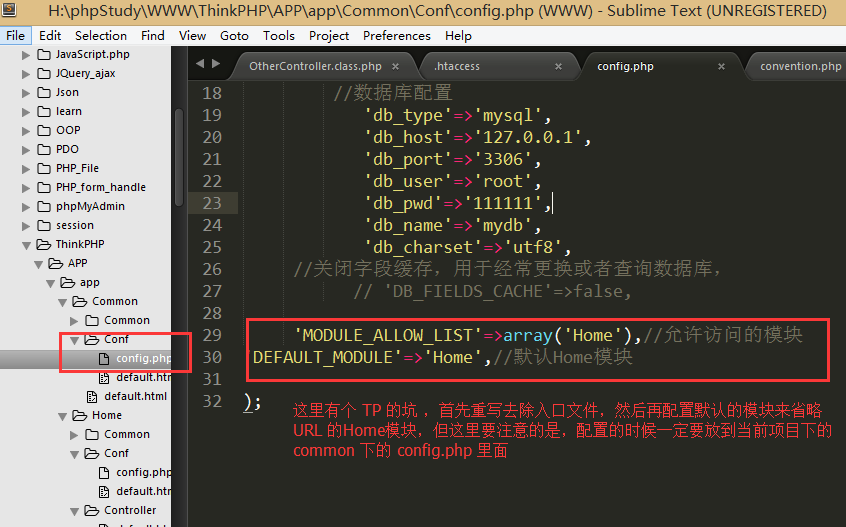
1. I()方法是获取GET、POST方法传回来的值，I方法有自带的’htmlspecialchars’ 来进行过滤，例子：
2. I(‘post.username’) 等同于 $\_POST[‘username’];
3. I(‘post.username’,’’,’strip\_tag’)//表示如果post.username没有值则返回 空字符，并进行 ‘strip\_tag’ 过滤

9、表单安全：

1、所有从前台获取的数据都要经过create()方法来创建，他会自动进行过滤，默认的是 htmlspecialchars 进行过滤 ;  
2、对表单字段的检测，可以通过 insertFields 和 updateFields 来进行对 insert 数据 和 update 数据的时候进行 数据表的字段的验证,（记得是数据库的实际，实际字段）；  
3、表单的令牌是防止重复提交数据，可以进行配置，实际上我也不怎么了解；  
4、防止SQL注入的好方法是采用where()方法里面的字段绑定，然后数据尽量用数组，如：$Model->where("id=%d and username='%s' and xx='%f'",array($id,$username,$xx))->select();  
5、目录安全：可以在入口文件用 define('DIR\_SECURE\_FILENAME','default.html');  
6、保护文件的意思是伪静态网址，写在 .htaccess 文件上;  
7、文件上传的安全是要对文件的类型，后缀，大小进行验证;

8、防止XSS的攻击是利用Cookie来获取客户信息，进而攻击，但TP里面有自带移除 XSS 的；

10、TP重写URL 的一个坑：



11、在TP部署的时候，可以利用的入口文件进行 模块(比如：Home)，控制器(比如：Admin)，操作方法(比如：Index,test)，内容如下：

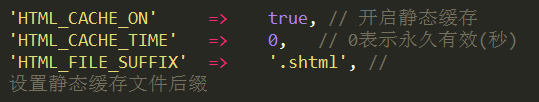
Define(‘BIND\_MODULE’,’Admin’);

define(‘BIND\_CONTROLLER’,’User’);

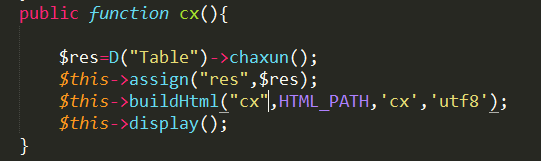
define(‘BIND\_ACTION’,’test’);

然后再利用重写规则，在 .htaccess 文件配置，然后就可以直接更简单的省略URL的访问了

12、静态缓存：(有个要留意的地方是:比如在index方法里写了$this->buildHtml()来访问cx()方法，结果只是显示了cx.html里面的html标签，涉及到标量的都不会显示，但如果在本方法里面缓存查询出来的结果就不会。)首先我们在入口文件那里：define(“HTML\_PATH”, ”./HTML/”); 然后我们可以在config.php配置文件那里进行后缀、有效期的设置：



然后进行控制器那里写 $this->buildHtml(“缓存文件名”,HTML\_PATH,”控制器:方法”,”utf8”);



13、(第三方类库引入)在Library目录下的Vendor下创建一个目录test，然后不需要用namespace ，然后在引入这类文件的时候直接使用import(“Vendor.test.Index”),并注意带根号

