//\*\*\*\*\*测试模板消息

public function commithf($pid,$nickname,$content){

//任务完成通知 start

$commit = Db::table('commit')->where('id','eq',$pid)->find();

$formid = $commit['formid'];

$user\_id = $commit['user\_id'];

$recruitmeid = $commit['recruitmeid'];

$merchantsid = $commit['merchantsid'];

$user = User::where('id','eq',$user\_id)->find();

$openid = $user['openid'];

$addtime = date('Y-m-d H:i:s',time()); //当前时间

$template\_id = 'OjhExCpuLsOFdVDXHtiSfRlR4vryMn5bx5hJs9kx5vY'; //模板id

$time=date('Y-m-d H:i:s',time());

$datainfo = [

'keyword1'=>[

'value'=>"xxx提示",

'color'=>'#000000'

],

'keyword2'=>[

'value'=>"'".$nickname."'",

'color'=>'#173177'

],

'keyword3'=>[

'value'=>"“".$content."”",

'color'=>'#173177'

]

,

'keyword4'=>[

'value'=>$time,

'color'=>'#173177'

],

];

$page = 'pages/recruitment/seemore/seemore?recruitmeid=1';

$res = $this->send\_template($openid,$template\_id,$formid,$datainfo,$user\_id,$page);

//任务完成通知end

}

//发送模板消息

/\*\*

\*@param $applet\_id 小程序ID

\*@param $user\_id 用户ID

\*@param $form\_id 表单ID

\*@param $data 消息模板

\*/

public function send\_template($openid,$template\_id,$formid,$datainfo,$user\_id,$page){

//form\_id 表单提交场景下，为submit事件带上的formId；支付场景下，为本次支付的prepay\_id

//获取token值

$config = Db::table('wx\_config')->where('id','eq','1')->find();

$url="https://api.weixin.qq.com/cgi-bin/token?grant\_type=client\_credential&appid=".$config['x\_appid']."&secret=".$config['x\_appsecret'];

$result=$this->curl\_get($url);

$access\_token=$result['access\_token'];

//拼装URL

$url="https://api.weixin.qq.com/cgi-bin/message/wxopen/template/send?access\_token=".$access\_token;

//拼装模板消息

$template\_data=[

'touser'=>$openid,

'template\_id'=>$template\_id,

'page'=>$page,

'form\_id'=>$formid,

'data'=>$datainfo,

'emphasis\_keyword'=>'',

];

$template\_data=json\_encode($template\_data);

$result = $this->curl\_post\_https($url,$template\_data);

return $result;

}

//get请求

public function curl\_get($url){

$curl = curl\_init();

curl\_setopt($curl, CURLOPT\_URL, $url);

curl\_setopt($curl, CURLOPT\_HEADER, 0);

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, false);//这个是重点。

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYHOST, FALSE);

$data = curl\_exec($curl);

curl\_close($curl);

$data=json\_decode($data,true);

return $data;

}

/\* PHP CURL HTTPS POST \*/

public function curl\_post\_https($url,$data){ // 模拟提交数据函数

$curl = curl\_init(); // 启动一个CURL会话

curl\_setopt($curl, CURLOPT\_URL, $url); // 要访问的地址

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, 0); // 对认证证书来源的检查

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYHOST, 2); // 从证书中检查SSL加密算法是否存在

curl\_setopt($curl, CURLOPT\_USERAGENT, $\_SERVER['HTTP\_USER\_AGENT']); // 模拟用户使用的浏览器

curl\_setopt($curl, CURLOPT\_FOLLOWLOCATION, 1); // 使用自动跳转

curl\_setopt($curl, CURLOPT\_AUTOREFERER, 1); // 自动设置Referer

curl\_setopt($curl, CURLOPT\_POST, 1); // 发送一个常规的Post请求

curl\_setopt($curl, CURLOPT\_POSTFIELDS, $data); // Post提交的数据包

curl\_setopt($curl, CURLOPT\_TIMEOUT, 30); // 设置超时限制防止死循环

curl\_setopt($curl, CURLOPT\_HEADER, 0); // 显示返回的Header区域内容

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1); // 获取的信息以文件流的形式返回

$tmpInfo = curl\_exec($curl); // 执行操作

if (curl\_errno($curl)) {

echo 'Errno'.curl\_error($curl);//捕抓异常

}

curl\_close($curl); // 关闭CURL会话

return $tmpInfo; // 返回数据，json格式

}