

## 目录

目录	1
简介	14
传统桥梁	14
CRMChat的好处	14
应用场景	14
接入方式	14
开源交流群（群人数已满200+，请添加客服企业微信，邀请进群）	14
安装	15
PHP设置	15
步骤总结	15
步骤详解：	15
站点配置	19
步骤总结	19
步骤详解	19
一. 点击宝塔左侧菜单网站,点击项目站点名称	19
二. 进入 网站目录 ,设置运目录为public,然后点击 保存	19
三. 进入 PHP版本 ,选择纯静态,点击 切换	19
四. 需要配置https的请查看,不需要可跳过;	20
五. 配置反向代理	20
六. 进入下一步,启动服务	25
运行服务	26
步骤总结	26
步骤详解	26
一. 检测命令行PHP版本,如果不再7.1~7.4之间请按照以下教程切换;	26
二. 启动命令	26
三. 运行完成,下一步进入程序一键安装	27
程序安装	28
安装步骤	28
步骤详解	28
一.浏览器新建窗口,输入您的域名,确认,系统会跳转自动进入安装界面,点击接受进入下一步	28
二.检测环境是否符合系统要求,确认无误后,点击下一步.如有X号等错误提示,请根据系统要求修改服务器环境配置后重启swoole,刷新当前页面	29
三.填写数据库用户名、数据库名、数据库密码	28
四.填写Redis数据库密码	30
五. 填写管理员密码后,点击创建数据.程序开始自动安装	31
六.等待程序安装,安装完成后建议删除install目录下所有文件除了install.lock文件不能删除!	31
七.安装程序完成.执行:sudo -u www php think swoole restart执行完后,就可以访问后台和前端了	32
使用文档	33
新手入门	33
1、登录后台	33
2、添加客服	33
3、登录客服	34
用户端访问体验咨询吧	38
站点引用	40
第一步 获取js代码	40
第二步 请进入自己客服系统后台复制代码,添加到您网址页面内	40
灵活的超链接使用	43
获取超链接	43
使用场景	43
超链接参数说明	43

深入对接客服	44
第一步引入js	44
调用客服功能	44
APP打包	46
APP打包和调试	46
配置	46
一、配置APP应用ID	46
二、配置APP图标	46
三、APP模块配置	46
四、APP权限配置	46
调试	47
一、浏览器调试	47
打包	49
一、申请打包证书	49
二、安卓打包	49
三、IOS打包	50
四、上传IOS应用，实际上ios打包和安卓没有太大区别，只要证书申请下来，打包也不是什么问题，至于打包后怎么上传IOS应用，请看以下说明	51
五、IOS应用发布	53
消息推送	55
APP消息推送	55
流程	55
开始配置	55
一、开通uniPush	55
二、配置uniPush	56
三、安卓厂商推送	57
四、其他厂商推送开通指南	62
UNIAPP调试	63
UNIAPP调试	63
配置	63
H5调试	63
真机调试	63
功能介绍	64
接口文档	64
chat	64
总后台/公用接口(无需授权)	64
GET 获取验证码	64
返回结果	64
返回数据结构	64
POST 登录	64
请求参数	65
返回结果	65
返回数据结构	65
GET 获取登录信息	65
返回结果	65
返回数据结构	65
总后台/统计	65
GET 统计	65
请求参数	65
返回结果	66
返回数据结构	66

GET 客户首页统计	66
请求参数	66
返回结果	66
返回数据结构	66
总后台/客服管理/客服管理	66
GET 客服登录	66
请求参数	66
返回结果	66
返回数据结构	66
GET 修改客服表单	66
请求参数	66
返回结果	67
返回数据结构	67
GET 获取添加客服表单	67
请求参数	67
返回结果	67
返回数据结构	67
DELETE 删除客服	67
请求参数	67
返回结果	67
返回数据结构	67
PUT 修改客服	67
请求参数	68
返回结果	68
返回数据结构	68
GET 聊天记录	68
请求参数	68
返回结果	69
返回数据结构	69
PUT 修改客服状态	69
请求参数	69
返回结果	69
返回数据结构	69
POST 添加客服	69
请求参数	70
返回结果	70
返回数据结构	70
GET 客服列表	70
请求参数	70
返回结果	71
返回数据结构	71
GET 查看对话	71
请求参数	71
返回结果	71
返回数据结构	71
总后台/客服管理/反馈管理	71
GET 获取修改用户反馈	71
请求参数	71
返回结果	71
返回数据结构	71
GET 反馈列表	71

请求参数	71
返回结果	72
返回数据结构	72
DELETE 删除用户反馈	72
请求参数	72
返回结果	72
返回数据结构	72
PUT 修改用户反馈	72
请求参数	72
返回结果	73
返回数据结构	73
总后台/客服管理/话术资源	73
GET 获取话术详情	73
请求参数	73
返回结果	73
返回数据结构	73
GET 话术列表	73
请求参数	73
返回结果	73
返回数据结构	73
POST 保存话术 (创建话术)	73
请求参数	74
返回结果	74
返回数据结构	74
DELETE 删除话术	74
请求参数	74
返回结果	74
返回数据结构	74
PUT 修改话术	74
请求参数	75
返回结果	75
返回数据结构	75
GET 获取修改话术	75
请求参数	75
返回结果	75
返回数据结构	76
GET 获取话术创建	76
请求参数	76
返回结果	76
返回数据结构	76
总后台/客服管理/话术分类	76
GET 获取话术创建	76
请求参数	76
返回结果	76
返回数据结构	76
GET 获取修改话术分类	76
请求参数	76
返回结果	77
返回数据结构	77
GET 取话术分类列表	77



请求参数	77
返回结果	77
返回数据结构	77
POST 保存话术分类	77
请求参数	77
返回结果	77
返回数据结构	78
PUT 修改话术分类	78
请求参数	78
返回结果	78
返回数据结构	78
DELETE 删除话术分类	78
请求参数	78
返回结果	78
返回数据结构	78
GET 获取话术分类详情	79
请求参数	79
返回结果	79
返回数据结构	79
总后台/系统管理/管理员	79
GET 退出登陆	79
请求参数	79
返回结果	79
返回数据结构	79
PUT 修改管理员状态	79
请求参数	79
返回结果	79
返回数据结构	80
总后台/系统管理/菜单管理	80
GET 获取权限菜单创建接口	80
请求参数	80
返回结果	80
返回数据结构	80
GET 获取权限菜单列表接口	80
请求参数	80
返回结果	80
返回数据结构	80
POST 保存权限菜单接口	80
请求参数	80
返回结果	80
返回数据结构	81
DELETE 删除权限菜单接口	81
请求参数	81
返回结果	81
返回数据结构	81
PUT 修改权限菜单接口	81
请求参数	81
返回结果	81
返回数据结构	81
GET 获取修改权限菜单接口	81

请求参数	81
返回结果	81
返回数据结构	82
总后台/系统管理/应用管理	82
PUT 重置应用token	82
请求参数	82
返回结果	82
返回数据结构	82
PUT 修改应用接口	82
请求参数	82
返回结果	82
返回数据结构	82
DELETE 删除应用接口	82
请求参数	82
返回结果	83
返回数据结构	83
POST 保存应用接口	83
请求参数	83
返回结果	83
返回数据结构	83
GET 获取应用列表接口	83
请求参数	83
返回结果	84
返回数据结构	84
GET 获取修改应用接口	84
请求参数	84
返回结果	84
返回数据结构	84
GET 获取应用创建接口	84
请求参数	84
返回结果	84
返回数据结构	84
总后台/附件相关	84
GET 图片附件列表	84
请求参数	84
返回结果	85
返回数据结构	85
PUT 修改图片名称	85
请求参数	85
返回结果	86
返回数据结构	86
POST 删除图片	86
请求参数	86
返回结果	86
返回数据结构	86
PUT 移动图片分类	86
请求参数	87
返回结果	87
返回数据结构	87
POST 上传图片	87

请求参数	87
返回结果	87
返回数据结构	87
总后台/附件相关/附件分类管理	88
GET 附件分类列表	88
请求参数	88
返回结果	88
返回数据结构	88
POST 保存附件分类	88
请求参数	88
返回结果	89
返回数据结构	89
PUT 修改附件分类	89
请求参数	89
返回结果	89
返回数据结构	89
DELETE 删除附件分类	89
请求参数	89
返回结果	89
返回数据结构	90
GET 获取修改附件分类	90
请求参数	90
返回结果	90
返回数据结构	90
总后台/用户管理/用户分组	90
GET 获取分组创建接口	90
请求参数	90
返回结果	90
返回数据结构	90
DELETE 删除分组接口	91
请求参数	91
返回结果	91
返回数据结构	91
PUT 修改分组接口	91
请求参数	91
返回结果	91
返回数据结构	91
POST 保存分组接口	92
请求参数	92
返回结果	92
返回数据结构	92
GET 获取分组列表接口	92
请求参数	92
返回结果	92
返回数据结构	92
GET 获取修分组签接口	93
请求参数	93
返回结果	93
返回数据结构	93
总后台/用户管理/用户标签	94

POST 保存标签接口	94
请求参数	94
返回结果	94
返回数据结构	94
GET 获取标签列表接口	94
请求参数	94
返回结果	94
返回数据结构	94
GET 获取标签创建接口	95
请求参数	95
返回结果	95
返回数据结构	95
GET 获取修改标签接口	95
请求参数	95
返回结果	95
返回数据结构	95
PUT 修改标签接口	95
请求参数	95
返回结果	96
返回数据结构	96
DELETE 删除标签接口	96
请求参数	96
返回结果	96
返回数据结构	96
总后台/用户管理/用户标签/标签分类	96
POST 保存标签分类接口	96
请求参数	96
返回结果	96
返回数据结构	97
GET 获取标签分类列表接口	97
请求参数	97
返回结果	97
返回数据结构	97
PUT 修改标签分类接口	97
请求参数	97
返回结果	97
返回数据结构	98
DELETE 删除标签分类接口	98
请求参数	98
返回结果	98
返回数据结构	98
GET 获取标签分类创建接口	98
请求参数	98
返回结果	98
返回数据结构	98
GET 获取修改标签分类接口	98
请求参数	98
返回结果	98
返回数据结构	99
总后台/用户管理/用户列表	99

GET 获取修改用户表单	99
请求参数	99
返回结果	99
返回数据结构	99
GET 获取全部标签	100
请求参数	100
返回结果	100
返回数据结构	100
GET 获取全部分组	100
请求参数	100
返回结果	101
返回数据结构	101
GET 用户列表	101
请求参数	101
返回结果	101
返回数据结构	101
PUT 批量修改用户标签	102
请求参数	102
返回结果	102
返回数据结构	103
PUT 修改用户	103
请求参数	103
返回结果	103
返回数据结构	103
PUT 批量修改用户分组	103
请求参数	104
返回结果	104
返回数据结构	104
总后台/维护管理/配置分类	104
GET 获取配置分类列表接口	104
请求参数	104
返回结果	104
返回数据结构	104
客服后台/用户标签	105
GET 获取用户信息	105
请求参数	105
返回结果	105
返回数据结构	105
GET 获取用户标签	105
请求参数	105
返回结果	106
返回数据结构	106
PUT 保存用户标签	106
请求参数	106
返回结果	106
返回数据结构	106
客服后台/公用接口	106
POST 客服登录	106
请求参数	107
返回结果	107

返回数据结构	107
POST 退出登录	107
请求参数	107
返回结果	107
返回数据结构	107
客服后台/客户列表	107
GET 获取当前客服客户列表	107
请求参数	107
返回结果	108
返回数据结构	108
PUT 设置用户标签	108
请求参数	108
返回结果	109
返回数据结构	109
GET 用户标签	109
请求参数	109
返回结果	109
返回数据结构	109
GET 和客服聊天过的用户	109
请求参数	109
返回结果	110
返回数据结构	110
客服后台/用户分组	110
PUT 修改用户分组	110
请求参数	110
返回结果	110
返回数据结构	110
GET 获取所有分组	110
请求参数	110
返回结果	110
返回数据结构	111
客服后台/个人中心	111
GET 获取当前登录客服信息	111
请求参数	111
返回结果	111
返回数据结构	111
PUT 修改当前登录客服	111
请求参数	111
返回结果	112
返回数据结构	112
POST 修改图像	112
请求参数	112
返回结果	112
返回数据结构	112
客服后台/客服工作台	112
DELETE 删除分类	112
请求参数	112
返回结果	113
返回数据结构	113
PUT 编辑分类	113

请求参数	113
返回结果	113
返回数据结构	113
POST 添加话术	113
请求参数	114
返回结果	114
返回数据结构	114
GET 客服话术	114
请求参数	114
返回结果	114
返回数据结构	114
GET 当前客服详细信息	114
请求参数	115
返回结果	115
返回数据结构	115
GET 聊天记录	115
请求参数	115
返回结果	115
返回数据结构	115
POST 保存分类	115
请求参数	115
返回结果	116
返回数据结构	116
POST 客服转接 (有问题)	116
请求参数	116
返回结果	116
返回数据结构	116
GET 转接客服列表	116
请求参数	116
返回结果	116
返回数据结构	117
DELETE 删除话术	117
请求参数	117
返回结果	117
返回数据结构	117
PUT 修改话术	117
请求参数	117
返回结果	117
返回数据结构	118
客服后台/统计	118
GET 获取当前客服统计(首页)	118
请求参数	118
返回结果	118
返回数据结构	118
GET 获取客户统计	118
请求参数	118
返回结果	119
返回数据结构	119
APP	119
POST 上传图片	119

请求参数	119
返回结果	119
返回数据结构	119
GET 获取客服广告	119
请求参数	119
返回结果	119
返回数据结构	120
POST 保存客服反馈内容	120
请求参数	120
返回结果	120
返回数据结构	120
GET 获取反馈页面广告位内容	120
请求参数	120
返回结果	120
返回数据结构	121
GET 获取缓存	121
请求参数	121
返回结果	121
返回数据结构	121
GET 获取聊天记录	121
请求参数	121
返回结果	121
返回数据结构	121
POST 保存缓存	122
请求参数	122
返回结果	122
返回数据结构	122
数据模型	122
目录结构	123
PHP目录接口	123
后端文件目录结构	123
APP目录结构	123
crmeb 目录结构	126
VUE目录结构	129
命令行介绍	130
Think-swoole命令	130
SWOOLE 命令详解	130
启动服务	130
停止服务	130
重启服务	130
平滑重启服务	130
客服系统启动命令	130
应用KEY命令	131
应用KEY命令	131
生成新KEY,不修改配置文件	131
重置/新建KEY	131
命令行安装命令	132
命令行一键安装	132
安装命令	132
启动swoole,已启动请重启swoole	133



恭喜,安装完成啦	133
开发文档	134
后台前端打包	134
常见问题	135
安装后出现：数据获取失败	135
客服端登录成功，用户端还是进入了客服反馈	136

# 简介

客服系统属于CRMEB技术团队开发，客服系统是客户和企业的桥梁，CRMChat为您的企业建立一座连接客户的桥梁。

## 传统桥梁

但是目前很多企业都做了各种渠道的宣传，大部分企业都是采用 1、展示企业电话 2、QQ 3、微信 4、企业微信 5、留言系统 6、表单提交 7、第三方客服系统

## CRMChat的好处

1、独立部署数据私有； 2、客户管理； 3、一套系统解决覆盖全网客户咨询； 4、提供源代码可以根据自己业务进行定制开发

## 应用场景

可以对接其它任何系统例如wap站、PC站点、小程序、APP、公众号等。客服系统聊天主要分为用户端和客服端 用户端支持：wap站、PC站点、小程序、公众号 客服端：PC端、APP端

## 接入方式

对接方式： 1、A链接/生成二维码 方便网址添加咨询客服图片或者文字直接打开客服进行聊天，这个非常灵活各种站点都可以使用； 2、js引入 后台可以获取js代码嵌入网页中，网页就拥有了客服聊天入口，也可以深度对接自己站点的用户数据。 3、深度开发接入

关注CRMchat开源项目获取最新更新包<http://https://gitee.com/ZhongBangKeJi/CRMChat>，如果您有帮助，请帮忙点下star 🌟 感谢 🙏



扫一扫上面的二维码图案，加我为朋友。

# 安装

## PHP设置

### 步骤总结

一、安装PHP插件：fileinfo、redis、swoole4。二、删除PHP对应版本中的 proc\_open 禁用函数。

### 步骤详解：

1. 进入宝塔面板点击 软件商城 ,点击 PHP设置 ,这里以 PHP7.3 为例;

The screenshot shows the 'Software Store' (软件商店) interface in the Baiduwan control panel. A sidebar on the left contains navigation options like '网站', 'FTP', '数据库', '监控', '安全', '文件', '计划任务', '软件商店', '面板设置', and '退出'. The main area features a search bar and filters for application categories. A table lists various software applications, with the PHP versions section highlighted. The row for 'php PHP-7.3' is circled in red, and a red arrow points to the '更新' (Update) button in its '操作' (Action) column.

软件名称	开发商	说明	价格	到期时间	位置	状态	首页显示	操作
Apache	官方	世界排名第一、快速、可靠并且可通过简单的API扩充	免费	--				安装
MySQL 5.7.25	官方	MySQL是一种关系数据库管理系统!	免费	--			更新   设置   卸载	
php PHP-7.4	官方	PHP是世界上最好的编程语言	免费	--				安装
php PHP-7.3	官方	PHP是世界上最好的编程语言	免费	--			更新   设置   卸载	
php PHP-7.2	官方	PHP是世界上最好的编程语言	免费	--			设置   卸载	
php PHP-7.1	官方	PHP是世界上最好的编程语言	免费	--			更新   设置   卸载	
php PHP-7.0	官方	PHP是世界上最好的编程语言	免费	--			设置   卸载	
php PHP-5.6	官方	PHP是世界上最好的编程语言	免费	--			设置   卸载	
php PHP-5.5	官方	PHP是世界上最好的编程语言	免费	--				安装
php PHP-5.4	官方	PHP是世界上最好的编程语言	免费	--				安装
php PHP-5.3	官方	PHP是世界上最好的编程语言	免费	--			设置   卸载	
php PHP-5.2	官方	PHP是世界上最好的编程语言	免费	--				安装
FTP Pure-FTPd 1.0.49	官方	PureFTPd是一款专注于程序健壮和软件安全的免费FTP服务器软件	免费	--			设置   卸载	
宝塔云控平台(Beta)	官方	云控平台免费版包含多机管理、SSH终端、资产管理、拨测监控。确认自己有需要这些功能才安装使用。	免费	--				安装

2. 进入安装扩展,安装: fileinfo、redis、swoole4 扩展插件

### php-7.3管理

服务	名称	类型	说明	状态	操作
安装扩展	ionCube	脚本解密	用于解密ionCube Encoder加密脚本!	✘	安装
配置修改	fileinfo	通用扩展	若可用内存小于1G, 可能会安装不上	✔	卸载
上传限制	opcache	缓存器	用于加速PHP脚本!	✘	安装
超时限制	memcached	缓存器	比memcache支持更多高级功能	✘	安装
配置文件	redis	缓存器	基于内存亦可持久化的Key-Value数据库	✔	卸载
禁用函数	apcu	缓存器	脚本缓存器	✘	安装
性能调整	imagemagick	通用扩展	Imagick高性能图形库	✘	安装
负载状态	xdebug	调试器	开源的PHP程序调试器	✘	安装
Session配置	imap	邮件服务	邮件服务器必备	✘	安装
日志	exif	通用扩展	用于读取图片EXIF信息	✘	安装
慢日志	intl	通用扩展	提供国际化支持	✘	安装
phpinfo	<ul style="list-style-type: none"> <li>Redis扩展仅支持一个PHP版本安装使用, 若在其它PHP版本已安装redis扩展, 请勿再装</li> <li>请按实际需求安装扩展,不要安装不必要的PHP扩展,这会影响PHP执行效率,甚至出现异常</li> <li>opcache/xcache/apc等脚本缓存扩展,请只安装其中1个,否则可能导致您的站点程序异常</li> </ul>				

服务	imap	邮件服务	邮件服务器必备	✘	安装
安装扩展	exif	通用扩展	用于读取图片EXIF信息	✘	安装
配置修改	intl	通用扩展	提供国际化支持	✘	安装
上传限制	xsl	通用扩展	xsl解析扩展	✘	安装
超时限制	Swoole4	通用扩展	swoole4.x版本	✔	卸载
配置文件	yaf	框架	Yaf是一个C语言编写的PHP框架	✘	安装
禁用函数	phalcon	框架	phalcon是一个C语言编写的PHP框架	✘	安装
性能调整	mongodb	通用扩展	Mongodb数据库连接驱动	✘	安装
负载状态	yac	缓存器	高性能无锁共享内存Cache	✘	安装
Session配置	sg11	脚本加密	php代码混淆加密	✔	卸载
日志	pgsql	通用扩展	php官方扩展(需提前安装postgresql)	✘	安装
慢日志	pdo_pgsql	通用扩展	php官方扩展(需提前安装postgresql)	✘	安装
phpinfo					

- Redis扩展仅支持一个PHP版本安装使用，若在其它PHP版本已安装redis扩展，请勿再装
- 请按实际需求安装扩展,不要安装不必要的PHP扩展,这会影响PHP执行效率,甚至出现异常
- opcache/xcache/apc等脚本缓存扩展,请只安装其中1个,否则可能导致您的站点程序异常

服务

安装扩展

配置修改

上传限制

超时限制

配置文件

**禁用函数**

性能调整

负载状态

Session配置

日志

慢日志

phpinfo

添加要被禁止的函数名,如: exec 保存

chgrp	删除
chown	删除
popen	删除
proc_open	删除
pcntl_exec	删除
ini_alter	删除
ini_restore	删除
dl	删除
openlog	删除
syslog	删除

- 在此处可以禁用指定函数的调用,以增强环境安全性!
- 强烈建议禁用如exec,system等危险函数!

3. 进入 禁用函数 ,找到 proc\_open 删除

php-7.3管理

服务

当前状态: 开启 ▶

停止 重启 重载配置

安装扩展

配置修改

上传限制

超时限制

配置文件

禁用函数

性能调整

负载状态

Session配置

日志

慢日志

phpinfo

4. 进入 服务 ,选择重载配置
5. PHP配置完成.进入站点配置

## 站点配置

### 步骤总结

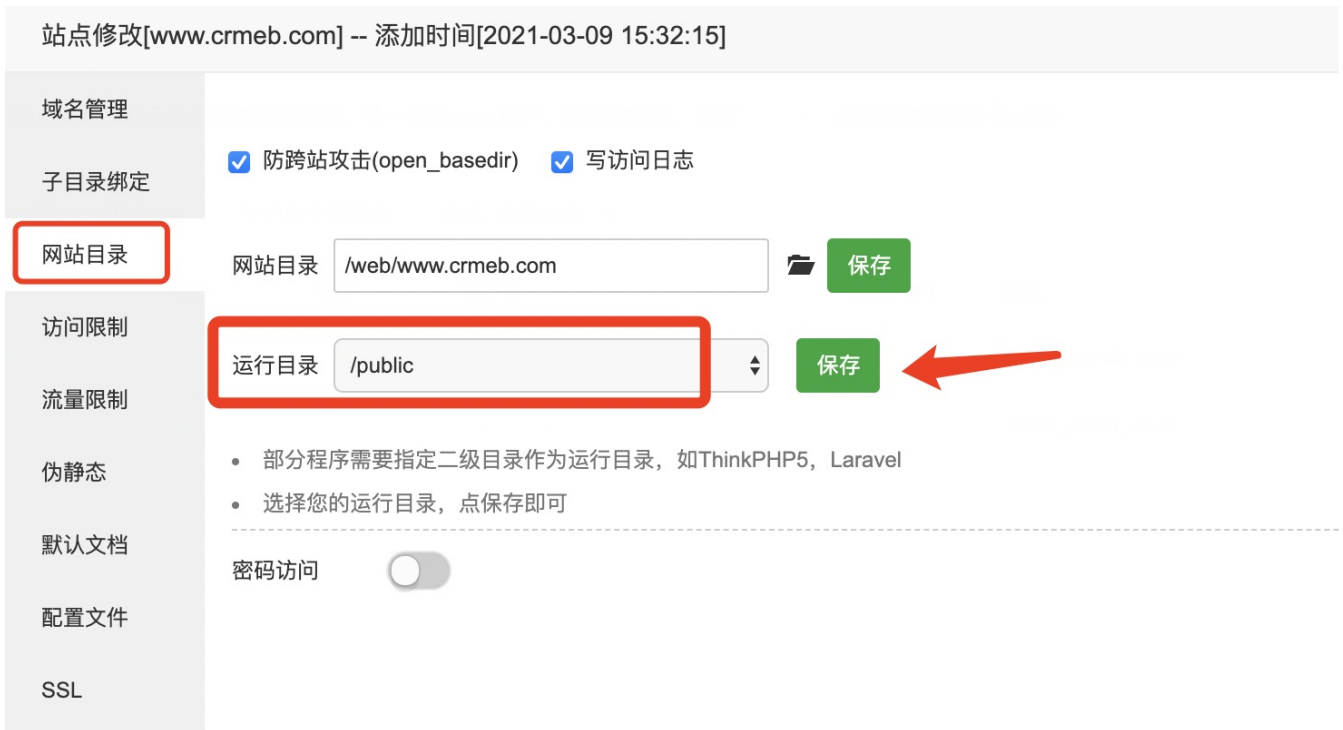
一、设置网站运行目录public，二、设置PHP版本选择纯静态。三、可选项如有需求则开启SSL,配置SSL证书，开启强制https域名。四、添加反向代理。

### 步骤详解

一. 点击宝塔左侧菜单网站,点击项目站点名称



二. 进入 网站目录 ,设置运行目录为 `public` ,然后点击 保存



三. 进入 PHP版本 ,选择纯静态,点击 切换

域名管理

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

SSL

**PHP版本**

Tomcat

重定向

PHP版本 纯静态 切换

- 请根据您的程序需求选择版本
- 若非必要,请尽量不要使用PHP5.2,这会降低您的服务器安全性;
- PHP7不支持mysql扩展, 默认安装mysqli以及mysql-pdo。

session隔离

- 开启后将会把session文件存放到独立文件夹独立文件夹, 不与其他站点公用存储位置
- 若您在PHP配置中将session保存到memcache/redis等缓存器时, 请不要开启此选项

四. 需要配置https的请查看,不需要可跳过;

进入SSL,配置SSL证书,配置好后点击保存.

没有证书的请查看教程: [Nginx SSL证书配置](#) Apache环境证书需要合并后再填写请看: [Apache SSL证书配置](#)

域名管理

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

**SSL**

PHP版本

商用证书 宝塔SSL Let's Encrypt 其他证书 关闭 证书夹 强制HTTPS

密钥(KEY) 证书(PEM格式)

.key 格式文件内容 .pem 或 .crt 格式文件内容

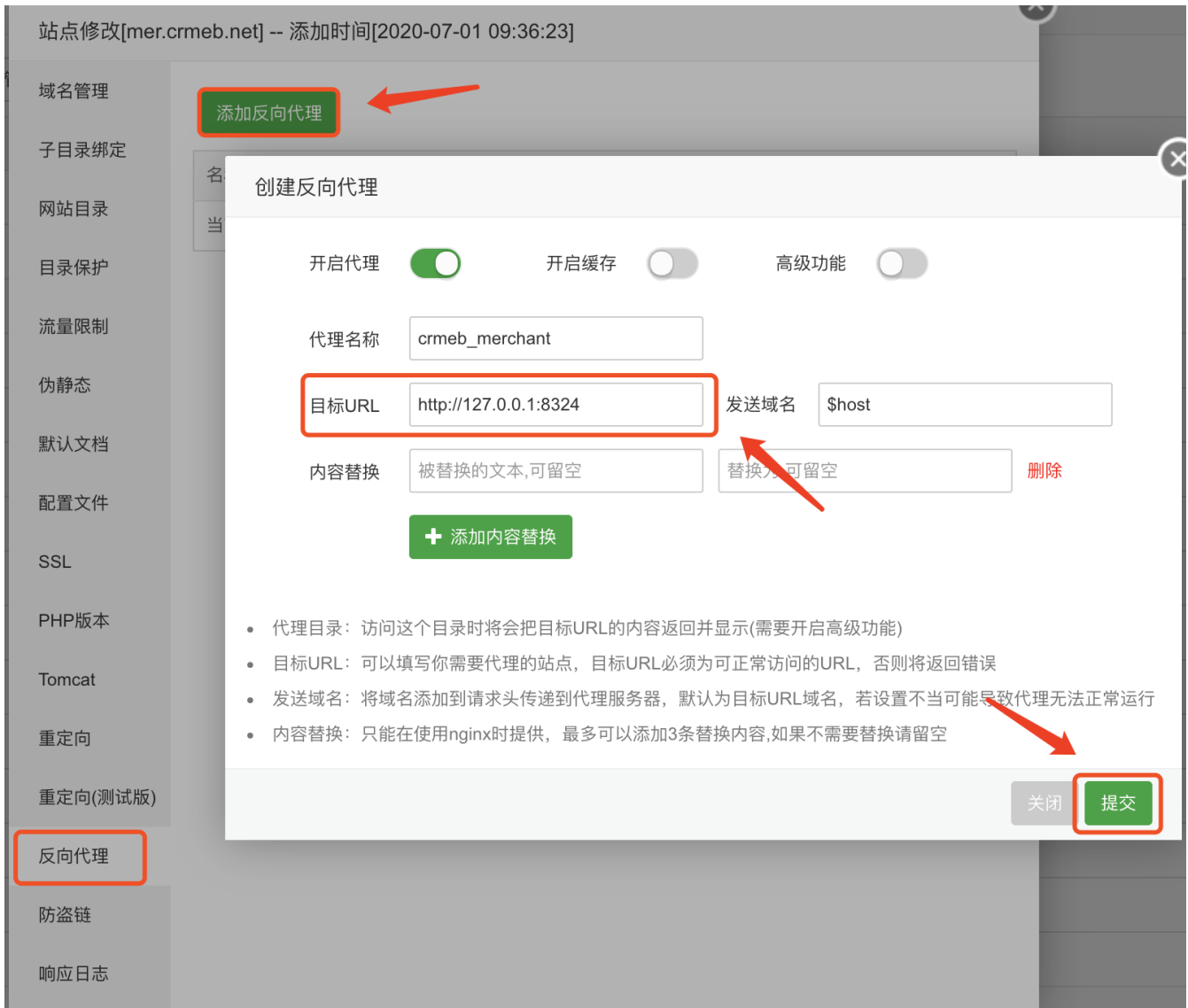
保存

五. 配置反向代理

nginx 和 apache反向代理配置二选一查看文档, 安装了哪个程序就看哪个的配置说明

nginx反向代理配置 添加反向代理,代理名称随意填写,尽量为英文,添加目标URL: `http://127.0.0.1:20108`,点击提交保存配置





如过 添加反向代理 报错失败请先删除 伪静态 里面的配置

配置ws反向代理,点击配置文件

站点修改[new.lfmn.fun] -- 添加时间[2021-02-25 16:59:34]

域名管理

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

SSL

PHP版本

Tomcat

重定向

反向代理

防盗链

响应日志

添加反向代理

<input type="checkbox"/>	名称	代理目录	目标url	缓存	状态	操作
<input type="checkbox"/>	crmeb_oa_new	/	http://127.0.0.1:8192	已关闭	运行中▶	配置文件   编辑   删除

编辑配置文件[crmeb\_oa\_new]

```

7   proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
8   proxy_set_header REMOTE-HOST $remote_addr;
9   }
10  location /
11  {
12     proxy_pass http://127.0.0.1:20108;
13     proxy_set_header Host $host;
14     proxy_set_header X-Real-IP $remote_addr;
15     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
16     proxy_set_header REMOTE-HOST $remote_addr;
17
18     add_header X-Cache $upstream_cache_status;
19
20     #Set Nginx Cache
21
22     add_header Cache-Control no-cache;
23     expires 12h;
24  }
25
26 #PROXY-BIND/

```

保存

- 此处为该负载均衡的配置文件, 若您不了解配置规则, 请勿随意修改。

制以下内容, 替换配置文件10~24行, 并保存(如果更换了端口, 就需要改变ip后面携带的端口号)

不固定是10~24行, 请根据自身情况。主要是替换掉location / {}, 花括号内的内容

```

location /
{
    proxy_pass http://127.0.0.1:20108;
    proxy_http_version 1.1;
    proxy_read_timeout 360s;
    proxy_redirect off;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header REMOTE-HOST $remote_addr;

    add_header X-Cache $upstream_cache_status;

    #Set Nginx Cache

    add_header Cache-Control no-cache;
    expires 12h;
}

```

编辑配置文件[crmeb\_oa\_new]

```
11 1
12 proxy_pass http://127.0.0.1:20108;
13 proxy_http_version 1.1;
14 proxy_read_timeout 360s;
15 proxy_redirect off;
16 proxy_set_header Upgrade $http_upgrade;
17 proxy_set_header Connection "upgrade";
18 proxy_set_header Host $host;
19 proxy_set_header X-Real-IP $remote_addr;
20 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
21 proxy_set_header REMOTE-HOST $remote_addr;
22
23 add_header X-Cache $upstream_cache_status;
24
25 #Set Nginx Cache
26
27     add_header Cache-Control no-cache;
28     expires 12h;
29 }
30
```

[保存](#)

- 此处为该负载均衡的配置文件,若您不了解配置规则,请勿随意修改。

替换后,点击保存

apache反向代理配置

添加完反向代理后,点击左侧 配置文件

站点修改[www.crmeb.com] -- 添加时间[2021-03-05 19:09:04]

域名管理

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

SSL

PHP版本

Tomcat

重定向

反向代理

提示: Ctrl+F 搜索关键字, Ctrl+S 保存, Ctrl+H 查找替换

```

1 <VirtualHost *:80>
2     ServerAdmin webmaster@example.com
3     DocumentRoot "/www/wwwroot/www.crmeb.com"
4     ServerName 6495258c.www.crmeb.com
5     ServerAlias www.crmeb.com
6     #errorDocument 404 /404.html
7     ErrorLog "/www/wwwlogs/www.crmeb.com-error_log"
8     CustomLog "/www/wwwlogs/www.crmeb.com-access_log" combined
9     #引用反向代理规则, 注释后配置的反向代理将无效
10
11     IncludeOptional /www/server/panel/vhost/apache/proxy/www.crmeb.com/*.conf
12
13     #DENY FILES
14     <Files ~ (\.user.ini|.htaccess|.git|.svn|.project|LICENSE|README.md)$>
15         Order allow,deny
16         Deny from all
17     </Files>
18
19     #PHP
20     <FilesMatch \.php$>
21         SetHandler "proxy:unix:/tmp/php-cgi-00.sock|fcgi://localhost"
22     </FilesMatch>
    
```

保存

此处为站点主配置文件,若您不了解配置规则,请勿随意修改.

在第:80端口下,添加代码

```

ProxyPass /ws ws://127.0.0.1:20108/
ProxyPassReverse /ws ws://127.0.0.1:20108/
    
```

复制 刚添加的两行代码和下面一行, 总共三行代码:

域名管理

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

提示: Ctrl+F 搜索关键字, Ctrl+S 保存, Ctrl+H 查找替换

```

1 <VirtualHost *:80>
2     ServerAdmin webmaster@example.com
3     DocumentRoot "/www/wwwroot/www.crmeb.com"
4     ServerName 6495258c.www.crmeb.com
5     ServerAlias www.crmeb.com
6     #errorDocument 404 /404.html
7     ErrorLog "/www/wwwlogs/www.crmeb.com-error_log"
8     CustomLog "/www/wwwlogs/www.crmeb.com-access_log" combined
9     #引用反向代理规则, 注释后配置的反向代理将无效
10     ProxyPass /ws ws://127.0.0.1:20199/
11     ProxyPassReverse /ws ws://127.0.0.1:20199/
12     IncludeOptional /www/server/panel/vhost/apache/proxy/www.crmeb.com/*.conf
13
14     #DENY FILES
    
```

域名管理

提示: Ctrl+F 搜索关键字, Ctrl+S 保存, Ctrl+H 查找替换

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

```

1 <VirtualHost *:80>
2     ServerAdmin webmaster@example.com
3     DocumentRoot "/www/wwwroot/www.crmeb.com"
4     ServerName 6495258c.www.crmeb.com
5     ServerAlias www.crmeb.com
6     #errorDocument 404 /404.html
7     ErrorLog "/www/wwwlogs/www.crmeb.com-error_log"
8     CustomLog "/www/wwwlogs/www.crmeb.com-access_log" combined
9     #引用反向代理规则, 注释后配置的反向代理将无效
10    ProxyPass /ws ws://127.0.0.1:20199/
11    ProxyPassReverse /ws ws://127.0.0.1:20199/
12    IncludeOptional /www/server/panel/vhost/apache/proxy/www.crmeb.com/*.conf
13
14    #DENY FILES
15    <Files ~ (\.user.ini|.htaccess|.git|.svn|.project|LICENSE|README.md)$>
16        Order allow,deny
    
```

复制

下拉查找443端口, 在其下方粘贴。

不使用https域名, 则不用配置443端口。无443端口, 请查看站点SSL证书配置。

域名管理

提示: Ctrl+F 搜索关键字, Ctrl+S 保存, Ctrl+H 查找替换

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

```

31 </VirtualHost>
32 <VirtualHost *:443>
33     ServerAdmin webmaster@example.com
34     DocumentRoot "/www/wwwroot/www.crmeb.com/public"
35     ServerName SSL.www.crmeb.com
36     ServerAlias www.crmeb.com
37     #errorDocument 404 /404.html
38     ErrorLog "/www/wwwlogs/www.crmeb.com-error_log"
39     CustomLog "/www/wwwlogs/www.crmeb.com-access_log" combined
40     ProxyPass /ws ws://127.0.0.1:20199/
41     ProxyPassReverse /ws ws://127.0.0.1:20199/
42     IncludeOptional /www/server/panel/vhost/apache/proxy/www.crmeb.com/*.conf
43
44     #SSL
45     SSL Engine On
    
```

443 端口

粘贴

Apache反向代理配置完成

六. 进入下一步, 启动服务

## 运行服务

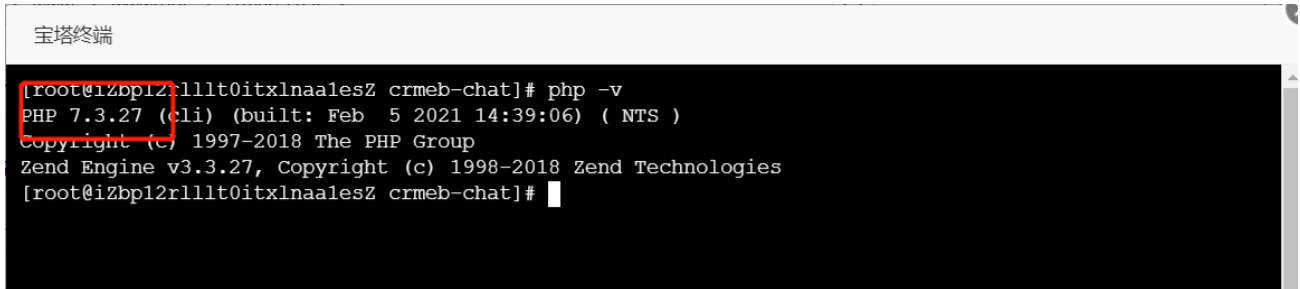
### 步骤总结

一、切换PHP命令行版本和站点一致。二、执行启动命令，启动swoole。

### 步骤详解

一. 检测命令行PHP版本,如果不再7.1~7.4之间请按照以下教程切换;

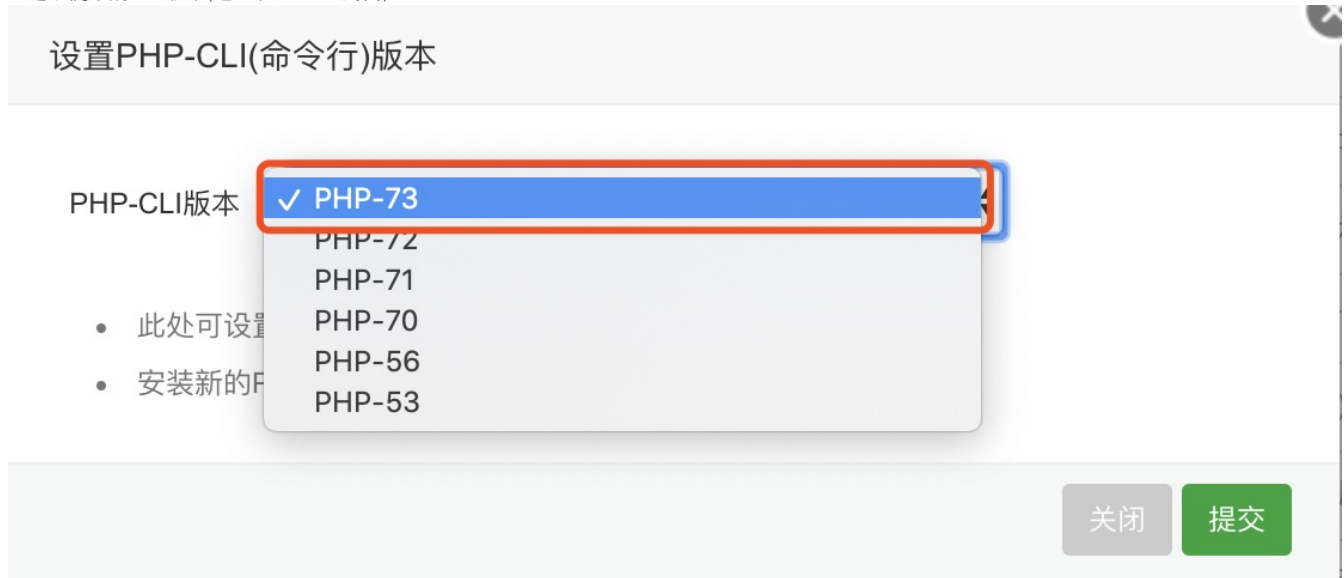
1. 检测PHP版本,如达到要求直接跳过,直接进入 启动命令



2.修改命令行PHP版本



3.选中使用的PHP版本(这里以PHP7.3为例)

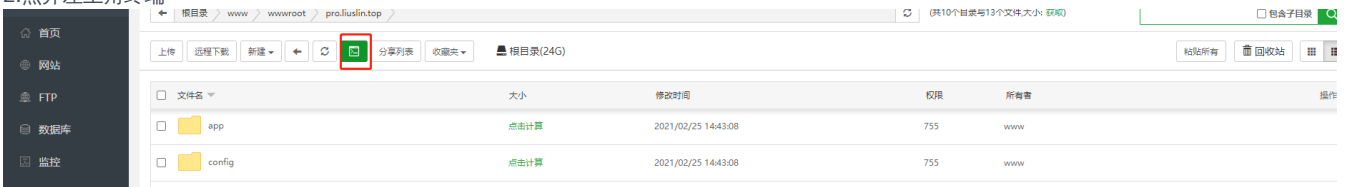


## 二. 启动命令

1.点击打开项目根目录



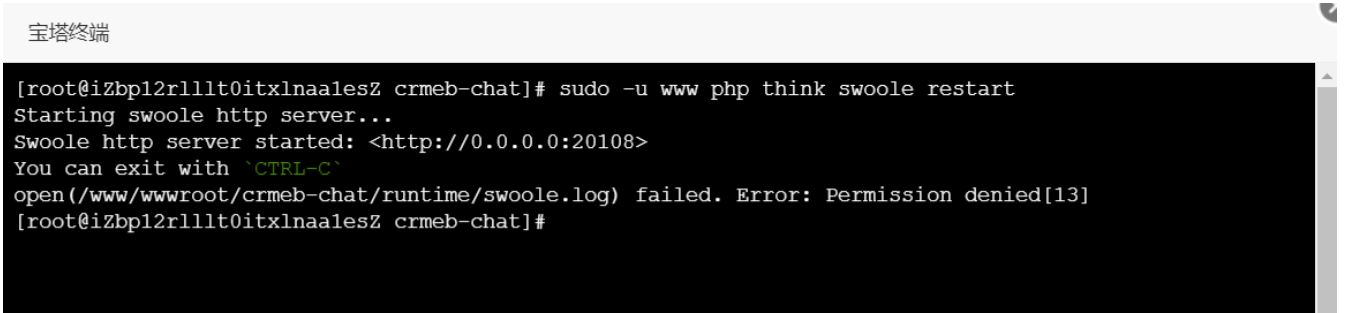
### 2. 点开左上角终端



### 3. 执行swoole启动命令

```
php think swoole restart
```

展示如下界面,启动成功



启动成功后,则可关闭退出;

三. 运行完成,下一步进入程序一键安装

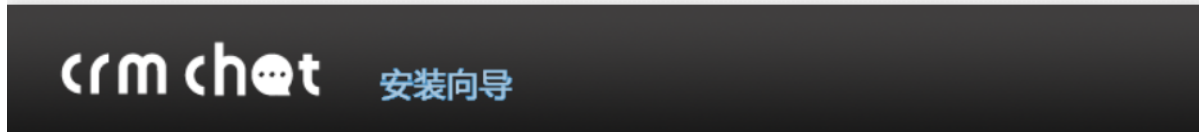
## 程序安装

### 安装步骤

一、输入域名，执行程序一键安装。二、安装完成后必须重启swoole 三、恭喜您，安装完成！多项目部署：同服务器多项目部署，请务必在点击最下方多项目部署配置，前些缓存前缀！

### 步骤详解

一.浏览器新建窗口,输入您的域名,确认,系统会跳转自动进入安装界面,点击接受进入下一步



### 软件许可协议

提示条款：  
本协议是您与西安众邦网络科技有限公司共同签署。  
**CRMEB客户管理+电商系统（以下称“CRMEB”），由西安众邦网络科技有限公司（以下称“众邦科技”）独创开发，版权所有Copyright (c) 2014-2024，众邦科技保留所有权利。CRMEB是国内最稳定、最强大、最先进的互联网电商平台解决方案之一，CRMEB基于 PHP + MySQL 的技术，采用ThinkPHP框架开发。CRMEB官方对此拥有最终修改权和解释权。**  
CRMEB官方网站：<http://www.crmeb.com>  
在使用CRMEB客户管理+电商系统（以下称“许可软件”或“本软件”）之前，请您仔细阅读本协议，特别是法律适用和争议解决条款，此等条款将以**粗体**标识，您需要重点阅读。如您对协议有任何疑问，可向客服咨询。如果您已下载、复制、安装或以其他方式使用该软件，则视为已经接受本协议。如果您不接受本协议的全部或部分条款，您将无权使用本软件。请立即终止安装、或以其他方式使用该软件，删除您已经安装或保留的该软件的任何组件。  
由于互联网高速发展，您与我们签署的本协议列明的条款并不能完整罗列并覆盖您与我们所有权利与义务，现有的约定也不能保证完全符合未来发展的需求。  
因此，《版权声明》及其他规则均为本协议的补充协议，与本协议不可分割且具有同等法律效力。如您使用许可软件，即视为同意上述补充协议。我们如修改本协议或其补充协议，协议条款修改后，请您仔细阅读并接受修改后的协议。

**接受**

二.检测环境是否符合系统要求,确认无误后,点击下一步.如有X号等错误提示,请根据系统要求修改服务器环境配置后重启swoole,刷新当前页面

重启命令: `sudo -u www php think swoole restart`



# crm chat 安装向导

## 1 检测环境

## 2 创建数据

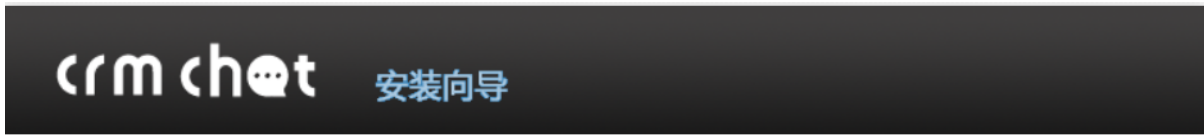
## 3 完成安装

环境检测	推荐配置	当前状态	最低要求
操作系统	类UNIX	✓ Linux	不限制
PHP版本	>7.1.0	✓ 7.3.27	7.1.0以上
附件上传	>2M	✓ 50M	不限制
session	开启	✓ 支持	开启
safe_mode	基础配置	[x]Off	启用
GD库	必须开启	[x]On bundled (2.1.0 compatible)	1.0
swoole	必须安装	✓ 已安装	4.0=>
redis	必须开启	✓ 已安装	启用
mysql	必须开启	✓ 已安装	启用
curl_init	必须扩展	[x]支持	启用
bcmath	必须扩展	[x]支持	启用
openssl	必须扩展	[x]支持	启用
fileinfo	必须扩展	[x]支持	启用
<b>目录、文件权限检查</b>		<b>写入</b>	<b>读取</b>
public/uploads	读写	✓ 可写	✓ 可读
runtime	读写	✓ 可写	✓ 可读
<b>函数检测必须开启</b>		<b>当前状态</b>	
file_put_contents		[x]开启	
imagefttext		[x]开启	

重新检测

下一步

三.填写数据库用户名、数据库名、数据库密码



- 1 检测环境
- 2 创建数据
- 3 完成安装

数据库信息

数据库服务器: 127.0.0.1 数据库服务器地址, 一般为127.0.0.1

数据库端口: 3306 数据库服务器端口, 一般为3306

数据库用户名: root

数据库密码:

数据库名: crmeb

数据库表前缀: eb\_ 建议使用默认, 同一数据库安装多个CRMEB时需修改

演示数据:

reids数据库信息

如果忘记数据库相关信息,可在宝塔后台数据库中查看



四.填写Redis数据库密码

reids数据库信息

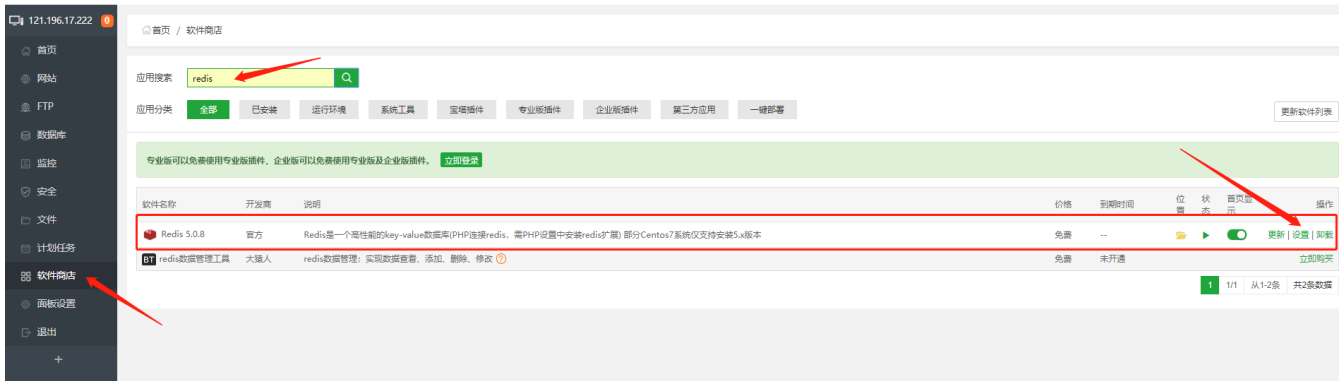
服务器地址: 127.0.0.1 reids服务器地址, 一般为127.0.0.1

端口号: 6379 reids端口,默认为6379

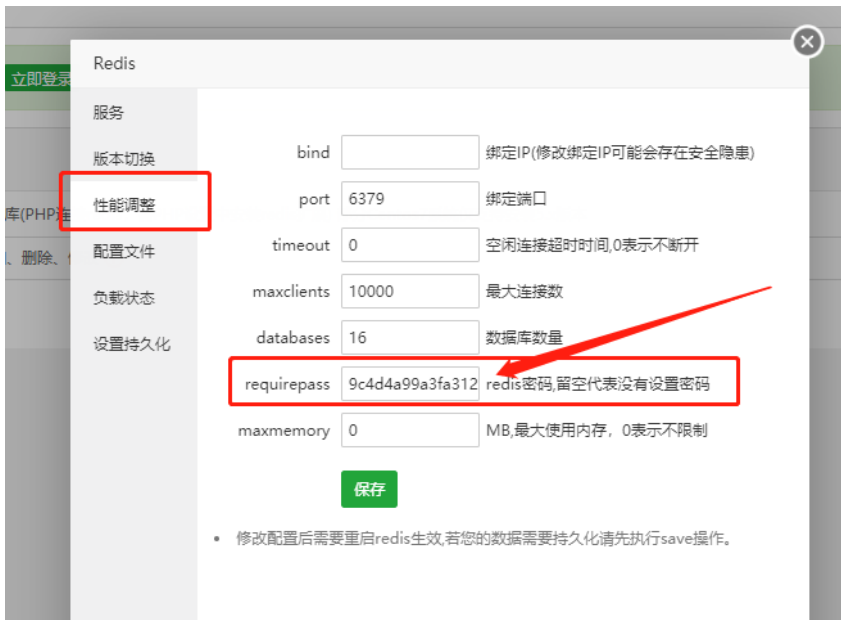
数据库: 0 reids数据库, 默认为0,一般不做更改

数据库密码:

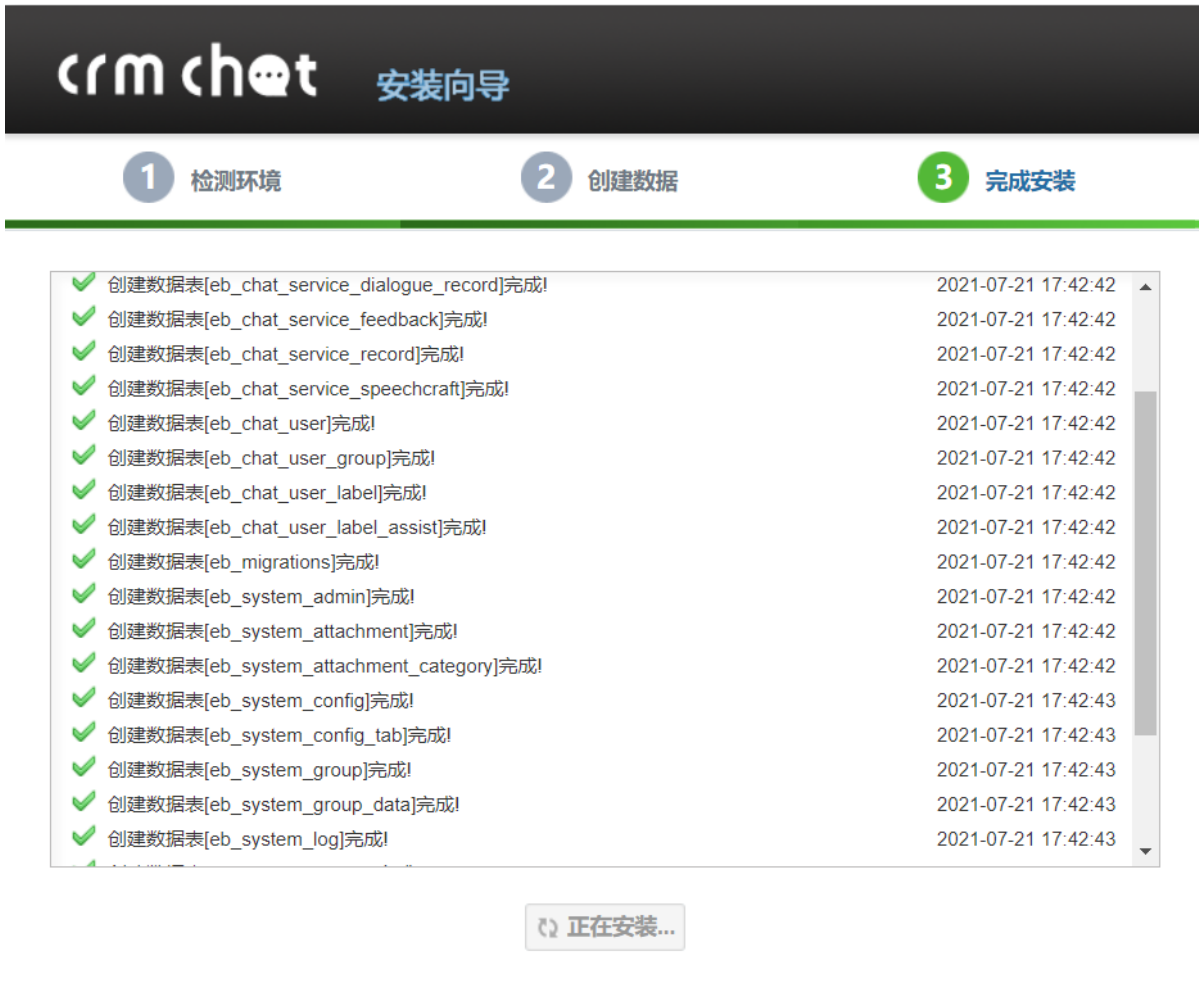
如果不知道密码,可在宝塔: 软件管理->Redis->设置->密码 中查看或者设置密码



复制密码(也可以直接留空,就不用填写配置密码)



五. 填写管理员密码后,点击创建数据,程序开始自动安装



六. 等待程序安装,安装完成后建议删除install目录下所有文件除了install.lock文件不能删除!



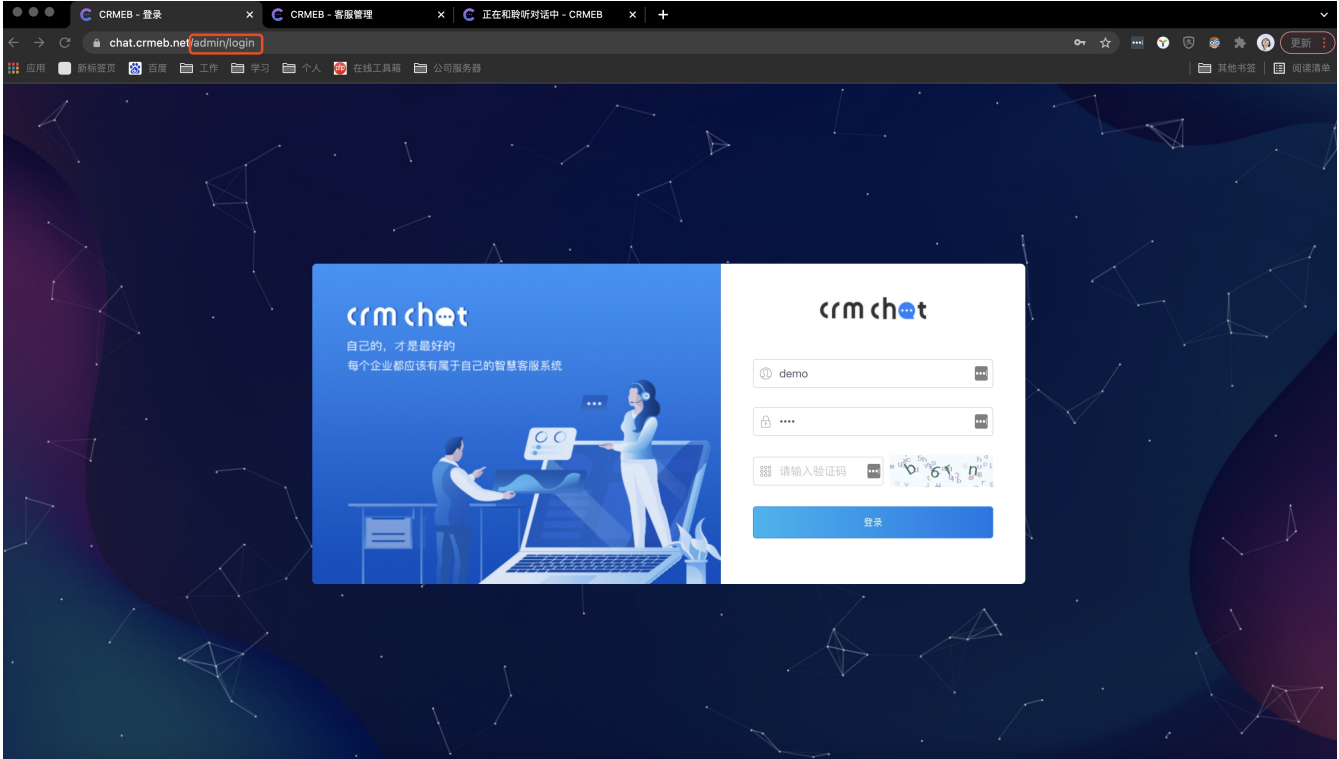
七.安装程序完成.执行: `sudo -u www php think swoole restart` 执行完后,就可以访问后台和前端了

# 使用文档

## 新手入门

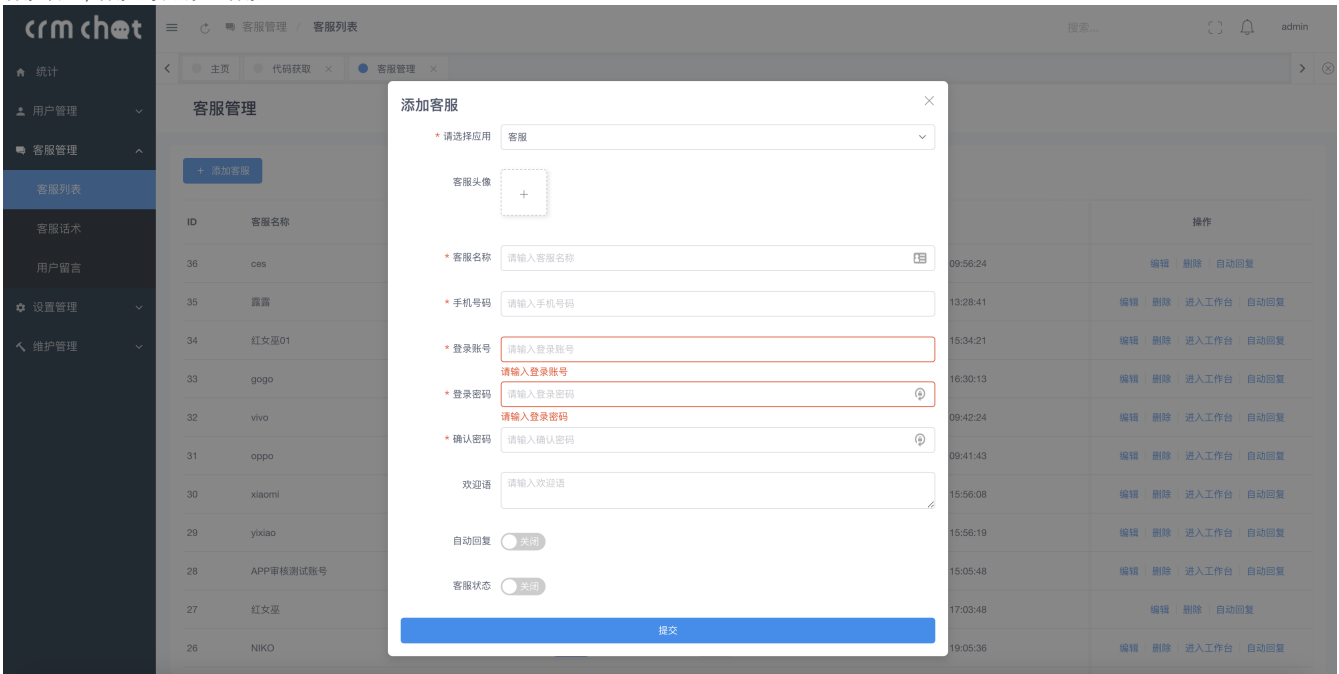
### 1、登录后台

<http://域名/admin> 访问管理后台



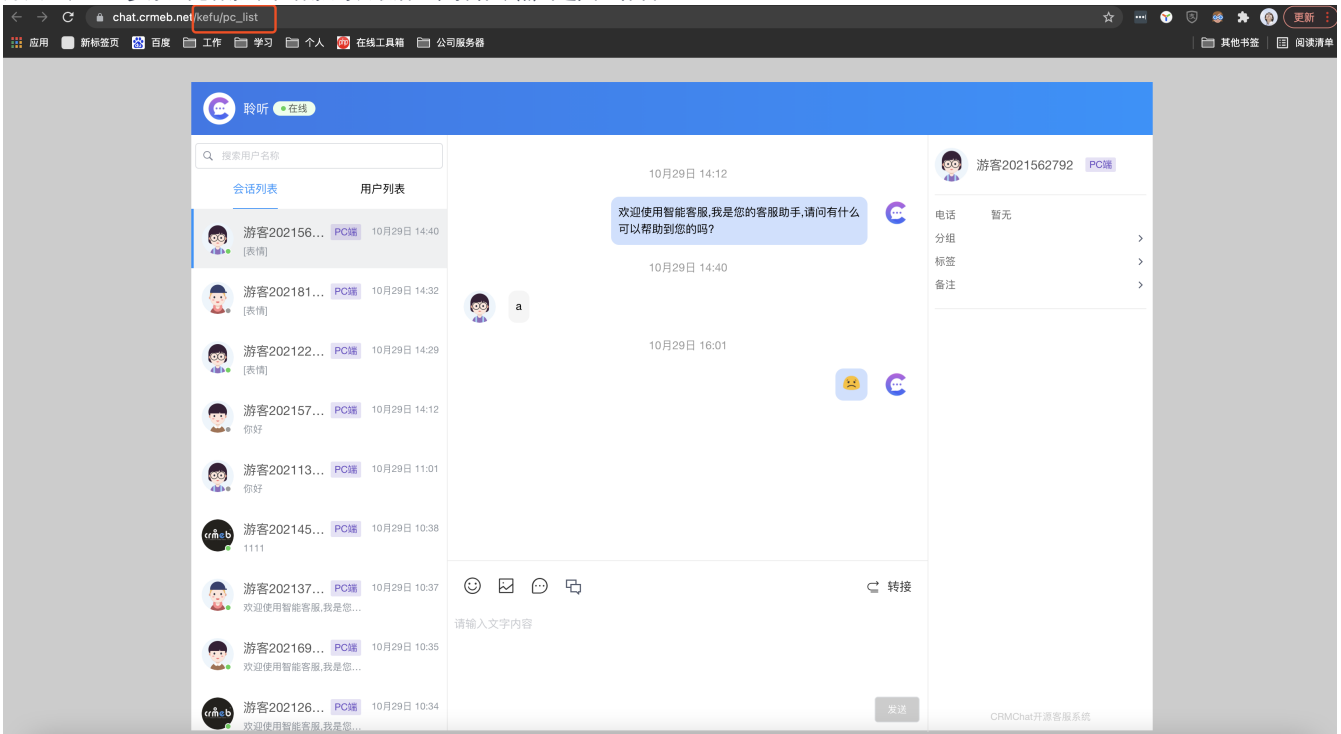
### 2、添加客服

客服管理/客服列表-添加客服

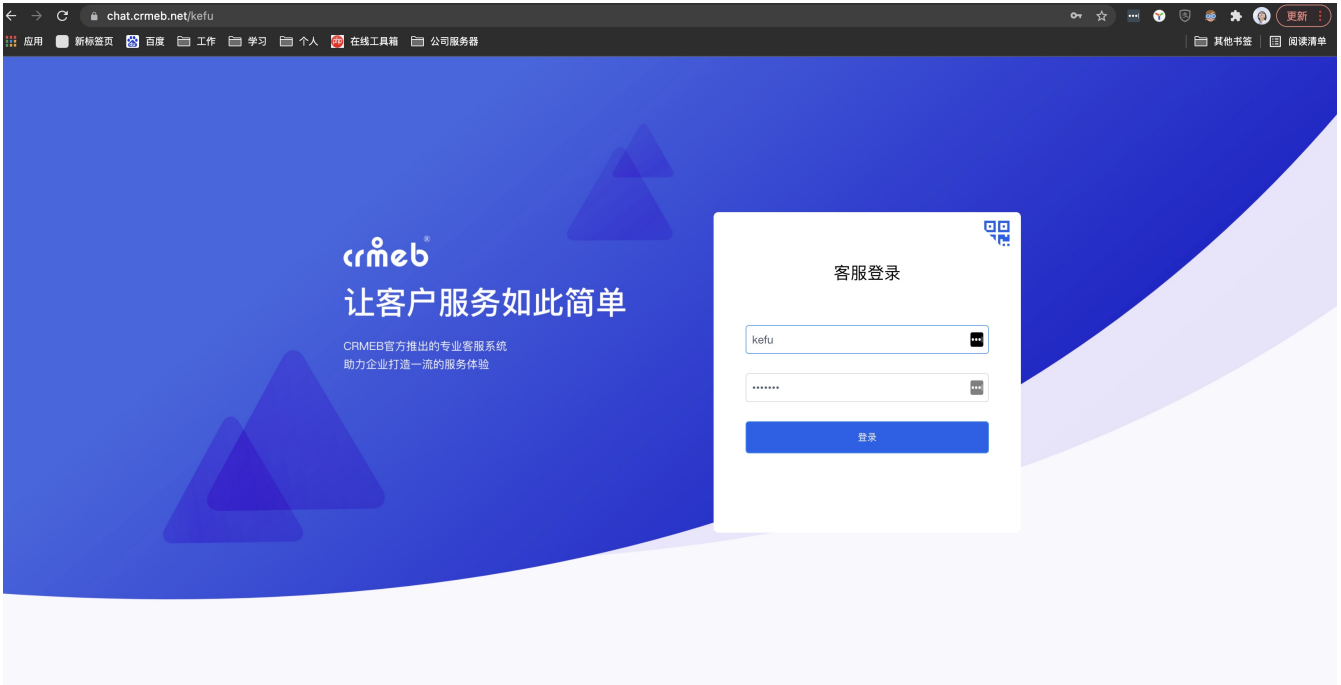


### 3、登录客服

方法一、上一步添加完客服，在客服列表最后一栏操作，点击进入工作台



方法二、直接访问客服 <http://域名/kefu>，输入刚添加客户的账号密码进行登录

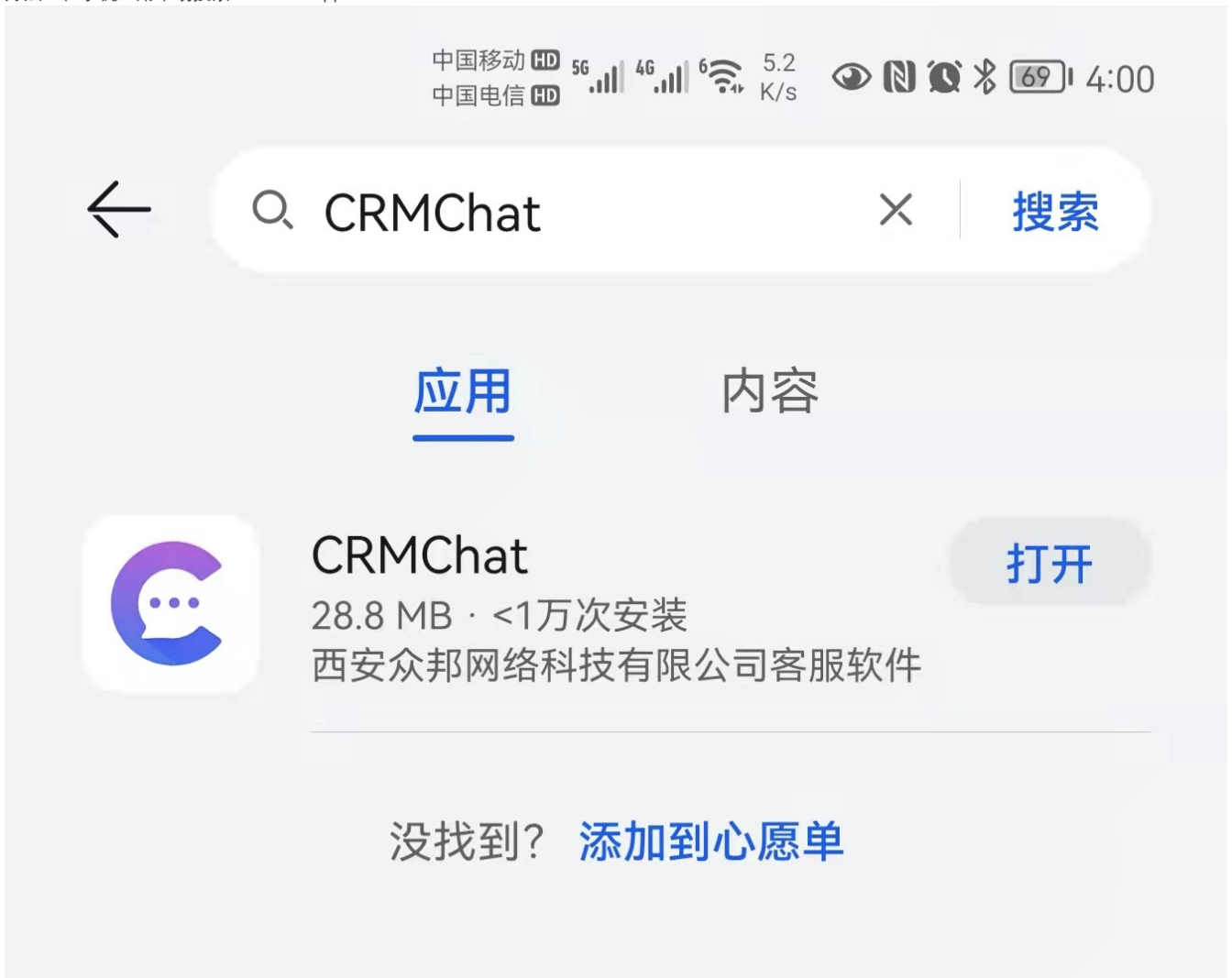


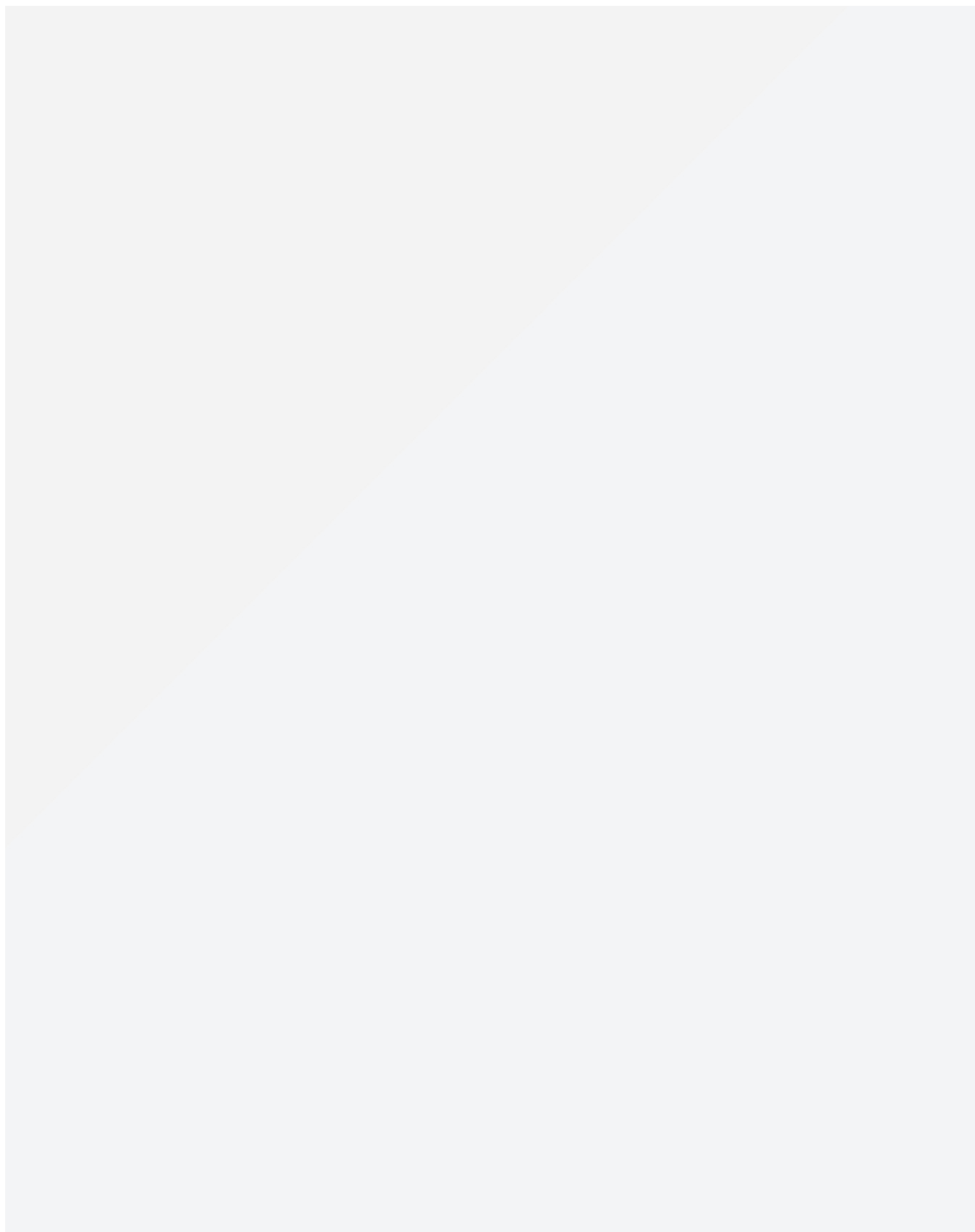
如果手机客服端登录可以使用APP扫码登录，APP右上角按钮



Copyright © 2020 西安众邦网络科技有限公司 | 客服系统

方法三、手机应用市场搜索CRMChat app





配置请求域名



## 设置请求域名

https:// chat.crmeb.net

---

取消

确定

注意：请根据您站点请求协议，选择http或https，再输入您站点的域名

## 用户端访问体验咨询吧

### 1、后台设置管理/获取代码

crm chat

设置管理 / 代码获取

统计

用户管理

客服管理

设置管理

系统设置

页面管理

管理权限

代码获取

维护管理

网页内嵌 **超链接** 定制开发 重置token

使用简介

A链接可以在网页中或内容中添加A链接使用，可以在自己站点使用，也可以发布的外站使用，或者自己生成链接二维码使用；

超链接参数说明

token:与后台交互的凭证  
noCanClose:PC端是否显示广告  
deviceId:手动适应浏览器类型Mobile移动端  
uid:用户ID  
nickName:用户昵称  
phone:用户手机号  
sex:用户  
avatar:用户  
openid:用户  
kefu\_id:客服ID

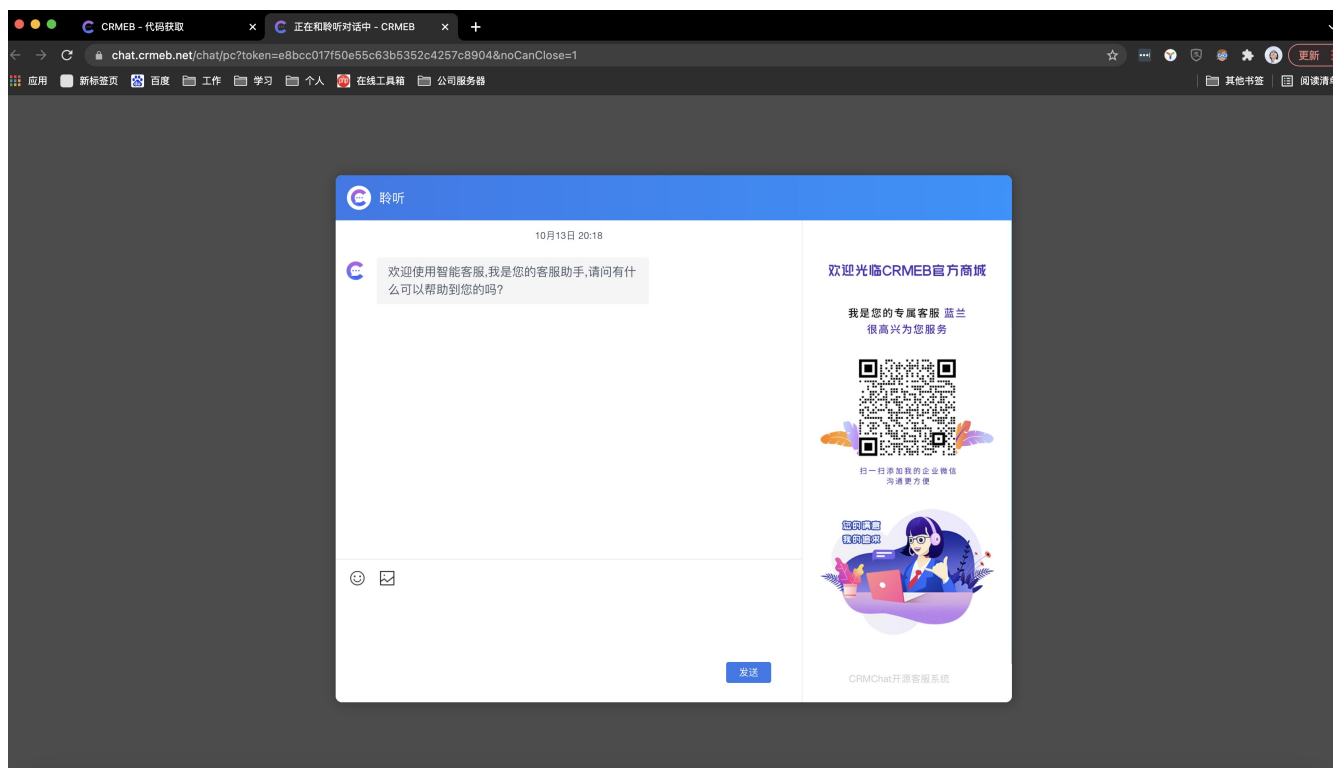
第一种，自动识别浏览器A链接使用

`https://chat.crmeb.net/chat/index?noCanClose=1&token=e8bcc017f50e55c63b5352c4257c8904`

点击体验 复制代码

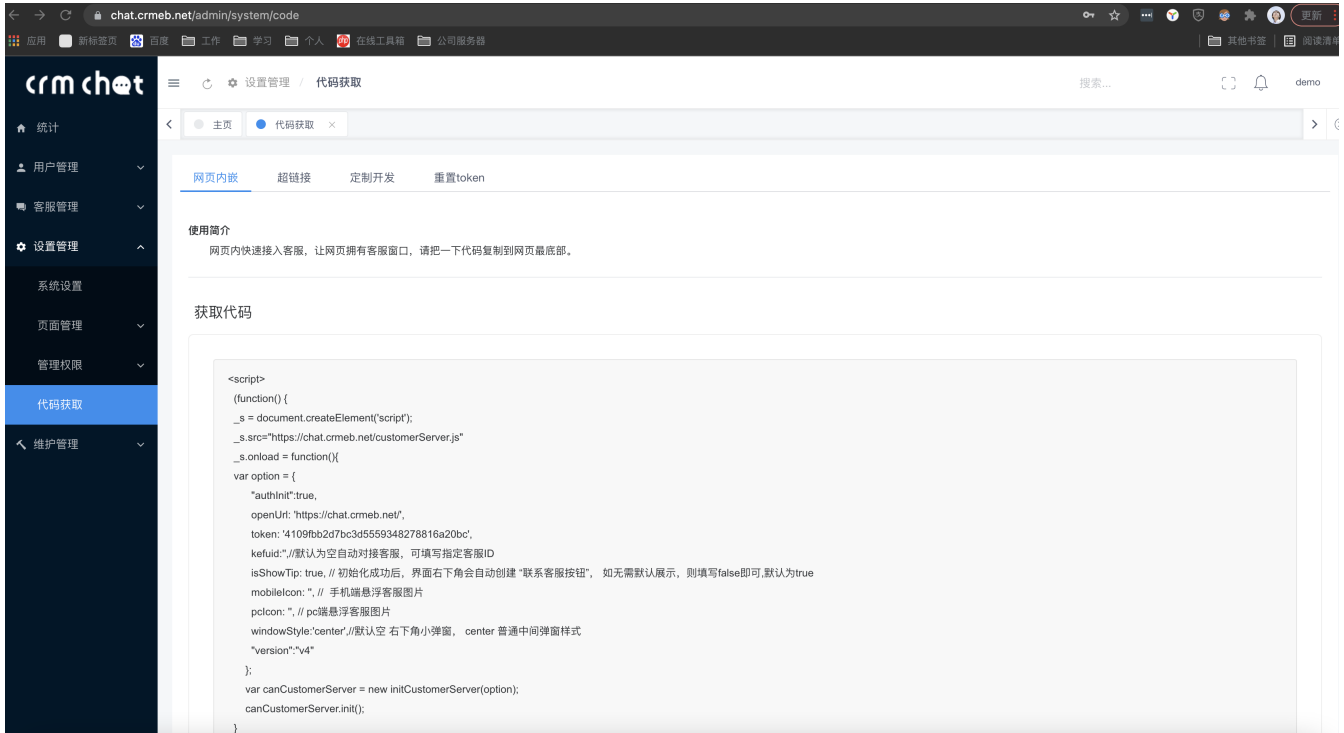
第二种，PC端A链接使用

### 2、看到用户的聊天界面



# 站点引用

## 第一步 获取js代码



```
<script>
(function() {
  _s = document.createElement('script');
  _s.src="https://chat.crmeb.net/customerServer.js"
  _s.onload = function(){
    var option = {
      "authInit":true,
      openUrl: 'https://chat.crmeb.net/',//这个自动获取您的网址
      token: '4109fbb2d7bc3d5559348278816a20bc',//必填自动获取系统token
      kefuId:"',//默认为空自动对接客服,可填写指定客服ID
      isShowTip: true, // 初始化成功后, 界面右下角会自动创建“联系客服按钮”, 如无需默认展示, 则填写false即可,默认为true
      mobileIcon: "", // 手机端悬浮客服图片
      pcIcon: "", // pc端悬浮客服图片
      windowStyle:'center',//默认空 右下角小弹窗, center 普通中间弹窗样式
      "version":"v4"
    };
    var canCustomerServer = new initCustomerServer(option);
    canCustomerServer.init();
  }
}
document.head.appendChild(_s)
})();
</script>
```

第二步 请进入自己客服系统后台复制代码，添加到您网址页面内

效果图

q.crmeb.net/?categoryid=all&sequence=0

新标签页 百度 工作 学习 个人 在线工具箱 公司服务器 其他书签 阅读清单

crmeb 搜索

首页 消息 发现 聆听

全部	226156
上班摸鱼	92
官方动态	96
Pro系统	700
多商户版	444
Java系统	59
知识付费	150
单商户V4	604
单商户V3	3189
CRMChat	1
合作与交流	673
会员下载	101
模版专区	47

所有 精华 已关注 类型 排序 发布

置顶 开源下载地址

置顶 小程序获取用户信息接口调整, 修复包汇总

置顶 CRMEB系统二开及安装配置视频教程

置顶 CRMEB服务市场限时免费入驻!

**CRMEB-吴汐** 5小时前 41

**【v4.3.0】虚拟商品可以使用线下支付的问题修复**

文件: app/services/order/StoreOrderServices.php

```
1537 $data['offline_pay_status'] = (int)sys_config( name: 'offline_pay_status' )
1540 $data['ywe_pay_status'] = (int)sys_config( name: 'balance_func_status' ) &&
1541 $data['pay_weixin_open'] = (int)sys_config( name: 'pay_weixin_open' ) ?? 0;
1542 $data['store_self_mention'] = (int)sys_config( name: 'store_self_mention' );
1543 if (isset($validCartInfo[0]['productInfo']['is_virtual']) && $validCartInfo[0]['productInfo']['is_virtual'] == 1) {
```

查看更多

赞 评论 分享

**CRMEB-ulo** 5小时前 16

**【标准版4.3】发货无法打印电子面单问题**

查看更多

**推荐内容**

- 1 多商户常见问题解决方案整理 589 精华
- 2 CRMEB知识付费系统演示(图片) 1494 精华
- 3 CRMEB3.x常见问题汇总【安装前必看...】 33331 精华
- 4 CRMEB知识付费系统v1.7【更新计划】 929 精华
- 5 【v4.2】CRMEB单商户v4.x版本常见... 1673 精华

换一批

扫一扫访问移动端 CRMEB社区

Powered By Discuz! Q Copyright © 2021 CRMEB社区 陕ICP备14011498号-3

在线咨询

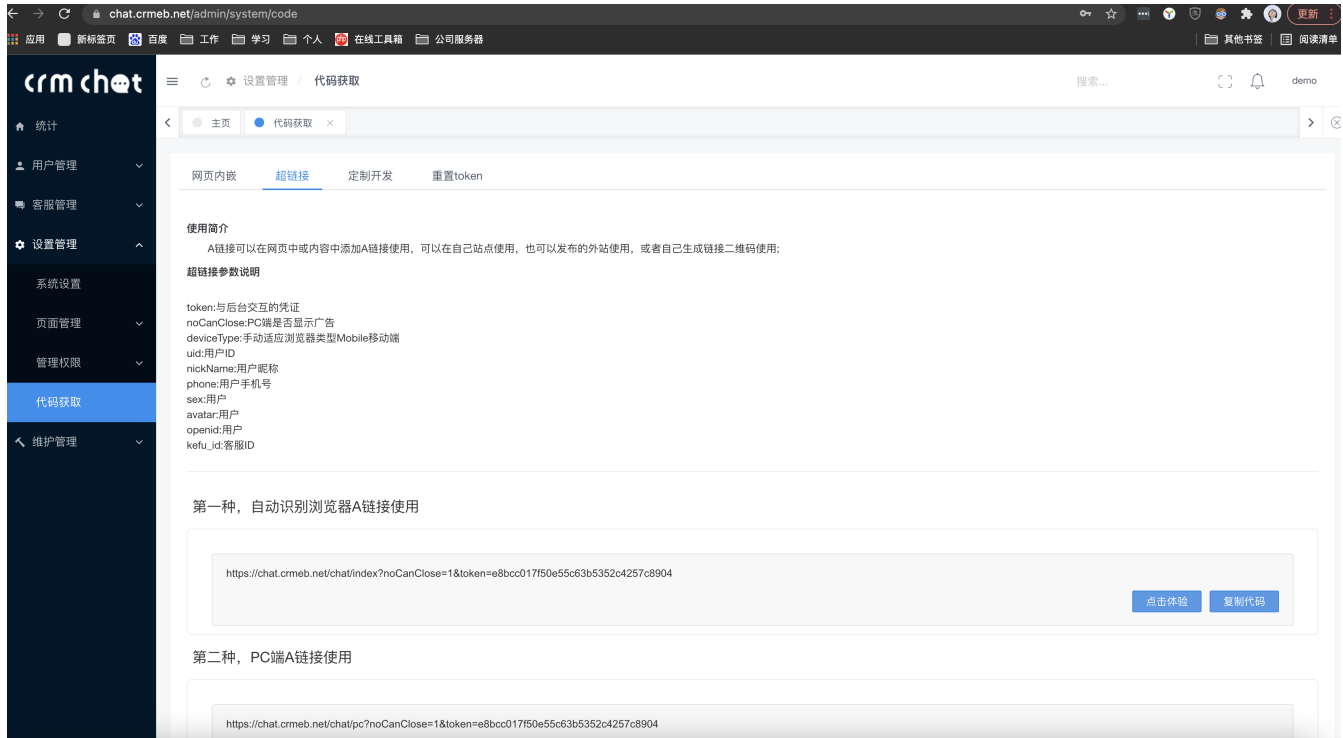


移动版效果图

更多效果查看：<https://chat.crmeb.net/demo/index.html>

## 灵活的超链接使用

### 获取超链接



### 使用场景

站外广告 站外新闻 站外博客 站内广告 站内链接 app中使用 小程序中使用 (使用说明：<https://developers.weixin.qq.com/miniprogram/dev/component/web-view.html>)

### 超链接参数说明

token:与后台交互的凭证  
noCanClose:PC端是否显示广告  
deviceType:手动适应浏览器类型Mobile移动端  
uid:用户ID  
nickName:用户昵称  
phone:用户手机号  
sex:用户性别  
avatar:用户头像  
openid:用户第三方授权的openid  
kefu\_id:客服ID

示例：<https://chat.crmeb.net/chat/index?noCanClose=1&token=e8bcc017f50e55c63b5352c4257c8904>

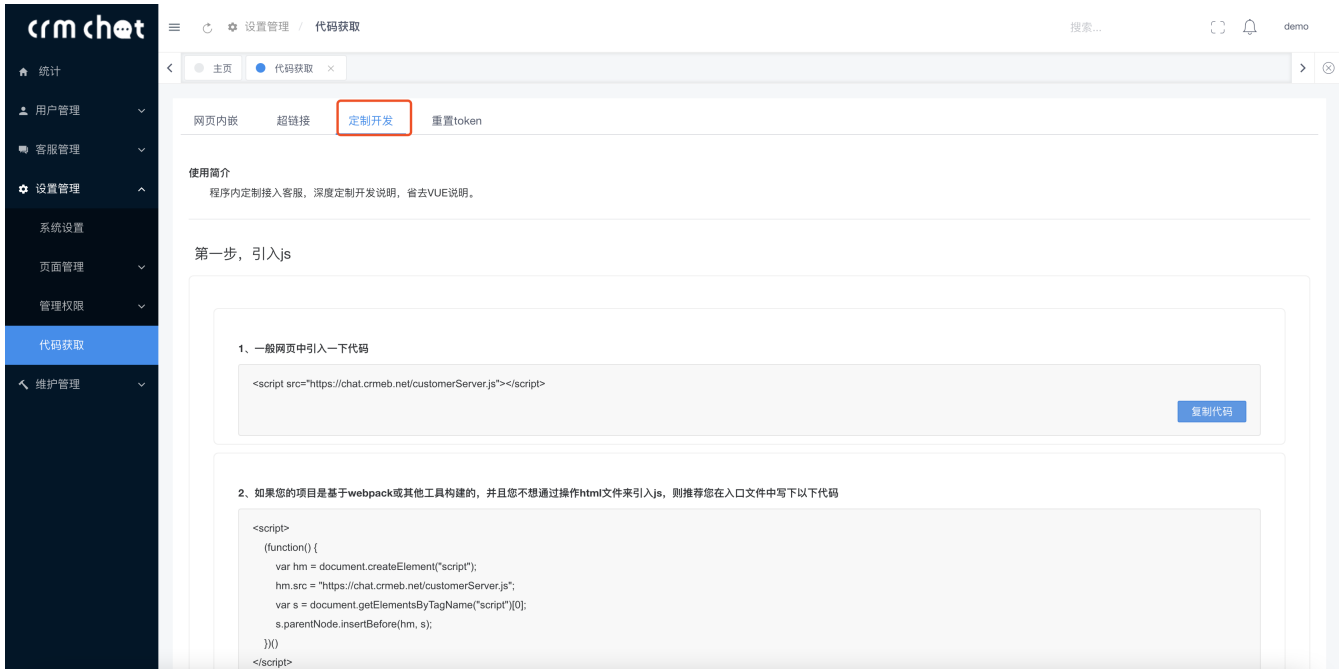
指定客服 [https://chat.crmeb.net/chat/index?noCanClose=1&token=e8bcc017f50e55c63b5352c4257c8904&kefu\\_id=1](https://chat.crmeb.net/chat/index?noCanClose=1&token=e8bcc017f50e55c63b5352c4257c8904&kefu_id=1)

[info]提示：token参数必须带的，后面新添加参数'&'隔开

例如要传用户昵称，用户ID <https://chat.crmeb.net/chat/index?noCanClose=1&token=e8bcc017f50e55c63b5352c4257c8904&uid=1&nickName=zhangsan>

## 深入对接客服

### 第一步引入js



#### 1、常规引用js方法

```
<script src="https://chat.crmeb.net/customerServer.js"></script>
```

#### 2、如果您的项目是基于webpack或其他工具构建的，并且您不想通过操作html文件来引入js，则推荐您在入口文件中写下以下代码

```
<script>
(function() {
  var hm = document.createElement("script");
  hm.src = "https://chat.crmeb.net/customerServer.js";
  var s = document.getElementsByTagName("script")[0];
  s.parentNode.insertBefore(hm, s);
})();
</script>
```

#### 调用客服功能

```
var option = {
  openUrl: "https://chat.crmeb.net", // 打开客服聊天框的地址，即：部署后台管理系统的地址，若未填写，则自动获取当前服务器的地址
  token: e8bcc017f50e55c63b5352c4257c8904, // token,与后台交互的凭证
  kefuId: "", // 默认为空自动对接客服，可填写指定客服ID
  isShowTip: true, // 初始化成功后，界面右下角会自动创建“联系客服按钮”，如无默认展示，则填写false即可，默认为true
  mobileIcon: "", // 手机端悬浮客服图片
  pcIcon: "", // pc端悬浮客服图片
  windowStyle: 'center', // 默认空 右下角小弹窗， center 普通中间弹窗样式
  domId: 'customerServerTip', // 展示在页面右下角联系客服的dom的id，可根据id获取到dom后自行修改样式，默认为customerServerTip
  insertDomNode: '.getCode_container', // SPA应用必填，html文件单独引入选项，表示插入客服弹窗的 dom节点，一般为当前界面的根节点，默认为body
  //设置客户信息
  sendUserData: {
    uid: '1', // 用户id
    nickName: "", // 用户昵称
    phone: "", // 用户联系方式
    sex: '1', // 用户性别
    avatar: "", // 用户头像 URL地址
  }
}
```



```

    openid: '//微信openid
  },
  //设置默认打开推送产品
  productInfo: {
    store_name: '蒙奇 D 路飞',
    stock: '库存',
    sales: '122', // 销量
    ficti: '10', // 赠送
    price: '100',
    image: '//产品图片
    proUrl:'//产品链接
  }
};

var canCustomerServer = new initCustomerServer(option);
//实例化加载客服组建
canCustomerServer.init();

//样式设置说明
canCustomerServer.setStyleOfCustomerServer(this.canCustomerServer.connentServerDom,{
  bottom:'300px',
  left:'50%'
});
// 调用打开客服弹窗的方法，如果isShowTip为false，就使用这个函数，当然也可以使用A链接
canCustomerServer.getCustomServer();

```

[info] 温馨提示，请到系统后台复制对应代码，以上代码请求地址和token请更换自己系统中的 例如公众号商城内深度对接客服示例：1、页面引入js

```
<script src="https://chat.crmeb.net/customerServer.js"></script>
```

## 2、参数负值

```

<script>
var option = {
  openUrl: "https://chat.crmeb.net", // 打开客服聊天框的地址，即：部署后台管理系统的地址，若未填写，则自动获取当前服务器的地址
  token: e8bcc017f50e55c63b5352c4257c8904, // token,与后台交互的凭证
  kefuId:"',//默认为空自动对接客服，可填写指定客服ID
  isShowTip: true, // 初始化成功后，界面右下角会自动创建“联系客服按钮”，如无需默认展示，则填写false即可,默认为true
  mobileIcon: "", // 手机端悬浮客服图片
  pcIcon: "", // pc端悬浮客服图片
  windowStyle:'center',//默认空 右下角小弹窗， center 普通中间弹窗样式
  //设置客户信息
  sendUserData: {
    uid: '1', // 用户id
    nickName: '聆听', // 用户昵称
    phone: '13688888888', // 用户联系方式
    sex: '1', // 用户性别
    avatar: 'https://thirdwx.qlogo.cn/mmpopen/vi_32/Q0j4TwGtftKia5uKc8B0wxcPUUykQ5qXiagQG75PVnIpb7wT1HoFibrNMbKniaZM1TLcPziAxHE
    openid: 'oRElr5M4KP_05PXWS9Nis2USNEvU'//微信openid
  }
};
//初始化
var canCustomerServer = new initCustomerServer(option);
//实例化加载客服组建
canCustomerServer.init();
//打开客服聊天框
canCustomerServer.getCustomServer();
</script>

```

## APP打包

### APP打包和调试

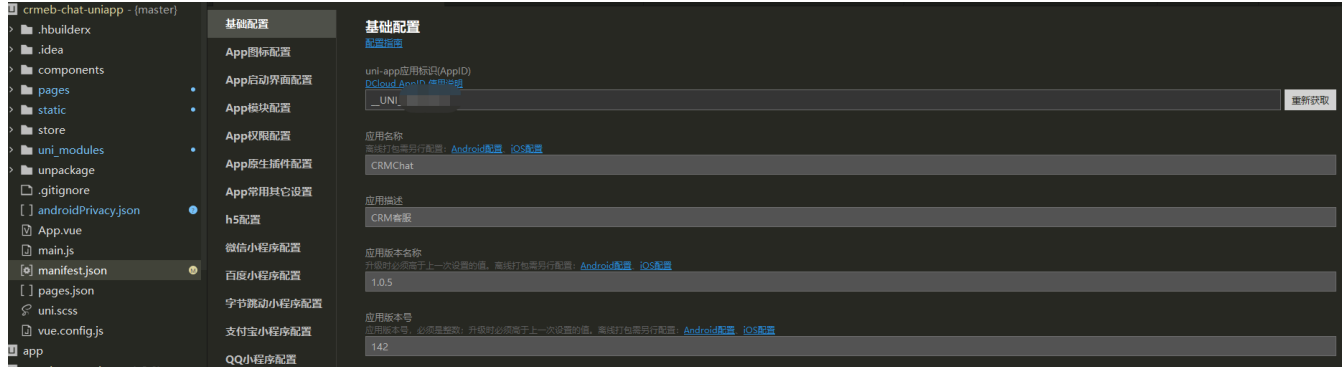
此项目由uniapp框架编写，需要使用Hbuilder X编译器进行运行和发布，支持浏览器调试，支持发行为APP和H5，其他发行方式不支持。需要Hubilder X安装less插件、sass插件、stylus编译。运行时请先安装插件。

百度网盘内APP源码存放位置：view/app 目录下

### 配置

#### 一、配置APP应用ID

打开编译器点击 manifest.json，需要重新获取应用标识。这一步是必须的，其他选项根据自身需求更换名称。



#### 二、配置APP图标

打开编译器点击 manifest.json，点击浏览选择一张 1024\*1024 png格式的图片，然后点击自动生成所有图标并替换会自动生成图标



#### 三、APP模块配置

打开编译器点击 manifest.json，点击左侧APP模块配置，找到Push(消息推送)并勾选，其他模块暂时不用配置

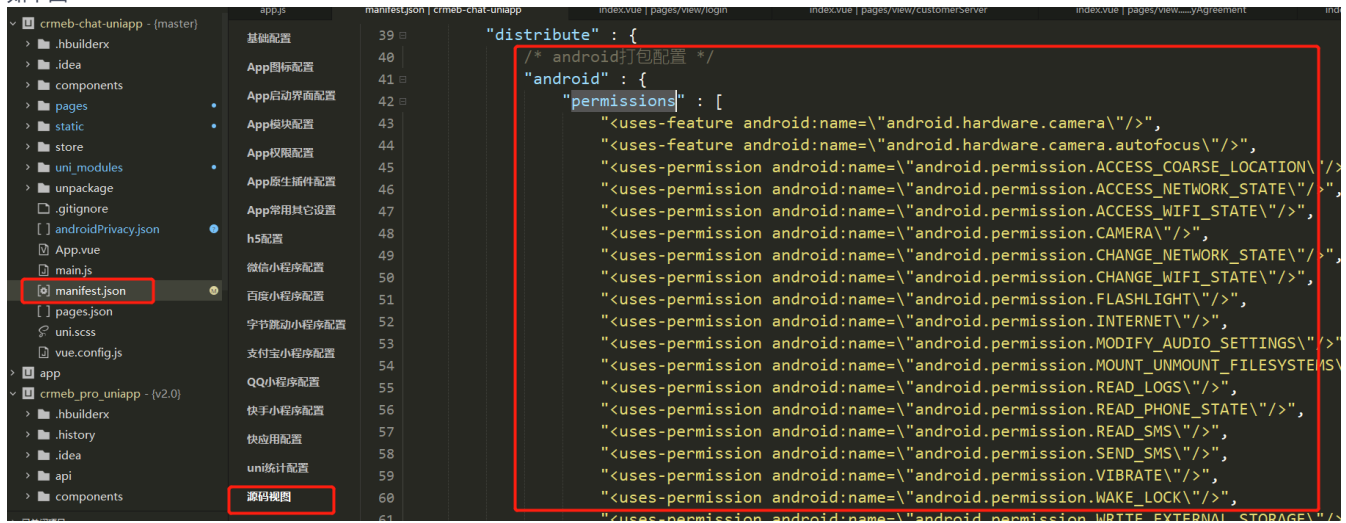


#### 四、APP权限配置

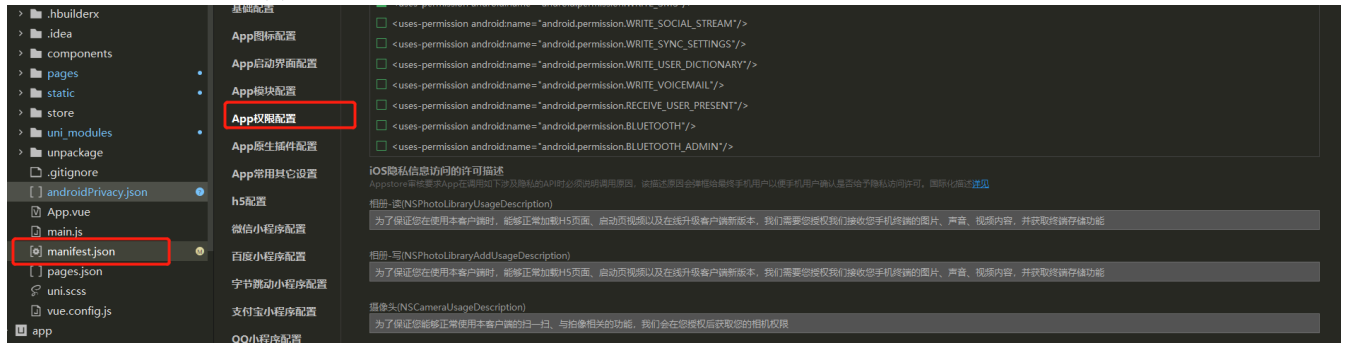
打开编译器点击 manifest.json，需要增加如下权限。点击源码视图，搜索：'android打包配置'粘粘在 permissions 对象里面

```
[  
  "<uses-feature android:name=\"android.hardware.camera\"/>\",  
  "<uses-feature android:name=\"android.hardware.camera.autofocus\"/>\",  
  "<uses-permission android:name=\"android.permission.ACCESS_COARSE_LOCATION\"/>\",  
  "<uses-permission android:name=\"android.permission.ACCESS_NETWORK_STATE\"/>\",  
  "<uses-permission android:name=\"android.permission.ACCESS_WIFI_STATE\"/>\",  
  "<uses-permission android:name=\"android.permission.CAMERA\"/>\",  
  "<uses-permission android:name=\"android.permission.CHANGE_NETWORK_STATE\"/>\",  
  "<uses-permission android:name=\"android.permission.CHANGE_WIFI_STATE\"/>\",  
  "<uses-permission android:name=\"android.permission.FLASHLIGHT\"/>\",  
  "<uses-permission android:name=\"android.permission.INTERNET\"/>\",  
  "<uses-permission android:name=\"android.permission.MODIFY_AUDIO_SETTINGS\"/>\",  
  "<uses-permission android:name=\"android.permission.MOUNT_UNMOUNT_FILESYSTEMS\"/>\",  
  "<uses-permission android:name=\"android.permission.READ_LOGS\"/>\",  
  "<uses-permission android:name=\"android.permission.READ_PHONE_STATE\"/>\",  
  "<uses-permission android:name=\"android.permission.READ_SMS\"/>\",  
  "<uses-permission android:name=\"android.permission.SEND_SMS\"/>\",  
  "<uses-permission android:name=\"android.permission.VIBRATE\"/>\",  
  "<uses-permission android:name=\"android.permission.WAKE_LOCK\"/>\",  
  "<uses-permission android:name=\"android.permission.WRITE_EXTERNAL_STORAGE\"/>\",  
  "<uses-permission android:name=\"android.permission.WRITE_SETTINGS\"/>\",  
  "<uses-permission android:name=\"android.permission.WRITE_SMS\"/>\"  
]
```

如下图



IOS隐私信息访问描述配置，点击左侧APP权限配置找到IOS隐私信息访问描述，需要配置说明：相册-读、相册-写、摄像头、访问本地网络



## 调试

调试可在浏览器上调试或者真机调试，真机调试需要使用数据线连接Hbuilder X编译器，安卓需要打开 USB调试模式

### 一、浏览器调试



### 设置请求域名

https:// chat.crmeb.net

取消

确定

## 打包

### 一、申请打包证书

- 安卓打包证书生成文档：<https://ask.dcloud.net.cn/article/35777>
- IOS打包证书生成文档：<https://ask.dcloud.net.cn/article/152>

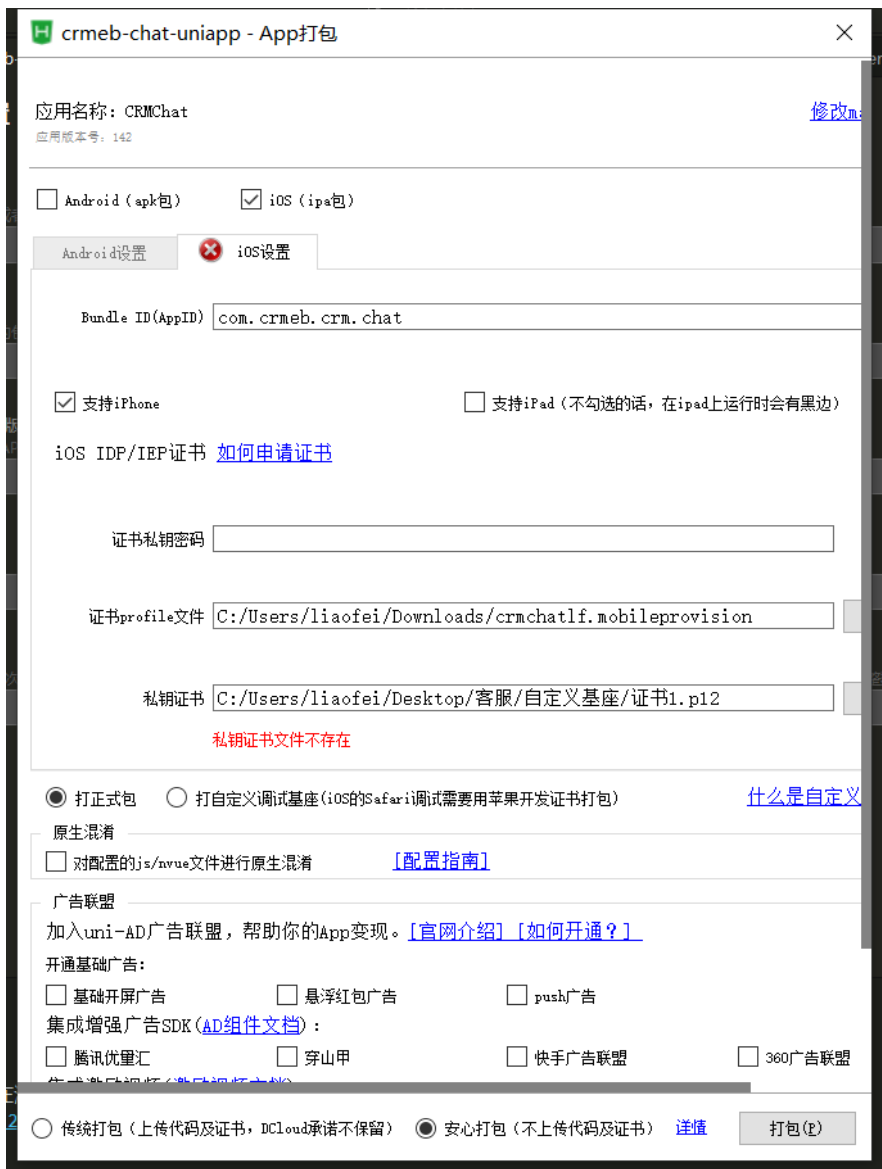
### 二、安卓打包

打开编译器，点击菜单上发发行->原生APP-云打包，输入申请的证书，勾选打正式包、安心打包。点击打包等待服务器打包完成。



### 三、IOS打包

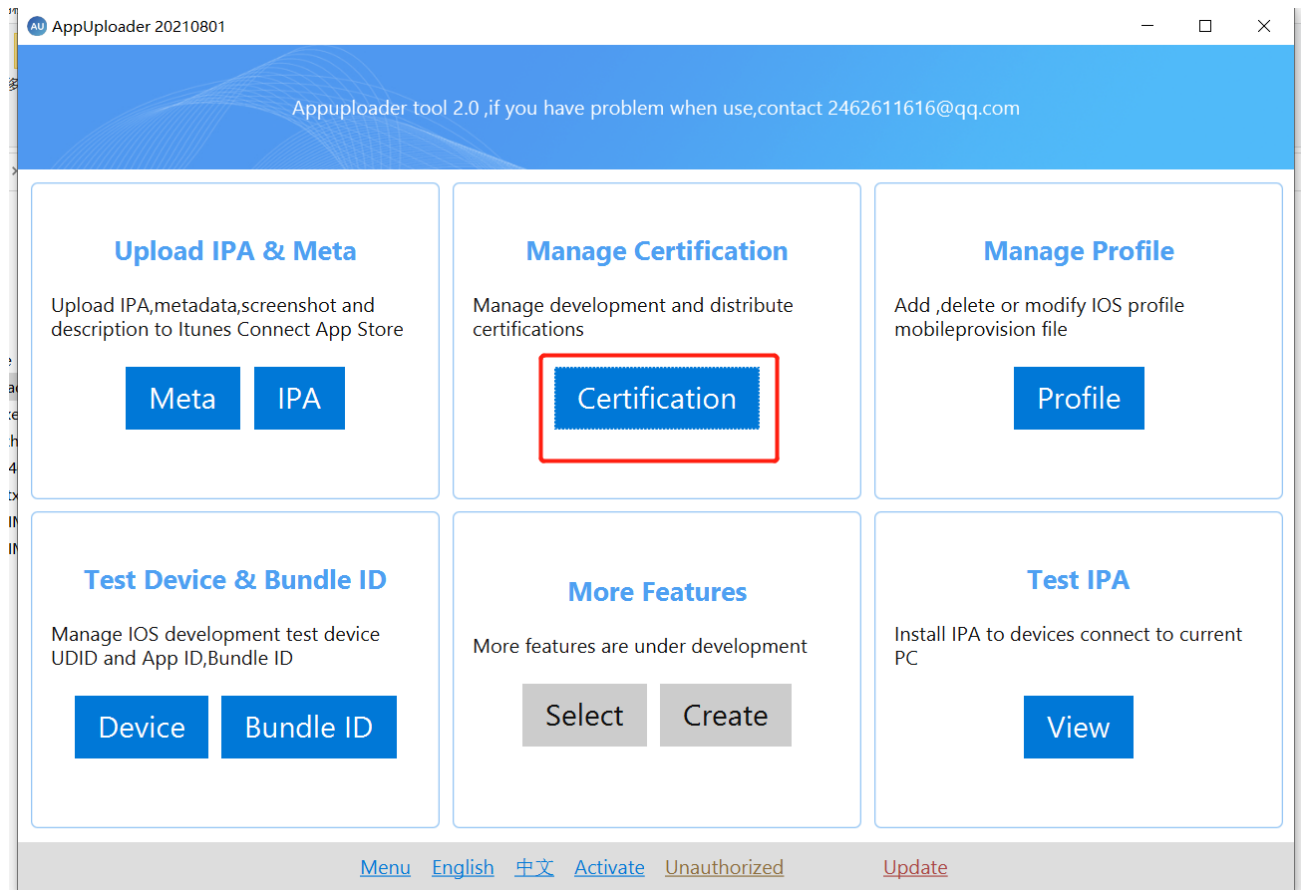
打开编译器, 点击菜单上发发行->原生APP-云打包, 输入申请的证书, 勾选打正式包、安心打包。点击打包等待服务器打包完成。打包方式和安卓差



不多一样

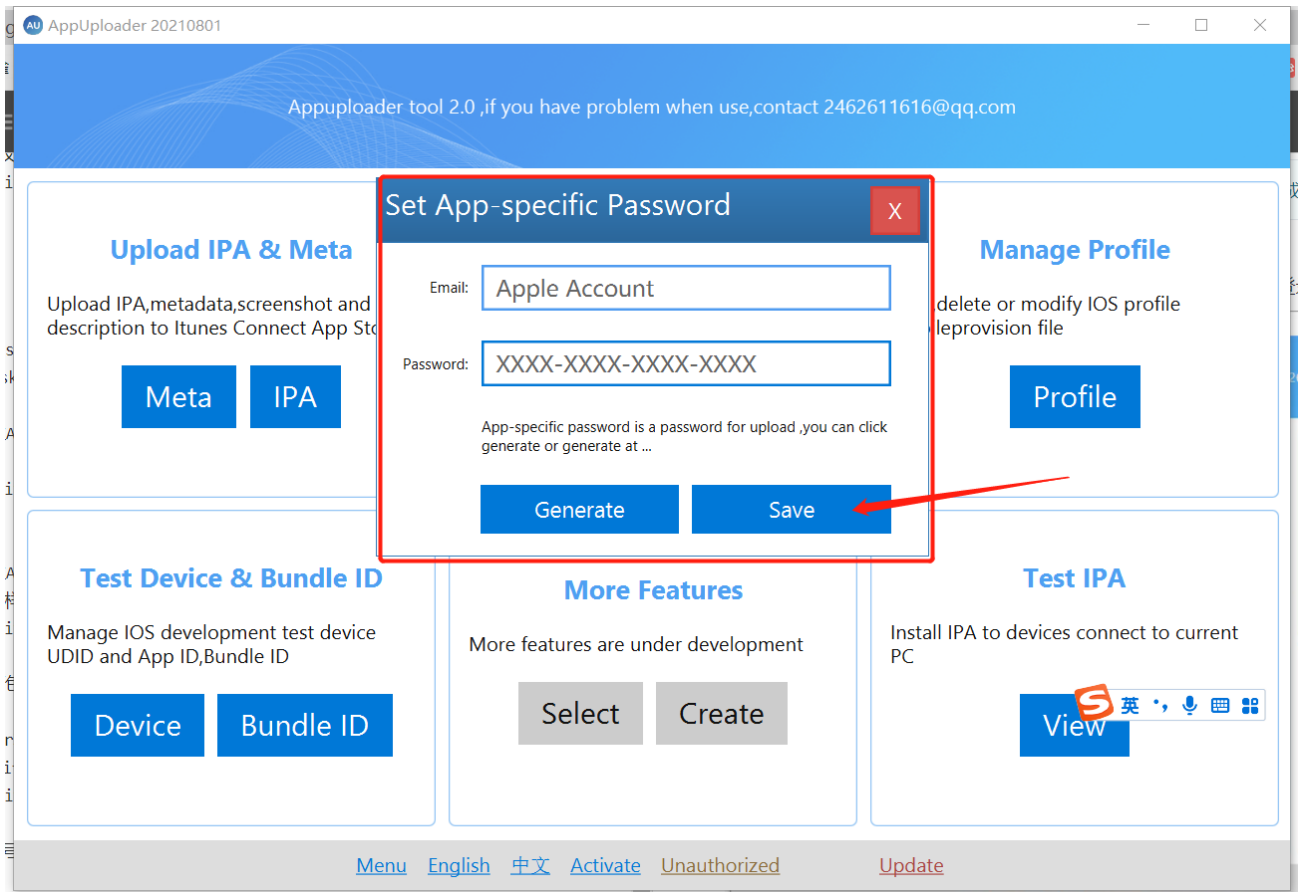
四、上传IOS应用，实际上ios打包和安卓没有太大区别，只要证书申请下来，打包也不是什么问题，至于打包后怎么上传IOS应用，请看以下说明

1. windows用户可以下载appuploader软件[appuploader官网](#)
2. 使用appuploader软件，点击certification弹出登录页面进行登录



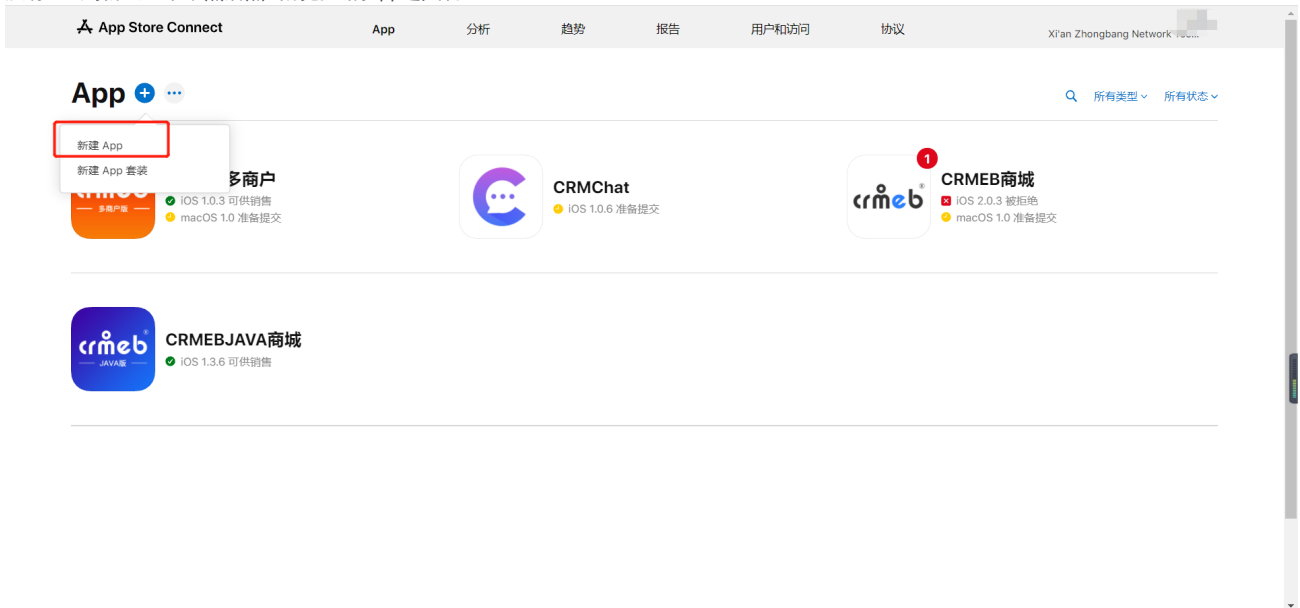
3. 点击IPA，输入apple账号email账号，输入app专用密码，如果没有请查看：[apple专用密码说明](#)。点击save保存专用密码。后进入上传APP界面，选择下载好的IPA文件上传



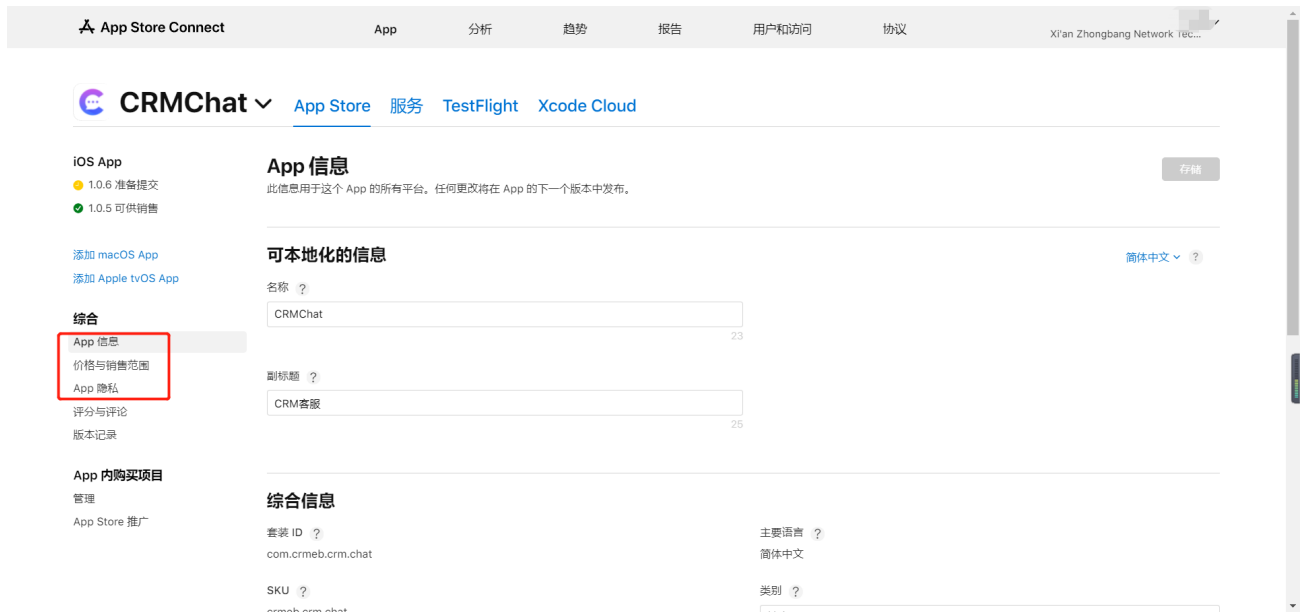


### 五、IOS应用发布

1. 登录APPLE应用管理后台<https://appstoreconnect.apple.com/login>
2. 没有APP则新建一个，然后点击刚创建的app进入管理



3. 需要填写综合项内的APP信息、价格与销售范围、APP隐私。APP隐私需要填写一个隐私政策地址，隐私内容如：<https://chat.crmeb.net/chat.html>。实际情况下可按照此隐私协议修改为自己的协议



4. 准备提交版本，需要填写APP预览和截图（iphone6.5和5.5显示屏至少两个）、关键字（根据自身填写）、描述（根据自身填写）、构建版本（这里的构建版本就是在刚才上传的ipa文件会展示到当前界面的TestFlight菜单里面，管理员可下载TestFlight应用进行测试）、App审核信息（利于过审）、版本发布（选择自动发布此版本）。填写好后点击存储，然后再点击提交以供审核。ios的一个版本就等待审核了。

## 消息推送

# APP消息推送

客服消息推送使用的是uniPush集成各家手机推送及个推，是个推的升级版。个推使用文档：<https://docs.getui.com/getui/start/product/>。

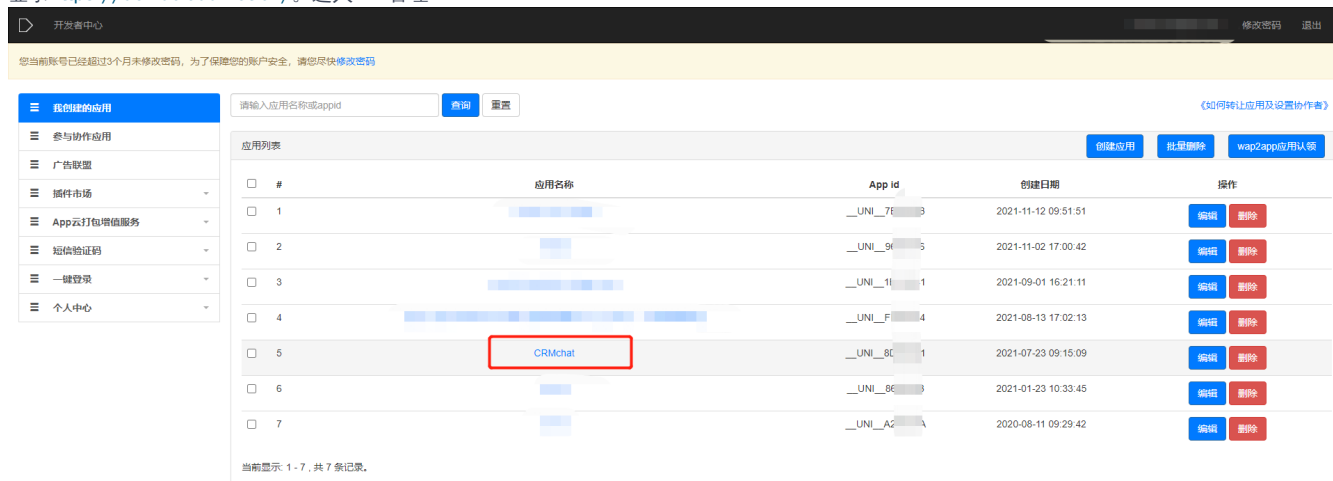
## 流程

1. 申请开通uniPush
2. 配置UNI Push应用设置
3. 复制uni Push内的配置信息到客服后台
4. 上传IOS推送p12证书
5. 安卓应用需要去各大厂商内商家APP然后再配置厂商推送配置
6. app离线推送和在线推送配置完成

## 开始配置

### 一、开通uniPush

登录<https://dev.dcloud.net.cn/>。进入APP管理



找到uni Push一栏，点击修改应用信息。填写好应用签名、包名、ios bundleid。点击开通。

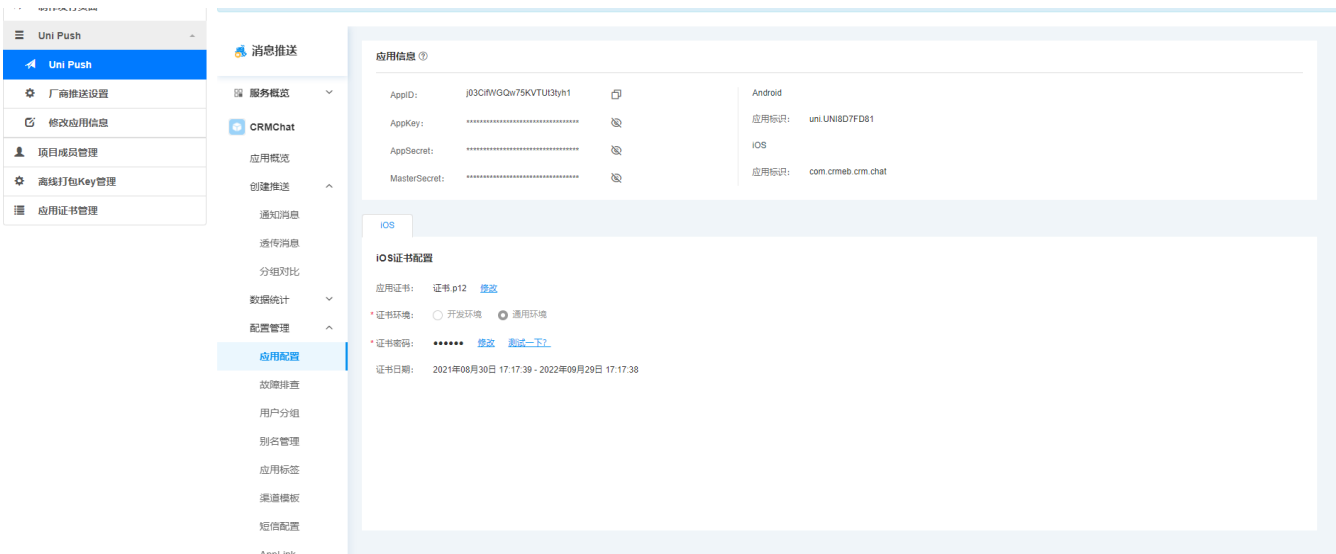


## 二、配置uniPush

开通uniPush后如下页面

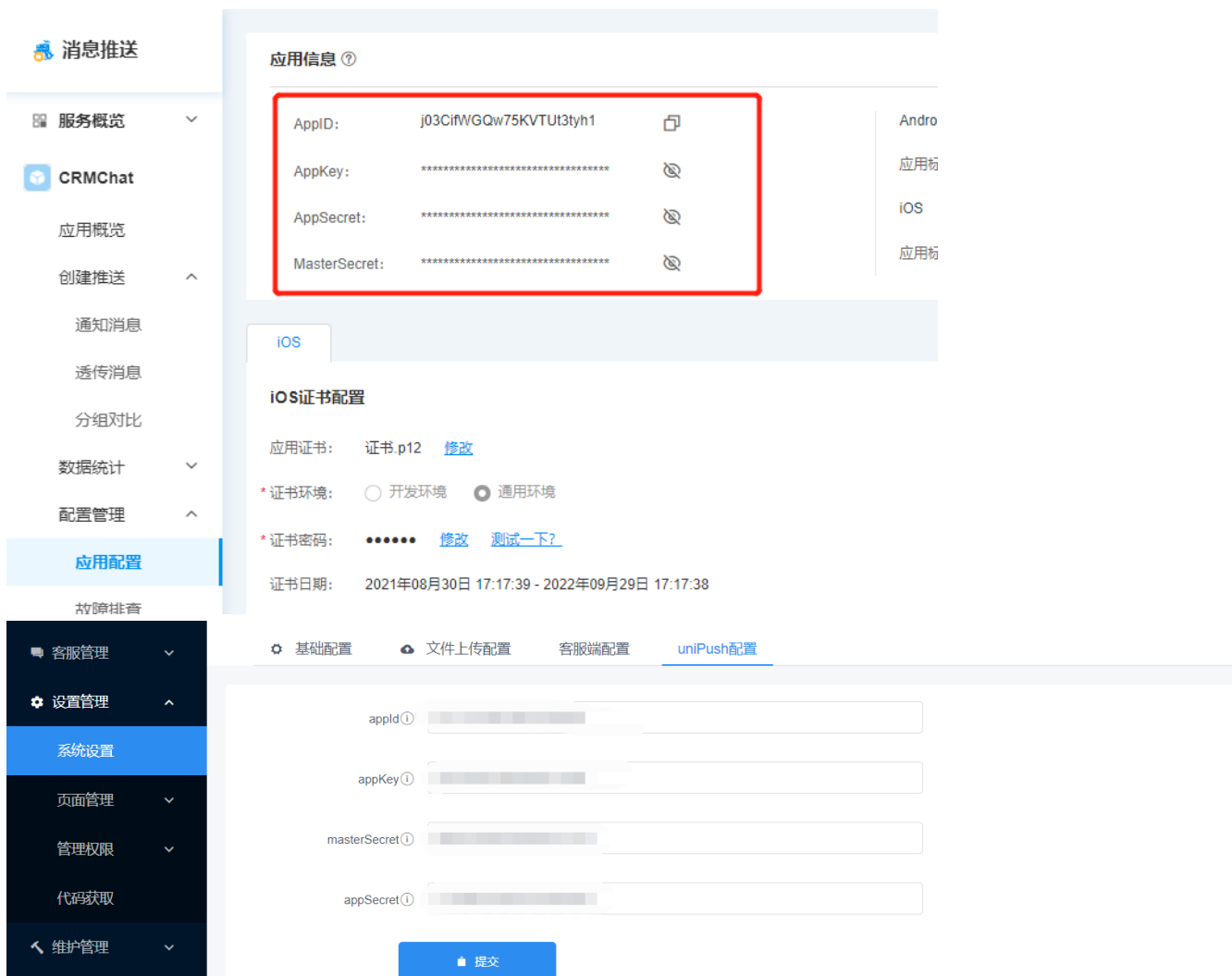


点击左侧的配置管理->应用配置



首先需要配置IOS的推送消息证书, 申请文档: <https://docs.getui.com/getui/mobile/ios/apns/>。这里的p12证书和打包的p12证书有点不同。申请下来后, 在上图中上传。

复制应用配置中的应用信息, 填写到客服后台的设置管理->系统设置->uniPush配置一栏



到此时，iOS的在线和离线推送全部配置完成。安卓的在线推送完成

### 三、安卓厂商推送

安卓厂商就是一个大杂烩，好多个大平台。需要每个平台内进行申请上架上架后再次申请消息推送就能使用，这里整理出几个大厂商的平台地址可供上架

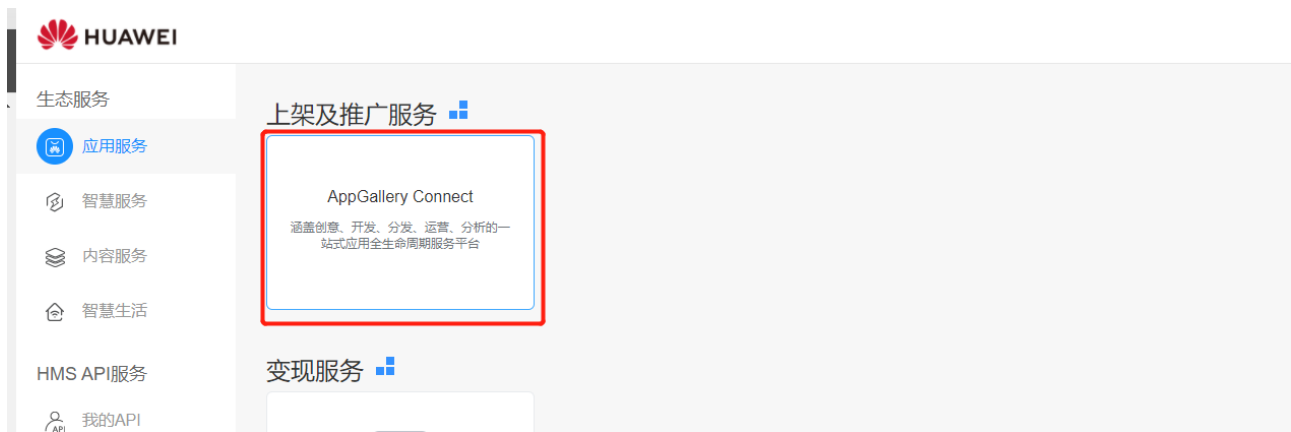
- 魅族
- OPPO
- 华为
- 小米
- VIVO
- google

除华为外，其他申请推送消息还算简单，这里主要说明华为上架APP和消息推送开通配置（华为审核能通过其他厂商也不是问题的）

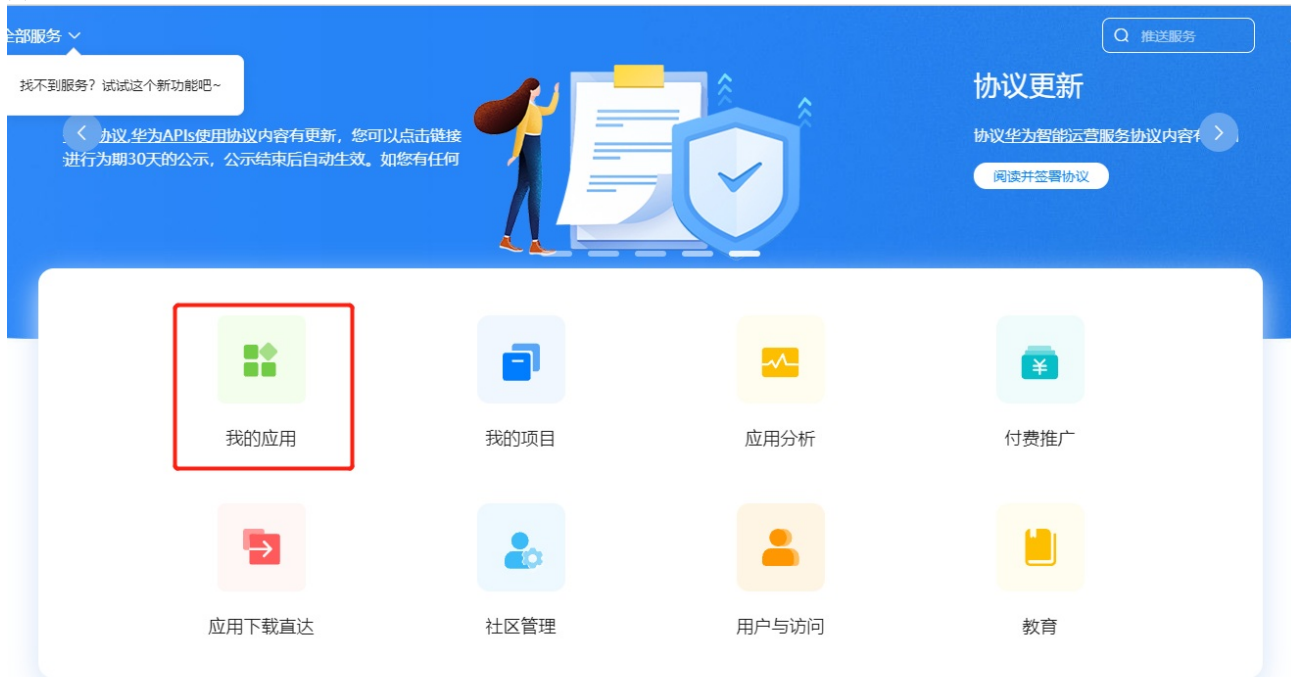
1. 登录进华为开发者管理平台点击管理中心



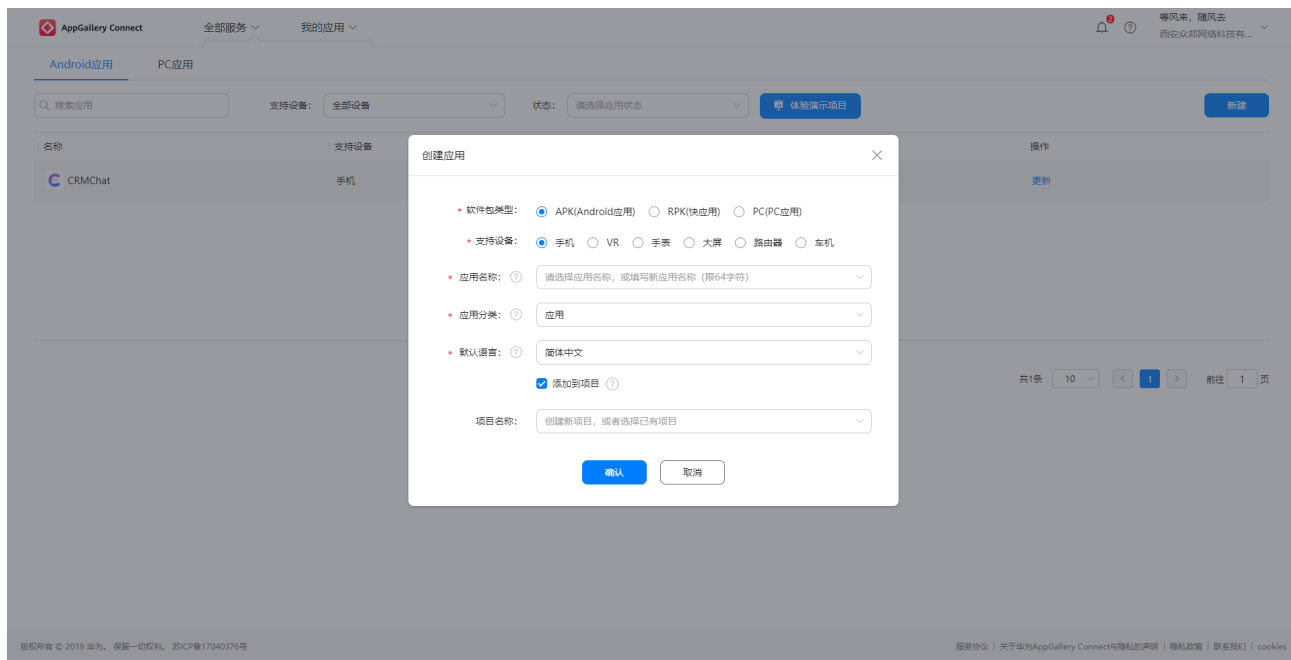
2. 点击应用服务->上架及推广服务



3. 点击我的应用



- 4. 进入我的应用列表后点击新建。添加应用名称、应用分类（选择应用）、默认语言（选择简体中文），一定要勾选添加到项目里面，然后输入项目名称（华为在创建应用后没有添加到项目，然后再去添加项目，关联不上当前的应用应该是华为平台的bug导致的，如果遇到就找客服人工处理下）



5. 点击列表上的应用，进入管理应用界面。需要填写：AppGallery华为应用市场(选择中国大陆)、是否开放式测试版本(根据自身选择)、软件版本(选择打包好的安卓app上传上去)、隐私政策网址(填写协议地址和ios的隐私协议一个地址就行)、应用版权证书或代理证书(软著和免责声明都需要)。填写好后点击提交审核。



6. 开通消息推送，点击我的项目，进入在第4步创建的项目



7. 查看应用信息中相应的华为 AppID、华为 SecretKey。该信息在之后步骤中将会使用，如下图



8. 在“项目设置 > 增长”中选择“推送服务”，点击“立即开通”，以此来开启华为侧推送服务状态





- 9. 在“项目设置 > 常规”中填写“SHA256证书指纹”，点击右侧对勾进行保存；若不知道具体值，请参考 [SHA256指纹证书获取](#)。
- 10. 在“项目设置 > 推送服务> 配置”中开通项目回执状态，配置个推侧的回调地址以及HTTPS证书。如下



回执名称可自定义 回调地址(杭州机房)

`https://thirdrcp-hz.getui.com/hw`

HTTPS证书

```
-----BEGIN CERTIFICATE-----
MIIGrDCCBZSgAwIBAgIQB0D9NX2MGYX7U16uPL4ZFtANBgkqhkiG9w0BAQsFADBF
MQswCQYDVQQGEWJVUzEVMBMGA1UEChMUMRGInaUNcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNIcnQuY29tMR4wHAYDVQQDEXVHZW9UcnVzdCBDTIBSU0EgQ0EgRzEw
HhcNMjEwNjA4MDAwMDAwWWhcNMjEwNjA4MDAwMDAwMjEwNjA4MDAwMDAwMjEw
MBAGA1UECAw5rWZ5rGf55yBMRlW EAYDVQQHDA nmna3lt57lullxjzAIBgNVBAoM
HuaVj+aXpeS6kuWKqOicOeS7veaciemZKOWFrOWPuDEUMBIGAlUEAwwLKi5nZXR1
aS5jb20wggeiMa0GCSqGSIB3DQEBAAQAA4IBDwAwggEKAoIBAQC1z29MoeLBrXa
v+uIHn7MunAmD3t3nKj1STyO+I+dmJlk/iz64Ussgwr2oyT7fy/LgzA217n2OQi
0sdA8TUTGEFvRV2ZEndjeWVvN2nb8mXPD/LfpwjIVXMu8b6hvwPbVvDi2XLPa01
```

```

uqxrd7luWGxqWbP9efbTUtARVG7wktlrBkoZ5J4aNT/XYHlvUbBsuq/RUCmwwxzm
vvwzvlll18c5CjZj2rBLwxdwP/pU3MHGyiQna0v8dRI9ME6G2QdzHPNfchhkTdB
M5/RVbfEIAnMFYtmsWI3qyef4HpxoTEFdUpIjyHu+Q52UC4EHMUDVyltKMpmpWT7
itRDaj/RAgMBAAGjggNNMIIDSTAFBgNVHSMGDAWgBSRn14x4Qn61gWffBzKpl
NC8MjAdBgNVHQ4EFgQub74ZxaQHDCGOajjVBnbCTYPHVkklQYDVR0RBBowGILL
Ki5nZXR1aS5jb22CCWdlHVPmNvbTAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0IBBYw
FAYIKwYBBQUHAWEGCCsGAQUFBwMCMHUGA1UdHwRuMGwwNKAyoDCGLmh0dHA6Ly9j
cmwzLmRpZ2ljZXJ0LmNvbS9HZW9UcnVzdENOUINBQ0FHMS5jcmwwNKAyoDCGLmh0
dHA6Ly9jcmw0LmRpZ2ljZXJ0LmNvbS9HZW9UcnVzdENOUINBQ0FHMS5jcmwwPgYD
VR0gBDCwNTAzBgZngQwBAglwKTAnBggrBgEFBQcCARYbaHR0cDovL3d3dy5kaWdp
Y2VydC5jb20vQ1BTMG8GCCsGAQUFBwEBBGMwYTAhBggrBgEFBQcwAYYVaHR0cDov
L29jc3AuZGNvY3NwLmNuMDwGCCsGAQUFBzAChjBodHRwOi8vY3JsLmRpZ2ljZXJ0
LWNuLmNvbS9HZW9UcnVzdENOUINBQ0FHMS5jcnQwDAYDVR0TAQH/BAIwADCCAX0G
CisGAQQB1nkCBAlEggFTBIIlBaQFnAHUAKXm+8J45OSHwVnOFy6V35b5XFzXgCvj5
TV0mXCVdx4QAAAF56aZtUAAABAMARjBEAiACZx790zhQCizmmN7zbWlMczAPT+p
Aof3L+Ew+g13SgIgm8fG372a/uyMK8QBGJoCbqYP6EA70ELD+yJy4WRUwDAAdQAI
RUUHVWUkVpY/oS/x922G4CMmY63A539dxoNcbuIPAgAAAXnppm0RAAAEAwBGMEQC
IDO+rFNYB6bfaK9zS7AoBM6ZNPx++WA13JcxaH03ssIRAIaam8XaEsCUOT7TNe/8
LjqMNsUFL9iAxQabtNvlhldbvqB3AFGjsPX9AXmcVm24N3iPDKR6zBsny/eeiEka
Df7UiwXIAAABeemmbW4AAQAEGwRglhAPWyMK37IqLxclln51T/fg1B/r9sZv4w
TpC8xvsl3nExAiEAyAylap5eITHOqXPF+E5gzyHkgOvF5zrjL6DdERXVG4wDQYJ
KoZihvcNAQELBQADggEBAG6W8Hn57NM3T/yqy+6gtEfkS5OEXbprMWeZc9VGbne
ULDk7KRaARDWT7TZEF3jbl0VOGc9sXYkizDbQYXM81eee5utGCaFESroWxZnVNgp
QVSe99iLmRzb01dB65GatedsG6wqL5hxSt+Kh9v00PlpK08/NGNQ9nKXdx6xUBFUy
//6XgBcEPIqDB6Pj4fWRTg/F9vLlpkcP41zYbj9MlyEBxSecnyIkaekJQC0KgY
xirrh8fUHM/5vc8f5Q6FU0AQAOZyNvr/dxRmjJCVcdiF35AUDzE5xSf8cOYQnYHo
tNhrH4+e8mXBSQANTLxGJf7E7EZTMSu4NLoa5xS65DQ=
-----END CERTIFICATE-----

```

点击“提交”，提示“成功”则表示配置成功

11. 进入<https://dev.dcloud.net.cn/>点击应用->uni push->厂商推送设置



四、其他厂商推送开通指南

个推厂商应用开通指南

## UNIAPP调试

## UNIAPP调试

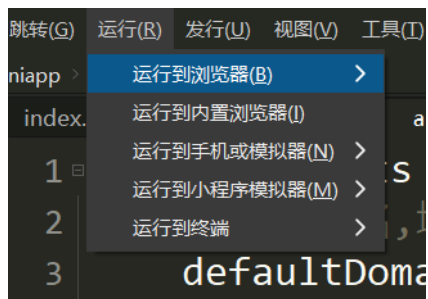
## 配置

在调试前可以选择性配置默认域名，配置文件地址：`/template/uniapp/pages/config/app.js`

```
module.exports = {
  //默认域名,域名格式例如:chat.crmeb.net
  defaultDomainName: '',
  //默认请求方式,可选值: https|http
  defaultRequestType: 'https',
  //默认ws链接方式,可选值: wss|ws
  defaultWsType: 'wss',
  //是否打开自定义配置域名,可选值:true|false;关闭后必须要配置默认域名(defaultDomainName)选项
  isDomainName: true,
}
```

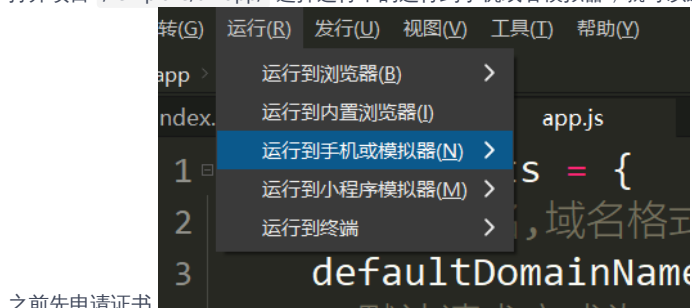
## H5调试

打开项目 `/template/uniapp/` 选择运行中的运行到浏览器，就可以进行调试



## 真机调试

打开项目 `/template/uniapp/` 选择运行中的运行到手机或者模拟器，就可以进行调试 手机调试可选自定义基座运行，需要先制作自定义基座，制作



之前先申请证书

# 功能介绍

## 接口文档

### chat

v1.0.0

## 总后台/公用接口(无需授权)

### GET 获取验证码

GET /api/admin/captcha\_pro

返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

#### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» img	string	true	none	验证码图片base64
»» key	string	true	none	验证码key

### POST 登录

POST /api/admin/login

Body 请求参数

```
type: object
properties:
  account:
    type: string
    description: 账号
  pwd:
    type: string
    description: 密码
  imgcode:
    type: string
    description: 验证码
  key:
    type: string
    description: 验证码key
required:
  - account
  - pwd
  - imgcode
  - key
```

请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» account	body	string	true	账号
» pwd	body	string	true	密码
» imgcode	body	string	true	验证码
» key	body	string	true	验证码key

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

GET 获取登录信息

GET /api/admin/login/info

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» slide	[string]	true	none	登录页轮播图
»» logo_square	string	true	none	透明logo
»» logo_rectangle	string	true	none	方形logo
»» login_logo	string	true	none	登录正常logo

## 总后台/统计

GET 统计

GET /api/admin/chart/sum

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 客户首页统计

GET /api/admin/chart

### 请求参数

名称	位置	类型	必选	说明
year	query	string	true	年
month	query	string	true	月
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 总后台/客服管理/客服管理

## GET 客服登录

GET /api/admin/chat/kefu/login/{id}

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	客服id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 修改客服表单

GET /api/admin/chat/kefu/{id}/edit

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	客服id
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取添加客服表单

GET /api/admin/chat/kefu/add

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## DELETE 删除客服

DELETE /api/admin/chat/kefu/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	客服id
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## PUT 修改客服

PUT /api/admin/chat/kefu/{id}

Body 请求参数

```

type: object
properties:
  avatar:
    type: string
  nickname:
    type: string
  account:
    type: string
  phone:
    type: string
  status:
    type: string
  notify:
    type: string
  customer:
    type: string
  password:
    type: string
  true_password:
    type: string
required:
- avatar
- account
- phone

```

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	客服id
Authori-zation	header	string	true	none
body	body	object	false	none
» avatar	body	string	true	none
» nickname	body	string	false	none
» account	body	string	true	none
» phone	body	string	true	none
» status	body	string	false	none
» notify	body	string	false	none
» customer	body	string	false	none
» password	body	string	false	none
» true_password	body	string	false	none

### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 聊天记录

GET /api/admin/chat/kefu/record/{id}

### 请求参数



名称	位置	类型	必选	说明
id	path	string	true	客服id
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

### PUT 修改客服状态

PUT /api/admin/chat/kefu/set\_status/{id}/{status}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	客服id
status	path	string	true	状态值
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

### POST 添加客服

POST /api/admin/chat/kefu

Body 请求参数

```

type: object
properties:
  avatar:
    type: string
    description: 客服头像
  uid:
    type: string
    description: 用户uid
  customer:
    type: string
  notify:
    type: string
  phone:
    type: string
    description: 手机号码
  account:
    type: string
    description: 账号
  password:
    type: string

```

```

description: 密码
true_password:
  type: string
description: 确认密码
nickname:
  type: string
status:
  type: string
required:
- avatar
- phone
- account
- password
- true_password
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» avatar	body	string	true	客服头像
» uid	body	string	false	用户uid
» customer	body	string	false	none
» notify	body	string	false	none
» phone	body	string	true	手机号码
» account	body	string	true	账号
» password	body	string	true	密码
» true_password	body	string	true	确认密码
» nickname	body	string	false	none
» status	body	string	false	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 客服列表

GET /api/admin/chat/kefu

### 请求参数

名称	位置	类型	必选	说明
page	query	string	true	页码
limit	query	string	true	展示数量
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 查看对话

GET /api/admin/chat/kefu/chat\_list

### 请求参数

名称	位置	类型	必选	说明
uid	query	string	false	none
to_uid	query	string	false	none
id	query	string	true	客服id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 总后台/客服管理/反馈管理

## GET 获取修改用户反馈

GET /api/admin/chat/feedback/{%24id}/edit

### 请求参数

名称	位置	类型	必选	说明
\$id	path	string	true	列表id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 反馈列表

GET /api/admin/chat/feedback

### 请求参数

名称	位置	类型	必选	说明
title	query	string	false	none
time	query	string	false	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

### DELETE 删除用户反馈

DELETE /api/admin/chat/feedback/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	列表id
Authori-zation	header	string	false	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

### PUT 修改用户反馈

PUT /api/admin/chat/feedback/{id}

Body 请求参数

```

type: object
properties:
  make:
    type: string
    description: 备注
  status:
    type: string
    description: 状态0=未查看, 1=已查看

```

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	列表id
Authori-zation	header	string	true	none
body	body	object	false	none

» make	body	string	false	备注
» status	body	string	false	状态0=未查看, 1=已查看

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## 总后台/客服管理/话术资源

## GET 获取话术详情

GET /api/admin/chat/speechcraft/{%24id}

## 请求参数

名称	位置	类型	必选	说明
\$id	path	string	true	话术列表id
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 话术列表

GET /api/admin/chat/speechcraft

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## POST 保存话术 (创建话术)

POST /api/admin/chat/speechcraft

Body 请求参数

```

type: object
properties:
  title:
    type: string
    description: 话术标题
  message:
    type: string
    description: 话术内容
  cate_id:
    type: string
    description: 全局话术分类id
  sort:
    type: string
    description: 排序
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» title	body	string	false	话术标题
» message	body	string	false	话术内容
» cate_id	body	string	false	全局话术分类id
» sort	body	string	false	排序

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## DELETE 删除话术

DELETE /api/admin/chat/speechcraft/{id}

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	列表id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 修改话术

PUT /api/admin/chat/speechcraft/{id}

Body 请求参数

```

type: object
properties:
  title:
    type: string
    description: 话术标题
  message:
    type: string
    description: 话术内容
  cate_id:
    type: string
    description: 全局话术分类id
  sort:
    type: string
    description: 排序
required:
  - title
  - cate_id
    
```

请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	false	none
body	body	object	false	none
» title	body	string	true	话术标题
» message	body	string	false	话术内容
» cate_id	body	string	true	全局话术分类id
» sort	body	string	false	排序

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

GET 获取修改话术

GET /api/admin/chat/speechcraft/{%24id}/edit

请求参数

名称	位置	类型	必选	说明
\$id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

## GET 获取话术创建

GET /api/admin/chat/speechcraft/create

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

## 总后台/客服管理/话术分类

### GET 获取话术创建

GET /api/admin/chat/speechcraftcate/create

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### GET 获取修改话术分类

GET /api/admin/chat/speechcraftcate/{%24id}/edit

请求参数

名称	位置	类型	必选	说明
\$id	path	string	true	f分类id
Authori-zation	header	string	false	none

返回示例



## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 取话术分类列表

GET /api/admin/chat/speechcraftcate

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## POST 保存话术分类

POST /api/admin/chat/speechcraftcate

## Body 请求参数

```

type: object
properties:
  name:
    type: string
    description: 分类名称
  sort:
    type: string
    description: 排序
required:
  - name

```

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» name	body	string	true	分类名称
» sort	body	string	false	排序

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 修改话术分类

PUT /api/admin/chat/speechcraftcate/{id}

#### Body 请求参数

```

type: object
properties:
  name:
    type: string
    description: 分类名称
  sort:
    type: string
    description: 排序
required:
  - name
    
```

#### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	分类id
Authori-zation	header	string	true	none
body	body	object	false	none
» name	body	string	true	分类名称
» sort	body	string	false	排序

#### 返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## DELETE 删除话术分类

DELETE /api/admin/chat/speechcraftcate/{id}

#### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	分类id
Authori-zation	header	string	true	none

#### 返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取话术分类详情

GET /api/admin/chat/speechcraftcate/{%24id}

### 请求参数

名称	位置	类型	必选	说明
\$id	path	string	true	分类id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 总后台/系统管理/管理员

## GET 退出登陆

GET /api/admin/setting/admin/logout

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 修改管理员状态

PUT /api/admin/setting/set\_status/{id}/{status}

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	管理员id
status	path	string	true	状态值
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

200	OK	成功	Inline
-----	----	----	--------

返回数据结构

## 总后台/系统管理/菜单管理

### GET 获取权限菜单创建接口

GET /api/admin/setting/menus/create

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### GET 获取权限菜单列表接口

GET /api/admin/setting/menus

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### POST 保存权限菜单接口

POST /api/admin/setting/menus

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

200	OK	成功	Inline
-----	----	----	--------

## 返回数据结构

## DELETE 删除权限菜单接口

DELETE /api/admin/setting/menus/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## PUT 修改权限菜单接口

PUT /api/admin/setting/menus/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取修改权限菜单接口

GET /api/admin/setting/menus/{id}/edit

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

## 总后台/系统管理/应用管理

### PUT 重置应用token

PUT /api/admin/app/reset/{id}

请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### PUT 修改应用接口

PUT /api/admin/app/{id}

请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### DELETE 删除应用接口

DELETE /api/admin/app/{id}

请求参数

名称	位置	类型	必选	说明
id	path	string	true	应用ID
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## POST 保存应用接口

POST /api/admin/app

## Body 请求参数

```

type: object
properties:
  icon:
    type: string
    description: 应用图标
    example: http://chat.lfmc.fun/uploads/attach/2021/07/20210708/90d8b487cea7ea2545cafcabfb6f6735.jpg
  name:
    type: string
    description: 应用名称
    example: 李星云
  introduce:
    type: string
    description: 应用简介
required:
  - icon
  - name

```

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» icon	body	string	true	应用图标
» name	body	string	true	应用名称
» introduce	body	string	false	应用简介

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取应用列表接口

GET /api/admin/app

## 请求参数

名称	位置	类型	必选	说明
----	----	----	----	----

Authori-zation	header	string	true	none
----------------	--------	--------	------	------

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取修改应用接口

GET /api/admin/app/{id}/edit

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	应用id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取应用创建接口

GET /api/admin/app/create

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 总后台/附件相关

## GET 图片附件列表

GET /api/admin/file/file

### 请求参数



名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» list	[object]	true	none	none
»»» att_id	integer	true	none	none
»»» name	string	true	none	名称
»»» att_dir	string	true	none	地址
»»» satt_dir	string	true	none	地址
»»» att_size	string	true	none	大小
»»» att_type	string	true	none	类型
»»» pid	integer	true	none	none
»»» time	integer	true	none	添加时间
»»» image_type	integer	true	none	附件类型
»»» module_type	integer	true	none	none
»»» real_name	string	true	none	none
»» count	integer	true	none	none

### PUT 修改图片名称

PUT /api/admin/file/file/update/{id}

Body 请求参数

```
type: object
properties:
  real_name:
    type: string
    description: " 图片名称"
```

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	附件id
Authori-zation	header	string	true	none

body	body	object	false	none
» real_name	body	string	false	图片名称

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 删除图片

POST /api/admin/file/file/delete

Body 请求参数

```

type: object
properties:
  ids:
    type: string
    description: 附件id
  required:
    - ids
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» ids	body	string	true	附件id

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 移动图片分类

PUT /api/admin/file/file/do\_mov

Body 请求参数

```

type: object
properties:
  pid:
    type: string
  images:
    type: string
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» pid	body	string	false	none
» images	body	string	false	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 上传图片

POST /api/admin/file/upload/{upload\_type}

Body 请求参数

```

type: object
properties:
  file:
    type: string
    format: binary
  pid:
    type: string
    description: 分类ID
    example: 分类ID
required:
  - file
    
```

### 请求参数

名称	位置	类型	必选	说明
upload_type	path	string	true	图片上传类型 1本地 2七牛云 3OSS 4COS
Authori-zation	header	string	true	none
body	body	object	false	none
» file	body	string(binary)	true	none
» pid	body	string	false	分类ID

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» src	string	true	none	图片地址

## 总后台/附件相关/附件分类管理

### GET 附件分类列表

GET /api/admin/file/category

#### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

#### 返回数据结构

### POST 保存附件分类

POST /api/admin/file/category

Body 请求参数

```

type: object
properties:
  pid:
    type: string
    description: 上级id
  name:
    type: string
    description: 分类名称
required:
  - pid
  - name

```

#### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» pid	body	string	true	上级id
» name	body	string	true	分类名称

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## PUT 修改附件分类

PUT /api/admin/file/category/{id}

## Body 请求参数

```

type: object
properties:
  pid:
    type: string
  name:
    type: string

```

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	分类id
Authori-zation	header	string	true	none
body	body	object	false	none
» pid	body	string	false	none
» name	body	string	false	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## DELETE 删除附件分类

DELETE /api/admin/file/category/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	分类id
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取修改附件分类

GET /api/admin/file/category/{%24id}/edit

## 请求参数

名称	位置	类型	必选	说明
\$id	path	string	true	分类id
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## 总后台/用户管理/用户分组

## GET 获取分组创建接口

GET /api/admin/user/group/create

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» rules	[object]	true	none	none
»»» type	string	false	none	none
»»» field	string	false	none	none
»»» value	string	false	none	none
»»» title	string	false	none	none

»»» props	object	false	none	none
»»» type	string	true	none	none
»»» placeholder	string	true	none	none
»»» validate	[object]	false	none	none
»»» message	string	false	none	none
»»» required	boolean	false	none	none
»»» type	string	false	none	none
»»» trigger	string	false	none	none
» title	string	true	none	none
» action	string	true	none	none
» method	string	true	none	none
» info	string	true	none	none
» status	boolean	true	none	none

## DELETE 删除分组接口

DELETE /api/admin/user/group/{id}

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 修改分组接口

PUT /api/admin/user/group/{id}

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 保存分组接口

POST /api/admin/user/group

Body 请求参数

```
type: object
properties:
  group_name:
    type: string
    description: 名称
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» group_name	body	string	false	名称

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取分组列表接口

GET /api/admin/user/group

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» list	[object]	true	none	none
»» id	integer	true	none	none



»» group_name	string	true	none	分组名称
» count	integer	true	none	分组总数

## GET 获取修分组签接口

GET /api/admin/user/group/{id}/edit

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» rules	[object]	true	none	none
»»» type	string	true	none	none
»»» field	string	true	none	none
»»» value	integer	true	none	none
»»» title	string	false	none	none
»»» props	object	false	none	none
»»»» type	string	true	none	none
»»»» placeholder	string	true	none	none
»»»» validate	[object]	false	none	none
»»»» message	string	false	none	none
»»»» required	boolean	false	none	none
»»»» type	string	false	none	none
»»»» trigger	string	false	none	none
» title	string	true	none	none
» action	string	true	none	none
» method	string	true	none	none
» info	string	true	none	none
» status	boolean	true	none	none

# 总后台/用户管理/用户标签

## POST 保存标签接口

POST /api/admin/user/label

### Body 请求参数

```
type: object
properties:
  cate_id:
    type: string
    description: 分类id
  label:
    type: string
    description: 标签名称
required:
- cate_id
- label
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» cate_id	body	string	true	分类id
» label	body	string	true	标签名称

### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取标签列表接口

GET /api/admin/user/label

### 请求参数

名称	位置	类型	必选	说明
cate_id	query	string	true	分类id
Authori-zation	header	string	true	none

### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取标签创建接口

GET /api/admin/user/label/create

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取修改标签接口

GET /api/admin/user/label/{id}/edit

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	获取修改标签表单
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 修改标签接口

PUT /api/admin/user/label/{id}

Body 请求参数

```

type: object
properties:
  cate_id:
    type: string
    description: 分类id
  label:
    type: string
    description: 标签
required:
  - cate_id
  - label

```

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none
body	body	object	false	none
» cate_id	body	string	true	分类id
» label	body	string	true	标签

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## DELETE 删除标签接口

DELETE /api/admin/user/label/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## 总后台/用户管理/用户标签/标签分类

## POST 保存标签分类接口

POST /api/admin/user/label/cate

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取标签分类列表接口

GET /api/admin/user/label/cate

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» data	[object]	true	none	none
»»» id	integer	true	none	分类id
»»» pid	integer	true	none	上级id
»»» owner_id	integer	true	none	所属类型
»»» name	string	true	none	分类名称
»»» sort	integer	true	none	排序
»»» type	integer	true	none	类型
»»» other	string	true	none	其他参数
»»» add_time	integer	true	none	添加时间
»» count	integer	true	none	总条数

## PUT 修改标签分类接口

PUT /api/admin/user/label/cate/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

200	OK	成功	Inline
-----	----	----	--------

## 返回数据结构

## DELETE 删除标签分类接口

DELETE /api/admin/user/label/cate/{id}

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取标签分类创建接口

GET /api/admin/user/label/cate/create

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取修改标签分类接口

GET /api/admin/user/label/cate/{id}/edit

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none

返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

200	OK	成功	Inline
-----	----	----	--------

返回数据结构

## 总后台/用户管理/用户列表

GET 获取修改用户表单

GET /api/admin/user/edit/{id}

请求参数

名称	位置	类型	必选	说明
id	path	string	true	用户id
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» rules	[object]	true	none	none
»»» type	string	true	none	none
»»» field	string	true	none	none
»»» value	string	true	none	none
»»» title	string	true	none	none
»»» props	object	true	none	none
»»»» type	string	true	none	none
»»»» maxLength	integer	false	none	none
»»»» title	string	false	none	none
»»»» src	string	false	none	none
»»»» icon	string	false	none	none
»»»» width	string	false	none	none
»»»» height	string	false	none	none
»»»» placeholder	string	true	none	none
»»»» multiple	boolean	false	none	none
»»» options	[object]	false	none	none
»»»» label	string	true	none	none

»»» value	integer	true	none	none
»» title	string	true	none	none
»» action	string	true	none	none
»» method	string	true	none	none
»» info	string	true	none	none
»» status	boolean	true	none	none

## GET 获取全部标签

GET /api/admin/user/label/all

### 请求参数

名称	位置	类型	必选	说明
id	query	string	false	用户id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	[object]	true	none	none
»» name	string	true	none	分类名称
»» id	integer	true	none	分类id
»» label	[object]	true	none	标签
»»» id	integer	true	none	标签id
»»» label	string	true	none	标签名称
»»» user_id	integer	true	none	用户id
»»» cate_id	integer	true	none	分类id
»»» create_time	string	true	none	创建时间

## GET 获取全部分组

GET /api/admin/user/group/all

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none



返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	[object]	true	none	none
»» id	integer	true	none	分组id
»» group_name	string	true	none	分组名称

GET 用户列表

GET /api/admin/user/index

请求参数

名称	位置	类型	必选	说明
nickname	query	string	false	昵称或手机号搜索
group_id	query	string	false	分组id
time	query	string	false	时间搜索
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» list	[object]	true	none	none
»»» id	integer	true	none	自增id
»»» uid	integer	true	none	用户端uid
»»» group_id	integer	true	none	分组id
»»» nickname	string	true	none	昵称

»»» avatar	string	true	none	头像
»»» phone	string	true	none	手机号
»»» last_ip	string	true	none	访问ip
»»» appid	string	true	none	应用id
»»» remarks	string	true	none	备注
»»» is_delete	integer	true	none	是否删除
»»» is_kefu	integer	true	none	是否客服
»»» is_tourist	integer	true	none	是否游客
»»» type	integer	true	none	来源类型
»»» sex	integer	true	none	性别
»»» create_time	string	true	none	创建时间
»»» update_time	string	true	none	更新时间
» count	integer	true	none	总条数

## PUT 批量修改用户标签

PUT /api/admin/user/batch/label

Body 请求参数

```

type: object
properties:
  "ids[]":
    type: string
    description: 用户id
  "label_id[]":
    type: string
    description: 标签id
  "un_label_id[]":
    type: string
    description: 取消的标签id
required:
  - ids[]
  - label_id[]
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» ids[]	body	string	true	用户id
» label_id[]	body	string	true	标签id
» un_label_id[]	body	string	false	取消的标签id

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## PUT 修改用户

PUT /api/admin/user/update/{id}

## Body 请求参数

```

type: object
properties:
  avatar:
    type: string
    description: 头像
  nickname:
    type: string
    description: 昵称
  group_id:
    type: string
    description: 分组
  remarks:
    type: string
    description: 备注
required:
  - avatar
  - nickname

```

## 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none
body	body	object	false	none
» avatar	body	string	true	头像
» nickname	body	string	true	昵称
» group_id	body	string	false	分组
» remarks	body	string	false	备注

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## PUT 批量修改用户分组

PUT /api/admin/user/batch/group

## Body 请求参数

```

type: object
properties:
  "ids[]":
    type: string
  group_id:
    type: string

```

```
required:
- ids[]
- group_id
```

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» ids[]	body	string	true	none
» group_id	body	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## 总后台/维护管理/配置分类

## GET 获取配置分类列表接口

GET /api/admin/setting/config\_class

## 请求参数

名称	位置	类型	必选	说明
status	query	string	false	状态
title	query	string	false	标题
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» list	[object]	true	none	none
»» id	integer	true	none	none

»»» pid	integer	true	none	上级id
»»» title	string	true	none	分类名称
»»» eng_title	string	true	none	分类名称
»»» status	integer	true	none	状态
»»» info	integer	true	none	说明
»»» icon	string	true	none	图标
»»» type	integer	true	none	类型
»»» sort	integer	true	none	排序
»»» children	[object]	false	none	none
»»» id	integer	true	none	none
»»»» pid	integer	true	none	none
»»»» title	string	true	none	none
»»»» eng_title	string	true	none	none
»»»» status	integer	true	none	none
»»»» info	integer	true	none	none
»»»» icon	string	true	none	none
»»»» type	integer	true	none	none
»»»» sort	integer	true	none	none
»» count	integer	true	none	none

## 客服后台/用户标签

### GET 获取用户信息

GET /api/kefu/user/info/{userId}

#### 请求参数

名称	位置	类型	必选	说明
userId	path	string	true	用户自增id
Authori-zation	header	string	true	none

返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

#### 返回数据结构

### GET 获取用户标签

GET /api/kefu/user/label/{userId}

#### 请求参数

名称	位置	类型	必选	说明
----	----	----	----	----

userId	path	string	true	用户自增ID
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 保存用户标签

PUT /api/kefu/user/label

Body 请求参数

```

type: object
properties:
  "user_ids[]":
    type: string
    example: 用户自增id
  label:
    type: string
    example: 标签名称
required:
  - user_ids[]
  - label
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» user_ids[]	body	string	true	none
» label	body	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 客服后台/公用接口

## POST 客服登录

POST /api/kefu/login

Body 请求参数

```

type: object
    
```

```
properties:
  account:
    type: string
    description: 账号
  password:
    type: string
    description: 密码
required:
- account
- password
```

### 请求参数

名称	位置	类型	必选	说明
body	body	object	false	none
» account	body	string	true	账号
» password	body	string	true	密码

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 退出登录

POST /api/kefu/user/logout

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 客服后台/客户列表

### GET 获取当前客服客户列表

GET /api/kefu/user/list

### 请求参数

名称	位置	类型	必选	说明
nickname	query	string	false	客户名称

Authori-zation	header	string	true	none
----------------	--------	--------	------	------

返回示例

成功

```
{
  "status": 200,
  "msg": "ok",
  "data": {
    "S": [
      {
        "id": 2,
        "appid": "202116257358989495",
        "user_id": 118,
        "to_user_id": 1,
        "nickname": "寿司",
        "avatar": "http://chat.lfmn.fun/uploads/attach/2021/07/20210708/90d8b487cea7ea2545cafcba6fb6f6735.jpg",
        "is_tourist": 0,
        "online": 0,
        "type": 0,
        "add_time": 0,
        "update_time": 0,
        "mssage_num": 0,
        "message": "",
        "message_type": 0,
        "initials": "S"
      }
    ]
  }
}
```

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 设置用户标签

PUT /api/kefu/user/label/{userId}

Body 请求参数

```
type: object
properties:
  "label_ids[]":
    type: string
  "un_label_ids[]":
    type: string
required:
  - label_ids[]
  - un_label_ids[]
```

### 请求参数

名称	位置	类型	必选	说明
userId	path	string	true	none
Authori-zation	header	string	true	none



body	body	object	false	none
» label_ids[]	body	string	true	none
» un_label_ids[]	body	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 用户标签

GET /api/kefu/user/label

### 请求参数

名称	位置	类型	必选	说明
id	query	string	true	用户user_id
Authori-zation	header	string	true	none

返回示例

成功

```
{
  "status": 200,
  "msg": "ok",
  "data": [
    {
      "id": 1,
      "label": "测试标签",
      "uid_id": 114,
      "cate_id": 0,
      "create_time": "2021-07-13 15:51:49",
      "user": []
    }
  ]
}
```

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 和客服聊天过的用户

GET /api/kefu/user/record

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	query	string	true	none

Authori-zation	header	string	true	none
----------------	--------	--------	------	------

返回示例

成功

```
{
  "status": 200,
  "msg": "ok",
  "data": []
}
```

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 客服后台/用户分组

### PUT 修改用户分组

PUT /api/kefu/user/group/{userId}/{id}

### 请求参数

名称	位置	类型	必选	说明
userId	path	string	true	用户自增id
id	path	string	true	分组id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

### GET 获取所有分组

GET /api/kefu/user/group

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

200	OK	成功	Inline
-----	----	----	--------

返回数据结构

## 客服后台/个人中心

### GET 获取当前登录客服信息

GET /api/kefu/user/userInfo

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### PUT 修改当前登录客服

PUT /api/kefu/user/userInfo

Body 请求参数

```

type: object
properties:
  nickname:
    type: string
    description: 昵称
  avatar:
    type: string
    description: 头像
  password:
    type: string
    description: 不修改返回6个星号
    example: "*****"
  phone:
    type: string
    description: 手机号
required:
  - nickname
  - avatar
  - phone
    
```

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» nickname	body	string	true	昵称
» avatar	body	string	true	头像

» password	body	string	false	不修改返回6个星号
» phone	body	string	true	手机号

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 修改图像

POST /api/kefu/upload

Body 请求参数

```

type: object
properties:
  file:
    type: string
    description: 图片
    format: binary
  filename:
    type: string
    description: 文件名称
required:
  - file
  - filename
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» file	body	string(binary)	true	图片
» filename	body	string	true	文件名称

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 客服后台/客服工作台

## DELETE 删除分类

DELETE /api/kefu/service/cate/{id}

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	分类id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 编辑分类

PUT /api/kefu/service/cate/{id}

Body 请求参数

```

type: object
properties:
  name:
    type: string
    description: 分类名称
  sort:
    type: string
    description: 排序
required:
  - name
    
```

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	none
Authori-zation	header	string	true	none
body	body	object	false	none
» name	body	string	true	分类名称
» sort	body	string	false	排序

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 添加话术

POST /api/kefu/service/speechcraft

Body 请求参数

```

type: object
properties:
    
```

```

title:
  type: string
  description: 话术标题
cate_id:
  type: string
  description: 分类全局话术id
message:
  type: string
  description: 话术内容
sort:
  type: string
  description: 排序
required:
- title
- message
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» title	body	string	true	话术标题
» cate_id	body	string	false	分类全局话术id
» message	body	string	true	话术内容
» sort	body	string	false	排序

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 客服话术

GET /api/kefu/service/speechcraft

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 当前客服详细信息

GET /api/kefu/service/info

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 聊天记录

GET /api/kefu/service/list

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 保存分类

POST /api/kefu/service/cate

Body 请求参数

```

type: object
properties:
  name:
    type: string
    description: 分类名称
  sort:
    type: string
    description: 排序
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» name	body	string	false	分类名称
» sort	body	string	false	排序

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## POST 客服转接 (有问题)

POST /api/kefu/service/transfer

#### Body 请求参数

```

type: object
properties:
  kefuToUserId:
    type: string
    description: 转接uid
  user_id:
    type: string
    description: 发送人uid
    
```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» kefuToUserId	body	string	false	转接uid
» user_id	body	string	false	发送人uid

#### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 转接客服列表

GET /api/kefu/service/transfer\_list

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

#### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline



### 返回数据结构

## DELETE 删除话术

DELETE /api/kefu/service/speechcraft/{id}

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	话术id
Authori-zation	header	string	true	none

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## PUT 修改话术

PUT /api/kefu/service/speechcraft/{id}

Body 请求参数

```

type: object
properties:
  title:
    type: string
    description: " 话术标题"
  cate_id:
    type: string
    description: 分类全局话术id
  message:
    type: string
    description: " 话术内容"
required:
- title
- cate_id
- message

```

### 请求参数

名称	位置	类型	必选	说明
id	path	string	true	话术id
Authori-zation	header	string	true	none
body	body	object	false	none
» title	body	string	true	话术标题
» cate_id	body	string	true	分类全局话术id
» message	body	string	true	话术内容

返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

## 客服后台/统计

### GET 获取当前客服统计(首页)

GET /api/kefu/statistics/index

请求参数

名称	位置	类型	必选	说明
year	query	string	true	年
month	query	string	true	月
Authori-zation	header	string	true	none

返回示例

成功

```
{
  "status": 200,
  "msg": "ok",
  "data": []
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

返回数据结构

### GET 获取客户统计

GET /api/kefu/statistics/all

请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

返回示例

成功

```
{
  "status": 200,
  "msg": "ok",
  "data": {
    "all": 1,
    "toDayKefu": 0,
    "month": 0,
    "toDayTourist": 0
  }
}
```

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## APP

## POST 上传图片

POST /api/mobile/service/upload

## Body 请求参数

```

type: object
properties:
  filename:
    type: string
    example: file
required:
  - filename

```

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» filename	body	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

## GET 获取客服广告

GET /api/mobile/service/adv

## 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

## 返回示例

## 返回结果

状态码	状态码含义	说明	数据模型
-----	-------	----	------

200	OK	成功	Inline
-----	----	----	--------

## 返回数据结构

### POST 保存客服反馈内容

POST /api/mobile/service/feedback

#### Body 请求参数

```

type: object
properties:
  rela_name:
    type: string
    description: 姓名
  phone:
    type: string
    description: 电话
  content:
    type: string
    description: 内容
required:
- rela_name
- phone
- content

```

#### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» rela_name	body	string	true	姓名
» phone	body	string	true	电话
» content	body	string	true	内容

#### 返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

## 返回数据结构

### GET 获取反馈页面广告位内容

GET /api/mobile/service/feedback

#### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

#### 返回示例

#### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取缓存

GET /api/mobile/service/cache/{key}

### 请求参数

名称	位置	类型	必选	说明
key	path	string	true	缓存key
Authori-zation	header	string	true	none

#### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## GET 获取聊天记录

GET /api/mobile/user/record

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none

#### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

状态码 200

名称	类型	必选	约束	说明
» status	integer	true	none	none
» msg	string	true	none	none
» data	object	true	none	none
»» serviceList	[string]	true	none	聊天记录
»» to_user_id	integer	true	none	聊天人id
»» uid	string	true	none	当前用户id
»» nickname	string	true	none	当前用户nickname
»» avatar	string	true	none	当前用户头像

» to_user_nickname	string	true	none	聊天人昵称
» to_user_avatar	string	true	none	聊天人头像
» site_name	string	true	none	页面标题
» is_tourist	string	true	none	是否为游客:1=是,0=否
» user_id	string	true	none	当前用户自增id

## POST 保存缓存

POST /api/mobile/service/cache

### Body 请求参数

```

type: object
properties:
  key:
    type: string
    description: 缓存key值
  value:
    type: string
    description: 缓存值
required:
  - key
  - value

```

### 请求参数

名称	位置	类型	必选	说明
Authori-zation	header	string	true	none
body	body	object	false	none
» key	body	string	true	缓存key值
» value	body	string	true	缓存值

### 返回示例

### 返回结果

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline

### 返回数据结构

## 数据模型

# 目录结构

## PHP目录接口

## 后端文件目录结构

目录结构只列出,框架之外的内容,如需了解其他目录结构请查看:Thinkphp6官网的文档说明

### APP目录结构

```
├─controller
│   ├── InstallController.php
│   └── admin 总后台模块
│       ├── Application.php 应用控制器
│       ├── AuthController.php 基类控制器
│       ├── Common.php 公用控制器
│       ├── Index.php 主页
│       ├── Login.php 登陆
│       └── Test.php 测试
│
│   ├── chat 客服
│       ├── Service.php 客服
│       ├── ServiceFeedback.php 反馈
│       ├── ServiceSpeechcraft.php 话术
│       └── ServiceSpeechcraftCate.php 话术分类
│
│   ├── file 文件
│       ├── Attachment.php 附件
│       └── AttachmentCategory.php 附件分类
│
│   ├── system 配置
│       ├── Admin.php 管理员
│       ├── Config.php 配置
│       ├── ConfigTab.php 配置分类
│       ├── Group.php 组合数据
│       ├── GroupData.php 组合数据数据
│       ├── Log.php 日志
│       ├── Menus.php 菜单
│       └── Role.php 权限
│
│   └── user 用户
│       ├── Group.php 分组
│       ├── Label.php 标签
│       ├── LabelCate.php 标签分类
│       └── User.php 用户
│
│   ├── kefu 客服模块
│       ├── AuthController.php 基础控制器
│       ├── Login.php 登陆
│       ├── Service.php 客服
│       ├── Statistics.php 统计
│       └── User.php 用户
│
│   └── mobile 移动端模块
│       ├── AuthController.php 基类控制器
│       ├── Feedback.php 反馈
│       └── Service.php 客服
│
├─dao Dao层
│   ├── ApplicationDao.php 应用
│   └── chat 客服
│       ├── ChatServiceAuxiliaryDao.php 客服关联
│       ├── ChatServiceDao.php 客服
│       ├── ChatServiceDialogueRecordDao.php 聊天记录日志
│       ├── ChatServiceFeedbackDao.php 发聩
│       └── ChatServiceRecordDao.php 聊天记录
```

```

| | ChatServiceSpeechcraftDao.php 话术
| | ChatUserDao.php 用户
| |
| | └─user 用户
| |     | ChatUserGroupDao.php 用户分组
| |     | ChatUserLabelAssistDao.php 标签辅助
| |     | ChatUserLabelDao.php 标签
| |
| | └─other 其他
| |     | AuxiliaryDao.php 辅助表
| |     | CacheDao.php 数据库缓存笔=表
| |     | CategoryDao.php 分类表
| |
| | └─system 系统配置
| |     | SystemMenusDao.php 菜单
| |     |
| |     └─admin 权限
| |         | AdminAuthDao.php 登陆dao
| |         | SystemAdminDao.php 管理员
| |         | SystemRoleDao.php 权限
| |
| |     └─attachment 附件
| |         | SystemAttachmentCategoryDao.php 附件分类
| |         | SystemAttachmentDao.php 附件
| |
| |     └─config 配置
| |         | SystemConfigDao.php 配置
| |         | SystemConfigTabDao.php 配置分类
| |         | SystemGroupDao.php 组合数据
| |         | SystemGroupDataDao.php 组合数据数据记录
| |
| |     └─log 日志
| |         | SystemFileDao.php 文件检测
| |         | SystemLogDao.php 日志记录
| |
| | └─http
| |     └─middleware 中间件
| |         | AllowOriginMiddleware.php 跨域中间件
| |         | InstallMiddleware.php 安装中间件
| |
| |         └─admin 总后台模块中间件
| |             | AdminAuthTokenMiddleware.php 登陆中间件
| |             | AdminCkeckRoleMiddleware.php 权限验证中间件
| |             | AdminLogMiddleware.php 日志中间
| |
| |         └─kefu 客服模块中间件
| |             | KefuAuthTokenMiddleware.php 客服登陆中间
| |
| |         └─mobile 移动端中间
| |             | MobileAuthTokenMiddleware.php 应用检测中间件
| |
| | └─jobs
| | └─lang 多语言
| |     | zh-cn.php
| |
| | └─listener 普通事件类存放
| | └─models 模型
| |     | Application.php 应用
| |     |
| |     └─chat 客服
| |         | ChatService.php
| |         | ChatServiceDialogueRecord.php
| |         | ChatServiceFeedback.php
| |         | ChatServiceRecord.php
| |         | ChatServiceSpeechcraft.php
| |         | ChatUser.php
| |         |
| |         └─user 客服用户
| |             | ChatUserGroup.php
| |             | ChatUserLabel.php
| |             | ChatUserLabelAssist.php

```



- ├──other 其他
  - │ Auxiliary.php
  - │ Cache.php
  - │ Category.php
- ├──system 系统配置
  - │ SystemMenus.php
  - ├──admin 权限
    - │ SystemAdmin.php
    - │ SystemRole.php
  - ├──attachment 附件
    - │ SystemAttachment.php
    - │ SystemAttachmentCategory.php
  - ├──config 配置
    - │ SystemConfig.php
    - │ SystemConfigTab.php
    - │ SystemGroup.php
    - │ SystemGroupData.php
  - └──log 日志
    - │ SystemFile.php
    - │ SystemLog.php
- ├──services 逻辑处理层
  - │ ApplicationServices.php 应用
  - ├──chat 客服
    - │ ChatServiceAuxiliaryServices.php
    - │ ChatServiceDialogueRecordServices.php
    - │ ChatServiceFeedbackServices.php
    - │ ChatServiceRecordServices.php
    - │ ChatServiceServices.php
    - │ ChatServiceSpeechcraftCateServices.php
    - │ ChatServiceSpeechcraftServices.php
    - │ ChatUserServices.php
    - └──user 客服用户
      - │ ChatUserGroupServices.php
      - │ ChatUserLabelAssistServices.php
      - │ ChatUserLabelCateServices.php
      - │ ChatUserLabelServices.php
  - ├──kefu 客服模块逻辑层
    - │ AuthServices.php
    - │ KefuServices.php
    - │ LoginServices.php
  - ├──other 其他
    - │ CacheServices.php
    - │ CategoryServices.php
  - └──system 系统配置
    - │ SystemMenusServices.php
    - ├──admin 权限
      - │ AdminAuthServices.php
      - │ SystemAdminServices.php
      - │ SystemRoleServices.php
    - ├──attachment 附件
      - │ SystemAttachmentCategoryServices.php
      - │ SystemAttachmentServices.php
    - ├──config 配置
      - │ SystemConfigServices.php
      - │ SystemConfigTabServices.php
      - │ SystemGroupDataServices.php

## 目录结构

```
|   |   SystemGroupServices.php
|   |   └─log 日志
|   |       ClearServices.php
|   |       SystemFileServices.php
|   |       SystemLogServices.php
|   └─validate 数据验证
|       |   Test.php
|       └─chat 客服
|           |   ChatServiceFeedbackValidate.php
|           |   ChatServiceValidate.php
|           └─kefu 客服总后台
|               |   LoginValidate.php
|               |   SpeechcraftValidate.php
|               └─system
|                   |   SystemAdminValidata.php
|                   |   SystemConfigValidata.php
|                   └─webscoket 长连接
|                       |   BaseHandler.php 基类时间
|                       |   Manager.php
|                       |   Ping.php 心跳
|                       |   Response.php socket response
|                       |   Room.php 房间
|                       |   SwooleWorkerStart.php swoole启动后置事件
|                       └─handler 时间回调处理
|                           |   AdminHandler.php
|                           |   KefuHandler.php
|                           |   UserHandler.php
|                       AppService.php
|                       build.php
|                       common.php
|                       event.php
|                       ExceptionHandle.php
|                       middleware.php
|                       provider.php
|                       Request.php
|                       Route.php
|                       └─service.php
```

## crmeb 目录结构

```
└─basic 基类存放
    |   BaseDao.php
    |   BaseExpress.php
    |   BaseJobs.php
    |   BaseManager.php
    |   BaseModel.php
    |   BaseServices.php
    |   BaseSmss.php
    |   BaseStorage.php
    |   BaseUpload.php
    └─command 命令行存放
        |   Dao.php
        |   Install.php
        |   Key.php
        |   Service.php
        └─stubs 模板
            |   dao.stub
            |   service.stub
```

- ├─exceptions 错误异常类
  - | AdminException.php
  - | ApiException.php
  - | AuthException.php
  - | DecryptException.php
  - | EncryptException.php
  - | MissingAppKeyException.php
  - | UploadException.php
- ├─interfaces 接口类存放
  - | HandlerInterface.php
  - | JobInterface.php
  - | ListenerInterface.php
  - | MiddlewareInterface.php
  - | ProviderInterface.php
- ├─listeners 系统事件订阅类存放
  - | InitSwooleLockListen.php
  - | Listener.php
  - | SwooleShutdownListen.php
  - | SwooleStartListen.php
  - | SwooleTaskListen.php
  - | TestListen.php
- ├─user
  - | UserLogin.php
- ├─services 系统内置服务
  - | AccessTokenServeService.php 一号通服务平台token
  - | CacheService.php 缓存
  - | FormBuilder.php 表单
  - | GroupDataService.php 组合数据
  - | HttpService.php curl
  - | SwooleTaskService.php swoole任务
  - | SystemConfigService.php 数据配置
  - | UploadService.php 上传
- ├─express 物流
  - | Express.php
  - └─storage
    - | Express.php
- ├─serve 一号通服务
  - | Serve.php
  - └─storage
    - | Crmeh.php
- ├─sms 短信服务
  - | Sms.php
  - └─storage
    - | Yunxin.php
- ├─upload 上传服务
  - | Upload.php
  - └─storage
    - | Cos.php
    - | Local.php
    - | Oss.php
    - | Qiniu.php
- ├─subscribes
  - | TaskSubscribe.php
- ├─traits
  - | ErrorTrait.php 错误
  - | JwtAuthModelTrait.php 中间件
  - | ModelTrait.php 模型

- QueueTrait.php 消息队列
- Schema.php
- ServicesTrait.php service搜索辅助
- TimeModelTrait.php 时间查询模型辅助

└─utils

- ApiErrorCode.php 接口返回状态
- Arr.php 数组帮助
- Blueprint.php
- Canvas.php 画图
- Captcha.php 验证码
- Character.php 中文排序
- Encrypter.php 加密解密
- Hook.php
- Json.php json返回
- JwtAuth.php jwt辅助
- Qrcode.php 二维码生成重写
- Queue.php 消息队列封装辅助
- Str.php 字符串辅助

目录结构

## VUE目录结构

.keep

## 命令行介绍

### Think-swoole命令

### SWOOLE 命令详解

#### 启动服务

```
php think swoole start
```

#### 停止服务

```
php think swoole stop
```

#### 重启服务

```
php think swoole restart
```

#### 平滑重启服务

```
php think swoole reload
```

#### 客服系统启动命令

请使用sudo更改命令启动用户,不然回出现某些缓存写入无法被www用户打开的问题

```
sudo -u www php think swoole restart
```

## 应用KEY命令

## 应用KEY命令

注意:应用key使用再客服应用加密上,重置过后所有客服聊天和用户聊天不能在被解析成功,需要重新生成TOKEN

生成新KEY,不修改配置文件

```
php think key --show
```

重置/新建KEY

```
php think key
```

## 命令行安装命令

### 命令行一键安装

#### 安装命令

```
php think install
```

1. 首先配置根目录下的.env文件中的数据库连接配置文件

2. 遇到如下提示请就是.env中的配置问题

```
[root@iZbp12rlllt0itxlnaaesZ chattest.lfmn.fun]# php think install
```

```
[think\db\Exception] PDOException
SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost' (using password: YES)
```

```
[root@iZbp12rlllt0itxlnaaesZ chattest.lfmn.fun]# █
```

3. 没有错误后回自动写入数据,如下图

```
[root@iZbp12rlllt0itxlnaaesZ chattest.lfmn.fun]# php think install
```

```
+----- [SQL安装] -----+
```

表名	执行结果	错误原因	时间
eb_application	ok	无错误	2021-07-22 17:38:04
eb_cache	ok	无错误	2021-07-22 17:38:04
eb_category	ok	无错误	2021-07-22 17:38:04
eb_chat_service	ok	无错误	2021-07-22 17:38:04
eb_chat_service_dialogue_record	ok	无错误	2021-07-22 17:38:04
eb_chat_service_feedback	ok	无错误	2021-07-22 17:38:04
eb_chat_service_record	ok	无错误	2021-07-22 17:38:04
eb_chat_service_speechcraft	ok	无错误	2021-07-22 17:38:04
eb_chat_user	ok	无错误	2021-07-22 17:38:04
eb_chat_user_group	ok	无错误	2021-07-22 17:38:04
eb_chat_user_label	ok	无错误	2021-07-22 17:38:04
eb_chat_user_label_assist	ok	无错误	2021-07-22 17:38:04
eb_migrations	ok	无错误	2021-07-22 17:38:04
eb_system_admin	ok	无错误	2021-07-22 17:38:04
eb_system_attachment	ok	无错误	2021-07-22 17:38:04
eb_system_attachment_category	ok	无错误	2021-07-22 17:38:04
eb_system_config	ok	无错误	2021-07-22 17:38:04
eb_system_config_tab	ok	无错误	2021-07-22 17:38:04
eb_system_group	ok	无错误	2021-07-22 17:38:04

4. 输入管理员账号密码



```
| eb_system_role | OK | 无错误 | 2021-07-22 17:46:36 |
+-----+-----+-----+-----+

+----- [创建管理员] -----+

请输入后台登陆账号,最少4个字符:
> admin

请输入登陆密码,密码为数字加字母/字母加符号/数字加字符的组合不能少于6位:
> zxc123

安装完成!!请妥善保管您的账号密码!
账号:admin
密码:zxc123
[root@iZbp12rlllt0itxlnaalesZ chattest.lfmn.fun]# █
```

- 5. 提示'安装完成',请保管号账号密码
- 6. 如果已经安装,并且还要重新安装会自动备份数据库到根目录下的 backup

启动swoole,已启动请重启swoole

```
sudo -u www php think swoole restart
```

恭喜,安装完成拉

## 开发文档

### 后台前端打包

打开项目：`template/admin`

安装项目：

```
npm install
```

调试：

```
npm run serve
```

打包：

```
npm run build
```

打包后静态文件存放目录：`template/admin/dist`

## 常见问题

安装后出现：数据获取失败

安装后必须重启swoole

```
php think swoole restart
```

## 客服端登录成功，用户端还是进入了客服反馈

1. 检查Socket连接是否成功
2. 重置后台token后重新复制连接访问用户端。
3. 检查network接口返回的是否有错误

