

# nya 留言板介绍文档

小程序简介：

人们一直以来都有对交流的需求，大到网红读私信解答困惑，小到闺蜜之间分享心情。电子邮件公共但是太麻烦，微信等聊天软件又太私密，有没有一个简易且不会进入个人生活的小程序与大家互动呢。**nya** 留言板就承担这种角色，不用害怕交流过深，操作也足够简易。

留言板主要分 3 块：自己的留言板，他人的留言板，个人中心。

在自己的留言板，实现了自己对自己留言箱的完全管理。

功能：

- 1) 可以设置公告，
- 2) 删除别人发给自己的留言，
- 3) 点击查看留言详情，可以进行回复，并允许重复回复作更改，可以收藏留言或者关注用户，
- 4) 有下拉刷新和触底加载更多，每加载一次多增最多 10 条留言，并按时间戳排序。

在他人的留言板，不开放管理权限，进入他人留言板的方式有 3 种：

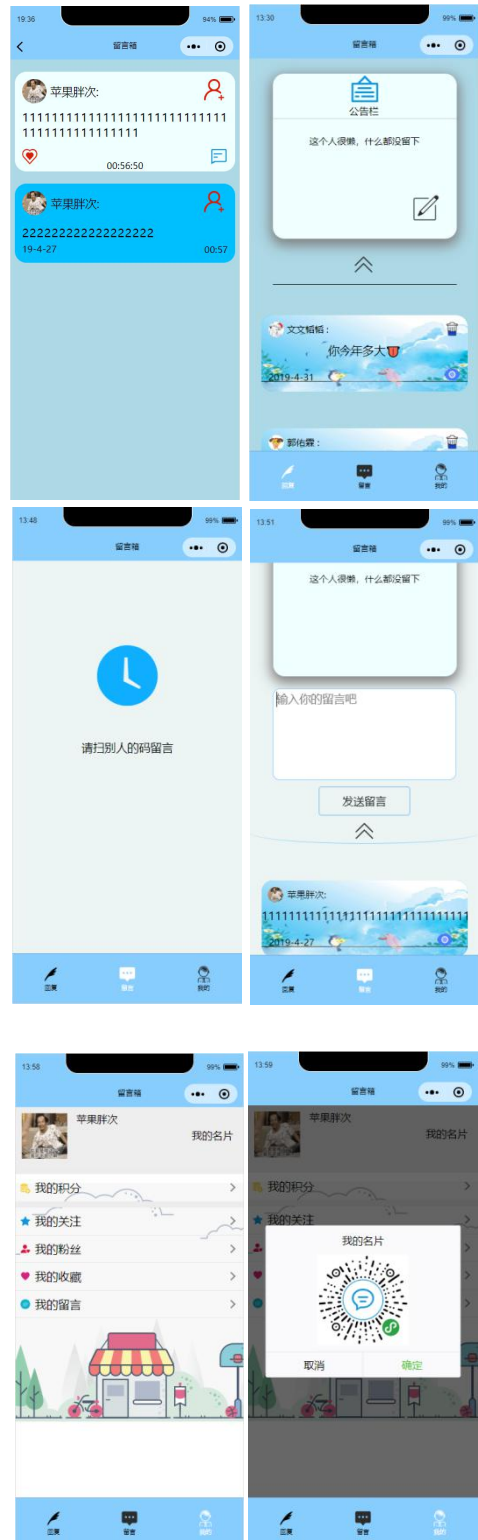
1. 微信分享、2. 扫个人中心（见下文）的小程序码、3. 在个人中心（见下文）的关注或粉丝列表，点击相应用户进入。

功能：

- 1) 查看公告，
- 2) 发送留言，
- 3) 点击查看留言详情，不可回复，可收藏或关注用户，
- 4) 同样有下拉刷新和触底加载更多，并时间排序分页的功能。

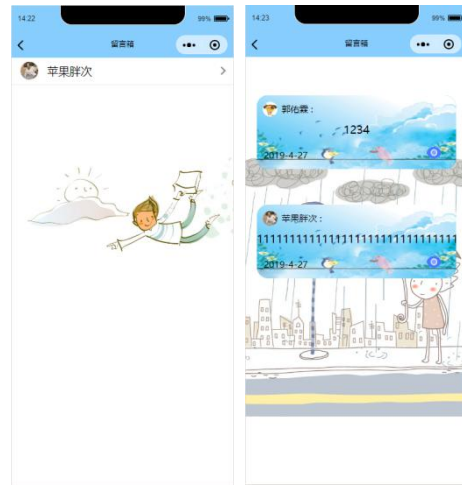
在个人中心，可以查看留言积分与回复积分，每次留言回复都加 10 分。

点击我的名片还可以加载自己的留言箱的二维码，别人扫码即可进入自己的留言板，并留言。



在我的关注和我的粉丝列表，可以点击进入他的留言箱。

在我的收藏和我的留言列表，可以查看收藏的留言以及自己发过的留言，自己发的留言也可以进行管理。



## 开发文档部分

数据库操作基本上都封成 `nyaAPI.js`，函数返回 `Promise` 对象，处理异步。

Api:

`getOpenid(): result<promise>`

调用同名的云函数获取当前用户的 `openid`

`getNoticeByOpenid(openid<string>): result<promise>`

通过 `openid` 查数据库 `userinfos` 获取公告

`getQuestionsByOpenid({ _openid<string>, page<number> }): result<promise>`

通过 `openid` 匹配 `_openid` 字段查数据库 `questions` 获取并打包处理留言，并通过 `page` 过滤已加载过的留言，确保不会重复加载

`updateNotice(addText<string>)`

通过调用云函数获取的 `openid` 查数据库 `userinfos` 并用 `addText` 更新字段 `notice` 公告内容

`deleteQuestionById(_id<string>)`

调用云函数 `deleteQuestionById` 来删除留言，防止权限问题

`getQuestionsByOpenidForOthers({ _openid<string>, page<number> }): result<promise>`

通过 `openid` 匹配 `_openid` 字段，且回复字段不为空字符串，查数据库 `questions` 获取并打包处理留言，并通过 `page` 过滤已加载过的留言，确保不会重复加载

`sendQuestionAPI({ question, answer_openid}): result<promise>`

给 `questions` 添加一条记录，包含留言与指向回答人的 `openid`，并记录日期

```
addUser({avatarUrl<string>,gender<number>,nickName<string>,_openid<string>}):result<promise>
```

给 `userinfos` 添加一条新的记录并初始化，相当于是注册

```
getMarkedQuestionsByOpenid({_openid<string>,page<number>,askInformation<array>}):result<promise>
```

通过用户 `openid` 查 `userinfos` 获取收藏字段 `mark`，再联查 `questions` 获取并打包处理所收藏的留言数组

```
getQuestionsByQuestionOpenid({_openid<string>,page<number>}):result<promise>
```

通过用户 `openid` 查 `questions` 匹配字段 `_openid` 获取并打包处理自己发过的留言数组

```
getUserByOpenid(_openid):result<promise>
```

通过获取的 `openid` 查 `userinfo` 获取用户信息

他人要调用此 API 时，要注意为了更好处理异步返回类型被包装成了 `Promise`，代码方式也改成 `Promise` 风格。

关于进入他人的留言箱的方式，通过给 `app.js` 里的全局变量里的 `toOpenid` 赋值实现，目前有 3 种带参传递赋值，第一种是分享小程序的页面传参（需上线小程序），第二种是扫个人中心的小程序码传参（也需上线小程序），第三种是在关注或粉丝列表点击用户，改变全局变量 `toOpenid` 的值为所点击的用户的 `openid`。

关于小程序码的获取，通过官方文档的 `wxacode.getUnlimited` 方法带参获取图片 `Buffer`，再将其解码，将图片保存文件至用户文件目录下，获得文件路径，从而获取到带参小程序码。