

# API接口

记录前后端通信格式以及其它通用变量格式。

## 服务器

- 1 用户名：ubuntu
- 2 IP地址：43.138.68.47
- 3 密码：:1aX0knZ4(0]]

## 服务器路由API

数组格式上传必须要先用JSON.stringify()

注意写urls.py时加参数name:

```
1 urlpatterns = [  
2     path("admin/", admin.site.urls),  
3     # path("", views.helloView),  
4     path("login/", views.login,name='login'),  
5     path("read/", views.read,name='read'),  
6     path("search/", views.search,name='search'),  
7     path("detail/", views.detail,name='detail'),  
8     path("list/", views.getlist,name='getlist'),  
9 ]
```

## 数据库

### 用户信息

#### 用户数据格式

```
1  
2 user: {  
3     openid: char, primary, unique, max_length=100 // 用户openid, 主键, 唯一, 最大长度100
```

```

4     create_time: datetime, auto_add_now          // 用户创建时间，自动设置
      为第一次创建的时间
5     status: integer, default=0                  // 用户登录状态，1是登录
      状态，实际上没用
6     nickname: char, max_length=100, default=""   // 用户昵称
7     headicon: image                             // 用户头像
8     reserved_time: integer, default=10           // 首选项，保存时长，默认
      10天
9     read_keep: integer, default=1                // 首选项，录音是否保存，
      默认保存
10    wdlistnumber: integer, default=0             // 用户单词本数量，默认为
      0
11 }

```

😄 请不要将这里的字段名称和下面的globalData中的键值搞混，谢谢。（因为写数据库的人nzyb）

## 伪用户

发送登录请求时，如果登录凭证（code）为3e5428-ff58yj5，那么就会返回假装正确的结果，格式如下：

```

1 {
2   'openid': "88hrt-j37db-x56kt-fkyou",
3   'session_key': "As your wish",
4   'unionid': 'usotuki',
5   'errmsg': '',
6   'errcode': 0,
7 }

```

😄 （看到f\*\*\*u的eqw就是正确登录了，因为是usotuki）

## 数据库数据查看

通过docker命令行查看。

```

1 sudo docker ps -a          // 查看正在运行的容器
2 docker exec -it [mysql_container_id] /bin/bash

```

```
3 sudo docker exec -it 8e0087a1236a /bin/bash
```

进入docker命令行后：

```
1 mysql -u root -p
2 输入密码：12345
```

进入mysql后：

```
1 use test;           // 使用test数据库
2 show tables;        // 查看该数据库下所有表格
3 select * from [table_name]; // 查看某表格下所有数据
4 drop table [table_name]; // 删除表格
5 truncate [table_name] // 清空表格
```

## 用户相关表单

### 单词本

每一个单词本是通过函数create\_my\_wordlist构造而来，需要传参openid与序号index。

```
1 {
2     owner_openid: ForeignKey, to='user', related_name='list'+index // 外键，
    相关到用户
3     content: char, max_length=100 // 内容
4 }
```

需要通过单词本wb访问用户，使用wb.owner\_openid.[属性]就可以访问该单词本对应的用户的信息。（用户信息见上“用户数据格式”）。

需要通过用户user访问单词本，只需要通过user.list[x]（x=1,2,3...）就可以访问到该单词本对应的表。

### 朗读记录

每个用户拥有唯一的朗读记录表，由create\_my\_readingrecord构造。

```
1 {
2     owner_openid: ForeignKey, to='user', related_name='readingrecord'
```

```

3     content: char, max_length=100
4     rdnumber: integer, default=0           // 朗读次数
5     cornumber: integer                    // 正确次数
6     date: datetime, auto_now=True         // 最后朗读时间
7     lastrd: char                          // 最后朗读结果，形如 '[0,1,1,0]'
8     lastres: integer, default=0           // 最后一次朗读是否正确，正确为1
9 }

```

## 查询记录

每个用户拥有唯一的查询记录表，由create\_my\_searchrecord构造。

```

1 {
2     owner_openid: ForeignKey, to='user', related_name='readingrecord'
3     content: char, max_length=100
4     schnumber: integer, default=0         // 查询次数
5     date: datetime, auto_now=True        // 最后查询时间
6 }

```

## 单词表信息

存放一个用户所有的单词表信息，由create\_my\_wordlistinfo构造。

```

1 {
2     owner_openid: ForeignKey, to='user', related_name='readingrecord'
3     name: char, max_length=100           // 单词表名称
4     intro: char, max_length=1000        // 单词表介绍
5     image: image                         // 单词表封面
6     date: datetime                       // 单词表创建时间
7 }

```

## 路由

### /login

用于用户登录。

### POST


## 请求体

```
1 {
2     "url": "43.138.68.47:8000/login/", //登录域名
3     "method": "POST", //请求方式
4     "data": {
5         "code": res.code //发送用户登录凭证
6     }
7 }
```

参考：

<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/login/wx.login.html>

## 返回

 返回内容改为带有globalData的数据

请求方式正确，并且传送的登录凭证有效，返回上述微信端口响应的直接结果。（json格式）

```
1 {
2     "openid": "xxxxxxx",
3     "session_key": "xxxxxx",
4     "unionid": "xxxxxx",
5     "errcode": 0,
6     "errmsg": "xxxxxx",
7     "user_data": {
8         // globalData对应后端返回内容
9     }
10 }
```

传送登录凭证无效，返回格式：

```
1 {
2     "errcode": xxx,
3     "errmsg": "xxx",
4 }
```

请求方式错误，返回体：

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

## /login/headicon

用于登陆时请求用户头像。

### POST

#### 请求体

```
1 {
2   "openid": "xxxxxxx"
3 }
```

### 返回

如果请求方式正确，且用户存在，则返回对应的用户头像的文件。

如果用户不存在，返回：

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

请求方式错误，返回：

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

## /search

### POST

## 请求体

```
1 {
2   "url": "43.138.68.47:8000/search", //登录域名
3   "method": "POST", //请求方式
4   data{
5     "openid": "dskadhkskada", //用户唯一标识
6     "searchWord": "xxxxxx",
7   }
8 }
```

## 成功响应

返回格式 ()

```
1 {
2   "code": 0,
3   "info": "Success in word searching",
4   "results": ["赤い【あかい◎】", "", .....] //单词名列表
5 }
```

## 失败响应

返回体:

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

## /detail

获取单词详情

## POST

请求体

```
1 {
2   "url": "43.138.68.47:8000/detail", //登录域名
3   "method": "POST", //请求方式
```

```

4     data{
5         "openid": "dskadhkskada",    //用户唯一标识
6         "word": "赤い【あかい◎】",    //单词名
7     }
8 }

```

## 成功响应

返回格式 ()

```

1  {
2      "code": 0,
3      "info": "Success in word searching",
4      "detail": {                                //单词详情的字典
5          "name": "赤い【あかい◎】",            //本体名字
6          "accent": 0,                          //未标声调时用-1
7          "kana": ["あ", "か", "い"],           //单拍平假名列表
8          "taka": [0, 1, 1],                    //真实假名高低, 不显示促音
9          "wave": [0, 1, 1],                   //每拍假名的高低, 2表示促音
10         "type": "形",                        //词性, 可能为空
11         "exsent": [                          //解释及例句的词典的列表, 可能为空
12             {
13                 "explan": "红色的",            //解释
14                 "sentJap": ["時が虚しく過ぎる。"], //例句日文列表, 可能
15                 "sentChi": ["虚度时光。"]      //例句中文列表
16             },
17             {
18                 .....
19             }
20         ]
21     }
22 }

```

## 失败响应

返回体:

```

1  {
2      "code": "405",
3      "message": "Method not allowed"
4  }

```



## /list

获取针对训练队列

## POST

### 请求体

```
1 {
2     "url": "43.138.68.47:8000/list",    //域名
3     "method": "POST",                  //请求方式
4     data{
5         "openid": "dskadhkskada",    //用户唯一标识
6         "type":2, //为0-10之间的数字，对应0型音到8型音、促音、长音
7     }
8 }
```

### 成功响应

返回格式 ()

```
1 {
2     "code": 0,
3     "info": "Success in word searching",
4     "medicine": [    //单词详情列表
5         {'name': 'シンクロメッシュ【しんくろめっしゅ】', 'accent': -1, 'kana':
6         ['し', 'ん', 'く', 'ろ', 'め', 'っ', 'しゅ'], 'taka': [0, 1, 1, 1, 1, 1, 1],
7         'wave': [0, 1, 1, 1, 1, 2, 1], 'type': '', 'exsent': [{'explan': '同步啮合的；齿
8         轮的同时咬合装置；同步配合；同步啮合', 'sentJap': [], 'sentChi': []}]},
9         {'name': '一発必中【いっぱつひっちゅう】', 'accent': -1, 'kana': ['い',
10        'っ', 'ぱ', 'っ', 'ひ', 'っ', 'ちゅ', 'う'], 'taka': [0, 1, 1, 1, 1, 1, 1, 1],
11        'wave': [0, 2, 1, 1, 1, 2, 1, 1], 'type': '', 'exsent': [{'explan': '1回きりの勝
12        負は必ず命中すること。', 'sentJap': [], 'sentChi': []}, {'explan': '一举成名。',
13        'sentJap': [], 'sentChi': []}, {'explan': '一炮而响。', 'sentJap': [], 'sentChi':
14        []}]}, {'name': 'チップボード【ちっぷぼーど】', 'accent': -1, 'kana': ['ち', 'っ',
15        'ぷ', 'ぼ', 'ー', 'ど'], 'taka': [0, 1, 1, 1, 1, 1], 'wave': [0, 2, 1, 1, 1,
16        1], 'type': '', 'exsent': [{'explan': '木屑板', 'sentJap': [], 'sentChi': []}]},
17        {'name': '雨降って地固まる【あめふってじかたまる】', 'accent': -1, 'kana':
18        ['あ', 'め', 'ふ', 'っ', 'て', 'じ', 'か', 'た', 'ま', 'る'], 'taka': [0, 1, 1, 1,
19        1, 1, 1, 1, 1, 1], 'wave': [0, 1, 1, 2, 1, 1, 1, 1, 1, 1], 'type': '',
20        'exsent': [{'explan': '谚语：雨后土地变坚固。比喻破坏之后始有建设，战争之后必有和平，不
21        打不相交等意。', 'sentJap': [], 'sentChi': []}]},
22    ]
23 }
```

```

8      {'name': '速効【そっこう◎】', 'accent': 0, 'kana': ['そ', 'っ', 'こ',
'う'], 'taka': [0, 1, 1, 1], 'wave': [0, 2, 1, 1], 'type': '名', 'exsent':
[{'explan': '速効。（ききめがはやく現れること。）', 'sentJap': [], 'sentChi': []}]},

9      {'name': 'ウォークールッキー【うおーきーるっきー】', 'accent': -1, 'kana':
['うお', 'ー', 'き', 'ー', 'る', 'っ', 'き', 'ー'], 'taka': [0, 1, 1, 1, 1, 1, 1,
1], 'wave': [0, 1, 1, 1, 1, 2, 1, 1], 'type': '', 'exsent': [{'explan': '手提式
电视摄影机, 便携式摄像机', 'sentJap': [], 'sentChi': []}]},
10     {'name': '秘密結社【ひみつけっしゃ④】', 'accent': 4, 'kana': ['ひ', 'み',
'っ', 'け', 'っ', 'しゃ'], 'taka': [0, 1, 1, 1, 0, 0], 'wave': [0, 1, 1, 1, 2,
0], 'type': '名', 'exsent': [{'explan': '秘密結社。（外部に対し、その存在・構成人員・
目的・活動などを秘匿している団体。）', 'sentJap': ['彼を秘密結社に加入させる。'],
'sentChi': ['使他加入秘密結社。']}]},
11     {'name': 'オポッサム【おぼっさむ②】', 'accent': 2, 'kana': ['お', 'ぼ',
'っ', 'さ', 'む'], 'taka': [0, 1, 0, 0, 0], 'wave': [0, 1, 2, 0, 0], 'type':
'名', 'exsent': [{'explan': 'opossum〈动〉负鼠。（有袋目オポッサム科の哺乳類の総称。ネ
ズミに似た外形で、キタオポッサムは、体長40～65センチ、尾長25～50センチ。体の上面は灰色の霜
降り。雑食性で夜行性。木登りがうまい。）', 'sentJap': [], 'sentChi': []}]},
12     {'name': 'オフリミット【おふりみっと】', 'accent': -1, 'kana': ['お', 'ふ',
'り', 'み', 'っ', 'と'], 'taka': [0, 1, 1, 1, 1, 1], 'wave': [0, 1, 1, 1, 2,
1], 'type': '', 'exsent': [{'explan': '界限外; 范围外; 限度外; 禁止入内',
'sentJap': [], 'sentChi': []}]}, {'name': '仏閣【ぶつかく◎】', 'accent': 0,
'kana': ['ぶ', 'っ', 'か', 'く'], 'taka': [0, 1, 1, 1], 'wave': [0, 2, 1, 1],
'type': '名', 'exsent': [{'explan': '佛堂, 寺院。（寺の建物 。寺院。）', 'sentJap':
['神社仏閣。'], 'sentChi': ['神社佛阁。']}]},
13     ]
14 }

```

## 失败响应

返回体:

```

1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }

```

## /read

用于发音评测。

## POST

## 请求体

```
1 {
2     "url": "43.138.68.47:8000/read",    //登录域名
3     "method": "POST",                  //请求方式
4     data{
5         "openid": "dskadhkskada",      //用户唯一标识
6         "file": ...,                    //录音文件，MP3格式
7         "wave": [0, 1, 1, 0],           //正确声调序列，2表示高音
8         "word": "赤い【あかい◎】"
9     }
10
11 }
```

## 成功响应

类似以下格式，后端确定

```
1 {
2     "code": 0,
3     "info": "Success in word analyzing",
4     "accent": [0, 1, 1, 0]
5 }
6
```

## 失败响应

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

/setting

/headIcon

POST

请求体

```
1 {
```

```
2     "url": "43.138.68.47:8000/setting/headIcon/",    //登录域名
3     "method": "POST",                               //请求方式
4     data {
5         "openid": "dskadhkskada",    //用户唯一标识
6         "file": ...,                //头像文件
7         "name":带后缀名的url,
8     }
9 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "Headicon successfully changed!",
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /nickname

### POST

### 请求体

```
1 {
```

```
2     "url": "43.138.68.47:8000/setting/nickname/",           // 访问路由
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6         "nickname": "new_nickname"           // 新用户昵称
7     }
8 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "Nickname successfully changed!"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /timeline

### POST

### 请求体

```
1 {
2     "url": "43.138.68.47:8000/setting/timeline/",           // 访问路由
```

```
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6         "timeline": 12          // 新保存时间
7     }
8 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "Timeline successfully changed!"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /record

### POST

### 请求体

```
1 {
2     "url": "43.138.68.47:8000/setting/record/",          // 访问路由
3     "method": "POST",
```

```
4     data {
5         "openid": "xxxx",
6         "recordOn": "false"           // 新保存时间  内容为"false" or "true"
7     }
8 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "Record button successfully changed!"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

### 发送参数recordOn错误

```
1 {
2     "code": "400",
3     "message": "Bad request: invalid request argument",
4 }
```

/wb

这是与单词本相关的路由。

## /create

创建单词本,id编号从1开始, 索引还是0开始

新增默认填充

```
1 {
2     id: this.data.wordbooks.length + 1, // 生成新的ID
3     type:1, // 0表示私密, 1表示公开
4     name: `未命名单词本${this.data.wordbooks.length + 1}`,
5     intro: "这是单词本的介绍",
6     coverUrl: "../images/dictimage/dictimage3.png"
7 }
```

## POST

请求体

```
1 {
2     "url": "43.138.68.47:8000/wb/create/", // 访问路由
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6     }
7 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "New wordbook successfully created!"
4 }
```

## 失败响应

请求方式错误

```
1 {
2     "code": "405",
```



```
3     "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /info

更改单词本名称与介绍

POST

请求体

```
1 {
2     "url": "43.138.68.47:8000/wb/info/",           // 访问路由
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6         "index": 2,                               // 单词本序号
7         "name": "xxx",                            // 单词本名称
8         "intro": "xxx",                           // 单词本介绍
9     }
10 }
```

成功响应

```
1 {
2     "code": 0,
3     "message": "Wordbook information successfully changed!"
4 }
```

失败响应

请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

单词本不存在

```
1 {
2   "code": "404",
3   "message": "Wordbook not found",
4 }
```

## /type

更改单词本公开属性

## POST

请求体

```
1 {
2   "url": "43.138.68.47:8000/wb/type/",           // 访问路由
3   "method": "POST",
4   data {
5     "openid": "xxxx",
6     "index": 2,                                   // 单词本序号
7     "type": 0,                                    // 单词本是否公开 0或1
8   }
9 }
```

成功响应

```
1 {
2   "code": 0,
3   "message": "Wordbook information successfully changed!"
4 }
```

## 失败响应

请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

单词本不存在

```
1 {
2   "code": "404",
3   "message": "Wordbook not found",
4 }
```

## /remove

删除单词本

POST

请求体

```
1 {
2   "url": "43.138.68.47:8000/wb/remove/",           // 访问路由
3   "method": "POST",
4   data {
```

```
5      "openid": "xxxx",
6      "index": 2,           // 单词本序号
7  }
8 }
```

## 成功响应

```
1 {
2   "code": 0,
3   "message": "Wordbook removed successfully!"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

### 单词本不存在

```
1 {
2   "code": "404",
3   "message": "Wordbook not found",
4 }
```

## /image

## /get

得到单词本封面图片

## POST

```
1 {
2     "url": "43.138.68.47:8000/wb/image/get/",           // 访问路由
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6         "index": 2,           // 单词本序号
7     }
8 }
```

返回：图像的二进制流，图片名为后端存储的文件名

## /set

修改单词本封面图片

## POST

```
1 {
2     "url": "43.138.68.47:8000/wb/image/set/",           //登录域名
3     "method": "POST",           //请求方式
4     data {
5         "openid": "dskadhkskada",           //用户唯一标识
6         "index": 2,           // 单词本序号
7         "image": ...,           //封面文件
8         "name":带后缀名的url,
9     }
10 }
```

返回：code:0 成功

## /word

这是与单词添加与删除相关的路由。

## /add

添加单词到单词本

## POST

## 请求体

```
1 {
2   "url": "43.138.68.47:8000/word/add/",      // 访问路由
3   "method": "POST",
4   data {
5     "openid": "xxxx",
6     "index": 2,          // 单词本序号
7     "word": "あかい..." // 单词
8   }
9 }
```

## 成功响应

```
1 {
2   "code": 0,
3   "message": "Word added successfully!"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

### 单词本不存在

```
1 {
```

```
2     "code": "404",
3     "message": "Wordbook not found",
4 }
```

单词已经添加

```
1 {
2     "code": 0,
3     "message": "Word already exists",
4 }
```

## /remove

将单词从单词本中删除

POST

请求体

```
1 {
2     "url": "43.138.68.47:8000/word/remove/",           // 访问路由
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6         "index": 2,                                     // 单词本序号
7         "word": "あかい..."                          // 单词
8     }
9 }
```

成功响应

```
1 {
2     "code": 0,
3     "message": "Word removed successfully!"
4 }
```

失败响应

请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

单词本不存在

```
1 {
2   "code": "404",
3   "message": "Wordbook not found",
4 }
```

单词不存在

```
1 {
2   "code": "404",
3   "message": "Word doesn't exist",
4 }
```

## /friends

这是与好友相关的路由。

## /list

通过openid获取关注的用户信息列表。

## POST

请求体

```
1 {
2   "url": "43.138.68.47:8000/friends/list/", // 访问路由
```



```
3     "method": "POST",
4     data {
5         "openid": "xxxx",
6     }
7 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "data": {
4         result:[
5             ["1234","萝莉控","111.png",0],..... // 关注用户的UID, 昵称, 头像
6             名 (头像路径),是否被关注0为否
7         ]
8     }
9 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /headicon

通过uid获取关注的用户头像。

## POST

### 请求体

```
1 {
2   "url": "43.138.68.47:8000/friends/headicon/",           // 访问路由
3   "method": "POST",
4   data {
5     "openid": "",
6     "uid": "xxxx",
7   }
8 }
```

## 成功响应

```
1 {
2   "code": 0,
3   "data": 文件流，类似登录获取头像
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

## /unfollow

通过uid取消关注。

## POST

### 请求体

```
1 {
2     "url": "43.138.68.47:8000/friends/unfollow/", // 访问路由
3     "method": "POST",
4     data {
5         "openid": "",
6         "uid": "xxxx",
7     }
8 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "successfully unfollow"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /follow

通过uid关注。

## POST

### 请求体

```
1 {
2     "url": "43.138.68.47:8000/friends/follow/", // 访问路由
3     "method": "POST",
4     data {
5         "openid": "",
6         "uid": "xxxx",
7     }
8 }
```

## 成功响应

```
1 {
2     "code": 0,
3     "message": "successfully follow"
4 }
```

## 失败响应

### 请求方式错误

```
1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }
```

### 用户不存在

```
1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }
```

## /uidsearch

通过uid搜索用户。

## POST

### 请求体

```

1 {
2     "url": "43.138.68.47:8000/friends/uidsearch/", // 访问路由
3     "method": "POST",
4     data {
5         "openid": "",
6         "uid": "xxxx",
7     }
8 }

```

## 成功响应

```

1 {
2     "code": 0,
3     "data": {
4         result:[
5             ["1234","萝莉控","111.png",0],..... // 关注用户的UID，昵称，头像
6             名（头像路径），是否被关注0为否
7         ]
8     }

```

## 失败响应

### 请求方式错误

```

1 {
2     "code": "405",
3     "message": "Method not allowed"
4 }

```

### 用户不存在

```

1 {
2     "code": "401",
3     "message": "User Unauthorized",
4 }

```

## /namesearch

通过昵称搜索用户。

## POST

### 请求体

```
1 {
2   "url": "43.138.68.47:8000/friends/namesearch/", // 访问路由
3   "method": "POST",
4   data {
5     "openid": "",
6     "name": "xxxx",
7   }
8 }
```

### 成功响应

```
1 {
2   "code": 0,
3   "data": {
4     result:[
5       ["1234","萝莉控","111.png",0],..... // 关注用户的UID，昵称，头像
6       名（头像路径），是否被关注0为否
7     ]
8   }
9 }
```

### 失败响应

#### 请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

#### 用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

## /info

通过uid搜索用户。

### POST

#### 请求体

```
1 {
2   "url": "43.138.68.47:8000/friends/info/",           // 访问路由
3   "method": "POST",
4   data {
5     "openid": "",
6     "uid": "xxxx",
7   }
8 }
```

#### 成功响应

```
1 {
2   "code": 0,
3   "data": {
4     info:{
5       nickname:"",
6       following:0,           // 是否已经关注, 0未关注
7     },
8     wordbooks:[
9       {
10        id: 1,
11        following:0,          // 是否已经关注, 0未关注
12        name: "单词本1",
13        type:1,
14        intro: "这是单词本的介绍",
15        coverUrl: "../images/dictimage/dictimage1.png",
16        words: [
17          "太陽光発電【たいようこうはつでん⑧】 ",
18        ],]
19 }
```

#### 失败响应

请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

## /subscribe

通过uid和单词本id订阅。

POST

请求体

```
1 {
2   "url": "43.138.68.47:8000/friends/subscribe/", // 访问路由
3   "method": "POST",
4   data {
5     "openid": "",
6     "uid": "xxxx",
7     "id": "",
8   }
9 }
```

成功响应

```
1 {
2   "code": 0,
3   "message": "successfully subscribe"
4 }
```

失败响应

请求方式错误



```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

## /unsubscribe

通过uid和单词本id取消订阅。

POST

请求体

```
1 {
2   "url": "43.138.68.47:8000/friends/unsubscribe/", // 访问路由
3   "method": "POST",
4   data {
5     "openid": "",
6     "uid": "xxxx",
7     "id": "",
8   }
9 }
```

成功响应

```
1 {
2   "code": 0,
3   "message": "successfully subscribe"
4 }
```

失败响应

请求方式错误

```
1 {
2   "code": "405",
3   "message": "Method not allowed"
4 }
```

用户不存在

```
1 {
2   "code": "401",
3   "message": "User Unauthorized",
4 }
```

## /wbcover

### POST

```
1 {
2   "url": "43.138.68.47:8000/friends/wbcover/",           // 访问路由
3   "method": "POST",
4   data {
5     "openid": "",
6     "uid": "xxxx",           // 好友uid
7     "id": "",               // 好友单词本id
8   }
9 }
```

返回文件名以及二进制文件

## 小程序相关

```
1 AppID(小程序ID):wx5ef6067498a5d631
2 AppSecret:f72baea608e953e3d5850999d9667309
```

## 本地用户数据

存储在globalData中，登录时拉取。

## 后端返回格式

```
1 {
2   file:头像,
3   info:{
4     openid:"",
5     nickname:"",
6     timeline:10,           //首选项, 保存期限等
7     recordOn:"true",
8     wbnun:11,
9     followees:[uid,],
10  },
11  flwbs:[
12    {
13      owner_uid: "xxx",
14      id: 3,
15      name: "单词本1",
16      intro: "这是单词本的介绍",
17      coverUrl: "../..../images/dictimage/dictimage1.png",
18      words: [
19        "太陽光発電【たいようこうはつでん⑧】 ",
20        "太陽光発電【たいようこうはつでん⑧】 ",
21        // 继续添加更多单词
22      ]
23    },
24    wordbooks:[
25      {
26        id: 1,
27        name: "单词本1",
28        intro: "这是单词本的介绍",
29        coverUrl: "../..../images/dictimage/dictimage1.png",
30        words: [
31          "太陽光発電【たいようこうはつでん⑧】 ",
32          "太陽光発電【たいようこうはつでん⑧】 ",
33          // 继续添加更多单词
34        ]
35      },
36      {
37        id: 2,
38        name: "单词本2",
39        intro: "这是单词本的介绍",
40        coverUrl: "../..../images/dictimage/dictimage2.png",
41        words: [
42          "太陽光発電【たいようこうはつでん⑧】 ",
43          "太陽光発電【たいようこうはつでん⑧】 ",
44        ]
45      },
46    ],
```

```

47     wordHistory:[
48         ["太阳光発電【たいようこうはつでん⑧】 ",
49         "2003/11/20",           //最后查询的时间
50         3                       //查询次数
51     ],
52 ],
53     readHistory:[
54         ["太阳光発電【たいようこうはつでん⑧】 ",
55         "2003/11/20",           //最后查询的时间
56         "[0,1,1,1,1,1,1,1,0,0]",           //最后朗读读音记录
57         1,                       //最后朗读正确性
58     ],
59 ],
60 }

```

## 前端全局globalData

```

1  globalData{
2      userHead:"lujing",
3      info:{
4          openid:"",
5          uid:"",
6          nickname:"",
7          timeline:10,           //首选项，保存期限等
8          recordOn:"true",
9          wbnum:11,
10         followees:[uid,],
11     },
12     flwbs:[
13         {
14             uid:"",
15             id:"",
16             name: "单词本1",
17             intro: "这是单词本的介绍",
18             coverUrl: "../..../images/dictimage/dictimage1.png",
19             words: [
20                 "太阳光発電【たいようこうはつでん⑧】 ",
21                 "太阳光発電【たいようこうはつでん⑧】 ",
22                 // 继续添加更多单词
23             ]
24         },
25     ],
26     wordbooks:[
27         {
28             id: 1,
29             type:0,               // 单词本属性，0表示私密，1表示公开可订阅

```

```

29     name: "单词本1",
30     intro: "这是单词本的介绍",
31     coverUrl: "../../images/dictimage/dictimage1.png",
32     words: [
33         "太陽光発電【たいようこうはつでん⑧】 ",
34         "太陽光発電【たいようこうはつでん⑧】 ",
35         // 继续添加更多单词
36     ]
37 },
38 {
39     id: 2,
40     name: "单词本2",
41     intro: "这是单词本的介绍",
42     coverUrl: "../../images/dictimage/dictimage2.png",
43     words: [
44         "太陽光発電【たいようこうはつでん⑧】 ",
45         "太陽光発電【たいようこうはつでん⑧】 ",
46     ]
47 },
48 ],
49 wordHistory:[
50     ["太陽光発電【たいようこうはつでん⑧】 ",
51     "2003/11/20",          //最后查询的时间
52     3                      //查询次数
53     ],
54 ],
55 readHistory:[
56     ["太陽光発電【たいようこうはつでん⑧】 ",
57     "2003/11/20",          //最后查询的时间
58     "[0,1,1,1,1,1,1,1,0,0]",//最后朗读结果
59     1,                    //最后朗读正确性
60     ],
61 ],
62 }

```

## 前端图片缓存

以单词本为例，图片名称（服务器格式）存放在coverUrl当中。处理后真实路径放在realUrl中用于页面渲染。

```

1 // 查看当前单词本封面缓存位置
2 let localUrl = wx.getStorageSync("wb" + wbs[i].uid + 'id' + wbs[i].id);
3 // 生成标准封面缓存路径

```

```

4 const target = wx.env.USER_DATA_PATH + "/" + wbs[i].coverUrl;
5     // 检查是否更新了封面
6     if (localUrl != target) {
7 // 检查缓存位置文件是否存在
8 if (localUrl) {
9     fs.access({

```

## 发音界面read

传参时list=true表示是训练队列，list=false表示是单个单词，后面分别接单词详情字典列表和单词详情字典。

```

1 //队列传参
2 testread: function () {
3     const exam = [{
4         "name": "赤い【あかい◎】",
5         "accent": 0,
6         "kana": ["あ", "か", "い"],
7         "taka": [0, 1, 1],
8         "wave": [0, 1, 1]
9     },
10    {'name': '太陽光発電【たいようこうはつでん⑧】', 'accent': 8, 'kana': ['た',
    'い', 'よ', 'う', 'こ', 'う', 'は', 'つ', 'で', 'ん'], 'taka': [0, 1, 1, 1, 1, 1,
    1, 1, 0, 0], 'wave': [0, 1, 1, 1, 1, 1, 1, 1, 0, 0], 'type': '名', 'exsent':
    [{'explan': '太陽の光エネルギーを電気エネルギーに変換する発 電方式。太陽電池など。',
    'sentJap': [], 'sentChi': []}]}
11 ]
12     let para = JSON.stringify(exam);
13     wx.navigateTo({
14         url: '../read/read?list=true&info=' + encodeURIComponent(para)
15     });
16 },
17 //单个单词传参
18 testread2: function () {
19     const exam = {
20         "name": "赤い【あかい◎】",
21         "accent": 0,
22         "kana": ["あ", "か", "い"],
23         "taka": [0, 1, 1],
24         "wave": [0, 1, 1]
25         .....
26     }
27     let para = JSON.stringify(exam);
28     wx.navigateTo({
29         url: '../read/read?list=false&info=' + encodeURIComponent(para)

```

```

30     });
31   },
32   //发音结果
33   testread3: function () {
34     const exam = {
35       "name": "赤い【あかい◎】",
36       "accent": 0,
37       "kana": ["あ", "か", "い"],
38       "taka": [0, 1, 1],
39       "wave": [0, 1, 1]
40     }
41     let para = JSON.stringify(exam);
42     let result=JSON.stringify([1,0,0]);
43     wx.navigateTo({
44       url: '../read/read?list=false&info=' +
         encodeURIComponent(para)+'&result='+encodeURIComponent(result),
45     });
46   },

```

## 词典相关

### 词典格式

```

1  词典存储格式json
2  键值对例如
3  "赤い【あかい◎】":{
4      "name":"赤い【あかい◎】",           //本体名字
5      "accent":0,                          //未标声调时用-1
6      "kana":["あ","か","い"],            //单拍平假名列表
7      "taka":[0,1,1]                      //真实假名高低,不显示促音
8      "wave":[0,1,1],                    //每拍假名的高低,2表示促音
9      "type":"形",                        //词性,可能为空
10     "exsent":[                          //解释及例句的词典的列表,可能为空
11         {
12             "explan":"红色的",          //解释
13             "sentJap":["時が虚しく過ぎる。"], //例句日文列表,可能为空
14             "sentChi":["虚度时光。"]      //例句中文列表
15         },
16         {
17             .....
18         }
19     ]
20 }

```

## 查询方式

```
1 使用dic_func.py中的Dic类
2      '''
3      字典类
4      只要创建一个示例，然后调用函数就行
5      funcs:
6          search_results
7          get_detail
8          medical_list
9      '''
```

### 1. 单词（假名、汉字，甚至声调圈号数字、外来语原型）搜索

就是搜索含有该部分的单词列表

```
1 search_results
2     返回样例，词典中键字符串序列
3     ['鼻を明かす【はなをあかす】',
4     '鼻元思案【はなもとじあん】',
5     '鼻息が荒い【はないきがあらう】',
6     'ICカード【あいしーカード】']
```

### 2. 单词详情获取

根据单词名获得单词详情

```
1 get_detail
2     返回样例，参见词典格式
3     {'name': 'aconitine【あこにちん】',
4     'accent': -1,
5     'kana': ['あ', 'こ', 'に', 'ち', 'ん'],
6     'taka': [0, 1, 1, 1, 1],
7     'wave': [0, 1, 1, 1, 1],
8     'type': '',
9     'exsent': [{'explan': '〈化〉乌头碱.', 'sentJap': [], 'sentChi': []}]}
```

### 3. 训练队列获取



```
1  medical_list(self,type):
2      '''
3      param:type=k, 为0-10之间的数字，对应0型音到8型音、促音、长音
4      return:字典列表，包含最多十个对应类型的单词详情字典
5      '''
```

## 语音分析相关

### 需求库

运行环境:Ubuntu 20.04 python3

可能需要的库：（因为我自己之前下载过，如果按照以下部分下载了没有用的话看报错信息找库下载）

```
import seaborn
```

```
sudo apt update
```

```
sudo apt install ffmpeg
```

```
pip3 install pydub
```

### 调用方法

运行方法：python3 final.py your-media-path your-json-path

后两个分别是录音文件地址和包含“accents”项的json文件地址，如果在相同文件夹下直接输入文件名即可；

脚本运行完成后会生成以下文件：

yourname.wav（如果源文件不是wav格式）

yourname.json（初次分析和平滑后的语音图）

accept.json（接收到的json包）

accept1.json（返回的包）

如果不想跑脚本，调用final.py中的main也可，参数仍然是 your-media-path ， your-json-path

返回结果为accept1.json，格式为：“accents”:[0,1,2,1]类似。