

## TrojanNow Server

1. URL: <https://trojannow-server.herokuapp.com/>
2. 接口总览

| 功能         | 接口                    | 说明                                     |
|------------|-----------------------|--|
| 注册         | /index/signup         | 用户注册                                   |
| 登录         | /index/login          | 用户登录                                   |
| 获取微博列表     | /blog/list            | 获取微博，分为公共和匿名模式                         |
| 发表微博       | /blog/new             | 发表微博，分为公共和匿名权限，并且可以记录环境信息              |
| 查看评论       | /comment/list         | 获取所选微博id的相关评论                          |
| 发表评论       | /comment/new          | 评论所选微博                                 |
| 查看信息列表     | /message/list         | 获取最新发送或接收的信息列表（类似微信主信息页面内容）            |
| 查看信息       | /message/messages     | 获取与某个好友的所有消息记录                         |
| 发送信息       | /message/new          | 发送信息                                   |
| 信息通知       | /message/notification | 通过安卓service调用，用于检测是否收到新消息以及其数目         |
| 查看粉丝或关注的人  | /friend/list          | 获取粉丝列表或你关注的人列表，（获取的都为他们的用户昵称user_name） |
| 计算粉丝或关注人数目 | /friend/sum           | 获取粉丝总数或关注的人总数，返回为一个数字                  |
| 关注某人       | /friend/add           | 添加对某人的关注                               |
| 取消关注某人     | /friend/delete        | 取消对某人的关注                               |

### 3. 具体信息

下面为具体POST参数以及返回的JSON数据，也可通过chrome postman应用进行查看JSON。

#### (1) 注册登录模块

| post参数名（注册）  | 说明           |
|--------------|--------------|
| user_account | string, 用户账号 |
| user_pwd     | string, 用户密码 |
| user_name    | string, 用户昵称 |
| user_email   | string, 电邮   |

登录部分 POST参数为：user\_account, user\_pwd

返回JSON格式：(注册)

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": ""
}
```

返回JSON格式：(登录)

```
{
  "status": "true",
  "message": "login successfully!",
  "result": {
    "user_account": "shenzui",
    "user_name": "jim",
    "user_email": "1234567@qq.com",
    "user_pic": "",
    "user_sid": "7862ag17okn7jg0gufvo33f1c1"
  }
}
```

## (2)微博模块

### a. 获取微博列表

| post参数   | 说明   |
|----------|--|
| last_id  | int, 用于分页, 记录当前页的最后一条微博的id, 第一次查询时应赋值为0            |
| list_num | int, 用于控制每次查询的微博条数                                 |
| level    | int, 用于区分模式, level = 1时为public模式, level = 2时为匿名模式。 |
| user_x   | int, 当level = 2时的设备当前的x坐标, 当level = 1时可忽略user_x    |
| user_y   | int, 当level = 2时的设备当前的y坐标, 当level = 1时可忽略user_y    |

返回JSON格式:

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": ""
}
```

### b. 发表微博

| post参数      | 说明   |
|-------------|--|
| mb_uaccount | string, 你(当前使用者)的账号                              |
| mb_content  | string, 微博内容                                     |
| mb_time     | string, 发送微博时的时间, 统一格式(比如 04-13-2015 23:20)      |
| mb_level    | int, 选择发送微博的模式, 为1时关闭坐标功能, 为2时需获取当前位置x, y的坐标     |
| mb_location | string, 设备当前所在的具体位置, 文字信息, 可选参数                  |
| mb_degree   | int, 通过设备传感器获取的温度, 可选参数                          |
| mb_weather  | string, 通过其他天气网站接口获取的天气文字信息, 可选参数                |
| mb_x        | int, 当前设备的x坐标, level = 1时忽略mb_x, level = 2时为必选参数 |
| mb_y        | 同mb_x  |

返回JSON格式：

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": ""
}
```

### (3) 评论模块

#### a. 查看某条微博的评论

| post参数 | 说明           |
|--------|--------------|
| mb_id  | 待查看评论的微博的id号 |

返回JSON格式：

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": [
    {
      "comt_id": "评论的id号",
      "comt_mid": "被评论的微博id",
      "comt_uaccount": "评论人账号",
      "comt_content": "评论内容",
      "comt_time": "评论时间",
      "comt_uname": "评论人昵称"
    },
    {
      .....
    }
  ]
}
```

## b. 发表评论

| post参数        | 说明                   |
|---------------|----------------------|
| comt_mid      | int, 评论的微博id         |
| comt_uaccount | string, 你（当前使用者）的账号  |
| comt_content  | string, 评论内容         |
| comt_time     | string, 评论时间, 统一格式时间 |

返回JSON格式:

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": ""
}
```

## (4) 消息模块

### a. 查看消息列表

| post参数       | 说明          |
|--------------|-------------|
| user_account | 你（当前使用者）的账号 |

返回JSON格式:

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": [
    {
      "message_id": "信息的id号",
      "message_from": "发信人账号",
      "message_to": "收信账号",
      "message_content": "信息内容",
      "message_time": "发信息时间时间",
      "message_status": "收信人是否已读, 0为未读, 1为已读",
      "message_usr_name": "收信人昵称"
    },
    {
      .....
    }
  ]
}
```

## b. 查看信息

| post参数         | 说明          |
|----------------|-------------|
| user_account   | 你（当前使用者）的账号 |
| friend_account | 收信人账号       |

返回JSON格式：

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": [
    {
      "message_id": "信息的id号",
      "message_from": "发信人账号",
      "message_to": "收信账号",
      "message_content": "信息内容",
      "message_time": "发信息时间时间"
    },
    {
      .....
    }
  ]
}
```

## c. 发送消息

| post参数          | 说明          |
|-----------------|-------------|
| message_from    | 你（当前使用者）的账号 |
| message_to      | 收信人账号       |
| message_content | 消息内容        |
| message_time    | 发消息时间       |

返回JSON格式：

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": ""
}
```

#### d. 信息通知

| post参数       | 说明          |
|--------------|-------------|
| user_account | 你（当前使用者）的账号 |

返回JSON格式：

```
{
  "status": "true/false",
  "message": "提示信息",
  "result": { "new_message_num": "新消息数目" }
}
```

#### (5) 好友模块

##### a. 查看粉丝或关注的人

| post参数         | 说明  |
|----------------|---|
| user_account   | 当前用户的账号   |
| follow_fan_all | 字符串，当为follow时表示查看关注的人列表<br>当为fan时表示查看粉丝列表<br>当为all时表示查看相互关注的人列表（好友列表，目前未实现，待开发） |

返回JSON格式

```
{
  "status": "true",
  "message": "got friend list successfully!",
  "result": [
    {
      "friend_account": "你的账号",
      "friend_name": "粉丝或关注人的用户昵称"
    },
    {
      "friend_account": "你的账号",
      "friend_name": "粉丝或关注人的用户昵称"
    },
    {
      .....
    }
  ]
}
```

b. 获取粉丝或关注人总数

| post参数         | 说明   |
|----------------|--|
| user_account   | 当前用户的账号  |
| follow_fan_all | 字符串，当为follow时表示获取关注的人的总数<br>当为fan时表示获取粉丝的总数<br>当为all时表示查看互粉的总数（好友总数，目前未实现，待开发） |

返回JSON格式

```
{
  "status": "true",
  "message": "got num successfully!",
  "result": {
    "friend_account": "粉丝或关注人的数目"
  }
}
```

c. 关注某人，取消关注（/friend/add ， /friend/delete ）

| post参数         | 说明              |
|----------------|-----------------|
| user_account   | 使用者（你）的账号       |
| friend_account | 要关注的人/取消关注的人的账号 |

返回JSON格式

```
{
  "status": "true",
  "message": "Add successfully!",
  "result": ""
}
```

或

```
{
  "status": "true",
  "message": "Delete successfully!",
  "result": ""
}
```

ps: 当出现非法删除/添加时message返回repeated add 或repeated delete