



# 飞书puppet开发

基于开放 API 封装 Wechaty 接口下的飞书  
聊天机器人

汇报人：马田慧

指导老师：范蕊



# 目录

01

项目介绍

---

02

需求分析

---

03

项目进展

---

04

中后期进展

01

# 项目介绍



# 项目介绍

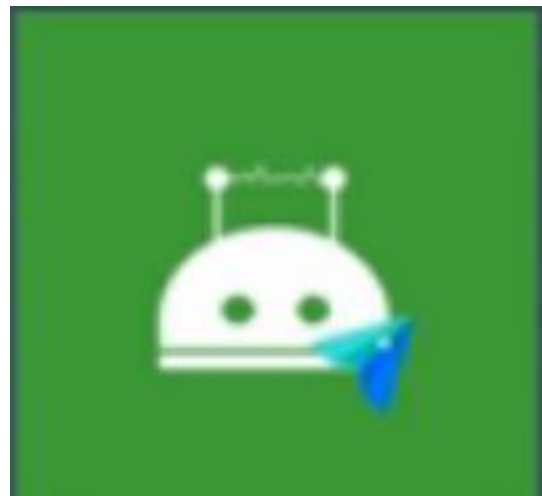
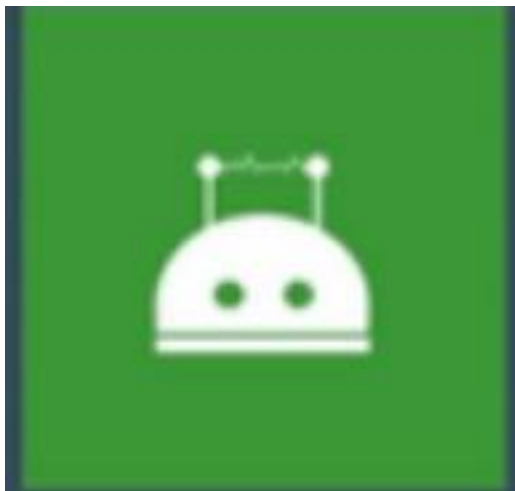
飞书：是字节跳动2016年自研的新一代一站式协作平台，是保障字节跳动全球五万人高效协作的办公工具；  
飞书官方提供API支持开发不同类型的机器人收发消息，是对国内Chatbot最开放的IM系统。





## 项目介绍

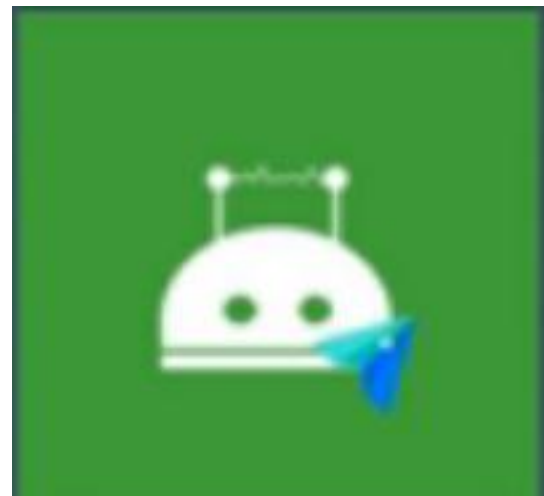
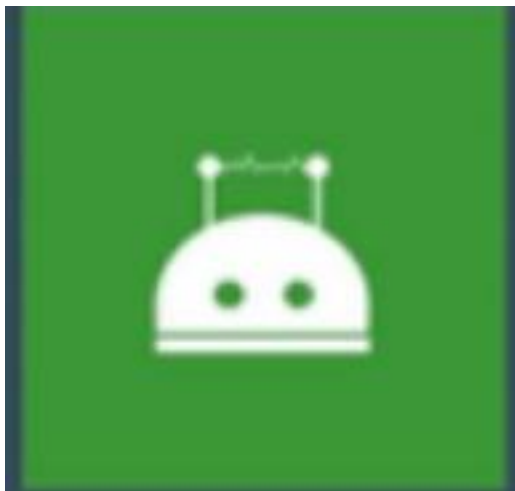
- wechaty是全球最大的开源聊天机器人框架，希望实现基于同一套应用层代码实现不同软件的聊天机器人。
- 飞书越来越开放便于企业管理，更符合wechaty的理念，更便于实现wechaty的高效功能。
- 我们希望实现通用的api设计，更方便今后的开发人员在飞书上使用wechaty





## 项目介绍

- 在暑假2020的第一代发展上，我们得到了wechaty的基本雏形，结合早起wechaty的应用开发出基本版本
- 但原有的api上飞书官方有更新
- 代码没有实现类包装，形式上有待整合
- 需要进一步完善相关的函数来实现代码复用，以及之后的npm包成熟发布



02

# 需求分析

Lorem Ipsum is simply dummy text of the printing and typesetting industry.



# 需求分析

原有框架



1

- 更新api版本, 熟悉飞书api和wechaty。.

2

- 对于其中函数进行完善更新

3

- 连接Contact、Message、Room等类, 尝试实现复用
- 撰写文档、example
- .发布npm包



03

# 项目进展

Lorem Ipsum is simply dummy text of the printing and typesetting industry.



# 项目进展

- 更新api版本，熟悉飞书api和wechaty。
- 实现之前未实现的函数

```
8     data: data ? data! : 'ding-dong',
9   }
10   this.emit('dong', eventDongPayload)
11 }
12
13 async contactPhone(): Promise<void> {
14   const _token = await this.getTenantAccessToken(this.appId, this.appSecret)
15   const payload: { [key: string]: string } = {}
16   const response = await axios({
17     url: 'https://open.feishu.cn/open-apis/contact/v3/users',
18     headers: {
19       Authorization: 'Bearer ' + _token,
20     },
21     method: 'GET',
22   })
23   if (response.data.code === 0) {
24     const contactlist = response.data.data.items
25     for (const i in contactlist) {
26       const obj = eval(contactlist[i])
27       payload[obj.name] = obj.mobile
28     }
29   } //TODO
30 }
31
32 contactCorporationRemark(): Promise<void> {
33   log.warn('There is no need to use this method \'logout\' in a lark bot.')
34   throw new Error('Method not implemented.')
35 }
36 }
```

正在下载 OpenDebugAD7 (Windows x86/x64)

```
async contactList(): Promise<string[]> {
  const _token = await this.getTenantAccessToken(this.appId, this.appSecret)
  console.log('tenantaccesstoken:' + _token);
  // let chat_id:string
  //let items:any=this.roomList()
  const response = await axios({
    headers: {
      Authorization: 'Bearer ' + _token,
    },
    method: 'GET',
    url: 'https://open.feishu.cn/open-apis/im/v1/chats/' + 'oc_c9245b00280a066d7d872671f96f859d' + '/members', //TODO
  })
  const authedEmployee: string[] = []
  if (response.data.code === 0) {
    const authedEmployee = response.data.data.items
    console.log('employee list:-----');
    console.log(authedEmployee);
    return authedEmployee
  } else {
    console.log('authedEmployee is empty');
    return authedEmployee
  }
}
```

protected contactRawPayload(): Promise<any> {



# 项目进展

- 更新api版本，熟悉飞书api和wechaty。
- 实现之前未实现的函数
- 官方网站测试

API调试台

Request

只看必填 ☒

访问凭证

t-6b47794ba2869d751f6fd9ad8d0822a044629df2

路径参数

chat\_id

oc\_c9245b00280a066d7d872671f96f859d

查询参数

发起调用

SDK示例

调用结果

API调用结果 200

```
{
  "code": 0,
  "data": {
    "has_more": false,
    "items": [
      {
        "member_id": "ou_84be6549e0170e62178fc9ce7d43f362",
        "member_id_type": "open_id",
        "name": "吴京京",
        "tenant_key": "2c6a32f81195575e"
      },
      {
        "member_id": "ou_5c288fd8a4306ad6ec71eab9c322eb18",
        "member_id_type": "open_id",
        "name": "高原",
        "tenant_key": "2c6a32f81195575e"
      },
      {
        "member_id": "ou_c3b4fac396850cc1a77c93d72d1da4a7",
        "member_id_type": "open_id",
        "name": "郑波",

```



## 项目进展

- 更新api版本，熟悉飞书api和wechaty。
- 实现之前未实现的函数

```
contactPayload (id: string, payload: ContactPayload): void
```

```
contactPayload (id: string): ContactPayload
```

```
contactPayload (id: string, payload?: ContactPayload): void | ContactPayload {  
  log.silly('Mocker', 'contactPayload(%s%s)', id, payload ? ', ' + JSON.stringify(payload) : '')
```

```
  if (payload) {  
    this.cacheContactPayload.set(id, payload)  
    return  
  }
```

```
  payload = this.cacheContactPayload.get(id)  
  if (!payload) {  
    throw new Error('no payload found for id ' + id)  
  }  
  return payload  
}
```

```
roomPayload (id: string, payload: RoomPayload): void
```

```
roomPayload (id: string): RoomPayload
```

```
roomPayload (id: string, payload?: RoomPayload): void | RoomPayload {  
  log.silly('Mocker', 'roomPayload(%s%s)', id, payload ? ', ' + JSON.stringify(payload) : '')
```

```
  if (payload) {  
    this.cacheRoomPayload.set(id, payload)  
    return  
  }
```

```
  payload = this.cacheRoomPayload.get(id)  
  if (!payload) {  
    throw new Error('no payload found for id ' + id)  
  }  
  return payload  
}
```

04

中后期进展



## 项目介绍

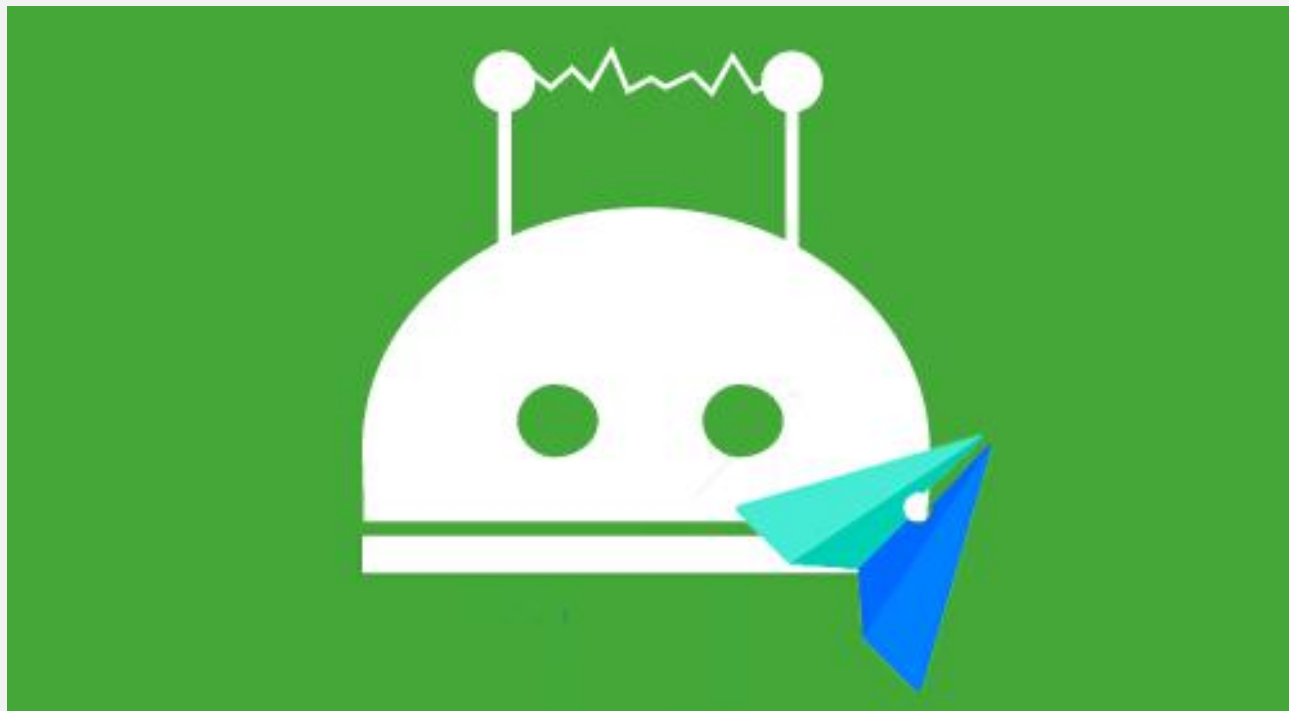
- 实现payload对接，对接wechaty的接口，参考
- 做代码测试
- 发布npm包
- 参考：<https://github.com/wechaty/wechaty-puppet-mock/blob/main/src/puppet-mock.ts>

```
async contactRawPayload (id: string): Promise<ContactPayload> {  
  log.verbose('Puppetlark', 'contactRawPayload(%s)', id)  
  let payload= this.messageStore[id]  
  if(payload){  
    this.cacheContactPayload.set(id, payload)  
  }  
  payload=this.cacheContactPayload.get(id)  
  if(!payload){  
    throw new Error('no payload found for id' + id)  
  }  
  return payload  
}  
  
async contactRawPayloadParser (payload: ContactPayload): Promise<ContactPayload> {  
  return payload  
}
```



## 实际应用

- 飞书方面权限配置要求
- 飞书的官方文档也在持续更新，需要关注官方文档的变化
- 可以将飞书方面不同的功能调用不同的api





Graduation thesis defense PPT template.

# 非常感谢您的观看

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

汇报人：马田慧

指导老师：范蕊