



# 期中答辩

---

技术实习生-余思荷  
北京句子科技有限公司  
16/12/2021





权限名称	scope	类别
获取授权码	code	授权事件
获取token	access_token	授权事件
刷新refresh_token	renew_refresh_token	授权事件
生成client_token	client_access_token	授权事件
静默授权码	login_id	授权事件
获取用户信息	user_info	授权事件
粉丝列表	fans.list	授权事件
关注列表	following.list	授权事件
回复视频评论	item.comment	webhook事件





# 期中进度





权限名称	scope	类别
评论列表	item.comment	webhook事件
评论回复列表	item.comment	webhook事件
接收视频评论事件	item.comment	webhook事件





模块	功能	能否支持引用	备注
ContactSelf	自我联系		官方接口未提供相关功能
Contact	联系人		可以提供联系人列表
Conversation	会话		官方接口未提供相关功能
Message	消息		实现dingdong逻辑
Room	群组		官方接口未提供相关功能
Room Invitation	进群邀请		官方接口未提供相关功能
Friendship	好友		官方接口未提供相关功能
Tag	标签		官方接口未提供相关功能





## puppet-mock可用模块中必须有的几个函数：

### Message:

- 1, messageSendText:  
发送文本消息
- 2, messageRawPayloadParser  
将原始payload进行一次转换➡得到wechaty标准化的数据结构
- 3, messageRawPayload  
(根据ID调用) 取出原始payload

### Contact:

- 1, contactList () 最重要的联系人列表
  - 2, contactRawPayloadParser
  - 3, contactRawPayload
- 



# Message\_messageSendText

```
override async messageSendText ( conversationId: string, text: string): Promise<string | void> {  
  const data = await axios.post(https://open.douyin.com/enterprise/im/message/send/?open\_id=9d34ba  
    headers:  
    { 'Content-Type': 'application/json' },  
    message_type: 'text', // 发送消息的种类, 其他 image图片; video视频; card卡片  
    to_user_id: '75742155-bddc-4982-87e6-6be93a5591a6',  
    // eslint-disable-next-line sort-keys  
    content: JSON.stringify({ text : 'dong' })),  
  })  
  // console.log("私信回复结束!")  
  return data.data  
}
```

给指定用户发送“dong”

向抖音开发平台发送post请求



## Message\_messageRawPayload

```
public override async messageRawPayload (messageId: string): Promise<any> {  
    return this.messageStore[messageId]  
}
```



将messageId存到messageStore中





# Message\_messageRawPayloadParser

```
public override async messageRawPayloadParser (rawPayload: any): Promise<MessagePayload> {  
  // tiktokmessage Payload -> Puppet message payload  
  const _types: { [key: string]: MessageType } = {  
    // file: MessageType.Attachment,  
    // image: MessageType.Image,  
    text: MessageType.Text,  
  }  
  
  const payload: MessagePayload = {  
    fromId: rawPayload.from_user_id,  
    id: rawPayload.open_id,  
    text: rawPayload.content.text,  
    timestamp: Date.now(),  
    toId: rawPayload.to_user_id, // TODO  
    type: _types[rawPayload.content.type],  
  }  
  
  return payload  
}
```


函数功能：将原始payload进行一次转换→得到wechaty标准化的数据结构

wechaty puppet指定的数据结构：

- 1, 消息来源
- 2, 机器人 id
- 3, 消息内容
- 4, 发送时间
- 5, 消息去向
- 6, 消息类型



## 私信接口只有抖音企业号才可以使用

1. 尝试沙盒环境调用私信接口
  2. 与开发平台技术人员咨询对接
  3. 申请基础企业号尝试调用私信接口
  4. 与开发平台技术人员咨询对接
  5. 与市场部的同事对接
  6. 正式企业号申请
  7. 抖音企业号人工审核
  8. 关联平台身份
  9. 申请成为自研开发者
  10. 申请一个有互动权限的网络应用(waiting~)
- 

# Contact\_contactList ()

```
override async contactList (): Promise<string[]> {  
  // eslint-disable-next-line prefer-const  
  let token = this.token  
  const User_fans = axios.get('https://open.douyin.com/fans/list/', {  
    params: {  
      open_id: this.id,  
      count: 1,  
      access_token: token  
    }  
  })  
  
  console.log(`-----调用getFans函数 → get请求粉丝列表-----\n`)  
  console.log("正在获取粉丝列表...")  
  console.log(`\n`)  
  console.log("    ##粉丝基本情况##    ")  
  console.log(`\n`)  
  console.log(`\n`)  
  const fansInfo = (await User_fans).data['data']  
  console.log(fansInfo)  
  return fansInfo  
}
```

## Contact\_contactList ()

```
await new Promise<void>((resolve) => {
  const interval = setInterval(() => {
    if (this.loginStatus === 'logged in') {
      clearInterval(interval)
      resolve()
    }
  }, 500)
})
}
```

t-tiktok.ts    TS ding-dong-bot.ts ×

> TS ding-dong-bot.ts > ...

// eslint-disable-next-line @typescript-eslint/no-floating-promises

**puppet.contactList()**



# 未来计划







计划一：预计三天内抖音开放平台会给出一个回复  
届时，预估下周可以顺利的实现puppet中的dingdong  
逻辑

计划二：完善代码，尽可能多的调用可用接口，熟悉  
puppet源码

计划三：用wechaty调用puppet中的ding-dong逻辑

