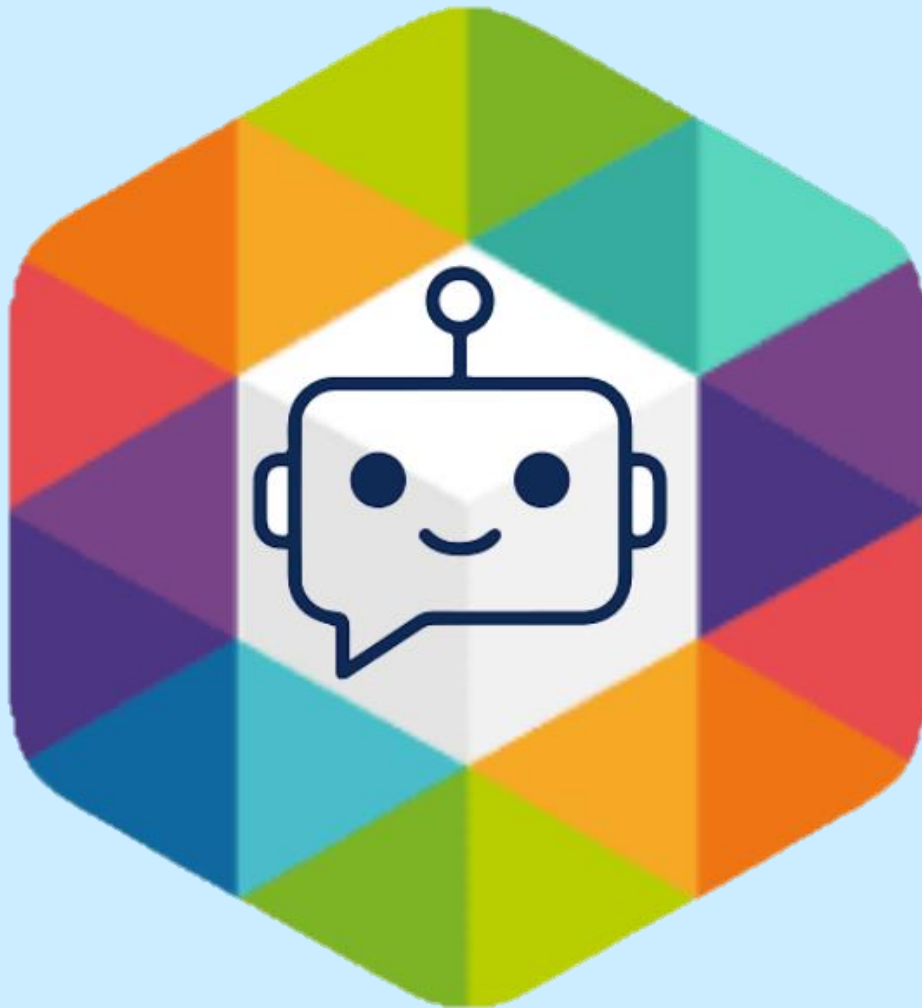


# بسم الله الرحمن الرحيم

مستندات کتابخانه روبیکا پای

نسخه ۵.۰.۳



## فهرست مطالب

مقدمه ----- ۵

فصل اول ----- ۸

۱. چت ها ----- ۹

۲. پیوی ها ----- ۱۱

۳. گروه ها ----- ۱۲

۴. کانال ها ----- ۱۳

۵. سرویس ها ----- ۱۴

۶. ربات ها ----- ۱۵

فصل دوم ----- ۱۶

۱. آشنایی با انواع فایل ها و پردازش آسان متن ها ----- ۱۷

۲. کار با فایل ها ----- ۲۰

## فهرست مطالب

### فصل سوم ----- ۲۱

۱. انواع رویداد ها در گروه ها ----- ۲۲
۲. کار با رویداد ها ----- ۲۳
۳. وب سوکت ----- ۲۵
۴. کار با وب سوکت ----- ۲۷

### فصل چهارم ----- ۳۰

۱. ابزار ها ----- ۳۱
۲. ابزار های پر استفاده ----- ۳۲

## فصل پنجم ----- ۳۳

۱. متد ها ----- ۳۴

۲. متد های پر استفاده ----- ۳۶

۳. متد های گروه ها ----- ۴۰

۴. متد های کانال ها ----- ۴۴

۶. مخاطبین

مقدمه

**کتابخانه روبیکا پای چیست؟** کتابخانه روبیکا پای یک فریمورک برای ایجاد یا ساخت ربات روی حساب کاربری در روبیکا میباشد، نه فقط حساب کاربری بلکه در آینده هم نیز قرار است که این کتابخانه یا فریمورک برای ساخت و میزبانی ربات های روبیکا هم استفاده شود!

**شناسه حساب کاربری یا آث ۱ چیست؟** این شناسه که از طرف سرور روبیکا برای حساب کاربری شما در نظر گرفته شده است نیز با این شناسه میتوان کل حساب کاربری را مدیریت کرد، درست مانند تلگرام البته با کمی تفاوت.

**هشدار: شناسه حساب کاربری (آث) خود را به هیچکس ندهید، حتی کسانی که در پیامرسان روبیکا خود را به جای پشتیبانی روبیکا معرفی میکنند.**

توکن ربات چیست؟ توکن ربات شناسه ای است که از طرف بات فادر ۲ یا پدر ربات نیز دریافت میشود و برای مدیریت ربات، به این شناسه نیاز است.

**هشدار: چون توکن ربات برای مدیریت ربات مورد نظر استفاده میشود نیز این توکن را نباید به شخصی داد.**

روبیکاپای ۳ تقریبا تمام متد های سرور روبیکا را فراهم کرده است تا شما عزیزان با استفاده از این مستندات یا داکيومنت بتوانید حساب کاربری خود را مدیریت و یا... کنید، در حال حاضر روبیکاپای تنها فریمورک پرسرعت و قوی و آسان در سطح اینترنت است، همینطور اینکه از روبیکاپای نیز به خاطر مستندات روان و آسانش هم یاد میشود.

## ویژگی های روبیکاپی

- سرعت بسیار بالا
- کتابخانه کاملاً آسان و ایسینک
- مستندات بسیار عالی
- آماده بودن و نصب سریع
- استفاده در تمام دستگاه ها

نحوه نصب کتابخانه:

برای نصب دستور

- `pip install rubpy`

را در ترمینال خود وارد کنید البته دقت کنید که قبل از آن پایتون نسخه ۳.۸ به بالاتر روی دستگاه شما نصب باشد.

نویسنده(ها): شایان حیدری

## فصل اول

در فصل اول قرار است که با انواع چت ها آشنا شویم،  
در واقع فصل اول برای آشنا شدن با نوع چت ها، شناسه  
چت ها و... میباشد.

اگر ما نحوه تشخیص نوع چت را یاد نگیریم ممکن است  
با مشکلات بسیاری رو به رو بشویم، پس این فصل را با  
دقت و کامل خوانده و تمرین کنید و سپس بعد از آن  
فصل های جدیدتر از آن را مطالعه کنید.



در هر پیامرسان، هر چت شناسه مخصوص خود را دارد و این یعنی اینکه اگر شناسه یک چت مشابه یک چت دیگر باشد، پیامرسان با مشکل رو به رو میشود اما این موضوع بسیار مهم است و در این درس ما با انواع شناسه های چت ها آشنا میشویم.

در پیامرسان روبیکا تقریبا ما ۵ نوع شناسه داریم که اعم از شناسه های پیوی ها، گروه ها، کانال ها، ربات ها و... هستند؛ در واقع شناسه ها مانند نام رفتار میکنند.

**نکته: شناسه های هر چت حاوی ۳۲ حرف و عدد متغیر هستند.**

شناسه ها در روبیکا

object\_guid

یا

guid

نام دارند که معنی گوئید -راهنما- است و معنی آبجکت  
گوئید نیز -راهنمای شی- است.

از شناسه ها برای ارسال پیام، ویرایش پیام، خروج از  
گروه یا کانال و ... استفاده میشود.

حداکثر ما انسان ها با دوستان و آشنایان و ... در پیوی  
با هم دیگر نیز صحبت میکنیم، هر پیوی شناسه  
منحصر به فرد خود را دارد و در پیامرسان روبیکا شناسه  
های پیوی ها با

u

شروع میشوند، برای مثال:

u0Di iJm0d0de32ff7ea8d1j82a1837e

این نیز یک شناسه چت برای پیوی در پیامرسان روبیکا  
میباشد.

درست از نام این نوع چت معلوم است که به درد چه کارهایی میخورد، مثلاً صحبت گروهی با دوستان در گروه یکی از پر استفاده ترین کارها میباشد.

یی

شناسه های گروه ها با

g

شروع میشوند، برای مثال:

g0Di iJm0d0de32ff7ea8d1j82a1837e

این یک شناسه گروه میباشد.

کانال ها هم شناسه مخصوص خود را دارند، شاید برای مدیریت کانال با حساب کاربری خود، به شناسه کانال نیاز داشته باشیم و البته شناسه کانال با

C

شروع میشود، برای مثال:

c0DiiJm0d0de32ff7ea8d1j82a1837e

این یک شناسه کانال است.

زمانی که ما در پیامرسان روبیکا برای بار دوم یا چندمین بار در حساب کاربری خود وارد میشویم (لاگین میکنیم) از سرویس اعلانات برای ما پیامی ارسال میشود که به ما اطلاع میدهد برای مثال با دستگاه لینوکسی خود وارد حساب کاربری خود شده ایم. سرویس های اعلانات هم شناسه های منحصر به فرد خود را دارند، برای مثال:

شناسه های سرویس های اعلانات با

S

شروع میشوند.

در پیامرسان روبیکا، ربات هایی مانند ربات پشتیبانی  
روبیکا وجود دارند که پیام های ربات ها از دکمه های  
شیشه ای و... تکمیل شده اند.

شناسه های ربات ها با

S

شروع میشوند.

## فصل دوم

ما در فصل دوم قرار است که با نوع(تایپ) های فایل های ارسالی و دریافتی و پردازش متن های ارسالی و دریافتی آشنا شویم مثل پیام هایی که حاوی متادیتا(جیسان های تنظیم متن های بولد و ایتالیک و...) و یا ...



در وب سوکت کتابخانه روبیکا پای ما قادر هستیم که با ایجاد متغیر ها، انواع داده ها و اطلاعات از جمله متن ها، نوع فایل ها و ... را مشخص کنیم.

برای مثال ما اینجا یک کد دریافت پیام از وب سوکت را تعریف میکنیم:

```
from rubpy import Client

app = Client('my_auth')

@app.newUpdatesHandler
async def my_bot(message):
    text = await message.text()
    print(text)
```

در واقع ما به این صورت با استفاده از وب سوکت پیام هایی که دریافت میشوند را گرفته و از آنها متن ها را استخراج میکنیم!

## فصل دوم بخش اول آشنایی با انواع فایل ها و پردازش آسان متن ها

حالا اینجا ما دیتای جیسان دریافتی از وب سوکت را  
پرینت(چاپ) میکنیم:

```
from rubpy import Client

app = Client('my-auth')

@app.newUpdatesHandler
async def my_bot(message):
    print(await message.show())
```

اگر شناسه حساب کاربری خود را به درستی وارد کرده  
باشیم و پیام جدیدی در حساب خود دریافت کنیم برای  
ما بدین صورت:

```
{'message_id': '333325396513245', 'action': 'New', 'message': {'message_id':  
'333325396513245', 'text': '.', 'time': '1668685102', 'is_edited': False,  
'type': 'Text', 'author_type': 'User', 'author_object_guid':  
'u0DiiJm0d0de32ff7ea8d13182a1837e'}, 'updated_parameters': [], 'timestamp':  
'1668685102', 'prev_message_id': '332982243962245', 'object_guid':  
'u0DiiJm0d0de32ff7ea8d13182a1837e', 'type': 'User', 'state': '1668685042'}
```

دریافت میشود و پرینت میشود، در واقع این دیتای  
مورد نیاز ما و اصلی برای استخراج نوع فایل یا استخراج  
متن و ... میباشد!

در بخش قبلی ما نحوه دریافت پیام ها را یاد گرفتیم و اگر نوعی فایل را دریافت کنیم کلید فایل اینلاین به دیتای جیسان ما اضافه میشود!

```
{'message_id': '333348405355245', 'action': 'New', 'message':
{'message_id': '333348405355245', 'time': '1668686425', 'is_edited':
False, 'sticker': {'emoji_character': '😍', 'w_h_ratio': '1.0',
'sticker_id': '5e1da5c9196212d79486cb44', 'file': {'file_id':
'499872475', 'mime': 'png', 'dc_id': '32', 'access_hash_rec':
'827565584630061998226959902175', 'file_name': 'sticker.png'},
'sticker_set_id': '5e1da57872ab7b37fbb6da71'}, 'type': 'Sticker',
'author_type': 'User', 'author_object_guid':
'u0DiiJm0d0de32ff7ea8d13182a1837e'}, 'updated_parameters': [],
'timestamp': '1668686425', 'prev_message_id': '333325396513245',
'object_guid': 'u0DiiJm0d0de32ff7ea8d13182a1837e', 'type': 'User',
'state': '1668686365'}
```

انواع فایل مانند گیف، عکس، ویدیو و ... را میتوانیم دریافت و استفاده کنیم!

در فصل سوم قرار است که ما با انواع رویداد ها آشنا شویم و جدا از آن درک کنیم که وب سوکت چیست و چگونه میتوان در فریمورک روبیکا پای از وب سوکت بهره برد، البته این فصل یکی از مهم ترین فصل هایی است که باید بیاموزید؛ چرا که اگر مفاهیم وب سوکت و رویداد ها را درک نکنید در آینده کار با کتابخانه برای شما مشکل ساز میشود.

حتما در گروه ها با پیغام "کاربر به گروه پیوست" یا "کاربر از گروه حذف شد" و ... مواجه شده اید؛ این پیغام در واقع یک رویداد یا ایونت است که در مواقع خاص مانند حذف کاربر به وجود می آید.

رویداد یا ایونت ها را در گروه ها دریافت میکنیم، اما در کتابخانه چگونه یک رویداد را تشخیص دهیم و کنترل کنیم؟ با ما همراه باشید تا چگونگی کنترل ایونت ها را از سه بخش یاد بگیریم:

رویداد ها را میتوان از سه بخش دریافت کرد که بخش اول کنترلر وب سوکت یا همان هندلر وب سوکت است که با این تیکه کد میتوانیم آپدیت های جدید ایونت ها را تماشا کنیم:

```
from rubpy import Client

app = Client('AUTH')

@app.newUpdatesHandler
async def my_bot(message):
    print(await message.show())
```

مثلا اگر یک شخص از گروه خارج شود جیسان زیر را به ما نشان میدهد:

```
{ 'message_id': '348861656782263', 'action': 'New', 'message':  
, 'message_id': '348861656782263', 'text': 'یک عضو گروه را ترک کرد',  
'time': '1670152084', 'is_edited': False, 'type': 'Event',  
'event_data': { 'type': 'LeaveGroup', 'performer_object': { 'type':  
'User', 'object_guid': 'u0Dqrce096d45d9aa824a3ec3b052159' } } },  
'updated_parameters': [], 'timestamp': '1670152084',  
'prev_message_id': '348861656670263', 'object_guid':  
'g0BaQJ803b7033905aeef49e0ef3e157', 'type': 'Group', 'state':  
'1670152024' }
```

روش های دیگری در متد های دیگر هم برای دریافت رویداد ها است که آنها را در کانال تلگرامی نیز تعریف کرده ایم.

کانال تلگرام ما:

[@rubika\\_library](https://t.me/rubika_library)



وب سوکت چیست؟ اکثر افراد ارسال پیام با واتساپ یا تلگرام را تجربه کرده اند و البته شاید در ذهن آنها یا شما این سوال پیش آمده است که چرا بدون در نظر گرفتن سرعت اینترنت، سرعت ارسال پیام ها در این پیامرسان ها زیاد است؟ این سوال سوال خوبی است! در واقع اتصال وب سوکت باعث افزایش سرعت بین کلاینت و سرور برای انتقال داده ها و اطلاعات میشود، حالا اینکه واقعا وب سوکت چیست را در ادامه توضیح خواهیم داد:

تا قبل از اینکه وب سوکت به وجود بیاید، تمامی ارتباطات میان کاربران و سرور تنها بر عهده اچ تی تی پی بود، اما زمانی که وب سوکت پا پیش گذاشت دنیای پیامرسان ها و شبکه های اجتماعی نیز تغییر یافت، در وب سوکت داده های داینامیک شده به راحتی میتوانند بین اتصالات وب سوکتی که پر سرعت هستند جریان پیدا کنند؛ وب سوکت بر

خلاف روش هایی که از قبل آن وجود داشت، پروتکلی بر مبنای تی سی پی است که کاملاً متفاوت با اچ تی تی پی است.

در وب سوکت اتصال به صورت دو طرفه کامل برقرار میشود و این اتصال دائمی است و حالا ما برای دریافت سریع پیام های جدید از روبیکا از وب سوکت در کتابخانه استفاده میکنیم.

در فصل های قبل و بخش های قبل مثال های زیادی از وب سوکت زدیم که این کمک میکند شما این قسمت را بهتر متوجه شوید، خوشبختانه کتابخانه روبیکا پای اتصال بین وب سوکت روبیکا و کاربر را بسیار ساده کرده است و فقط نیاز است با چند خط کد، کلاینت خود را برای دریافت پیام های جدید آماده کنید؛ مثال:

```
from rubpy import Client

app = Client('AUTH')

@app.newUpdatesHandler
async def my_bot(message):
    print(await message.show())
```

با این تیکه کد ما میتوانیم جدیدترین پیام ها را در پیامرسان دریافت کنیم و پردازش کنیم.

در کلاسی که از آرگومان هندلر (کنترلر آپدیت ها) دریافت میکنیم متدهای کاربردی وجود دارد که ما آنها را برای شما در ادامه تعریف کرده ایم:

پاسخ به پیام:

```
await message.reply('text')
```

آیا پیام از گروه ارسال شده است:

```
await message.of_group()
```

آیا پیام از پیوی ارسال شده است:

```
await message.of_user()
```

دریافت آیدی پیام ها:

```
await message.id()
```

دریافت متن پیام دریافتی:

```
await message.text()
```

دریافت نوع پیام دریافتی:

```
await message.type()
```

آیا پیام جدید، پیام متنی است:

```
await message.is_text()
```

نوع ارسال کننده پیام:

```
await message.author_type()
```

دریافت گویند مکانی که پیام از آنجا دریافت شده:

```
await message.chat_id()
```

آیا پیام فوروارد شده است:

```
await message.is_forward()
```

آیا متن پیام ارسالی حاوی تبلیغات است:

```
await message.hasAds()
```

## فصل چهارم

در فصل چهارم قرار است با یک سری ابزار کاربری آشنا شویم که برای ساده تر کردن کارهای ما نیز استفاده میشوند؛ یک کلاس در کتابخانه روبیکا پای وجود دارد که اینگونه وارد میشود:

```
from rubpy import Utils
```

در این کلاس چند متد کاربردی وجود دارد که در این فصل کار با آنها را یاد خواهیم گرفت.

ممکن است برای دریافت ساینز عکس یا دریافت فرمت فایل از لینک دانلودی به مشکل بر بخوریم اما ما باز هم کار را برای شما ساده کرده ایم.

مواردی متد وجود دارد که برای رفع نیاز شما و کتابخانه آماده کرده ایم، برخی از این متد ها برای دریافت اطلاعات عکس ها و ویدیو ها و... است و برخی دیگر برای تجزیه و تحلیل متن ها و متد های دیگر...

البته ما به متد خاصی نیاز نداریم، متد های کاربردی در بخش دوم فصل چهارم نیز تعریف شده اند.

یک ابزار بسیار کاربردی وجود دارد، این متد میتواند فرمت فایل را از لینک دانلودی برای شما نیز خارج کند و خروجی را برای شما برگرداند اما چطور این متد را استفاده کنیم؟

تیکه کد زیر این کار را انجام میدهد:

```
from rubpy import Utils

utils = Utils()

mime = await
utils.getMimeFromURL('http://example.com/file.txt'
)

print(mime)
```

دقت کنید که

```
await
utils.getMimeFromURL('http://example.com/file.txt'
)
```

باید داخل یک فانکشن ایسینک استفاده شود!



فصل پنجم مهم ترین فصل است زیرا در این فصل مباحث پیشرفته کتابخانه نیز آموزش داده میشوند و با متدهایی همچون ارسال پیام، دریافت پیام ها، ویرایش پیام ها و... آشنا خواهیم شد و به خوبی یاد خواهیم گرفت که چگونه از متدهای کتابخانه استفاده کنیم و یا حتی فایلی ارسال کنیم و موارد دیگر اما مطلبی که اینجا تاکید میشود این است که حتما آموزشات را با دقت بخوانید و تمرین کنید که به خوبی یاد بگیرید.

به فانکشن هایی که در یک کلاس هستند، متد میگویند و معنی متد روش است. ما برای ارسال پیام، به متد ارسال پیام نیاز داریم.

اما چطور متد ها را فراخوانی کنیم؟

طبیعتا ابتدا باید هندلر وب سوکت خود را بسازیم و سپس با متد ها کار کنیم، پس با ما همراه باشید:

```
from rubpy import Client

app = Client('AUTH')

@app.newUpdatesHandler
async def my_bot(message):
    print(await message.show())
```

با تیکه کد بالا پیام های جدید را دریافت میکنیم، البته فراموش نکنید که در خط سوم به کلمه آث، شناسه حساب کاربری خود را قرار دهید.

با تیکه کد پایین به پیام هایی که محتوای متنی آنها "تست" باشد پاسخ میدهیم، البته اگر پیام ها از پیوی دریافت شده باشند:

```
from rubpy import Client

app = Client('AUTH')

@app.newUpdatesHandler
async def my_bot(message):
    if await message.of_user():
        if await message.is_text():
            text = await message.text()
            if text == 'تست':
                await message.reply('بله')
```

در واقع فانکشن هایی که ما از ارگومان میسیج دریافت میکنیم، به آنها متد گفته میشود!

در این بخش ما با متدهایی آشنا میشویم که از همه بیشتر برای ما کاربرد دارند و تعداد دفعات زیادی از آنها استفاده میکنیم.

مثالهای زیرین را با دقت تماشا کنید چون اینها مثالهایی از کار کردن با متدها و... است.

ارسال پیام متنی:

```
from rubpy import Client

app = Client('auth')

async def main():
    await app.sendText('chat_id', 'text',
                       'reply_to_message_id')

app.run(main)
```

در آرگومان اول گویند و دوم متن را میدهند، میسج آیدی مهم نیست.

حذف پیام ها:

```
await app.deleteMessages('chat_id',  
['message_id'])
```

به دو آرگومان نیاز است، گویند و میسج ایدی که در کد بالا مشخص است و البته یک آرگومان دیر هم برای تنظیم اینکه فقط برای من یا همه پاک شود دارد!

دریافت اطلاعات پیام ها با آیدی:

```
await app.getMessagesByID('chat_id',  
['message_id'])
```

دقت کنید که باید آیدی پیام ها را داخل لیست به متد بدهید

دریافت پیام های داخل یک چت:

```
await app.getMessagesInterval('chat_id',  
['middle_message_id'])
```

دریافت اطلاعات کاربر:

```
await app.getUserInfo('user_guid')
```

دریافت اطلاعات با آیدی:

```
await app.getObjectByUsername('username')
```

دریافت چت ها:

```
await app.getChats()
```

دریافت آپدیت های چت ها:

```
await app.getChatsUpdates()
```

این متد معمولاً برای دریافت جدیدترین آپدیت ها و پیام های جدید استفاده میشود

حذف یک کاربر از گروه:

```
await app.banGroupMember('group_guid',  
'member_guid')
```

حذف یک کاربر از لیست سیاه گروه:

```
await app.banGroupMember('group_guid',  
'member_guid', action='Unset')
```

دریافت همه ادمین های گروه:

```
await app.getGroupAdminMembers('group_guid')
```

دریافت چت آیدی ادمین های گروه:

```
await app.getGroupAdminMembers(  
    'group_guid',  
    get_admin_guids=True  
)
```

دریافت اطلاعات گروه:

```
await app.getGroupInfo('group_guid')
```

متد افزودن عضو به گروه:

```
await app.addGroupMembers('group_guid',  
['member_guid'])
```

با استفاده از این متد میتوانیم با گویند یک کاربر، آن را در گروه عضو کنیم!

متد تنظیم دسترسی های کاربران عادی در گروه:

```
await app.setGroupDefaultAccess('group_guid',  
['access'])
```

میتوانید تنظیم کنید که کاربران دسترسی ارسال پیام در گروه را داشته باشند یا خیر

دریافت لیست ممبر های گروه:

```
await app.getGroupAllMembers('group_guid')
```

با این متد میتوانید لیست عضو های یک گروه را دریافت کنید.



## دریافت لینک گروه:

```
await app.getGroupLink('group_guid')
```

اگر حساب شما در گروه مورد نظر ادمین باشد و دسترسی دریافت لینک گروه را داشته باشد، این متد لینک گروه را برای شما بر میگرداند

## تنظیم لینک جدید برای گروه:

```
await app.setGroupLink('group_guid')
```

اگر شما ادمین گروه مورد نظر لینک را عوض میکند و لینک جدید را به شما بر میگرداند

## دریافت لیست سیاه گروه:

```
await app.getBannedGroupMembers('group_guid')
```

تنظیم تایمر برای گروه:

```
await app.setGroupTimer('group_guid', 'time')
```

با استفاده از این متد میتوان تایمر برای گروه تنظیم کرد

افزودن ادمین جدید به گروه:

```
await app.setGroupAdmin('group_guid',  
'member_guid', ['access'])
```

حذف ادمین:

بزودی...

عضویت در گروه:

```
await app.joinGroup('link')
```

دریافت اطلاعات گروه قبل از عضویت:

```
await app.groupPreviewByJoinLink('link')
```

خروج از گروه:

```
await app.leaveGroup('group_guid')
```

دریافت لیست منشن گروه:

```
await app.getGroupMentionList('group_guid')
```

این متد برای دریافت لیست کاربران و برای منشن کردن آنها استفاده میشود

دریافت تعداد کاربران آنلاین در گروه:

```
await app.getGroupOnlineCount('group_guid')
```

افزودن عضو به کانال:

```
await app.addChannelMembers('channel_guid',  
['member_guid'])
```

دریافت لیست ممبر های کانال:

```
await app.getChannelAllMembers('channel_guid')
```

با این متد میتوانید لیست عضو های یک کانال را دریافت کنید.

دریافت اطلاعات کانال:

```
await app.getChannelInfo('channel_guid')
```

دریافت لینک کانال:

```
await app.getChannelLink('channel_guid')
```

اگر حساب شما در کانال مورد نظر ادمین باشد و دسترسی دریافت لینک کانال را داشته باشد، این متد لینک کانال را برای شما بر میگرداند

تنظیم لینک جدید برای کانال:

```
await app.setChannelLink('channel_guid')
```

اگر شما ادمین کانال مورد نظر باشید لینک را عوض میکند و لینک جدید را به شما بر میگرداند البته دقت کنید که لینک را یا تبدیل به خصوصی میکند یا لینک خصوصی جدید میسازد

دریافت لیست سیاه کانال:

بزودی...

عضویت در کانال با لینک خصوصی:

```
await app.joinChannelByLink('channel_link')
```

افزودن ادمین جدید به کانال:  
بزودی...

حذف ادمین:  
بزودی...

عضویت در کانال با گویند:

```
await app.joinChannelAction('channel_guid')
```

دریافت اطلاعات کانال قبل از عضویت (کانال های خصوصی):

```
await app.channelPreviewByJoinLink('link')
```

خروج از کانال:

```
await app.joinChannelAction('channel_guid', 'Leave')
```

اضافه کردن مخاطب جدید:

```
await app.addAddressBook('989...',  
'first_name', 'last_name: default None')
```

دریافت مخاطبین:

```
await app.getContacts()
```

تغییر رمز تایید دو مرحله ای:

```
await app.changePassword('new_hint',  
'new_password', 'old_password')
```

در آرگومان اول باید راهنمای تایید دو مرحله ای را وارد کنید و در آرگومان دوم باید پسورد جدید را وارد کنید و در آرگومان آخر پسورد قبلی

تغییر شماره حساب کاربری:

```
await app.requestChangePhoneNumber('phone')
```

خاموش کردن تایید دو مرحله ای:

```
await app.turnOffTwoStep('pass')
```

دریافت نشست ها:

```
await app.getMySessions()
```



جستجو در پیامرسان:

```
await app.searchGlobalObjects('text')
```

دریافت اطلاعات لینک ها:

```
await app.getLinkFromAppUrl('app_url')
```

این متد برخی مواقع کاربرد دارد، مثلاً برای دریافت اطلاعات یک پست کانال عمومی یا پست رویینو کاربرد دارد

آپلود آوتار (عکس پروفایل):

```
await app.uploadAvatar('object_guid', 'image-path')
```

در آرگومان دوم باید مسیر فایل عکس خود را قرار دهید

جهت آموزش و مستندات بیشتر به کانال تلگرام ما  
مراجعه کنید:

[t.me/rubika library](https://t.me/rubika_library)

## آخرین تغییرات و آموزشات

...