aiookru

Release 0.1.1.post1

May 08, 2020

Contents:

1 Usage **3**
- 1.1 Client application . . . 3
- 1.2 Server application . . . 3

2 Installation **5**

3 Supported Python Versions **7**
- 3.1 Getting Started . . . 7
- 3.2 Authorization . . . 8
- 3.3 Session . . . 9
- 3.4 REST API . . . 13

4 Indices and tables **21**

aiookru is a python ok.ru API wrapper. The main features are:

- authorization (Authorization Code, Implicit Flow, Password Grant, Refresh Token)
- REST API methods

CHAPTER 1

Usage

To use ok.ru API you need a registered app and ok.ru account. For more details, see aiookru Documentation.

1.1 Client application

Use `ClientSession` when REST API is needed in:

- client component of the client-server application
- standalone mobile/desktop application

i.e. when you embed your app's info (application key) in publicly available code.

```
from aiookru import ClientSession, API

session = ClientSession(app_id, app_key, access_token, session_secret_key)
api = API(session)

events = await api.events.get()
friends = await api.friends.get()
```

Pass `session_secret_key` and `access_token` that were received after authorization. For more details, see authorization instruction.

1.2 Server application

Use `ServerSession` when REST API is needed in:

- server component of the client-server application
- requests from your servers

```
from aiookru import ServerSession, API

session = ServerSession(app_id, app_key, app_secret_key, access_token)
api = API(session)

events = await api.events.get()
friends = await api.friends.get()
```

Pass `app_secret_key` and `access_token` that was received after authorization. For more details, see authorization instruction.

CHAPTER 2

Installation

```
pip install aiookru
```

or

CHAPTER 3

Supported Python Versions

Python 3.5, 3.6, 3.7 and 3.8 are supported.

3.1 Getting Started

3.1.1 Installation

If you use pip, just type

```
$ pip install aiookru
```

You can install from the source code like

```
$ git clone https://github.com/KonstantinTogoi/aiookru.git
$ cd aiookru
$ python setup.py install
```

3.1.2 Account

To create an app you need to:

1. Sign up in ok.Ru and link an email to your account to receive emails with app data.
2. Obtain developer rights at https://ok.ru/devaccess.

After obtaining developer rights you will get a link to add apps or external sites. Open Games and select "My downloads" on top.

3.1.3 Application

After signing up visit the OK.Ru REST API documentation page and create a new application: https://apiok.ru/en/dev/app/create.

To use client OAuth authentication it must be enabled in the app settings.

Save **app_id** (aka **application_id**), **app_key** (aka **application_key**) and **app_secret_key** (aka **application_secret_key**) for user authorization and executing API requests.

```
app_id = 'your_client_id'
app_key = 'your_private_key'
app_secret_key = 'your_secret_key'
```

3.2 Authorization

The preferred way to authorize is an `async with` statement. After authorization the session will have the following attributes:

- `access_token` aka `session_key`, always
- `session_secret_key` if Implicit Grant / Password Grant used
- `permission_granted` if Implicit Grant used
- `state` if Implicit Grant used
- `token_type` if Code Grant / Refresh Token Grant used
- `refresh_token` if Code Grant used
- `expires_in` if Code Grant / Implicit Grant / Refresh Token Grant used

3.2.1 Authorization Code Grant

```
from aiookru import CodeSession, API

app_id = 123456
app_key = 'abcde'
app_secret_key = 'xyz'

async with CodeSession(app_id, app_key, app_secret_key, code, redirect_uri) as␣
↪session:
    api = API(session)
    ...
```

About OAuth 2.0 Authorization Code Grant: https://oauth.net/2/grant-types/authorization-code/

For more details, see https://apiok.ru/ext/oauth/server

3.2.2 Implicit Grant

```
from aiookru import ImplicitSession, API

app_id = 123456
```

(continues on next page)

(continued from previous page)

```
app_key = 'abcde'
app_secret_key = ''

async with ImplicitSession(app_id, app_key, app_secret_key, login, passwd, scope) as␣
↪session:
    api = API(session)
    ...
```

About OAuth 2.0 Implicit Grant: https://oauth.net/2/grant-types/implicit/

For more details, see https://apiok.ru/ext/oauth/client

3.2.3 Password Grant

```
from aiookru import PasswordSession, API

app_id = 123456
app_key = 'abcde'
app_secret_key = ''

async with PasswordSession(app_id, app_key, app_secret_key, login, passwd) as session:
    api = API(session)
    ...
```

About OAuth 2.0 Password Grant: https://oauth.net/2/grant-types/password/

3.2.4 Refresh Token

```
from aiookru import RefreshSession, API

app_id = 123456
app_key = 'abcde'
app_secret_key = 'xyz'

async with RefreshSession(app_id, app_key, app_secret_key, refresh_token) as session:
    api = API(session)
    ...
```

About OAuth 2.0 Refresh Token: https://oauth.net/2/grant-types/refresh-token/

For more details, see https://apiok.ru/ext/oauth/server

3.3 Session

The session makes **GET** requests when you call instances of `APIMethod` class that are returned as attributes of an `API` instance.

3.3.1 Request

By default, the session (`CodeSession`, `ImplicitSession`, `PasswordSession`, `RefreshSession`) tries to infer which signature generation circuit to use:

- if `app_secret_key` is not empty string - server-server signature generation circuit is used
- else if `session_secret_key` is not empty string - client-server signature generation circuit is used
- else exception is raised

You can explicitly set a signature generation circuit for signing requests by passing to `API` one of the sessions below.

Client-Server signature generation circuit

Let's consider the following example of API request with client-server signature:

```
from aiookru import TokenSession, API

session = TokenSession(
    app_id=123456,
    app_key='ABCDEFGHIGKLMNOPK',
    app_secret_key='',
    access_token='-s-2GUXOAvQYI7-RfxsZtV1wezsdtVPv92xfuaSQ8.SAIV1O2ywYra2-
↪3ywes5St2yvcuZSr9UUWN2TtbWtWKVTuAy8',
    session_secret_key='ae5362b5b588cc7294c2414d71b74d5d',
)
api = API(session)

events = await api.events.get()
```

It is equivalent to **GET** request:

```
https://api.ok.ru/fb.do
    ?application_key=ABCDEFGHIGKLMNOPK
    &format=json
    &method=events.get
    &sig=03a41413523ea8092507949d6e711963
    &access_token=-s-2GUXOAvQYI7-RfxsZtV1wezsdtVPv92xfuaSQ8.SAIV1O2ywYra2-
↪3ywes5St2yvcuZSr9UUWN2TtbWtWKVTuAy8
```

The following steps were taken:

1. `session_secret_key` used as secret key
2. sorted request parameters and secret key were concatenated: `application_key=ABCDEFGHIGKLMNOPKformat=jsonmethod=events.getae5362b5b588cc7294c2414d71b74d5d`
3. signature `03a41413523ea8092507949d6e711963` calculated as MD5 of the previous string
4. signature appended to **GET** request parameters
5. `access_token` appended to **GET** request parameters

ClientSession

`ClientSession` is a subclass of `TokenSession`. Use it as a client session without authorization. Use `session_secret_key` and `access_token` that were already received.

```
from aiookru import ClientSession, API

session = ClientSession(app_id, app_key, access_token, session_secret_key)
api = API(session)
...
```

ImplicitClientSession

`ImplicitClientSession` is a subclass of `ImplicitSession`. Use it as a client session with authorization (Implicit Flow).

```
from aiookru import ImplicitClientSession, API

async with ImplicitClientSession(app_id, app_key, login, passwd, scope) as session:
    api = API(session)
    ...
```

PasswordClientSession

`PasswordClientSession` is a subclass of `PasswordSession`. Use it as a client session with authorization (Password Grant).

```
from aiookru import PasswordClientSession, API

async with PasswordClientSession(app_id, app_key, login, passwd) as session:
    api = API(session)
    ...
```

Server-Server signature generation circuit

Let's consider the following example of API request with server-server signature:

```
from aiookru import TokenSession, API

session = TokenSession(
    app_id=123456,
    app_key='ABCDEFGHIGKLMNOPK',
    app_secret_key='ABC123DEF456GHI789JKL123',
    access_token='-s-84W-s3egarWUsbkq-IWTucuedzTKT8VUXIA.s4Xx8IW7',
    session_secret_key='',
)
api = API(session)

events = await api.events.get()
```

It is equivalent to **GET** request:

```
https://api.ok.ru/fb.do
    ?application_key=ABCDEFGHIGKLMNOPK
    &format=json
    &method=events.get
    &sig=232c8eb921951c4dba9b72606f9ddb4c
    &access_token=-s-84W-s3egarWUsbkq-IWTucuedzTKT8VUXIA.s4Xx8IW7
```

The following steps were taken:

1. `b1a2b89707a94624c43afae67d59274c` used as secret key, it was calculated as MD5(`access_token` + `app_secret_key`)
2. sorted request parameters and secret key were concatenated: `application_key=ABCDEFGHIGKLMNOPKformat=jsonmethod=events.getb1a2b89707a94624c43afae67d59274c`

3. signature `232c8eb921951c4dba9b72606f9ddb4c` calculated as MD5 of the previous string
4. signature appended to **GET** request parameters
5. `access_token` appended to **GET** request parameters

ServerSession

`ServerSession` is a subclass of `TokenSession`. Use it as a server session without authorization. Use your `app_secret_key` and `access_token` that was already received.

```
from aiookru import ServerSession, API

session = ServerSession(app_id, app_key, app_secret_key, access_token)
api = API(session)
...
```

CodeServerSession

`CodeServerSession` is a subclass of `CodeSession`. Use it as a server session with authorization (Authorization Code).

```
from aiookru import CodeServerSession, API

async with CodeServerSession(app_id, app_key, app_secret_key, code, redirect_uri) as␣
↪session:
    api = API(session)
    ...
```

RefreshServerSession

`RefreshServerSession` is a subclass of `RefreshSession`. Use it as a server session with authorization (Refresh Token).

```
from aiookru import RefreshServerSession, API

async with RefreshServerSession(app_id, app_key, app_secret_key, refresh_token) as␣
↪session:
    api = API(session)
    ...
```

3.3.2 Response

By default, a session after executing request returns response's body as `dict` if executing was successful, otherwise it raises exception.

You can pass `pass_error` parameter to `TokenSession` for returning original response (including errors).

3.3.3 Error

In case of an error, by default, exception is raised. You can pass `pass_error` parameter to `TokenSession` for returning original error's body as `dict`:

```
{
    "error_code": 100,
    "error_data": 1,
    "error_msg": "PARAM : Either session_key or uid must be specified"
}
```

All error codes are available here: https://apiok.ru/en/dev/errors.

3.4 REST API

List of all methods is available here: https://apiok.ru/en/dev/methods/rest/.

3.4.1 Executing requests

For executing API requests call an instance of `APIMethod` class. You can get it as an attribute of `API` class instance or as an attribute of other `APIMethod` class instance.

```
from aiookru import API

api = API(session)

events = await api.events.get()  # events for current user
friends = await api.friends.get()  # current user's friends
```

Under the hood each API request is enriched with parameters to generate signature:

- `application_key`
- `format`
- `method`

and with the following parameters after generating signature:

- `sig`
- `access_token`

3.4.2 Methods

* - only for Android and iOS applications, for external applications is required.

apps

method	Session	Required permissions
checkVipOfferStatus	OAuth / WEB	VALUABLE_ACCESS
getAppPromoInfo	prohibited	VALUABLE_ACCESS
getPlatformCatalogNodeTop	OAuth / WEB	VALUABLE_ACCESS
getPlatformCatalogNodesTop	OAuth / WEB	VALUABLE_ACCESS
getPlatformNew	OAuth / WEB	VALUABLE_ACCESS
getPlatformTop	OAuth / WEB	VALUABLE_ACCESS
getTop	OAuth / WEB / Without Session *	VALUABLE_ACCESS
removeAppPromoInfo	prohibited	VALUABLE_ACCESS
setAppPromoInfo	prohibited	VALUABLE_ACCESS
setVipOfferStatus	OAuth / WEB	VALUABLE_ACCESS
setVipOffers	OAuth / WEB	VALUABLE_ACCESS

bookmark

method	Session	Required permissions
add	OAuth / WEB	VALUABLE_ACCESS
delete	OAuth / WEB	VALUABLE_ACCESS

callbacks

method	Session	Required permissions
payment	prohibited	VALUABLE_ACCESS

communities

method	Session	Required permissions
getMembers	OAuth / WEB	VALUABLE_ACCESS

discussions

method	Session	Required permissions
get	OAuth / WEB	VALUABLE_ACCESS
getAttachedResources	OAuth / WEB	VALUABLE_ACCESS
getComment	OAuth / WEB	VALUABLE_ACCESS
getCommentLikes	OAuth / WEB	VALUABLE_ACCESS
getComments	OAuth / WEB	VALUABLE_ACCESS
getDiscussionComments	OAuth / WEB	VALUABLE_ACCESS
getDiscussionCommentsCount	OAuth / WEB	VALUABLE_ACCESS
getDiscussionLikes	OAuth / WEB	VALUABLE_ACCESS
getDiscussions	OAuth / WEB	VALUABLE_ACCESS
getDiscussionsNews	OAuth / WEB	VALUABLE_ACCESS
getList	OAuth / WEB	VALUABLE_ACCESS

events

method	Session	Required permissions
get	OAuth / WEB	VALUABLE_ACCESS

friends

method	Session	Required permissions
appInvite	OAuth / WEB	APP_INVITE
get	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getAppUsers	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getAppUsersOnline	OAuth / WEB	VALUABLE_ACCESS
getBirthdays	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getByDevices	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getMutualFriends	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getOnline	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getRelatives	OAuth / WEB	VALUABLE_ACCESS
getRelativesV2	OAuth / WEB	VALUABLE_ACCESS
getSuggestions	OAuth / WEB / Without Session *	VALUABLE_ACCESS

group

method	Session	Required permissions
getCounters	OAuth / WEB / Without Session *	
getInfo	OAuth / WEB / Without Session *	
getMembers	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getStatOverview	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getStatPeople	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getStatTopic	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getStatTopics	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getStatTrends	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getUserGroupsByIds	OAuth / WEB / Without Session *	
getUserGroupsV2	OAuth / WEB / Without Session *	
pinGroupFeed	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
setMainPhoto	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT,PHOTO_C

interests

method	Session	Required permissions
get	OAuth / WEB	VALUABLE_ACCESS

market

method	Session	Required permissions
add	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
addCatalog	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
delete	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
deleteCatalog	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
edit	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
editCatalog	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getByCatalog	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getByIds	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getCatalogsByGroup	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getCatalogsByIds	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
getProducts	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
pin	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
reorder	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
reorderCatalogs	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
setStatus	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT
updateCatalogsList	OAuth / WEB	VALUABLE_ACCESS,GROUP_CONTENT

mediatopic

method	Session	Required permissions
getByIds	OAuth / WEB	VALUABLE_ACCESS
getPollAnswerVoters	OAuth / WEB	VALUABLE_ACCESS
getRepublishedTopic	OAuth / WEB	VALUABLE_ACCESS
post	OAuth / WEB / Without Session *	

notifications

method	Session	Required permissions
sendFavPromo	prohibited	VALUABLE_ACCESS
sendMass	prohibited	VALUABLE_ACCESS
sendSimple	prohibited	VALUABLE_ACCESS
stopFavPromo	prohibited	VALUABLE_ACCESS
stopSendMass	prohibited	VALUABLE_ACCESS
updateFavPromo	prohibited	VALUABLE_ACCESS

payment

method	Session	Required permissions
appCashback	prohibited	VALUABLE_ACCESS
getUserAccountBalance	OAuth / WEB	
getUserAccountBonusBalance	OAuth / WEB	VALUABLE_ACCESS
getVipStatus	OAuth / WEB	VALUABLE_ACCESS

photos

method	Session	Required permissions
addAlbumLike	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT,LIKE
addPhotoLike	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT,LIKE
createAlbum	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
deleteAlbum	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
deletePhoto	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
deleteTags	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
editAlbum	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
editPhoto	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
getAlbumInfo	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
getAlbumLikes	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
getAlbums	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getInfo	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
getPhotoInfo	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
getPhotoLikes	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
getPhotoMarks	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT
getPhotos	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getTags	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
getUserAlbumPhotos	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
getUserPhotos	OAuth / WEB / Without Session *	VALUABLE_ACCESS,PHOTO_CONTENT
setAlbumMainPhoto	OAuth / WEB	VALUABLE_ACCESS,PHOTO_CONTENT

photosV2

method	Session	Required permissions
commit	OAuth / WEB / Without Session *	PHOTO_CONTENT
getUploadUrl	OAuth / WEB / Without Session *	PHOTO_CONTENT

places

method	Session	Required permissions
reverseGeocode	OAuth / WEB	VALUABLE_ACCESS
validate	OAuth / WEB	VALUABLE_ACCESS

sdk

method	Session	Required permissions
getEndpoints	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getInstallSource	prohibited	VALUABLE_ACCESS
getNotes	prohibited	VALUABLE_ACCESS
init	OAuth / WEB	VALUABLE_ACCESS
reportPayment	OAuth / WEB	VALUABLE_ACCESS
reportStats	OAuth / WEB	VALUABLE_ACCESS
resetNotes	OAuth / WEB	VALUABLE_ACCESS
sendNote	OAuth / WEB	VALUABLE_ACCESS

search

method	Session	Required permissions
tagContents	OAuth / WEB	
tagMentions	OAuth / WEB	
tagSearch	OAuth / WEB	

share

method	Session	Required permissions
fetchLink	OAuth / WEB	VALUABLE_ACCESS
fetchLinkV2	OAuth / WEB	VALUABLE_ACCESS

stream

method	Session	Required permissions
delete	OAuth / WEB	VALUABLE_ACCESS
isSubscribed	OAuth / WEB	VALUABLE_ACCESS
markAsSpam	OAuth / WEB	VALUABLE_ACCESS

url

method	Session	Required permissions
getInfo	OAuth / WEB / Without Session *	VALUABLE_ACCESS

users

method	Session	Required permissions
deleteGuests	OAuth / WEB	VALUABLE_ACCESS
getAdditionalInfo	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getCallsLeft	OAuth / WEB / Without Session *	
getCurrentUser	OAuth / WEB	GET_EMAIL
getGames	OAuth / WEB	VALUABLE_ACCESS
getGuests	OAuth / WEB	VALUABLE_ACCESS
getHolidays	OAuth / WEB	VALUABLE_ACCESS
getInfo	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getInfoBy	OAuth / WEB	VALUABLE_ACCESS
getInvitableFriends	OAuth / WEB	VALUABLE_ACCESS
getLoggedInUser	OAuth / WEB	
getMobileOperator	OAuth / WEB / Without Session *	VALUABLE_ACCESS
hasAppPermission	OAuth / WEB / Without Session *	
isAppUser	OAuth / WEB / Without Session *	
removeAppPermissions	OAuth / WEB / Without Session *	
setStatus	OAuth / WEB	SET_STATUS
updateMask	OAuth / WEB / Without Session *	VALUABLE_ACCESS
updateMasks	prohibited	
updateMasksV2	prohibited	

video

method	Session	Required permissions
delete	OAuth / WEB	VALUABLE_ACCESS,VIDEO_CONTENT
getUploadUrl	OAuth / WEB / Without Session *	VALUABLE_ACCESS,VIDEO_CONTENT
subscribe	OAuth / WEB	VALUABLE_ACCESS,VIDEO_CONTENT
update	OAuth / WEB / Without Session *	VALUABLE_ACCESS,VIDEO_CONTENT

widget

method	Session	Required permissions
getWidgetContent	OAuth / WEB / Without Session *	VALUABLE_ACCESS
getWidgets	OAuth / WEB / Without Session *	VALUABLE_ACCESS

CHAPTER 4

Indices and tables

- genindex
- modindex