

---

# **Python Ulord-platform SDK API Documentation**

*Release 0.0.1*

**Ulord-PuJi**

**Jul 31, 2018**



---

## Contents:

---

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Including . . . . .	1
1.2	Featured Projects . . . . .	1
1.3	TODO list . . . . .	1
1.4	Contribute . . . . .	2
<b>2</b>	<b>What'snew</b>	<b>3</b>
<b>3</b>	<b>Install</b>	<b>5</b>
<b>4</b>	<b>Example</b>	<b>7</b>
4.1	Get development permissions . . . . .	7
4.2	Init . . . . .	7
4.3	Config . . . . .	7
4.4	Create your required interface . . . . .	8
<b>5</b>	<b>ulordapi</b>	<b>9</b>
5.1	udfs . . . . .	9
5.2	config . . . . .	11
5.3	daemonCLI . . . . .	11
5.4	errcode . . . . .	11
5.5	manage . . . . .	11
5.6	up . . . . .	14
5.7	user . . . . .	17
5.8	utils . . . . .	21
5.9	version . . . . .	24
5.10	webServer . . . . .	24
<b>6</b>	<b>About py-ulord-api</b>	<b>27</b>
<b>7</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Python Module Index</b>	<b>31</b>



Ulord-platform HTTP Client Library

Before reading the document you should pay attention to the following:

Important: The legacy py-ulord-api package/module will only work for Ulord-platform 0.0.1 and Python 2.7.

Note: This library constantly has to change to stay compatible with the Ulord-platform HTTP API. Currently, this library is tested against Ulord-platform v0.0.1. You may experience compatibility issues when attempting to use it with other versions of Ulord-platform.

## 1.1 Including

This package has three functions,including cli, py-api and web-API.

py-api is a python package.It support most functions.\*\*Suggest you to use this\*\*.

Cli is a simple package for python api.

Web-API is also a simple package for python package.

## 1.2 Featured Projects

waiting...

## 1.3 TODO list

- ☒ add TODO list
- ☐ some docs
- ☐ support python3

- [ ] more friendly expression
- [ ] add multi-threading download
- [ ] add unit test
- [ ] docker environment

## 1.4 Contribute

### 1.4.1 Bug reports

You can submit bug reports using the [GitHub issue tracker](#).

### 1.4.2 Pull requests

Pull requests are welcome. Before submitting a new pull request, please waiting. . .

### 1.4.3 Want to read this repository?

Some places to get you started. (WIP)

Senior programmer Main file: [ulordapi/user.py](#)

Junior Programmer Main file: [ulordapi/user.py](#)

CLI Commands: [ulordapi/daemonCLI.py](#)

Web Server: [ulordapi/webServer.py](#)

## CHAPTER 2

---

What'snew

---

waiting...





## CHAPTER 3

---

### Install

---

> *import*: haven't completed!

Install with pip(not completed!):

```
$ pip install ulordapi
```

Or you can use this repository to set up

```
git clone https://github.com/UlordChain/py-ulord-api.git
cd py-ulord-api
python setup.py install
```



This is a tutorial that uses SDK to develop a blog demo.

### 4.1 Get development permissions

You should register on the ulord-platform. And then you need to create a application. You will get a appkey and a secret. The two things are important. You mustn't lose them.

### 4.2 Init

Create a role according to your ability using your appkey and secret. SDK has two roles. One is junior and the other is senior. Both of them have to init using your appkey and secret.

Senior has base API about UDFS and ulord-platform. UDFS is a distributed storage platform. Using it you can upload your resource and download easily. The API about ulord-platform packages for ulord-platform http interface.

Junior has all API of Senior. And it also has a base database. You can use database whatever you want to use. Default is sqlite. It achieves base on the flask-sqlalchemy. And you can build your own database structure.

```
from ulordapi.user import Junior
junior = Junior(appkey="5d42b27e581c11e88b12f48e3889c8ab", secret=
    "5d42b27f581c11e8bf63f48e3889c8ab")
```

### 4.3 Config

You should make your own config. Reading config module and create your own config.

```
blog_config = {
    'baseconfig':{
        'config_file':'E:\ulord\ulord-blog-demo\config'
    },
    'logconfig':{
        'log_file_path': 'E:\ulord\ulord-blog-demo\blogdemo.log'
    }
}
junior.config_edit(blog_config)
```

## 4.4 Create your required interface

Example regist(This demo is using flask):

```
@app.route('/user/regist',methods=['POST'])
def regist():
    """
    user regist

    :return: user token
    """
    username = request.json.get('username')
    password = request.json.get('password')
    cellphone = request.json.get('cellphone')
    email = request.json.get('email')
    if username is None or password is None:
        # missing arguments
        return jsonify(return_result(60100))
    args = junior.decrypt([username, password, cellphone, email])
    if args:
        result = junior.user_regist(username=args[0],password=args[1],
        ↪cellphone=args[2],email=args[3])
        return jsonify(result)
    else:
        return jsonify(return_result(60100))
```

## 5.1 udfs

### 5.1.1 ulordapi.udfs

#### udfs

```
class ulordapi.udfs.udfs.Udfs
    udfs class,including some operations about daemon program

    get_pid()
        get daemon pid

        Returns daemon pid

    get_udfs()
        get udfs path according to the os

        Returns udfs path

    modify_config()
        modify udfs config

    start (daemon=None)
        start udfs.if daemon is true, the udfs is a daemon,else the udfs is starting the current thread.

        Parameters daemon (bool) – if wait

    start_command (cmd)
        start external command

        Parameters cmd (str/list) – shell command

        Returns popen

    start_init()
        init udfs, sleep 3 microsecond to wait the command finished.
```

**stop()**  
kill the udfs daemon

**class** `ulordapi.udfs.udfs.UdfsHelper` (*host='127.0.0.1', port='5001'*)  
download and upload files from Ulord

**cat** (*udfshash*)  
Retrieves the contents of a file identified by hash. :param udfshash: udfs hash :type udfshash:str :return: str(File contents)

**downloadfile** (*localfile*)  
query the localfile from DB and then download from the udfs

**Parameters** **localfile** (*str*) – a file path

**Returns** True or False

**downloadhash** (*filehash, filepath=None, Debug=False*)  
download file from the UDFS according to the udfs hash

**Parameters**

- **filehash** (*str*) – file udfs hash
- **filepath** (*str*) – the path to save the file
- **Debug** (*bool*) – if Debug print the cost time

**Returns** True or False

**list** (*filehash*)  
list the udfshash chunks

**Parameters** **filehash** (*str*) – a udfs hash

**Returns** a list chunks of the udfs hash

**resumableDownload** (*filehash, filename=None*)  
resumable download

**Parameters**

- **filehash** (*str*) – file UDFS hash
- **filename** (*str*) – file path to save the file

**Returns** True or False

**update** (*host='127.0.0.1', port='5001'*)  
update udfs dameon connection

**Parameters**

- **host** (*str*) – udfs daemon host
- **port** (*str/int*) – udfs daemon port

**upload\_file** (*local\_file*)  
upload the file to the udfs

**Parameters** **local\_file** (*str*) – a local file path

**Returns** Hash or False

**upload\_stream** (*stream*)  
upload the stream to the ulord

**Parameters** **stream** (*unicode*) – stream data

### Returns

## 5.2 config

```
class ulordapi.config.Config(*args, **kwargs)
    Bases: dict

    config class.Based on the dict.Update key-value,including subdict.

    edit (data)

    read (init=True)
        read config from config file and update config.

        Parameters init (bool) – check if the config is first save.Default is True, will save the config
            file.

    save ()
        save config to the config file.Just for the config,not support for it's sub config.
```

## 5.3 daemonCLI

```
class ulordapi.daemonCLI.client
ulordapi.daemonCLI.edit_config(*args, **kwargs)
ulordapi.daemonCLI.formatResult (func)
ulordapi.daemonCLI.main()
ulordapi.daemonCLI.show_config(*args, **kwargs)
```

## 5.4 errcode

```
ulordapi.errcode.return_result (errcode=0, reason=None, result=None)
    update errcode,reason,result and then return

    Parameters

    • errcode (int) – errcode
    • reason (str/dict) – reason, change it to the dict and update original reason
    • result (str/dict) – result, change it to the dict and update original result

    Returns errcode result
```

## 5.5 manage

```
class ulordapi.manage.Ads(**kwargs)
    Bases: ulordapi.manage.Resource

    Advertisement.Viewer will get a few ulord from the author when they view it.

    UPID
```

```
    amount
    body
    claimID
    date
    description
    id
    resource_type
    tags
    title
    userid
    views

class ulordapi.manage.Content(**kwargs)
    Bases: ulordapi.manage.Resource
    Content,viewer need to pay a few ulord to view.
    UPID
    amount
    body
    claimID
    date
    description
    id
    resource
    resource_type
    tags
    title
    userid
    views

class ulordapi.manage.Resource(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Model
    Resource table: create resource table
    UPID
    amount
    body
    claimID
    date
    description
    id
```



```
resource
resource_type
tags
title
userid
views

class ulordapi.manage.Tag(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Model
    tag table
    id
    pre1
    pre2
    resource
    tagname

class ulordapi.manage.User(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Model
    User table: create user database.

    activity
    balance
    boughts
    cellphone
    email
    hash_password(password)
        Hash user's password and save hash result.

        Parameters password(str) – user's password

    id
    password_hash
    pay_password
    timestamp
    token
    username
    verify_password(password)
        verify password

        Parameters password(str) – user's password

        Returns True or False.

    wallet

ulordapi.manage.create()
    Create database.
```

## 5.6 up

**class** ulordapi.up.UlordHelper (*appkey=None, ulord\_secret=None*)

Bases: object

a helper to request the ulord paltform

**calculate\_sign** (*dt=None*)

encrypt the request data according to the secret

**Parameters** *dt* (*dict*) – data need to be encrypted, Default is None

**Returns** Usign

**checkisbought** (*payer, claim\_ids*)

query the personal balance from the ulord platform

**Parameters**

- **payer** (*str*) – payer wallet name
- **claim\_ids** (*list*) – resource claim id

**Returns** errcode.You can query from the errcode dict.

**delete** (*id, password*)

delete resource on the ulord-platform

**Parameters**

- **id** (*int*) – resource ulord-platform DB ID
- **password** (*str*) – user wallet password

**Returns** errcode.You can query from the errcode dict.

**get** (*url*)

get from the ulord-platform

**Parameters** *url* (*str*) – request's url

**Returns** return result.You can query the errcode.

**paytouser** (*username*)

activity send some ulords to the user

**Parameters** *username* (*str*) – user wallet name

**Returns** errcode.You can query from the errcode dict.

**post** (*url, data*)

post to the ulord-platform

**Parameters**

- **url** (*str*) – request's url
- **data** (*dict*) – post data

**Returns** return result.you can query the errcode

**publish** (*data*)

publish data to the ulord platform

**Parameters** *data* (*dict*) – data needed to be published.Key includes au-  
thor,title,tags,udfs\_hash,price,content\_type,despay\_password

**Returns** errcode.You can query from the errcode dict.

**queryBillingDetailByID** (*claim\_id*, *page=1*, *num=10*)  
query all billing details according to the claimID

**Parameters**

- **claim\_id** (*str*) – resource on the ulord-chain ID
- **page** (*int*) – which page of result do you want to view?Default is 1.
- **num** (*int*) – how many pieces of data of result do you want to view?Default is 10.

**Returns** errcode.You can query from the errcode dict.

**queryStatisticsByID** (*claim\_ids*)  
query resource statistics information by ID

**Parameters** **claim\_ids** (*list*) – need to be query ID list.

**Returns** errcode.You can query from the errcode dict.

**query\_resorc\_by\_ID** (*ids*)  
query resource list according to the id list

**Parameters** **ids** (*list*) – need to be query id list

**Returns** errcode.You can query from the errcode dict.

**querybalance** (*payer*, *pay\_password*)  
query the personal balance from the ulord platform

**Parameters**

- **payer** (*str*) – payer wallet name
- **pay\_password** (*str*) – payer wallet password

**Returns** errcode.You can query from the errcode dict.

**querybillings** (*username*, *start*, *end*)  
get billings info

**Parameters**

- **username** (*str*) – current user wallet name
- **start** (*str*) – start time.2018-03-29
- **end** (*str*) – end time.2018-03-29

**Returns** errcode.You can query from the errcode dict.

**querybillingsdetail** (*author*, *page=1*, *num=10*)  
query the billings detail.Union the income and outgo

**Parameters**

- **author** (*str*) – current user wallet name
- **page** (*int*) – which page of result do you want to view?Default is 1.
- **num** (*int*) – how many pieces of data of result do you want to view?Default is 10.

**Returns** errcode.You can query from the errcode dict.

**queryincomebillings** (*author*, *start*, *end*, *page=1*, *num=10*, *category=2*)  
get income billings info

**Parameters**

- **author** (*str*) – current user wallet name
- **start** (*str*) – start time.2018-03-29
- **end** (*str*) – end time.2018-03-29
- **page** (*int*) – which page of result do you want to view?Default is 1.
- **num** (*int*) – how many pieces of data of result do you want to view?Default is 10.
- **category** (*int*) – resource type. 0—common resource,1—ads,2—all

**Returns** errcode.You can query from the errcode dict.

**queryoutgobillings** (*author, start, end, page=1, num=10, category=2*)  
get outgo billings info

**Parameters**

- **author** (*str*) – current user wallet name
- **start** (*str*) – start time.2018-03-29
- **end** (*str*) – end time.2018-03-29
- **page** (*int*) – which page of result do you want to view?Default is 1.
- **num** (*int*) – how many pieces of data of result do you want to view?Default is 10.
- **category** (*int*) – resource type. 0—common resource,1—ads,2—all

**Returns** errcode.You can query from the errcode dict.

**querypublishnum** (*author, start, end*)  
query the number of resourced which has published

**Parameters**

- **author** (*str*) – current user wallet name
- **start** (*str*) – start time.2018-03-29
- **end** (*str*) – end time.2018-03-29

**Returns** errcode.You can query from the errcode dict.

**queryresource** (*page=1, num=10, \*\*kwargs*)  
query the resource list from the ulord platform.method is get

**Parameters**

- **page** (*int*) – which page do you want to view?Default is 1.
- **num** (*int*) – how many pieces of datas in one page?.Default is 10.
- **kwargs** (*key-value*) – key-value query

**Returns** errcode.You can query from the errcode dict.

**queryuserpublished** (*wallet\_username, page=1, num=10*)  
query user published from ulort platform

**Parameters**

- **wallet\_username** (*str*) – auther wallet name
- **page** (*int*) – which page of result do you want to view?Default is 1.

- **num** (*int*) – how many pieces of data of result do you want to view?Default is 10.

**Returns** errcode.You can query from the errcode dict.

**regist** (*username, password*)

regist wallet address from the ulord platform

**Parameters**

- **username** (*str*) – wallet name
- **password** (*str*) – wallet password

**Returns** errcode.You can query from the errcode dict

**transaction** (*payer, claim\_id, pay\_password, isads=False*)

record the transaction to the ulord platform

**Parameters**

- **payer** (*str*) – payer wallet name
- **claim\_id** (*str*) – resource claim id
- **pay\_password** (*str*) – payer wallet password
- **isads** (*bool*) – check the resource if a Ad.

**Returns** errcode.You can query from the errcode dict.

**update** (*data*)

update data from the ulord-platform

**Parameters** **data** (*dict*) – data needed to be updated.Key includes id,pay\_password,title,tags,udfs\_hash,price,content\_type,despay\_password

**Returns** errcode.You can query from the errcode dict.

## 5.7 user

**class** ulordapi.user.Developer (*appkey, secret*)

Bases: *ulordapi.up.UlordHelper*

basic develoer class to execute some functions

**config\_edit** (*args=None*)

config operations

**Parameters** **args** (*list/dict*) – a list or a dict.Update config

**Returns** args dict

**config\_show** (*args=None*)

show config.Default show all.

**Parameters** **args** (*list/dict*) – keys

**Returns** config according to the keys

**decrypt** (*args*)

check if the arg is encrypted.If encrypted return decrypted arg,else return None.

**Parameters** **args** (*str/str list*) – arg need to be checked,

**Returns** decrypted arg or arg(unicode)

**request** (*method, url, data=None*)  
advanced command. request to the ulord-platform

**Parameters**

- **method** (*str*) – request method, like post, get and so on.
- **url** (*str*) – request url
- **data** (*dict*) – request data

**Returns** return result. you can query the errcode

**udfs\_cat** (*udfshashes*)  
view the udfs hash comment

**Parameters** **udfshashes** (*list*) – udfs hashes

**Returns** dict udfshash - True/False

**udfs\_download** (*udfshashes*)  
download file from ulord. udfs is a udfs list

**Parameters** **udfshashes** (*list*) – udfs hashes

**Returns** udfshash - True/False

**udfs\_upload** (*fileinfos*)  
upload file into ulord. fileinfo is a file list

**Parameters** **fileinfos** (*list/str*) – file need to upload to the ulord-platform

**Returns** dict fileinfo - True/False

**class** `ulordapi.user.Junior` (*appkey, secret*)  
Bases: `ulordapi.user.Developer`

Junior programmer to develop his application. Using default database.

**create\_database** (*path=None*)  
create database

**Parameters** **path** (*str*) – database path. Warning: It's a dir

**Returns** True/False

**pay\_ads** (*wallet, claim\_id, authname*)  
user view ads

**Parameters**

- **wallet** (*str*) – user wallet name
- **claim\_id** (*str*) – resource claim id
- **authname** (*str*) – resource auth name

**Returns** errcode. You can query from the errcode.

**pay\_resources** (*payer, claim\_id, password, encrypted=False*)  
user pay resource

**Parameters**

- **payer** (`User`) – payer
- **claim\_id** (*str*) – resource claim id
- **password** (*str*) – payer password

**Returns** errcode.You can query from the errcode.

**query** (*sql*)

advanced command self query

**Parameters** **sql** (*str*) – sql sentence

**Returns** query result

**resource\_views** (*claim\_id*)

add resource view

**Parameters** **claim\_id** (*str*) – resource claim\_id

**Returns** resource current view

**resource\_publish** (*title, udfshash, amount, tags, des, usercondition*)

user publish resource

**Parameters**

- **title** (*str*) – resource title
- **udfshash** (*str*) – resource uplorded to the UDFS hash
- **amount** (*float*) – resource price
- **tags** (*list*) – resource tag
- **des** (*str*) – resource description
- **usercondition** (*dict*) – a condition which need to query user

**Returns** errcode.You can query from the errcode.

**resource\_update** (*id, pay\_password, encrypted=True, title=None, body=None, price=None, tags=None, des=None*)

user update data to the ulord-platform

**Parameters**

- **id** (*str*) – data id
- **pay\_password** (*str*) – auther wallet pay password
- **encrypted** (*bool*) – if encrypt password
- **title** (*str*) – resource title
- **body** (*str*) – resource
- **price** (*float*) – resource price
- **tags** (*list*) – resource tags
- **des** (*str*) – resource description

**Returns** errcode.You can query from the errcode.

**start\_web** ()

start web server

**user\_activity** (*token=None, username=None*)

activity.App developer send some ulords to the user.

**Parameters**

- **token** (*str*) – user token
- **username** (*str*) – user name

**Returns** True or False.

**user\_info\_query** (*username=None, token=None*)  
user information

**Parameters**

- **username** (*str*) – username.Default is none.
- **token** (*str*) – user token.Default is none.

**Returns** dict.User info

**user\_infor\_modify** (*password, username=None, token=None, encrypted=False, cellphone=None, email=None, new\_password=None*)  
user information

**Parameters**

- **username** (*str*) – username.Default is none.
- **token** (*str*) – user token.Default is none.
- **encrypted** (*bool*) – if encrypted password.If ture,all password will be decrypted.
- **password** (*str*) – user passowrd to confirm account
- **cellphone** (*str*) – user new cellphone
- **email** (*str*) – user new email
- **new\_password** (*str*) – user new password

**Returns** dict.User info

**user\_login** (*username, password, encrypted=False*)  
user login

**Parameters**

- **username** (*str*) – user name
- **password** (*str*) – user password
- **encrypted** (*bool*) – check if the password is encrypted

**Returns** user token

**user\_logout** (*token=None, username=None*)  
user logout.Change user's timestamp

**Parameters**

- **token** (*str*) – user token
- **username** (*str*) – user name

**Returns** username or errcode.You can query the errcode dict.

**user\_regist** (*username, password, cellphone=None, email=None, wallet=None, pay\_password=None, encrypted=False*)  
user regist

**Parameters**

- **username** (*str*) – user name
- **password** (*str*) – user password



- **cellphone** (*str*) – user cellphone.Default is None.
- **email** (*str*) – user email.Default is None.
- **wallet** (*str*) – user wallet name.Default is username.
- **pay\_password** (*str*) – user wallet password.Default is hash password.
- **encrypted** (*bool*) – check if the password encrypted

**Returns** user token

**class** `ulordapi.user.Senior` (*appkey, secret*)

Bases: `ulordapi.user.Developer`

Senior programmer to develop his application.

## 5.8 utils

**class** `ulordapi.utils.FileHelper`

file helper to operate the files

**changeName** (*originalname, newname*)

change file name from a hash to filename

**Parameters**

- **originalname** (*str*) – a file path, old file name.
- **newname** (*str*) – a file path, new file name.

**Returns** True or False

**getName** (*filename*)

get file name,including type

**Parameters** **filename** (*str*) – a file path

**Returns** file name

**getPureName** (*filename*)

get file name.Doesn't contain type.

**Parameters** **filename** (*str*) – a file path

**Returns** file name

**getSize** (*filename*)

get file size

**Parameters** **filename** (*str*) – a file path

**Returns** file size

**getType** (*filename*)

get file type

**Parameters** **filename** (*str*) – a file path

**Returns** file type

**mergeFile** (*filepath, chunks*)

merge chunks into a file.Used for the multi-threading download

**Parameters**

- **filepath** (*str*) – a file path.Final file.
- **chunks** (*list*) – chunk path list

**Returns** True or False

**readFile** (*filepath*)  
read file comment

**Parameters** **filepath** (*str*) – a file path

**Returns** file comment

**saveFile** (*filepath, source*)  
save source into a filepath. Create path if dir is not existed.

**Parameters**

- **filepath** (*str*) – a file path
- **source** (*unicode*) – message

**Returns** True or False

`ulordapi.utils.ListToDict` (*args*)  
change a list to dict

**Parameters** **args** (*list*) – a list

**Returns** dict

```
>>> a = [1,2,3,4]
>>> print(ListToDict(a))
{1: {2: {3: 4}}}
```

**class** `ulordapi.utils.RSAHelper` (*public\_path, private\_path*)  
Bases: object

rsa helper.Create private key and public key.

**decrypt** (*prikey, message*)  
using private key to decrypt message

**Parameters**

- **prikey** (*Crypto.PublicKey.RSA.RsaKey*) – private key
- **message** (*str*) – need to be decrypted

**Returns** decrypted message

**encry** (*pubkey, comment*)  
using pubkey to encrypt comment

**Parameters**

- **pubkey** (*Crypto.PublicKey.RSA.RsaKey*) – public key
- **comment** (*str*) – information

**Returns** encrypted data

**generate** ()  
generate public key and private key

**load** ()  
read key from the path

`ulordapi.utils.Update(orgen_dict, dict_father)`  
 update dict dict\_father to dict orgen\_dict

**Parameters**

- **orgen\_dict** (*dict*) – base dict
- **dict\_father** (*dict*) – need to update

**Returns** dict orgen\_dict which updates orgen\_dict

```
In [8]: dictb = {
...: "t":2,
...: "ttt":{
...: "123":1,
...: "a":2
...: }
...:
...: }

In [9]: dicta = {
...: "test": 1,
...: "ttt":{
...: "a":1
...: }
...: }

In [10]: Update(dictb, dicta)
Out[10]: {'t': 2, 'test': 1, 'ttt': {'123': 1, 'a': 1}}
```

`ulordapi.utils.generateMD5(value)`  
 generate value's md5

**Parameters** **value** (*str*) – need to encrypt

**Returns** value's md5 or ""

`ulordapi.utils.isCellphone(number)`  
 check if a cellphone

**Parameters** **number** (*str*) – cellphone

**Returns** True of False

`ulordapi.utils.isMail(mail)`  
 check if a mail

**Parameters** **mail** (*str*) – mail

**Returns** True of False

`ulordapi.utils.isUdfsHash(udfshash)`  
 check the string if a udfs hash simple check, TODO advanced check

**Parameters** **udfshash** (*str*) – a string

**Returns** True or False

`ulordapi.utils.json_load_byteified(file_handle)`  
 encode a json file handle :param file\_handle: a json file handle :type file\_handle: handle :return: encoded json(utf-8)

`ulordapi.utils.require(required, tag)`

a decorator function.current function need required function.Tag is a marker which mark the required function is running.

**Parameters**

- **required** (*function*) – required function
- **tag** (*bool*) – mark the required function if has been running.

**Returns** decorator

## 5.9 version

### 5.10 webServer

`ulordapi.webServer.activity()`

activity.Send 10 ulord to new user.

`ulordapi.webServer.add_views()`

add views according to the resource's title

**Returns** current views

`ulordapi.webServer.auth_login_required()`

check token

**Returns** current user

`ulordapi.webServer.blog_delete()`

delete blog :return: errcode.You can query from the errcode dict.

`ulordapi.webServer.blog_list()`

list all blog

**Returns** blog list

`ulordapi.webServer.blog_list_by_ID()`

list blogs by ID

**Returns** blog list

`ulordapi.webServer.blog_publish()`

publish blog

**Returns** claim id,ulord-platform DB ID

`ulordapi.webServer.blog_update()`

update blog

**Returns** claim id,ulord-platform DB id

`ulordapi.webServer.check_bought()`

check the resource if has been bought

**Returns** if has bought return resource hash,if not existed return null,if hasn't bought return none.

`ulordapi.webServer.encrypt()`

`ulordapi.webServer.get_billings()`

query user's billing

`ulordapi.webServer.get_billingsdetail()`  
query user billing detail

`ulordapi.webServer.get_expensebillings()`  
get user expense billing information

`ulordapi.webServer.get_incomebillings()`  
get user income billing information

`ulordapi.webServer.get_pubkey()`  
Get:generate publikey to fronted-end.Post:Check the message if crypted.  
**Returns** get-publickey/post-decrypted message

`ulordapi.webServer.get_userbalance()`  
get user balance  
**Returns** user balance

`ulordapi.webServer.get_userinfo()`  
get user infor  
**Returns** dict.User information

`ulordapi.webServer.get_userpublished()`  
get blog list the user has published

`ulordapi.webServer.get_userpublishednum()`  
get the num of the blogs that user has published

`ulordapi.webServer.login()`  
user login  
**Returns** user token

`ulordapi.webServer.logout()`  
user logout  
**Returns** success

`ulordapi.webServer.modify_userinfo()`  
Delete modify username.It will make publish error

`ulordapi.webServer.pay_ads()`  
get ulord from ads  
**Returns** ads hash

`ulordapi.webServer.pay_blogs()`  
user pay blogs to view  
**Returns** blog hash

`ulordapi.webServer.regist()`  
user regist  
**Returns** user token

`ulordapi.webServer.start()`



## CHAPTER 6

---

### About py-ulord-api

---

Ulord is a P2P value delivery public chain. Through building its blockchain underlying architecture and digital resource distribution protocols, it enables third-party developers to explore their own applications over open-source agreements to form a complete ecology of blockchain technology and applications. Based on various rules and protocols created by Ulord, it loads various types of digital resource application scenarios including text, pictures, music, video and software, providing a direct docking platform for information creators and consumers.

This is a toolkit to save your resource to the ulord-platform.





## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### U

- `ulordapi.config`, [11](#)
- `ulordapi.daemonCLI`, [11](#)
- `ulordapi.errcode`, [11](#)
- `ulordapi.manage`, [11](#)
- `ulordapi.udfs.udfs`, [9](#)
- `ulordapi.up`, [14](#)
- `ulordapi.user`, [17](#)
- `ulordapi.utils`, [21](#)
- `ulordapi.version`, [24](#)
- `ulordapi.webServer`, [24](#)



## A

activity (ulordapi.manage.User attribute), 13  
activity() (in module ulordapi.webServer), 24  
add\_views() (in module ulordapi.webServer), 24  
Ads (class in ulordapi.manage), 11  
amount (ulordapi.manage.Ads attribute), 11  
amount (ulordapi.manage.Content attribute), 12  
amount (ulordapi.manage.Resource attribute), 12  
auth\_login\_required() (in module ulordapi.webServer), 24

## B

balance (ulordapi.manage.User attribute), 13  
blog\_delete() (in module ulordapi.webServer), 24  
blog\_list() (in module ulordapi.webServer), 24  
blog\_list\_by\_ID() (in module ulordapi.webServer), 24  
blog\_publish() (in module ulordapi.webServer), 24  
blog\_update() (in module ulordapi.webServer), 24  
body (ulordapi.manage.Ads attribute), 12  
body (ulordapi.manage.Content attribute), 12  
body (ulordapi.manage.Resource attribute), 12  
boughts (ulordapi.manage.User attribute), 13

## C

calculate\_sign() (ulordapi.up.UlordHelper method), 14  
cat() (ulordapi.udfs.udfs.UdfsHelper method), 10  
cellphone (ulordapi.manage.User attribute), 13  
changeName() (ulordapi.utils.FileHelper method), 21  
check\_bought() (in module ulordapi.webServer), 24  
checkisbought() (ulordapi.up.UlordHelper method), 14  
claimID (ulordapi.manage.Ads attribute), 12  
claimID (ulordapi.manage.Content attribute), 12  
claimID (ulordapi.manage.Resource attribute), 12  
client (class in ulordapi.daemonCLI), 11  
Config (class in ulordapi.config), 11  
config\_edit() (ulordapi.user.Developer method), 17  
config\_show() (ulordapi.user.Developer method), 17  
Content (class in ulordapi.manage), 12  
create() (in module ulordapi.manage), 13

create\_database() (ulordapi.user.Junior method), 18

## D

date (ulordapi.manage.Ads attribute), 12  
date (ulordapi.manage.Content attribute), 12  
date (ulordapi.manage.Resource attribute), 12  
decrypt() (ulordapi.user.Developer method), 17  
decrypt() (ulordapi.utils.RSAHelper method), 22  
delete() (ulordapi.up.UlordHelper method), 14  
description (ulordapi.manage.Ads attribute), 12  
description (ulordapi.manage.Content attribute), 12  
description (ulordapi.manage.Resource attribute), 12  
Developer (class in ulordapi.user), 17  
downloadfile() (ulordapi.udfs.udfs.UdfsHelper method), 10  
downloadhash() (ulordapi.udfs.udfs.UdfsHelper method), 10

## E

edit() (ulordapi.config.Config method), 11  
edit\_config() (in module ulordapi.daemonCLI), 11  
email (ulordapi.manage.User attribute), 13  
encyr() (ulordapi.utils.RSAHelper method), 22  
encrypt() (in module ulordapi.webServer), 24

## F

FileHelper (class in ulordapi.utils), 21  
formatResult() (in module ulordapi.daemonCLI), 11

## G

generate() (ulordapi.utils.RSAHelper method), 22  
generateMD5() (in module ulordapi.utils), 23  
get() (ulordapi.up.UlordHelper method), 14  
get\_billings() (in module ulordapi.webServer), 24  
get\_billingsdetail() (in module ulordapi.webServer), 24  
get\_expensebillings() (in module ulordapi.webServer), 25  
get\_incomebillings() (in module ulordapi.webServer), 25  
get\_pid() (ulordapi.udfs.udfs.Udfs method), 9  
get\_pubkey() (in module ulordapi.webServer), 25

get\_udfs() (ulordapi.udfs.udfs.Udfs method), 9  
 get\_userbalance() (in module ulordapi.webServer), 25  
 get\_userinfo() (in module ulordapi.webServer), 25  
 get\_userpublished() (in module ulordapi.webServer), 25  
 get\_userpublishednum() (in module ulordapi.webServer), 25  
 getName() (ulordapi.utils.FileHelper method), 21  
 getPureName() (ulordapi.utils.FileHelper method), 21  
 getSize() (ulordapi.utils.FileHelper method), 21  
 getType() (ulordapi.utils.FileHelper method), 21

## H

hash\_password() (ulordapi.manage.User method), 13

## I

id (ulordapi.manage.Ads attribute), 12  
 id (ulordapi.manage.Content attribute), 12  
 id (ulordapi.manage.Resource attribute), 12  
 id (ulordapi.manage.Tag attribute), 13  
 id (ulordapi.manage.User attribute), 13  
 isCellphone() (in module ulordapi.utils), 23  
 isMail() (in module ulordapi.utils), 23  
 isUdfsHash() (in module ulordapi.utils), 23

## J

json\_load\_byteified() (in module ulordapi.utils), 23  
 Junior (class in ulordapi.user), 18

## L

list() (ulordapi.udfs.udfs.UdfsHelper method), 10  
 ListToDict() (in module ulordapi.utils), 22  
 load() (ulordapi.utils.RSAHelper method), 22  
 login() (in module ulordapi.webServer), 25  
 logout() (in module ulordapi.webServer), 25

## M

main() (in module ulordapi.daemonCLI), 11  
 mergeFile() (ulordapi.utils.FileHelper method), 21  
 modify\_config() (ulordapi.udfs.udfs.Udfs method), 9  
 modify\_userinfo() (in module ulordapi.webServer), 25

## P

password\_hash (ulordapi.manage.User attribute), 13  
 pay\_ads() (in module ulordapi.webServer), 25  
 pay\_ads() (ulordapi.user.Junior method), 18  
 pay\_blogs() (in module ulordapi.webServer), 25  
 pay\_password (ulordapi.manage.User attribute), 13  
 pay\_resources() (ulordapi.user.Junior method), 18  
 paytouser() (ulordapi.up.UlordHelper method), 14  
 post() (ulordapi.up.UlordHelper method), 14  
 pre1 (ulordapi.manage.Tag attribute), 13  
 pre2 (ulordapi.manage.Tag attribute), 13  
 publish() (ulordapi.up.UlordHelper method), 14

## Q

query() (ulordapi.user.Junior method), 19  
 query\_resource\_by\_ID() (ulordapi.up.UlordHelper method), 15  
 querybalance() (ulordapi.up.UlordHelper method), 15  
 queryBillingDetailByID() (ulordapi.up.UlordHelper method), 15  
 querybillings() (ulordapi.up.UlordHelper method), 15  
 querybillingsdetail() (ulordapi.up.UlordHelper method), 15  
 queryincomebillings() (ulordapi.up.UlordHelper method), 15  
 queryoutgobillings() (ulordapi.up.UlordHelper method), 16  
 querypublishnum() (ulordapi.up.UlordHelper method), 16  
 queryresource() (ulordapi.up.UlordHelper method), 16  
 queryStatisticsByID() (ulordapi.up.UlordHelper method), 15  
 queryuserpublished() (ulordapi.up.UlordHelper method), 16

## R

read() (ulordapi.config.Config method), 11  
 readFile() (ulordapi.utils.FileHelper method), 22  
 regist() (in module ulordapi.webServer), 25  
 regist() (ulordapi.up.UlordHelper method), 17  
 request() (ulordapi.user.Developer method), 17  
 require() (in module ulordapi.utils), 23  
 resource\_views() (ulordapi.user.Junior method), 19  
 Resource (class in ulordapi.manage), 12  
 resource (ulordapi.manage.Content attribute), 12  
 resource (ulordapi.manage.Resource attribute), 12  
 resource (ulordapi.manage.Tag attribute), 13  
 resource\_publish() (ulordapi.user.Junior method), 19  
 resource\_type (ulordapi.manage.Ads attribute), 12  
 resource\_type (ulordapi.manage.Content attribute), 12  
 resource\_type (ulordapi.manage.Resource attribute), 13  
 resource\_update() (ulordapi.user.Junior method), 19  
 resumableDownload() (ulordapi.udfs.udfs.UdfsHelper method), 10  
 return\_result() (in module ulordapi.errcode), 11  
 RSAHelper (class in ulordapi.utils), 22

## S

save() (ulordapi.config.Config method), 11  
 saveFile() (ulordapi.utils.FileHelper method), 22  
 Senior (class in ulordapi.user), 21  
 show\_config() (in module ulordapi.daemonCLI), 11  
 start() (in module ulordapi.webServer), 25  
 start() (ulordapi.udfs.udfs.Udfs method), 9  
 start\_command() (ulordapi.udfs.udfs.Udfs method), 9  
 start\_init() (ulordapi.udfs.udfs.Udfs method), 9  
 start\_web() (ulordapi.user.Junior method), 19

stop() (ulordapi.udfs.udfs.Udfs method), 9

## T

Tag (class in ulordapi.manage), 13  
tagname (ulordapi.manage.Tag attribute), 13  
tags (ulordapi.manage.Ads attribute), 12  
tags (ulordapi.manage.Content attribute), 12  
tags (ulordapi.manage.Resource attribute), 13  
timestamp (ulordapi.manage.User attribute), 13  
title (ulordapi.manage.Ads attribute), 12  
title (ulordapi.manage.Content attribute), 12  
title (ulordapi.manage.Resource attribute), 13  
token (ulordapi.manage.User attribute), 13  
transaction() (ulordapi.up.UlordHelper method), 17

## U

Udfs (class in ulordapi.udfs.udfs), 9  
udfs\_cat() (ulordapi.user.Developer method), 18  
udfs\_download() (ulordapi.user.Developer method), 18  
udfs\_upload() (ulordapi.user.Developer method), 18  
UdfsHelper (class in ulordapi.udfs.udfs), 10  
ulordapi.config (module), 11  
ulordapi.daemonCLI (module), 11  
ulordapi.errcode (module), 11  
ulordapi.manage (module), 11  
ulordapi.udfs.udfs (module), 9  
ulordapi.up (module), 14  
ulordapi.user (module), 17  
ulordapi.utils (module), 21  
ulordapi.version (module), 24  
ulordapi.webServer (module), 24  
UlordHelper (class in ulordapi.up), 14  
Update() (in module ulordapi.utils), 22  
update() (ulordapi.udfs.udfs.UdfsHelper method), 10  
update() (ulordapi.up.UlordHelper method), 17  
UPID (ulordapi.manage.Ads attribute), 11  
UPID (ulordapi.manage.Content attribute), 12  
UPID (ulordapi.manage.Resource attribute), 12  
upload\_file() (ulordapi.udfs.udfs.UdfsHelper method), 10  
upload\_stream() (ulordapi.udfs.udfs.UdfsHelper method), 10  
User (class in ulordapi.manage), 13  
user\_activity() (ulordapi.user.Junior method), 19  
user\_info\_query() (ulordapi.user.Junior method), 20  
user\_infor\_modify() (ulordapi.user.Junior method), 20  
user\_login() (ulordapi.user.Junior method), 20  
user\_logout() (ulordapi.user.Junior method), 20  
user\_regist() (ulordapi.user.Junior method), 20  
userid (ulordapi.manage.Ads attribute), 12  
userid (ulordapi.manage.Content attribute), 12  
userid (ulordapi.manage.Resource attribute), 13  
username (ulordapi.manage.User attribute), 13

## V

verify\_password() (ulordapi.manage.User method), 13  
views (ulordapi.manage.Ads attribute), 12  
views (ulordapi.manage.Content attribute), 12  
views (ulordapi.manage.Resource attribute), 13

## W

wallet (ulordapi.manage.User attribute), 13