arkesel$_pythonDocumentation$

Release 0.1.0

Wilham Opoku Danquah

Aug 23, 2021

CONTENTS:

CHAPTER

ONE

ARKESEL_PYTHON

Python Library for the Arkesel API

- Free software: MIT license
- Documentation: https://arkesel-python.readthedocs.io.

1.1 Arkesel API Python Library

A Python library for the Arkesel API. Allows you to do anything the Arkesel API does, but from within Python apps – send bulk messages , send OTP for phone-number/users authentication, creating and adding contacts to yout groups etc.

1.2 Documentation

- The documentation for the Arkesel API can be found here
- The documentation for the Arkesel Python library documentation can be found here

CHAPTER TWO

INSTALLATION

2.1 Stable release

To install arkesel_python, run this command in your terminal:

```
$ pip install arkesel_python
```

This is the preferred method to install arkesel_python, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for arkesel_python can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/wilham-lynce/arkesel_python
```

Or download the tarball:

```
$ curl -OJL https://github.com/wilham-lynce/arkesel_python/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER

THREE

USAGE

First of all you should ensure that you have an account with Arkesel and hence you do have an API key saved in your .env file / environment. If you don't have one then you can visit this link , create an acccount and login to proceed from there.

To use this Arkesel tool in your project:

```
pip instsall arkesel_python
```

To call classes in your code:

```
* from arkesel_python import ArkeselSMS
* from arkesel_python import SmsInfo
* from arkesel_python import ArkeselOtp
* from arkesel_python import Contacts
```

1. class ArkeselSMS has the following methods:

   ```
   sendSms
   scheduledSms
   webhookSms
   sandBox
   voiceSms
   send_group_sms
   ```

2. class ArkeselOTP has the following methods:

   ```
   sendOtp
   verifyOtp
   ```

3. class SmsInfo has the following methods:

   ```
   smsBalance
   smsDetails
   ```

4. class Contacts has the following methods:

   ```
   #. create_contact_group:

       create_contact_group(group_name: str):: python
       create_contact_group("TEST"):: python
   ```

(continues on next page)

(continued from previous page)

```
#. add_to_contact_group:

    add_contact_to_group(group_name: str, contacts: array):: python
    add_contact_to_group("TEST" , [{"phone_number":"0XXXXXXXXX"}]):: python
```

Sending Bulk SMS:

```
def sendBulkText():
    letter = ArkeselSMS()
    print (letter.sendSms("user" , "example text" , ["0XXXXXXXXX"]))
sendBulkText()
```

Sending Scheduled Bulk SMS:

```
def sendBulkText():
    send = ArkeselSMS()
    print (send.scheduledSms('Trial','just trying this',['0XXXXXXXXX'],"2021-07-01 12:07␣
↪PM"))
sendBulkText()
```

Sending Bulk SMS With Delivery Webhook:

```
def sendWithWebhook():
    send = ArkeselSMS()
    print (send.webhookSms('Trial','just trying this',['0XXXXXXXXX'],"https://aptinc.com/
↪sms/delivery_webhook"))
```

CHAPTER

FOUR

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/wilham-lynce/arkesel_python/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

arkesel_python could always use more documentation, whether as part of the official arkesel_python docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/wilham-lynce/arkesel_python/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *arkesel_python* for local development.

1. Fork the *arkesel_python* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/arkesel_python.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv arkesel_python
$ cd arkesel_python/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 arkesel_python tests
$ python setup.py test or pytest
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/wilham-lynce/arkesel_python/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_arkesel_python
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER

FIVE

HISTORY

5.1 0.1.0 (2021-08-13)

- First release on PyPI.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search