
SharePastes Documentation

Release 0.2.0

Vaidik Kapoor

Sep 27, 2017

Contents

1	What does SharePastes do?	3
2	Contents	5
2.1	Extending SharePastes	5
2.2	Indices and tables	8
3	Requirements	9
4	Installation	11
5	How to use?	13
6	Creating Shortcuts	15
7	Extending SharePastes	17
8	What's next?	19
9	Indices and tables	21

Helps you quickly share code snippets via various services like [Github Gist](#) and [PasteBin.com](#).

CHAPTER 1

What does SharePastes do?

SharePastes provides you with a simple and minimal command line tool that you can use to quickly share whatever you have copied to your clipboard.

Extending SharePastes

SharePastes can be easily extended to work with other similar services.

Deciding an identifier

You first need to decide an identifier for your service that will be used by SharePastes to identify your extension. For example:

- `gist` is for Github's Gist.
- `pastebin` is for Pastebin.com.

This identifier will be used to name your `module` and `class`.

Creating the module for your new extension

Lets assume that name of your extension's identifier is `gdocs` and you are writing this to get support for Google Docs. Then, the name of your module will be `gdocs.py` and the class that you will create in your module will also be `gdocs`.

Sample code for your extension's module (`gdocs.py`):

```
import json
import requests
import sys

from sharepastes import BaseSharePastes
from sharepastes import Config

# notice that the name of the class here is gdocs and it extends
# BaseSharePastes
```

```
class gdocs(BaseSharePastes):
    def __init__(self):
        pass

    # you have to implement api_call method. Make sure that you add an
    # argument to receive the text from your clipboard.
    def api_call(self, text):
        # do something awesome here
        # make API calls to Google Docs to create a new document

        # something like (obviously you must checkout GoogleDocs API docs):
        r = requests.post('http://docs.google.com/document',
                          data=json.dumps({ 'document': text }),
                          headers={ 'Authorization': 'Your auth token' })

        # check if it really happened
        if r.status_code != 201:
            # if not, then exit.
            print 'Error.'
            sys.exit(1)
        else:
            # if it worked, then somehow get the URL of the new document
            # from the response
            response = json.loads(r.text)
            print 'Done: ' + response.document_url

            # Return the document URL that you would want to share.
            # This URL will be copied to your clipboard by SharePastes so
            # that you can quickly paste it.
            return response.document_url
```

BaseSharePastes Class

class sharepastes.core.**BaseSharePastes**

Base class that must be extended while writing a module to extend SharePastes.

api_call(text)

Supposed to do the actual work of posting text to a service.

Parameters **text** (str) – the text that is to be posted.

Returns a valid URL if posting to the service succeeds.

Note: Must be implemented in the inherited class or else this method will raise a `NotImplementedError` exception.

Saving Configuration

Since you are dealing with APIs, you would want to store user credentials or API tokens or auth keys.

SharePastes provides a way to store such information in text files which are saved in `$HOME_DIR` by the name `.sharepastes.conf` on Mac and Linux and similarly on an equivalent location on Windows.

Note: The contents of the config file are JSON encoded.

Using Config (Example)

Lets say you want to save your auth token for Google Docs. This is what you may do:

```
import json
import requests
import sys

from sharepastes import BaseSharePastes
from sharepastes import Config

# notice that the name of the class here is gdocs and it extends
# BaseSharePastes
class gdocs(BaseSharePastes):
    def __init__(self):
        pass

    def _generate_key(self):
        # do something to get your API keys
        r = requests.post('http://www.google.com/authorize/',
                          auth=('username', 'password'))

        if r.status_code == 200:
            resp = json.loads(r.text)

            # get the Config object
            config = Config.get()

            # The config.config variable returns a dict
            # Use it to store your auth keys.
            # Create a new key for your extension by your extension's
            # unique identifier.
            config.config['gdocs'] = {
                'token' = resp.token
            }

            # finally, save it so that the changes are committed to the
            # conf file.
            config.save()

        else:
            print 'Error'
            sys.exit(1)

    def api_call(self, text):
        config = Config.get()
        try:
            auth_token = config.config['gdocs']['token']
        except KeyError:
            self._generate_key()
            auth_token = config.config['gdocs']['token']

        # do something awesome here
        # make API calls to Google Docs to create a new document
        # something like (obviously you must checkout GoogleDocs API docs):
        r = requests.post('http://docs.google.com/document',
                          data=json.dumps({ 'document': text }),
                          headers={ 'Authorization': auth_token })
```

Config Class

class `sharepastes.core.Config`
Handle configuration for your SharePastes extensions.

classmethod `get ()`
Get the static Config object.
Returns the singleton *Config* object.

load ()
Forcefully reload the configuration from the config file.

save ()
Save configuration to the config file.

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

CHAPTER 3

Requirements

- Python 2.7 (that's what I have tested it with).
- **xclip** command for Linux must be installed.
 - `yum install xclip` for **Fedora**.
 - `apt-get install xclip` for **Ubuntu**.

CHAPTER 4

Installation

```
sudo pip install https://github.com/vaidikkp/sharepastes/archive/master.zip
```


CHAPTER 5

How to use?

1. Copy anything to your clipboard.
2. Run the following command: `sharepastes --using <service-name>`
where <service-name> is the service that you want to use:
 - `gist` for Github's Gist
 - `pastebin` for Pastebin.com
3. After successful execution of the above command, you will get link to your post in your terminal and the same will be copied to your clipboard as well.
4. Go ahead and share it with whoever you want to. Simply use your Operating System's shortcut for pasting to paste the URL.

Creating Shortcuts

Obviously, you wouldn't want to run that command everytime you want to share something. So the best use of SharePastes is by creating a shortcut for the above command i.e. `sharepastes --using <service-name>`.

You may create shortcuts in:

1. **Linux**

- (a) GNOME - System Settings > Keyboard > Shortcuts

- (b) *If you use anyother desktop experience software and know the way to do this, please send a pull request.*

2. Mac OS - See [this link](#).

Note: before using your shortcut for the first time, make sure you use the command from the terminal first because SharePastes needs API keys of the service you intend to post your text to.

CHAPTER 7

Extending SharePastes

SharePastes currently supports only Github Gist and Pastebin. But, it can be extended to work with any other similar service that provides an API for the same.

CHAPTER 8

What's next?

1. Perhaps support for more used services.
2. OS notifications using something like PyNotify for Linux and similar for Mac OS and Windows.

CHAPTER 9

Indices and tables

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

A

`api_call()` (`sharepastes.core.BaseSharePastes` method), 6

B

`BaseSharePastes` (class in `sharepastes.core`), 6

C

`Config` (class in `sharepastes.core`), 8

G

`get()` (`sharepastes.core.Config` class method), 8

L

`load()` (`sharepastes.core.Config` method), 8

S

`save()` (`sharepastes.core.Config` method), 8