pyclip

Spencer Phillip Young

Mar 18, 2022

CONTENTS:

1 PyClip **3**
- 1.1 Installation . . . 3
- 1.2 Usage . . . 3
- 1.3 Platform specific notes/issues . . . 4

2 Acknowledgements **5**

3 PyClip API **7**
- 3.1 pyclip base module . . . 7
- 3.2 pyclip cli module . . . 8
- 3.3 pyclip macos_clip module . . . 8
- 3.4 pyclip util module . . . 8
- 3.5 pyclip win_clip module . . . 9
- 3.6 pyclip xclip_clip module . . . 9

4 Indices and tables **11**

Python Module Index **13**

Index **15**

To get started:

```
pip install pyclip
```

Requires Python 3.7+

CHAPTER

ONE

PYCLIP

Cross-platform clipboard utilities supporting both binary and text data.

Some key features include:

- A cross-platform API (supports MacOS, Windows, Linux)
- Can handle arbitrary binary data
- On Windows, some additional clipboard formats are supported

1.1 Installation

Requires python 3.7+

```
pip install pyclip
```

1.2 Usage

pyclip can be used in Python code

```
import pyclip

pyclip.copy("hello clipboard") # copy data to the clipboard
cb_data = pyclip.paste() # retrieve clipboard contents
print(cb_data)  # b'hello clipboard'
cb_text = pyclip.paste(text=True)  # paste as text
print(cb_text)  # 'hello clipboard'

pyclip.clear() # clears the clipboard contents
assert not pyclip.paste()
```

Or a CLI

```
# paste clipboard contents to stdout
python -m pyclip paste

# load contents to the clipboard from stdin
python -m pyclip copy < myfile.text
# same as above, but pipe from another command
some-program | python -m pyclip copy
```

Installing via pip also provides the console script `pyclip`:

```
pyclip copy < my_file.txt
```

This library implements functionality for several platforms and clipboard utilities.

- [x] MacOS
- [x] Windows
- [x] Linux on x11 (with `xclip`)
- [x] Linux on wayland (with `wl-clipboard`)

If there is a platform or utility not currently listed, please request it by creating an issue.

1.3 Platform specific notes/issues

1.3.1 Windows

- On Windows, the `pywin32` package is installed as a requirement.
- On Windows, additional clipboard formats are supported, including copying from a file (like if you right-click copy from File Explorer)

1.3.2 MacOS

MacOS has support for multiple backends. By default, the `pasteboard` package is used.

`pbcopy`/`pbpaste` can also be used as a backend, but does not support arbitrary binary data, which may lead to data being lost on copy/paste. This backend may be removed in a future release.

1.3.3 Linux

Linux on X11 requires `xclip` to work. Install with your package manager, e.g. `sudo apt install xclip` Linux on Wayland requires `wl-clipboard` to work. Install with your package manager, e.g. `sudo apt install wl-clipboard`

CHAPTER

TWO

ACKNOWLEDGEMENTS

Big thanks to Howard Mao for donating the PyClip project name on PyPI to this project.

CHAPTER

THREE

PYCLIP API

To support multiple platforms, PyClip's API is composed of several implementations.

Generally, the public API is provided by `__init__.py`:

`__init__.py`

`pyclip.__init__.clear`(**args*, ***kwargs*)

`pyclip.__init__.copy`(**args*, ***kwargs*)

`pyclip.__init__.paste`(**args*, ***kwargs*)

`pyclip.__init__.wrapif`(*f*)

Here you will find auto-generated API docs for modules contained herein. Loosely, each platform implementation will have its own module.

The base class from which all implementations are derived is in the *`pyclip.base`* module.

3.1 pyclip base module

Provides the abstract base class from which all clipboard implementations are derived.

class `pyclip.base.`**`ClipboardBase`**

Abstract base class for Clipboard implementations.

abstract clear()

abstract copy(*data*, *encoding=None*)

abstract paste(*encoding=None*, *text=None*, *errors=None*)

Return type `Union[str, bytes]`

exception `pyclip.base.`**`ClipboardException`**

exception `pyclip.base.`**`ClipboardSetupException`**

3.2 pyclip cli module

`pyclip.cli.main()`

3.3 pyclip macos_clip module

Provides clipboard for MacOS

class `pyclip.macos_clip.MacOSClip(_backend=None)`

Provides Clipboard functionality for MacOS.

Defers to one of two backends: `_PasteboardBackend` (the default) or `_PBCopyPBPasteBackend`.

clear()

Clear the clipboard contents

Returns

copy(*data*, *encoding=None*)

Parameters

- **data** (`Union[str, bytes]`) – data to copy to the clipboard
- **encoding** (`Optional[str]`) – this parameter is ignored on this backend

Returns

paste(*encoding=None*, *text=None*, *errors=None*)

Retrieve contents of the clipboard

Parameters

- **encoding** (`Optional[str]`) – same meaning as in `bytes.encode`. Implies `text=True`
- **text** (`Optional[bool]`) – if True, bytes object will be en
- **errors** (`Optional[str]`) – same meaning as in `bytes.encode`. Implies `text=True`.

Returns clipboard contents. Return value is bytes by default

or str if any of `encoding`, `text`, or `errors` is provided.

3.4 pyclip util module

`pyclip.util.detect_clipboard()`

Determine what implementation to use based on `sys.platform`

Return type *`ClipboardBase`*

3.5 pyclip win_clip module

Provides clipboard functionality for Windows via the `pywin32` package

exception `pyclip.win_clip.`**`ClipboardNotTextFormatException`**

exception `pyclip.win_clip.`**`UnparsableClipboardFormatException`**

class `pyclip.win_clip.`**`WindowsClipboard`**

`clear`()
Clear the clipboard contents

Return type `None`

Returns

`copy`(*data*, *encoding=None*)
Copy given string into system clipboard.

`paste`(*encoding=None*, *text=None*, *errors=None*)
Returns clipboard contents

Parameters

- **`encoding`** (`Optional[str]`) – same meaning as in `bytes.encode`. Implies `text=True`
- **`text`** (`Optional[bool]`) – if True, bytes object will be en
- **`errors`** (`Optional[str]`) – same meaning as in `bytes.encode`. Implies `text=True`.

Return type `Union[str, bytes]`

Returns clipboard contents. Return value is bytes by default or str if any of `encoding`, `text`, or `errors` is provided.

3.6 pyclip xclip_clip module

Provides the clipboard functionality for Linux via `xclip`

class `pyclip.xclip_clip.`**`XclipClipboard`**

`clear`()
Clear the clipboard contents

Returns

`copy`(*data*, *encoding=None*)
Copy data into the clipboard

Parameters

- **`data`** (`Union[str, bytes]`) – the data to be copied to the clipboard. Can be str or bytes.
- **`encoding`** (`Optional[str]`) – same meaning as in `subprocess.Popen`.

Return type `None`

Returns None

paste(*encoding=None*, *text=None*, *errors=None*)

Retrieve data from the clipboard

Parameters

- **encoding** (`Optional[str]`) – same meaning as in `subprocess.run`
- **text** (`Optional[bool]`) – same meaning as in `subprocess.run`
- **errors** (`Optional[str]`) – same meaning as in `subprocess.run`

Returns the clipboard contents. return type is binary by default. If any of `encoding`, `errors`, or `text` are specified, the result type is str

Other modules:

`__main__.py`

Simple entrypoint for calling pyclip via `python -m pyclip`. Uses *`pyclip.cli.main`* as entrypoint.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

pyclip.__init__, 7
pyclip.base, 7
pyclip.cli, 8
pyclip.macos_clip, 8
pyclip.util, 8
pyclip.win_clip, 9
pyclip.xclip_clip, 9

INDEX

C

clear() (*in module pyclip.__init__*), 7
clear() (*pyclip.base.ClipboardBase method*), 7
clear() (*pyclip.macos_clip.MacOSClip method*), 8
clear() (*pyclip.win_clip.WindowsClipboard method*), 9
clear() (*pyclip.xclip_clip.XclipClipboard method*), 9
ClipboardBase (*class in pyclip.base*), 7
ClipboardException, 7
ClipboardNotTextFormatException, 9
ClipboardSetupException, 7
copy() (*in module pyclip.__init__*), 7
copy() (*pyclip.base.ClipboardBase method*), 7
copy() (*pyclip.macos_clip.MacOSClip method*), 8
copy() (*pyclip.win_clip.WindowsClipboard method*), 9
copy() (*pyclip.xclip_clip.XclipClipboard method*), 9

D

detect_clipboard() (*in module pyclip.util*), 8

M

MacOSClip (*class in pyclip.macos_clip*), 8
main() (*in module pyclip.cli*), 8
module
    pyclip.__init__, 7
    pyclip.base, 7
    pyclip.cli, 8
    pyclip.macos_clip, 8
    pyclip.util, 8
    pyclip.win_clip, 9
    pyclip.xclip_clip, 9

P

paste() (*in module pyclip.__init__*), 7
paste() (*pyclip.base.ClipboardBase method*), 7
paste() (*pyclip.macos_clip.MacOSClip method*), 8
paste() (*pyclip.win_clip.WindowsClipboard method*), 9
paste() (*pyclip.xclip_clip.XclipClipboard method*), 9
pyclip.__init__
    module, 7
pyclip.base
    module, 7
pyclip.cli
    module, 8
pyclip.macos_clip
    module, 8
pyclip.util
    module, 8
pyclip.win_clip
    module, 9
pyclip.xclip_clip
    module, 9

U

UnparsableClipboardFormatException, 9

W

WindowsClipboard (*class in pyclip.win_clip*), 9
wrapif() (*in module pyclip.__init__*), 7

X

XclipClipboard (*class in pyclip.xclip_clip*), 9