
PyQTermWidget Documentation

Release 0.2

Henning Schroeder

Jul 24, 2017

Contents

1	How to use PyQTermWidget	3
2	TODO	7
3	How to get it	9
4	Indices and tables	11

Unlike the older pyqconsole this console widget works with PyQt4 and let's you embed a shell into your application.

The vt100 terminal emulation code is based on AjaxTerm and WebShell. All code is distributed under the General Public License 2.

Contents:

How to use PyQTermWidget

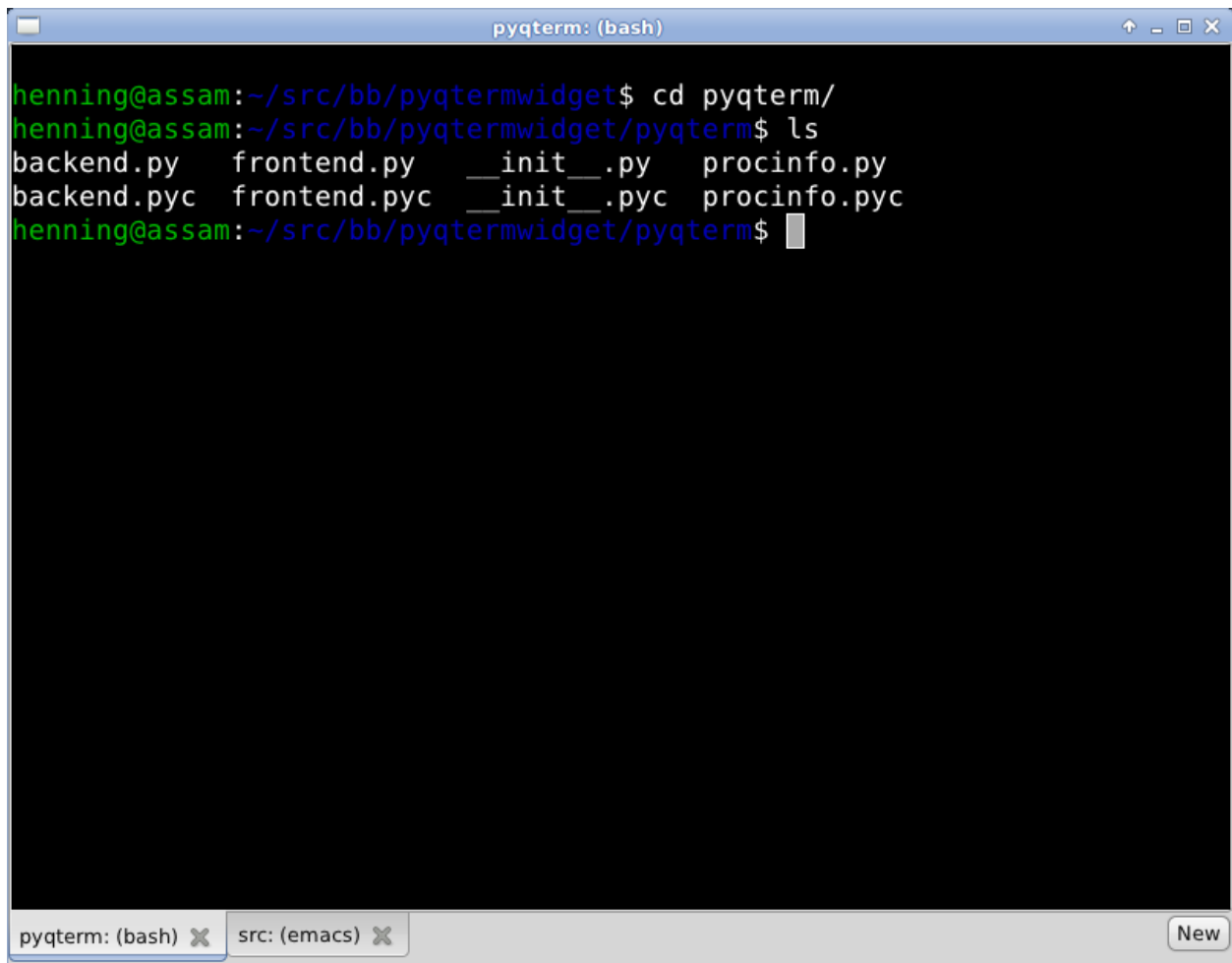
Here is a minimal example:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from PyQt4.QtGui import QApplication
from pyqterm import TerminalWidget

if __name__ == "__main__":
    app = QApplication(sys.argv)
    win = TerminalWidget()
    win.resize(800, 600)
    win.show()
    app.exec_()
```

Look into *demo.py* for a more complete example with a tabbed terminal application.



```
pyqterm: (bash)
henning@assam:~/src/bb/pyqtermwidget$ cd pyqterm/
henning@assam:~/src/bb/pyqtermwidget/pyqterm$ ls
backend.py  frontend.py  __init__.py  procinfo.py
backend.pyc frontend.pyc __init__.pyc  procinfo.pyc
henning@assam:~/src/bb/pyqtermwidget/pyqterm$
```

The constructor has the following signature:

- `def __init__(self, parent=None, command="/bin/bash", font_name="Monospace", font_size=18)`

The widget has the following methods:

- `execute(command="/bin/bash")`
- `send(string)`
- `stop()`
- `pid() -> process id (int)`
- `zoom_in()`
- `zoom_out()`
- `text_selection() -> string`
- `update_screen()`
- `is_alive() -> bool`
- `row_count() -> int`
- `column_count() -> int`
- `text() -> string`

TerminalWidget inherits directly from QWidget, so it has show, hide, setFont, etc.

The widget emits the following signals:

- session_closed()
- return_pressed()

CHAPTER 2

TODO

- add keyboard shortcuts to switch focus to other widgets
- get screen output as string
- history / scrolling
- mouse selection: tripple click (select line)
- keyboard shortcuts to insert from clipboard/xselection
- better rendering speed
- let user deactivate blinking cursor

CHAPTER 3

How to get it

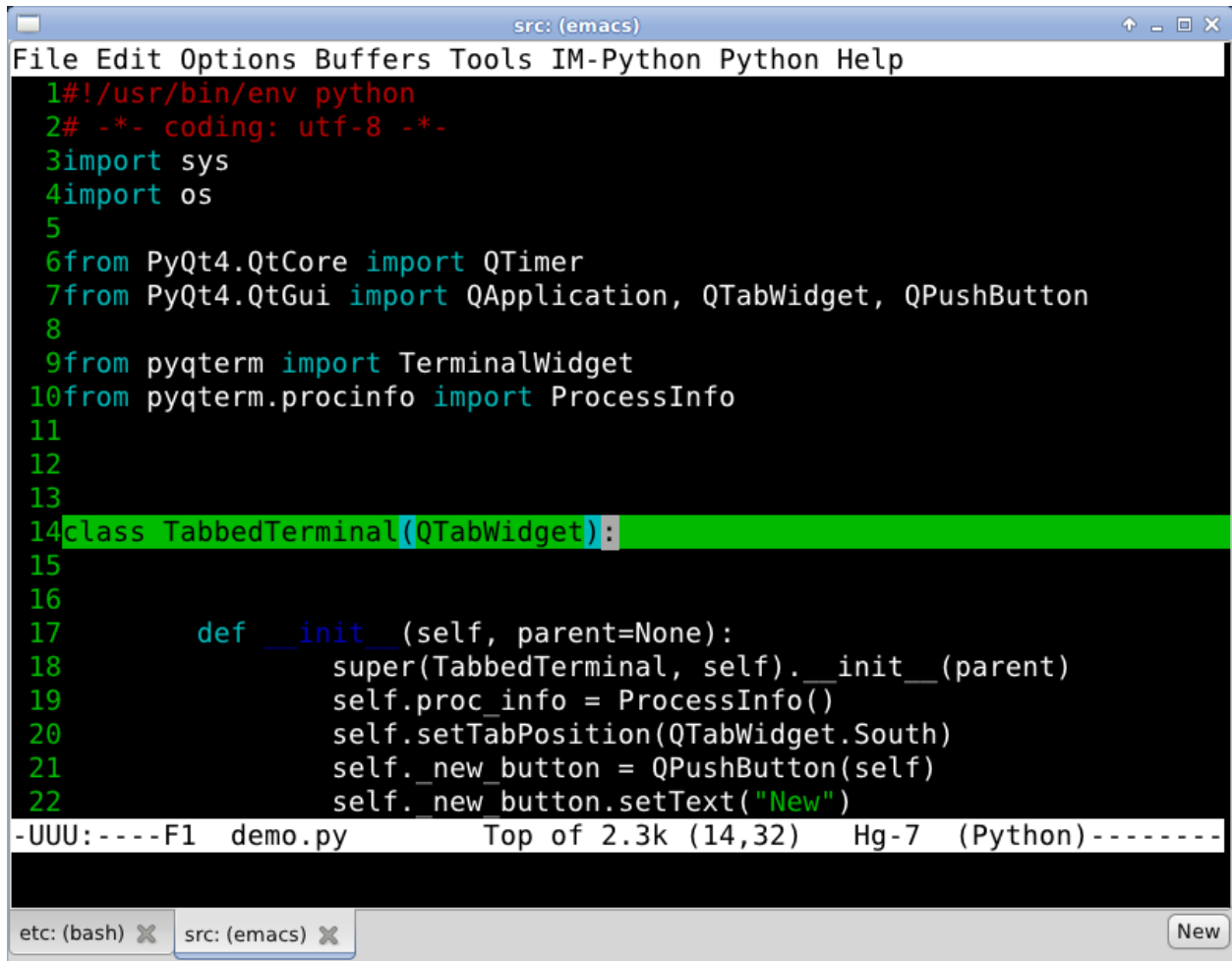
The latest snapshot is available at:

```
https://bitbucket.org/henning/pyqtermwidget/get/tip.tar.bz2
```

To clone the Mercurial repository do:

```
hg clone https://bitbucket.org/henning/pyqtermwidget
```

Or just got to the BitBucket overview page: <https://bitbucket.org/henning/pyqtermwidget>

The image shows a screenshot of an Emacs editor window titled 'src: (emacs)'. The window contains Python code for a class named 'TabbedTerminal'. The code is as follows:

```
1#!/usr/bin/env python
2# -*- coding: utf-8 -*-
3import sys
4import os
5
6from PyQt4.QtCore import QTimer
7from PyQt4.QtGui import QApplication, QTabWidget, QPushButton
8
9from pyqterm import TerminalWidget
10from pyqterm.procinfo import ProcessInfo
11
12
13
14class TabbedTerminal(QTabWidget):
15
16
17     def __init__(self, parent=None):
18         super(TabbedTerminal, self).__init__(parent)
19         self.proc_info = ProcessInfo()
20         self.setTabPosition(QTabWidget.South)
21         self._new_button = QPushButton(self)
22         self._new_button.setText("New")
```

The code is displayed on a black background with green text. The line numbers 1 through 22 are on the left. The class definition on line 14 is highlighted in green. At the bottom of the editor window, there is a status bar showing '-UUU:----F1 demo.py Top of 2.3k (14,32) Hg-7 (Python)-----'. Below the editor window, there are two tabs: 'etc: (bash)' and 'src: (emacs)'. A 'New' button is visible in the bottom right corner of the editor window.

The Python Packaging Index (PyPI) also contains releases: <http://pypi.python.org/pypi/pyqterm>

The generated HTML-documentation is hosted at <http://pyqtermwidget.rtd.org/> (PDF: <http://media.readthedocs.org/pdf/pyqtermwidget/latest/pyqtermwidget.pdf>).

CHAPTER 4

Indices and tables

- genindex
- modindex
- search