

---

# **AdafruitPyPortal Library Documentation**

*Release 1.0*

**Limor Fried**

**Mar 17, 2020**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Documentation</b>	<b>9</b>
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test .....	11
5.2	adafruit_pyportal .....	12
5.2.1	Implementation Notes .....	12
<b>6</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



CircuitPython driver for Adafruit PyPortal.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit library and driver bundle](#).





## CHAPTER 2

---

### Usage Example

---

See `examples/pyportal_simpletest.py`



## CHAPTER 3

---

### Contributing

---

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.



## CHAPTER 4

---

### Documentation

---

For information on building library documentation, please check out [this guide](#).



## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/pyportal\_simpletest.py

```
1  # NOTE: Make sure you've created your secrets.py file before running this example
2  # https://learn.adafruit.com/adafruit-pyportal/internet-connect#whats-a-secrets-file-
   ↳17-2
3  import board
4  from adafruit_pyportal import PyPortal
5
6  # Set a data source URL
7  TEXT_URL = "http://wifitest.adafruit.com/testwifi/index.html"
8
9  # Create the PyPortal object
10 pyportal = PyPortal(url=TEXT_URL, status_neopixel=board.NEOPIXEL)
11
12 # Set display to show REPL
13 board.DISPLAY.show(None)
14
15 # Go get that data
16 print("Fetching text from", TEXT_URL)
17 data = pyportal.fetch()
18
19 # Print out what we got
20 print("-" * 40)
21 print(data)
22 print("-" * 40)
```

## 5.2 adafruit\_pyportal

CircuitPython driver for Adafruit PyPortal.

- Author(s): Limor Fried, Kevin J. Walters

### 5.2.1 Implementation Notes

#### Hardware:

- Adafruit PyPortal

#### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_BusDevice](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

**class** `adafruit_pyportal.Fake_Requests` (*filename*)

For faking 'requests' using a local file instead of the network.

**json** ()

json parsed version for local requests.

**class** `adafruit_pyportal.PyPortal` (\*, *url=None*, *headers=None*, *json\_path=None*, *reg-exp\_path=None*, *default\_bg=0*, *status\_neopixel=None*, *text\_font=None*, *text\_position=None*, *text\_color=8421504*, *text\_wrap=False*, *text\_maxlen=0*, *text\_transform=None*, *json\_transform=None*, *image\_json\_path=None*, *image\_resize=None*, *image\_position=None*, *image\_dim\_json\_path=None*, *caption\_text=None*, *caption\_font=None*, *caption\_position=None*, *caption\_color=8421504*, *image\_url\_path=None*, *success\_callback=None*, *esp=None*, *external\_spi=None*, *debug=False*)

Class representing the Adafruit PyPortal.

#### Parameters

- **url** – The URL of your data source. Defaults to `None`.
- **headers** – The headers for authentication, typically used by Azure API's.
- **json\_path** – The list of json traversal to get data out of. Can be list of lists for multiple data points. Defaults to `None` to not use json.
- **regexp\_path** – The list of regexp strings to get data out (use a single regexp group). Can be list of regexps for multiple data points. Defaults to `None` to not use regexp.
- **default\_bg** – The path to your default background image file or a hex color. Defaults to `0x000000`.
- **status\_neopixel** – The pin for the status NeoPixel. Use `board.NEOPIXEL` for the on-board NeoPixel. Defaults to `None`, no status LED
- **text\_font** (*str*) – The path to your font file for your data text display.
- **text\_position** – The position of your extracted text on the display in an (x, y) tuple. Can be a list of tuples for when there's a list of `json_paths`, for example
- **text\_color** – The color of the text, in `0xRRGGBB` format. Can be a list of colors for when there's multiple texts. Defaults to `None`.



- **text\_wrap** – Whether or not to wrap text (for long text data chunks). Defaults to `False`, no wrapping.
- **text\_maxlen** – The max length of the text for text wrapping. Defaults to 0.
- **text\_transform** – A function that will be called on the text before display
- **json\_transform** – A function or a list of functions to call with the parsed JSON. Changes and additions are permitted for the `dict` object.
- **image\_json\_path** – The JSON traversal path for a background image to display. Defaults to `None`.
- **image\_resize** – What size to resize the image we got from the `json_path`, make this a tuple of the width and height you want. Defaults to `None`.
- **image\_position** – The position of the image on the display as an (x, y) tuple. Defaults to `None`.
- **image\_dim\_json\_path** – The JSON traversal path for the original dimensions of image tuple. Used with `fetch()`. Defaults to `None`.
- **success\_callback** – A function we’ll call if you like, when we fetch data successfully. Defaults to `None`.
- **caption\_text** (*str*) – The text of your caption, a fixed text not changed by the data we get. Defaults to `None`.
- **caption\_font** (*str*) – The path to the font file for your caption. Defaults to `None`.
- **caption\_position** – The position of your caption on the display as an (x, y) tuple. Defaults to `None`.
- **caption\_color** – The color of your caption. Must be a hex value, e.g. `0x808000`.
- **image\_url\_path** – The HTTP traversal path for a background image to display. Defaults to `None`.
- **esp** – A passed ESP32 object, Can be used in cases where the ESP32 chip needs to be used before calling the `pyportal` class. Defaults to `None`.
- **external\_spi** (*busio.SPI*) – A previously declared `spi` object. Defaults to `None`.
- **debug** – Turn on debug print outs. Defaults to `False`.

**fetch** (*refresh\_url=None*)

Fetch data from the url we initialized with, perform any parsing, and display text or graphics. This function does pretty much everything. Optionally update the URL

**get\_local\_time** (*location=None*)

Fetch and “set” the local time of this microcontroller to the local time at the location, using an internet time API.

**Parameters location** (*str*) – Your city and country, e.g. "New York, US".

**hide\_QR** ()

Clear any QR codes that are currently on the screen

**static image\_converter\_url** (*image\_url, width, height, color\_depth=16*)

Generate a converted image url from the url passed in, with the given width and height. `aio_username` and `aio_key` must be set in secrets.

**neo\_status** (*value*)

The status NeoPixel.

**Parameters value** – The color to change the NeoPixel.

**play\_file** (*file\_name*, *wait\_to\_finish=True*)  
Play a wav file.

**Parameters file\_name** (*str*) – The name of the wav file to play on the speaker.

**preload\_font** (*glyphs=None*)  
Preload font.

**Parameters glyphs** – The font glyphs to load. Defaults to `None`, uses alphanumeric glyphs if `None`.

**push\_to\_io** (*feed\_key*, *data*)  
Push data to an adafruit.io feed

**Parameters**

- **feed\_key** (*str*) – Name of feed key to push data to.
- **data** – data to send to feed

**sd\_check** ()  
Returns True if there is an SD card preset and False if there is no SD card. The `_sdcard` value is set in `_init`

**set\_background** (*file\_or\_color*, *position=None*)  
The background image to a bitmap file.

**Parameters file\_or\_color** – The filename of the chosen background image, or a hex color.

**set\_backlight** (*val*)  
Adjust the TFT backlight.

**Parameters val** – The backlight brightness. Use a value between 0 and 1, where 0 is off, and 1 is 100% brightness.

**set\_caption** (*caption\_text*, *caption\_position*, *caption\_color*)  
A caption. Requires setting `caption_font` in `init`!

**Parameters**

- **caption\_text** – The text of the caption.
- **caption\_position** – The position of the caption text.
- **caption\_color** – The color of your caption text. Must be a hex value, e.g. `0x808000`.

**set\_headers** (*headers*)  
Set the headers used by `fetch()`.

**Parameters headers** – The new header dictionary

**set\_text** (*val*, *index=0*)  
Display text, with indexing into our list of text boxes.

**Parameters**

- **val** (*str*) – The text to be displayed
- **index** – Defaults to 0.

**show\_QR** (*qr\_data*, \*, *qr\_size=1*, *x=0*, *y=0*, *hide\_background=False*)  
Display a QR code on the TFT

**Parameters**

- **qr\_data** – The data for the QR code.
- **qr\_size** (*int*) – The scale of the QR code.
- **x** – The x position of upper left corner of the QR code on the display.
- **y** – The y position of upper left corner of the QR code on the display.
- **hide\_background** – Show the QR code on a black background if True.

**wget** (*url, filename, \*, chunk\_size=12000*)

Download a url and save to filename location, like the command wget.

**Parameters**

- **url** – The URL from which to obtain the data.
- **filename** – The name of the file to save the data to.
- **chunk\_size** – how much data to read/write at a time.

**static wrap\_nicely** (*string, max\_chars*)

A helper that will return a list of lines with word-break wrapping.

**Parameters**

- **string** (*str*) – The text to be wrapped.
- **max\_chars** (*int*) – The maximum number of characters on a line before wrapping.



## CHAPTER 6

---

### Indices and tables

---

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



**a**

`adafruit_pyportal`, 11





## A

adafruit\_pyportal (*module*), 11

## F

Fake\_Requests (*class in adafruit\_pyportal*), 12

fetch() (*adafruit\_pyportal.PyPortal method*), 13

## G

get\_local\_time() (*adafruit\_pyportal.PyPortal method*), 13

## H

hide\_QR() (*adafruit\_pyportal.PyPortal method*), 13

## I

image\_converter\_url() (*adafruit\_pyportal.PyPortal static method*), 13

## J

json() (*adafruit\_pyportal.Fake\_Requests method*), 12

## N

neo\_status() (*adafruit\_pyportal.PyPortal method*), 13

## P

play\_file() (*adafruit\_pyportal.PyPortal method*), 14

preload\_font() (*adafruit\_pyportal.PyPortal method*), 14

push\_to\_io() (*adafruit\_pyportal.PyPortal method*), 14

PyPortal (*class in adafruit\_pyportal*), 12

## S

sd\_check() (*adafruit\_pyportal.PyPortal method*), 14

set\_background() (*adafruit\_pyportal.PyPortal method*), 14

set\_backlight() (*adafruit\_pyportal.PyPortal method*), 14

set\_caption() (*adafruit\_pyportal.PyPortal method*), 14

set\_headers() (*adafruit\_pyportal.PyPortal method*), 14

set\_text() (*adafruit\_pyportal.PyPortal method*), 14

show\_QR() (*adafruit\_pyportal.PyPortal method*), 14

## W

wget() (*adafruit\_pyportal.PyPortal method*), 15

wrap\_nicely() (*adafruit\_pyportal.PyPortal static method*), 15