pyanywhere Documentation

Release 0.0.0

Dull Bananas

Nov 21, 2019

Contents

CHAPTER 1

Get Started

These pages contain a brief overview of how this module works.

1.1 Introduction

This module is a wrapper of the PythonAnywhere API. Like the Discord.py library, it defines classes for everything, such as users and consoles.

The `User` class is very important because you have to use it to access anything else. If you are running this module on PythonAnywhere, you can use the `get_current_user` to create a `User` object that represents your user:

```
>>> from pyanywhere.users import get_current_user
>>> user = get_current_user()
>>> user.name
'joe_mama'
```

Every `User` object needs an API token. When you call `get_current_user`, it gets the token using the `API_TOKEN` environment variable. If you are running this module outside of PythonAnywhere, or you want to access the API through a different user, you can manually create a `User` object. The constructor accepts two arguments: the username and the API token:

```
>>> from pyanywhere.users import User
>>> user = User('username', 'XXXXXXX')
```

Like I said earlier, this object is necessaary to access other things. For example, this is how you print the names of all your consoles:

```
>>> for console in user.get_consoles():
...     print(console.name)
...
Bash console 14000012
Bash console 14042069
```

1.2 Consoles

This module has a `Console` class for dealing with consoles. It allows you to read from and write to consoles, as well as kill and start them. The `get_consoles` method in the `User` object is a generator that yields `Console` objects representing all of the consoles the user owns.

Every console on PythonAnywhere is started with an executable name, arguments, and working directory:

- The executable name is the command used to start the console, and when this executable stops running, the console is closed. For example, Bash console have `bash` set as the executable.
- The arguments specify the command line arguments passed to the executable. For example, if you want to start a Bash console with verbose mode enabled, you can set arguments to `'--verbose'`.
- The working directory specifies the initial working directory the console runs in. When you start a console using the PythonAnywhere website, this is always set to your home directory.

1.2.1 Reading from and writing to consoles

`Console` objects have two methods for I/O: `get_latest_output` and `send_input`. According to the official API help page, `get_latest_output` gives you approximately the 500 most recent characters of the console's output. It accepts a boolean argument named `replace_newlines` which tells whether or not to replace `\r\n` with `\n`. It is `False` by default. This code plays back the most recent output from a console:

```
>>> import time
>>> for char in console.get_latest_output(replace_newlines=True):
...     print(char, end='')
...     time.sleep(0.1)
...
(console output will slowly appear)
```

CHAPTER 2

API Reference

2.1 API Reference

2.1.1 Users

class `pyanywhere.users.`**`User`**(*name*, *token=None*)

The User class represents a PythonAnywhere user. It is the central object because it has to be used in order to access anything else. When running this module from PythonAnywhere, the `pyanywhere.users.get_current_user()` function can be used to automatically generate this object.

Parameters

- **`name`** – The username.
- **`token`** – The API token that is associated with the user.

`get_consoles`()

This method gets the consoles that are running on the user.

Returns A geterator yielding `pyanywhere.consoles.Console` objects.

`get_shared_consoles`()

A variant of `get_consoles()` that yields consoles shared with the user.

`start_console`(*exec_name*, *args*, *working_dir*)

This method starts a new console.

Parameters

- **`exec_name`** – The name of the executable used for the console. For example, you can set this to `'bash'` to start a Bash console.
- **`args`** – The command line arguments passed to `exec_name`. This must be a string.
- **`working_dir`** – The initial working directory of the console.

Returns A `pyanywhere.consoles.Console` object.

`pyanywhere.users.`**`get_current_user`**`()`

This function can be used to return a *`pyanywhere.users.User`* object if this module is running on PythonAnyhwere. It gets your username by getting your Linux username, and it gets your API token from the `API_TOKEN` environment variable.

2.1.2 Consoles

`class` `pyanywhere.consoles.`**`Console`**`(`*id*, *owner*, *executable*, *arguments*, *working_dir*, *name*, *url*, *frame_url*`)`

Represents a console. Can be retrieved from the *`pyanywhere.users.User`* class.

`get_latest_output``(`*replace_newlines=False*`)`

Gets the latest output from the console. This will contain CRLF (Windows, `\r\n`) newlines. According to the API's help page, this will return approximately 500 characters.

Parameters **`replace_newlines`** – Replace `\r\n` with `\n`.

Returns A string

`kill``()`

Kills the console.

`send_input``(`*data*`)`

Sends a string to the console's stdin (types into the console).

Parameters **`data`** – The string to send to the console.

CHAPTER 3

Links

- GitHub repository

Python Module Index

Index