pyman Documentation

Release 0.1.0a1

Mark Pittaway

Aug 04, 2018

Contents:

PyMan is a small library that allows you to build your own CLI to manage your projects. This way you do not have to remember commands to run, you simply navigate your CLI to run those commands.

CHAPTER 1

Introduction To PyMan

PyMan is a small library that allows you to build your own CLI to manage your projects. This way you do not have to remember commands to run, you simply navigate your CLI to run those commands.

1.1 Installing

PyMan is available from the PyPi repository.

This means that all you have to do to install PyMan is run the following in a console:

```
$ pip install pyman
Collecting pyman
  Using cached pyman-0.1.3-py2.py3-none-any.whl
Installing collected packages: pyman
Successfully installed pyman-0.1.3
```

1.2 Minimal Example

```
import pyman

pyman.Main("PyMan - Menu Example", [
    pyman.Action.Cmd( "Hello World", "echo 'Testing PyMan'" ),
    pyman.Action.Exit()
]).cli()
```

Example Output

```
================================================================
                      PyMan - Menu Example
================================================================
```

(continues on next page)

(continued from previous page)

```
Main Menu
--------------------
1) Hello World
2) Exit
--------------------

Choice:
```

1.3 How It Works

PyMan uses the idea of Pages and Actions. Each page is made up of a number of actions.

You start off by instanciating the 'Main Menu' class, providing the title you wish to display:

```
menu = pyman.Main("PyMan - Menu Example")
```

From here you add other Pages or Actions

```
menu.add([
    pyman.Action.Exit()
])
```

pyman.Action.Exit is one of the in-built actions. Other inbuilt actions include:

- *pyman.Actions.Cmd*
- *pyman.Actions.Back*
- *pyman.Actions.Exit*

And finally, you start the CLI with:

```
menu.cli()
```

CHAPTER 2

Actions

Each page is made up of a list of actions.

An action can be one of the inbuilt actions, a Page, or your own custom Action.

2.1 Inbuilt Actions

2.1.1 Terminal Command

This command allows you to execute a command in the terminal

```
menu.add([
    pyman.Actions.Cmd("Hello World", "echo 'Hello World'")
])
```

class pyman.Actions.**Cmd**(*name*, *cmd=''*)

Bases: *pyman.Action.Action*

Run a command in the terminal

cmd

The command to run

2.1.2 Back

This command will send you back one page in the menu

```
menu.add([
    pyman.Actions.Back()
])
```

class pyman.Actions.**Back**
Bases: *pyman.Action.Action*

Go back one page

2.1.3 Exit

This command will exit out of the CLI

```
menu.add([
    pyman.Actions.Exit()
])
```

class pyman.Actions.**Exit**
Bases: *pyman.Action.Action*

Exit the CLI

2.2 Action Class

class pyman.Action.**Action**(*name*)
Base Action class to be used by all actions

name
The display name for this action

parent
The parent page this action is assigned to, value assigned when added to a Page

menu
The main page for this CLI instance, value assigned when added to a Page

init()
This method is called when the Action is assigned to a Page

run()
This method is called when the Action is executed. Overwrite it to perform your specific action

CHAPTER 3

Pages

3.1 Main Page

class pyman.Main.**Main**(*title*, *actions=None*, *chars=None*)

Bases: *pyman.Page.Page*

title
The title to display for all menus

chars
A list of characters to use when displaying the menu.

- chars[0]: Character to use for the Title Border. Defaults to '='
- chars[1]: Character to use for the Menu Border. Defaults to '-'
- chars[2]: Characters to use for each entry in the menu. Defaults to ') '

current
A list that houses the stack of pages.

This is used then executing the Action *pyman.Actions.Back*

current_title
The current title that is being displayed

is_automated
A boolean that to determines if the CLI is running in automated mode.

This becomes True if you run pymain.Main.cli() with a list of commands, otherwise it is False.

3.2 Page Class

class pyman.Page.**Page**(*name*, *parent=None*, *menu=None*)

name
The display name for this page

parent
The parent page to this page

menu
The main page for this CLI instance

actions
This is a list where the attached Actions and Pages are stored

init()

choices()

add(*actions*)

add_action(*action*)

run(*index*)

CHAPTER 4

Inbuilt Pages

4.1 Documentation

This page allows you to generate HTML Documentation using Sphinx.

code:

```
menu.add([
    pyman.Doc()
])
```

menu:

```
================================================================
                      PyMan - Menu Example
================================================================

documentation
--------------------
1) Generate Multi-Page HTML
2) Generate Single-Page HTML
3) Clean
4) Back
--------------------

Choice:
```

4.2 VCS - Git

This page allows you to perform git commands on your project

code:

```
menu.add([
    pyman.Git()
])
```

menu:

```
================================================================
                      PyMan - Menu Example
================================================================

git
----------------------------------------
1) Commit
2) Commit File
3) Add File
4) Push
5) History
6) Back
----------------------------------------

Choice:
```

4.3 NoseTest

This page allows you to run Unit Tests using NoseTest

code:

```
menu.add([
    pyman.NoseTest()
])
```

menu:

```
================================================================
                      PyMan - Menu Example
================================================================

testing_(python_nosetest)
----------------------------------------
1) Without Stdout
2) With Stdout
3) Back
----------------------------------------

Choice:
```

4.4 PyPi

This page allows you to package your project and upload it to PyPi

code:

```
menu.add([
    pyman.PyPi()
])
```

menu:

```
================================================================
                      PyMan - Menu Example
================================================================

pypi
----------------------------------------
1) Package Source
2) Package Wheel
3) Upload
4) Back
----------------------------------------

Choice:
```

4.5 Webpack

This page allows you to build js distribution files and/or run webpack-serve

code:

```
menu.add([
    pyman.Webpack()
])
```

menu:

```
================================================================
                      PyMan - Menu Example
================================================================

webpack
----------------------------------------
1) Dev Server
2) Build
3) Back
----------------------------------------

Choice:
```

CHAPTER 5

Custom Page & Action

5.1 Page

To create your own custom page, you only need to implement two methods in your class, __init__ and init.

```
from pyman import Page, Actions

class MyPage(Page):
    def __init__(self):
        super(MyPage, self).__init__("My Custom Page")

    def init( self ):
        self.add([
            Actions.Cmd("Generate Multi-Page HTML", "cd docs; make html; cd .."),
            Actions.Cmd("Generate Single-Page HTML", "cd docs; make singlehtml; cd ..
↪"),
            Actions.Cmd("Clean", "cd docs; make clean; cd .."),
            Actions.Back()
        ])
```

5.2 Action

To create your own custom Action, you only need to implement the run method.

```
from pyman.Action import Action
from pyman import Screen
class MyAction(Action):
    def run(self):
        Screen.write("Custom functionality goes here")
```

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index