Watson - Console

Release 1.0.1

September 30, 2014

Contents

Create console commands with ease.

CHAPTER 1

Build Status

CHAPTER 2

Installation

```
pip install watson-console
```

CHAPTER 3

Testing

Watson can be tested with py.test. Simply activate your virtualenv and run `python setup.py test`.

CHAPTER 4

Contributing

If you would like to contribute to Watson, please feel free to issue a pull request via Github with the associated tests for your code. Your name will be added to the AUTHORS file under contributors.

CHAPTER 5

Table of Contents

5.1 Reference Library

5.1.1 watson.console.colors

watson.console.colors.**fail**(*string*, *terminate=True*)

Wraps a string in the terminal colors for fail.

Example:

```
fail('some text')  # colored text in terminal
```

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the color

watson.console.colors.**header**(*string*, *terminate=True*)

Wraps a string in the terminal colors for headers.

Example:

```
header('some text')  # colored text in terminal
```

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the color

watson.console.colors.**ok_blue**(*string*, *terminate=True*)

Wraps a string in the terminal colors for ok blue.

Example:

```
ok_blue('some text')  # colored text in terminal
```

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the color

watson.console.colors.**ok_green**(*string*, *terminate=True*)
Wraps a string in the terminal colors for ok green.

Example:

```
ok_green('some text')  # colored text in terminal
```

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the color

watson.console.colors.**warning**(*string*, *terminate=True*)
Wraps a string in the terminal colors for warning.

Example:

```
warning('some text')  # colored text in terminal
```

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the color

5.1.2 watson.console.command

class watson.console.command.**Base**
The base command that outlines the required structure for a console command.

Help is automatically invoked when the *-h* or *–help* option is used.

http://docs.python.org/dev/library/argparse.html#the-add-argument-method

Example:

```
# can be executed by 'script.py mycommand'
class MyCommand(BaseCommand):
    name = 'mycommand'

    def execute(self):
        return True

# can be executed by 'script.py mycommand -t something'
class MyCommand(BaseCommand):
    name = 'mycommand'
    arguments = [
        (['-t', '--test'], {'help': 'Do something with -t'})
    ]

    def execute(self):
        return True if self.parsed_args.t else False

# can be executed by 'script.py mycommand something'
class MyCommand(BaseCommand):
    name = 'mycommand'
    arguments = [
        {'dest': 'argument1', 'help': 'This is the help for the argument'}
    ]
```

```
    def execute(self):
        return True if self.parsed_args.argument1 else False
```

parsed_args
Returns the parsed arguments.

Returns list|dict depending on whether or not there have been named arguments.

watson.console.command.**find_commands_in_module**(*module*)
Retrieves a list of all commands within a module.

Returns A list of commands from the module.

5.1.3 watson.console.runner

exception watson.console.runner.**ConsoleError**
An error that should be raised from within the command.

class watson.console.runner.**Runner**(*argv=None*, *commands=None*)
A command line runner that allows new commands to be added and run on demand.

Commands can be added either as a fully qualified name, or imported.

Example:

```
runner = Runner(commands=['module.commands.ACommand'])
runner()
```

__init__(*argv=None*, *commands=None*)

add_command(*command*)
Convenience method to add new commands after the runner has been initialized.

Parameters **command** (*string|class*) – the command to add

add_commands(*commands*)
Convenience method to add multiple commands.

Parameters **commands** (*list|tuple*) – the commands to add

available_commands_usage
Returns the usage text for all commands.

This is used when no commands have been specified.

commands
A list of all commands added to the runner.

Returns OrderedDict containing all the commands.

execute()
Executes the specified command.

get_command(*command_name*)
Returns an initialized command from the attached commands.

get_command_usage(*command*)
Returns the usage string for an individual command.

name
Returns the name of the script that runner was executed from.

usage
Returns the usage text.

This is used when the -h or –help command is invoked.

5.1.4 watson.console.styles

watson.console.styles.**bold**(*string*, *terminate=True*)
Bolds text within the terminal.

Example:

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the styling

watson.console.styles.**format_style**(*string*, *start*, *end='\x1b[0m'*)
Formats text for usage within the terminal.

Example:

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the styling

watson.console.styles.**underline**(*string*, *terminate=True*)
Underlines text within the terminal.

Example:

Parameters

- **string** (*string*) – The string to wrap
- **terminate** (*boolean*) – Whether or not to terminate the styling

W