sufler Documentation

Release 0.0.1

Radoslaw Tomaszewski

Mar 29, 2018

Contents

Sufler - is the tool to help you with generating bash/zsh/powershell/fish autocompletions from YAML file.

CHAPTER 1

Quickstart

1.1 Installation

To install Sufler, open an interactive shell and run:

```
pip install sufler
```

or

```
bash -c "$(curl -fsSL https://raw.githubusercontent.com/limebrains/sufler/master/
↪install.bash)"
```

1.2 Using Sufler

To start using **Sufler**, you need to first install completions.

```
$ sufler install
```

Note: During installation. may appear message that ask for shell completer path if not detected automatically

You will have directory in your home dir where you can install your custom completions.

```
/Users/user/.sufler/
├── completions
│   ├── npm.yml
│   └── pip.yml
└── .config
```

There is repo which accepts PR's with common completions sufler-completions - github

After installation just reload shell, type installed command and press double time **Tab**.

CHAPTER 2

Creation of completion

For example, we want to add completion for command **food**, so we need to add his arguments in nodes after **:**

```
'food': &food
    'fruit': &fruit
        'orange': *fruit
        'banana': *fruit
        'strawberry': *fruit
        'grape':
            'green':
            'red':
        'grapefruit':
            '"ruby red"':
            'yellow':
        '--seedless=': &seedless
            'true': *food
            'false': *food

        '<Exec> ls':
          'rm':
```

Note: We can add reference to any node of tree. E.g. if we want to repeat completions from **'fruit'** after **'orange'**,

CHAPTER 3

Advanced

Sufler has implemented support for advanced markers:

- **TREE**
  Tree marker can be used in any other place for access to previously selected elements from tree.

```
'fruit': &fruit
    '<Exec> ls':
      'cat':
        '<Run> TREE~1 TREE~2':
```

  after

```
'fruit': &fruit
    '<Exec> ls':
      'cat':
        '<Run> cat README.md':
```

- **<File>**
  File marker allow to display in autocomplete all files in current directory.

```
'-r':
    '<File>': *food
```

  after

```
'-r':
    'Library/': *food
    'Desktop/': *food
    'Movies/': *food
    'Pictures/': *food
    'README.md': *food
```

  Note: File can autocomplete path to nested files if recursive parameter('<File rec>') is used.

- **<Exec>**
  Exec mark allow to get output of shell commands as completion.

```
'fruit': &fruit
    '-f':
        '<Exec> ls /': *food
```

after running the command it return

```
'fruit': &fruit
    '-f':
        'Library/': *food
        'System/': *food
        'Volumes/': *food
        'etc/': *food
        'Users/': *food
```

- **<Regex>**

  Regex mark check take regular expression and check that entered string match to expression. If True return what nested node as completion else suggest current node.

```
'--color':
    'red': *food
    'white': *food
    'blue': *food
    '<Regex>.+ack': *m
```

- **<Run>**

  Run mark allow to run any option that will be executed by sufler. In example we use earlier selections to complete and execute command.

```
'fruit': &fruit
    '<Exec> ls':
      'cat':
        '<Run> TREE~1 TREE~2':
```

after

```
$ fruit README.md cat &>/dev/null | sufler run 'cat README.md'
```

In example we use output of *ls* command(README.md) and use *cat* command to display content on screen.