localconfig Documentation

Release 0.2.7

Max Zheng

Jul 20, 2018

Contents

CHAPTER 1

localconfig

A simplified interface to ConfigParser using dot notion with data type / comment support.

CHAPTER 2

Feature Summary

- Simple access to config using dot notion and iterators
- Full compatibility with ConfigParser ini formats (as that is used as the backend)
- Data type support by intelligently guessing the data types based on value on read.
- Multiple config source input (read from string, file pointer, file, or list of them)
- Full comment support / retention on save
- Lazy reading of config sources for performance (only read when a config value is accessed)

CHAPTER 3

Quick Start Tutorial

To install:

```
pip install localconfig
```

Let's say we have a script named 'program' with the following config in ~/.config/program:

```
[Web Server]
# Server host
host = 0.0.0.0

# Server port
port = 8080

# Debug logging
debug = off
```

To read the config, simply do:

```
from localconfig import config

start_server(config.web_server.host, config.web_server.port, config.web_server.debug)

# Or use get method:
# start_server(config.get('Web Server', 'host'),
#              config.get('Web Server', 'port'),
#              config.get('web_server', 'debug'))  # Yes, 'web_server' also works␣
↪here!
#
# Or if the config is in docstring, read from it:
# config.read(__doc__)
#
# Or if the config file is elsewhere:
# config.read('/etc/path/to/config.ini')  # Non-existing file is ignored
#
```

(continues on next page)

(continued from previous page)

```
# Or read from a list of sources
# config.read(['string config', file_path, file_pointer, io.StringIO('config')])
#
# Or create another instance for another config:
# from localconfig import LocalConfig
# config2 = LocalConfig('/etc/path/to/another/config.ini')
```

Now, let's do some inspection:

```
# Iterate over sections and their keys/values
for section in config:
  print section                     # Web Server

  for key, value in config.items(section):
    print key, value, type(value)   # host 0.0.0.0 <type 'str'>
                                    # port 8080 <type 'int'>
                                    # debug False <type 'bool'>

sections = list(config)             # ['Web Server']

# Iterate over keys/values
for key, value in config.web_server:
  print key, value, type(value)     # Same output as above config.items()

items = list(config.web_server)     # [('host', '0.0.0.0'), ('port', 8080), ('debug',␣
↪False)]
items = dict(config.web_server)     # {'host': '0.0.0.0', 'port': 8080, 'debug': False}

# Check if a section or key is set - any non-existing section or key defaults to None.
if config.web_server or config.no_such_section:
  pass

if config.web_server and (config.web_server.port or config.web_server.no_such_key):
  pass
```

To add a section and set a value:

```
config.add_section('App Server', comment='Settings for application server')
config.app_server.host = 'localhost'

# Use `set` if you want to set a comment
config.set('App Server', 'port', 9090, comment='App server port')

# Set value for the DEFAULT section (default value for all other sections)
config.env = 'prod'
```

To write the config:

```
config.save()

# Or simply get the config as a string:
# config_str = str(config)
#
# Or save to a different location:
# config.save('/path/to/save/to.ini')
```

If we open ~/.config/program now, we would see:

CHAPTER 4

Supported Data Types

Data type is guessed based on the value and converted on read.

The following types are supported:

Type	Example Value
int	1
float	2.0
bool	true false yes no on off (case insensitive)
None	none (case insensitive)
str	Any other value not matched by above

CHAPTER 5

Remote Config

Check out: https://pypi.python.org/pypi/remoteconfig

CHAPTER 6

More

Documentation: http://localconfig.readthedocs.org/

PyPI Package: https://pypi.python.org/pypi/localconfig
GitHub Source: https://github.com/maxzheng/localconfig
Report Issues/Bugs: https://github.com/maxzheng/localconfig/issues

Connect: https://www.linkedin.com/in/maxzheng
Contact: maxzheng.os @t gmail.com

CHAPTER 7

API Documentation

7.1 LocalConfig

7.2 Utils

CHAPTER 8

Change Log

8.1 Version 1.1.2

- Allow user to specify custom interpolation instance

8.1.1 Version 1.1.1

- Remove test concurrency and change default env to cover,style

8.1.2 Version 1.1.0

- Add support for DEFAULT section
- Fix style issues
- Fix tox.ini
- Switch to 4 space indents

8.2 Version 1.0.2

- Switch to use configparser.read_file
- Update tox.ini

8.2.1 Version 1.0.1

- Migrate to Python 3.x

8.2.2 Version 0.4.2

- Return None instead of raising NoSectionError/NoOptionError
- Add long description / url
- Update tox.ini

8.2.3 Version 0.4.1

- Update tox.ini
- Update tox.ini to run test by default
- Ensure last source isn't set to empty string by checking sys.arg[0] And skip reading last source when it isn't set.
- Remove ln whitelist from tox
- Remove activate symlink
- Update doc

8.2.4 Version 0.4.0

- Lazy read configs on access and support list of sources
- Rename DotNotationConfig to LocalConfig

8.3 Version 0.3.6

- Fix bug in read for non-existing file

8.3.1 Version 0.3.5

- Only read config file if exists

8.3.2 Version 0.3.4

- Remove __new__ use as it does not work as expected and move SectionAccessor back into DotNotationConfig as private class
- Update changelog
- Add changelog to index

8.3.3 Version 0.3.3

- Add changelog

8.3.4 Version 0.3.2

- Add PyPi package link

CHAPTER 9

Indices and tables

- genindex
- modindex
- search