
xdgconfig

Release 1.2.1

Dogeek

Jun 18, 2021

CONTENTS:

1	How to use	1
1.1	Basic Usage	1
1.2	Setting default values	2
1.3	Using xdgconfig in CLIs	2
2	API Reference	3
3	Indices and tables	5
4	Indices and tables	7
	Index	9

HOW TO USE

1.1 Basic Usage

This library adds several objects that represent configuration for an app. These objects all have the same programming interface, with a few key differences:

- The serializer used is different for each `Config` object. `JsonConfig` serializes the configuration as JSON, while `TomlConfig` does it in a TOML markup.
- Depending on the platform, the configuration path differs
 - On windows, the configuration is stored in `~/AppData/Roaming/prog/config.ext`, with `prog`, and `config.ext` being set by the initializer of the `Config` object.
 - On UNIX-like systems (Cygwin, MacOS and Linux flavors), the configuration is stored in `~/.config/prog/config.ext`.

There are 5 different serializers available. Limitations may apply to different serializers.

- JSON (JSONC)
- TOML
- INI
- YAML
- XML

Some serializers may require third-party libraries to work properly. In that case, you can install them using the extra syntax (`pip3 install xdgconfig[xml]` for instance). The `json` and `ini` serializers are always available.

Each `Config` object accesses its values through a dictionary-like interface, for instance:

```
from xdgconfig import JsonConfig

# Will save in ~/.config/PROG/config
config = JsonConfig('PROG', autosave=True)
config['foo'] = 'bar'
print(config['foo'])
```

It also automatically adds dictionaries when needed, so the following syntax is valid :

```
config['bar']['baz'] = 'tar'
print(config)
# Prints {'bar': {'baz': 'tar'}, 'foo': 'bar'}
```

1.2 Setting default values

You may need some values in the configuration to have defaults. Not to worry, `xdgconfig` has you covered. You simply need to mixin an object with a `_DEFAULTS` class attribute with the `Config` object of your choosing. For instance :

```
from pathlib import Path
from pprint import pprint

from xdgconfig import JsonConfig

class DefaultConfig:
    _DEFAULTS = {
        'logger.level': 'info',
        'logger.verbosity': 3,
        'app.path': str(Path.cwd()),
        'app.credentials.username': 'user',
        'app.credentials.password': 'password',
    }

class Config(DefaultConfig, JsonConfig):
    ...

config = Config('PROG')
print(config['app']['credentials']['username']) # Prints 'user'
config['app']['credentials']['username'] = 'foobar'
print(config['app']['credentials']['username']) # Prints 'foobar'
```

In this example, accessing the value directly will return the default unless it has been set on the config object already.

1.3 Using xdgconfig in CLIs

`xdgconfig` comes with subparsers for most popular CLI libraries such as `click`, `argparse` and `typer`. Below is an example with the `argparse` library. These parsers are of course, optional, but provide a convenient way to add configuration to any CLI.

```
import argparse

from xdgconfig import JsonConfig
from xdgconfig.cli_tools import argparse_config

config = JsonConfig('PROG', 'config.json', autosave=True)

parser = argparse.ArgumentParser()
subparsers = parser.add_subparsers()
subparsers = argparse_config(subparsers, config)
args = vars(parser.parse_args())
args.pop('callback', lambda *a: None)(**args)
```

API REFERENCE

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

`xdgconfig.config.Config`
alias of `xdgconfig.config`.

`xdgconfig.defaultdict.defaultdict`
alias of `xdgconfig.defaultdict`.

`xdgconfig.mixins.JsonMixin`
alias of `xdgconfig.mixins`.

`xdgconfig.mixins.IniMixin`
alias of `xdgconfig.mixins`.

`xdgconfig.mixins.TomlMixin`
alias of `xdgconfig.mixins`.

`xdgconfig.mixins.XmlMixin`
alias of `xdgconfig.mixins`.

`xdgconfig.mixins.YamlMixin`
alias of `xdgconfig.mixins`.

`xdgconfig.JsonConfig`
alias of `xdgconfig`.

`xdgconfig.YamlConfig`
alias of `xdgconfig`.

`xdgconfig.XmlConfig`
alias of `xdgconfig`.

`xdgconfig.YamlConfig`
alias of `xdgconfig`.

`xdgconfig.IniConfig`
alias of `xdgconfig`.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

INDEX

C

`Config` (*in module `xdgconfig.config`*), 5

D

`defaultdict` (*in module `xdgconfig.defaultdict`*), 5

I

`IniConfig` (*in module `xdgconfig`*), 5

`IniMixin` (*in module `xdgconfig.mixins`*), 5

J

`JsonConfig` (*in module `xdgconfig`*), 5

`JsonMixin` (*in module `xdgconfig.mixins`*), 5

T

`TomlMixin` (*in module `xdgconfig.mixins`*), 5

X

`XmlConfig` (*in module `xdgconfig`*), 5

`XmlMixin` (*in module `xdgconfig.mixins`*), 5

Y

`YamlConfig` (*in module `xdgconfig`*), 5

`YamlMixin` (*in module `xdgconfig.mixins`*), 5