
CFG Saver

Aug 10, 2021

Contents:

1	Installation	3
2	Usage	5
3	Attribution	7
4	Indices and tables	9

`cfgsaver` is a python library to save and fetch configuration data in a standard or conventional path and format. By default, it saves the configuration data in *JSON* format to the `~/.config/<pkg_name>/cfg.json` where `~` represents the user's home directory.

CHAPTER 1

Installation

```
pip install cfsaver
```


Import the *cfgsaver* module to save/read configuration values in your source files:

```
from cfgsaver import cfgsaver

def save_config():
    # saves configuration data to ~/.config/<your_package>/cfg.json
    # unless cfgpath parameter is overridden:
    config = {'name': 'Prahlad',
             'language': 'Python',
             'framework': 'Flask'
            }
    cfgsaver.save('<your_package>', config)

def get_config():
    # gets configuration data from ~/.config/<your_package>/cfg.json
    # unless cfgpath parameter is overridden:
    config = cfgsaver.get("<your_package>") #returns None if config file doesn't_
↪exist
```

This will save the configuration data to *~/.config/<your_package>/cfg.json* on both linux and windows.

In case you want to ship some default configuration data packaged with your app, then read the below instructions carefully:

Ensure to include the path to your *cfg.json* in *MANIFEST.in* as follows (you'll have to copy this file from *~/.config/<your_package>* to your source directory for packaging purpose):

```
include <your_package>/cfg.json
```

Override the *PostInstall* class in your *setup.py* like this in order to save your config file to the user's machine after installation:

```
from setuptools import setup, find_packages
from setuptools.command.install import install
import shutil
```

(continues on next page)

```
class PostInstallCommand(install):
    """Post-installation for installation mode."""
    def run(self):
        install.run(self)
        fpath = os.path.join(self.install_lib, your_package)
        fpath = os.path.join(fpath, "cfg.json")
        tpath = os.path.join(os.path.expanduser("~"), ".config/<your_package>/
↪cfg.json")
        if not os.path.isdir(tpath):
            os.makedirs(tpath)
        shutil.move(fpath, tpath)

s = setup(
    name = "your_package",
    packages=find_packages(),
    ..
    ..
    ..
    cmdclass={
        'install': PostInstallCommand,
    },
)
```

CHAPTER 3

Attribution

CHAPTER 4

Indices and tables

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