project python sample

Release 1.0.0

Nov 08, 2019

Contents:

All issues and contributions should be done on ‘Gitlab <https://gitlab.com/Dom31/python-project-template>‘_. Github is used only as a mirror for visibility

CHAPTER 1

Python Package Template

This project is an example of implementation of:

- documentation (sphinx, selfhosted + readthedocs)
- testing (pytest) and coverage (pytest-cov)
- building a package (`setup.py`, `README.md`, `CHANGELOG.md`, `LICENSE.md`)
- command line interface with argparse
- badges for testing, packages, and documentation

Thank's to these site authors:

- https://gitlab.com/costrouc/python-package-template
- https://pypi.org/project/python_boilerplate_template/

Examples and comments:

- https://docs.astropy.org/en/stable/index.html
- https://stackoverflow.com/questions/193161/what-is-the-best-project-structure-for-a-python-application

1.1 Assumptions:

Gitlab will be used for the continuous deployment. See these features:

- pages
- CI/CD

1.2 Contributing

All contributions, bug reports, bug fixes, documentation improvements, enhancements and ideas are welcome. These should be submitted at the Gitlab repository. Github is only used for visibility.

The goal of this project is to in an opinionated way guide modern python packaging development for myself.

1.3 License

No licence (but why not : BSD, MIT, GPL v3,. . .)

CHAPTER 2

Installation and usage

2.1 Requirements

requirements.txt

Note: others external python modules may be mentioned in setup.py and docs/conf.py

2.2 Install with git

```
git clone https://gitlab.com/Dom31/python-package-template/python-package-template.git
```

2.3 Build

```
no build instructions
```

2.4 Tests

```
pytest --cov=people tests/
```

See also the section on tests in .gitlab-ci.yml **Note:** Batch file for windows (coverage output in HTML) : tests/people-cli/run_coverage_windows.bat

2.5 logbook messages

Logging is done for the following purposes:

```
Information gathering
Troubleshooting
Generating statistics
Auditing
Profiling
```

A logging.yaml has been implemented as recommended in http://zetcode.com/python/logging/

Thank's to this file, it is possible to choose the logbook type:

```
loggers:
  log_in_console:
    handlers: [console]
  log_in_console_and_file:
    handlers: [console_debug, file_handler]
  log_in_file:
    handlers: [file_handler]
root:
  handlers: [console_debug, file_handler]
  level: DEBUG
```

Python code:

```
log_cfg = safe_load(resource_stream('people', logging_gui.yaml))
logging.config.dictConfig(log_cfg)
# create (root) logger
# TODO: use loggers in yaml file as define them oas root logger (used in multiple␣
↪python sub modules)
logger = logging.getLogger()
```

More in

- https://gist.github.com/kingspp/9451566a5555fb022215ca2b7b802f19
- https://stackoverflow.com/questions/38323810/does-pythons-logging-config-dictconfig-apply-the-loggers-configuration-setti

2.6 Package data and meta data

If you are going to publish your package, then you probably want to give your potential users some more information about your package, including a description, the name of the author or maintainer, and the url to the package's home page. You can find a complete list of all allowed metadata in the setuptools docs.

The setup.py file has been filled as recommended in https://blog.godatadriven.com/setup-py.

The data file people.json is read thank's to the package

```
from pkg_resources import resource_stream
```

More in https://stackoverflow.com/questions/6028000/how-to-read-a-static-file-from-inside-a-python-package

2.7 GUI

Note on GUI: pygubu is used for its praticity: https://github.com/alejandroautalan/pygubu

Its designer is very simple to use:

```
pygubu-designer
```

Using code from https://github.com/beenje/tkinter-logging-text-widget it is possible to print log messages in a *TK widget console*

```
logger.debug('debug message')
logger.info('info message')
logger.warning('warn message')
logger.error('error message')
logger.critical('critical message')
```

2.8 Usages

```
python people/people_cli.py
python people/people_cli.py init
python people/people_cli.py file data/people.json
python peopleGUI/people_gui.py
```

2.9 Bug reports

Please report bugs and feature requests at https://gitlab.com/Dom31/python-project-template/issues

CHAPTER 3

Documentation

See the command lines in the section on documentation in .gitlab-ci.yml

The full documentation for CLI and API is available on Read-the-Docs

The configuration is in .readthedocs.yml and docs/conf.py

Note: Documentation is generated with the docs/source/Makefile (Linux) and docs/source/Make.bat (Windows).

TODO: use sphinx-automodapi in order to get a PDF Design Document with API classes

3.1 Code-Style

We use black as code formatter, so you'll need to format your changes using the black code formatter

Just run

```
cd python-gitlab/
pipinstall --user black
black .
```

to format your code according to our guidelines.

CHAPTER 4

Release notes

This page describes important changes between python-gitlab releases.

4.1 1.0

- Initialization. . . just a try. . .

CHAPTER 5

Indices and tables

- genindex
- modindex
- search