
gen_py_module Documentation

Release

https://github.com/vroncevic/gen_py_module/releases

Vladimir Roncevic <elektron.ronca@gmail.com>

Mar 23, 2022

Contents

1	gen_py_module	3
1.1	gen_py_module package	3
1.1.1	Subpackages	3
1.1.1.1	gen_py_module.module package	3
1.1.2	Module contents	7
2	Installation	9
3	Dependencies	11
4	Generation flow of py module	13
5	Tool structure	15
6	Copyright and licence	17
7	Indices and tables	19
	Python Module Index	21
	Index	23

gen_py_module is tool for generation PY Module.

Developed in `python` code: **100%**.

The README is used to introduce the modules and provide instructions on how to install the modules, any machine dependencies it may have and any other information that should be provided before the modules are installed.

1.1 gen_py_module package

1.1.1 Subpackages

1.1.1.1 gen_py_module.module package

Submodules

gen_py_module.module.module_selector module

Module module_selector.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_module is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_py_module is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ModuleSelector with attribute(s) and method(s). Selecting python template module for generating process.

class gen_py_module.module.module_selector.**ModuleSelector**

Bases: object

Defined class ModuleSelector with attribute(s) and method(s). Selecting python template module for generating process. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

choose_module - selecting type of module.
format_name - formatting name for file module.
__str__ - dunder method for ModuleSelector.

```
GEN_VERBOSE = 'GEN_PY_MODULE::MODULE::MODULE_SELECTOR'
```

```
classmethod choose_module (config, verbose=False)  
Select module type.
```

Parameters *verbose* (<bool>) – enable/disable verbose option.

Returns project template selected | None.

Return type <str> | <NoneType>

Exceptions None

gen_py_module.module.read_template module

Module read_template.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_module is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_py_module is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ReadTemplate with attribute(s) and method(s). Created API for reading a template and return a string representation.

```
class gen_py_module.module.read_template.ReadTemplate (verbose=False)  
Bases: ats_utilities.config_io.base_check.FileChecking
```

Defined class ReadTemplate with attribute(s) and method(s). Created API for reading a template and return a string representation. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
TEMPLATE_DIR - prefix path to templates.
__template_dir - template dir path.

methods

__init__ - initial constructor.
get_template - getter for template dir path.
read - read a template and return a content or None.
__str__ - dunder method for ReadTemplate.

```
GEN_VERBOSE = 'GEN_PY_MODULE::MODULE::READ_TEMPLATE'
```

```
TEMPLATE_DIR = '/../conf/template/'
```

```
VERBOSE = 'ATS_UTILITIES'
```

```
get_template ()  
Getter for template dir path.
```

Returns template dir path | None.

Return type <str> | <NoneType>

Exceptions None

read (*template_name*, *verbose=False*)

Read a template and return a content or None.

Parameters

- **template_name** (<str>) – template file name.
- **verbose** (<bool>) – enable/disable verbose option.

Returns template module content | None.

Return type <str> | <NoneType>

Exceptions ATSTypeError | ATSBadCallError

gen_py_module.module.write_template module

Module write_template.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_module is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_py_module is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class WriteTemplate with attribute(s) and method(s). Created API for writing a template content with parameters to a file.

class gen_py_module.module.write_template.**WriteTemplate** (*verbose=False*)

Bases: ats_utilities.config_io.base_check.FileChecking

Defined class WriteTemplate with attribute(s) and method(s). Created API for writing a template content with parameters to a file. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

__check_setup - check setup status.

methods

__init__ - initial constructor.

get_check_setup - getter for checking status.

write - write a template content with parameters to a file.

__str__ - dunder method for WriteTemplate.

GEN_VERBOSE = 'GEN_PY_MODULE::MODULE::WRITE_TEMPLATE'

VERBOSE = 'ATS_UTILITIES'

get_check_setup ()

Getter for checking status.

Returns boolean status, True (checked and ok) | False.

Return type <bool>

Exceptions None

write (*module_content*, *module_name*, *verbose=False*)

Write a template content with parameters to a file.

Parameters

- **module_content** (*<str>*) – template content.
- **module_name** (*<str>*) – file name.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type *<bool>*

Exceptions *ATSTypeError* | *ATSBadCallError*

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_module is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_py_module is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class GenModule with attribute(s) and method(s). Generate python module by template and parameters.

class `gen_py_module.module.GenModule` (*verbose=False*)

Bases: `ats_utilities.config_io.base_check.FileChecking`

Defined class GenModule with attribute(s) and method(s). Generate python module by template and parameters.

It defines:

attributes

`GEN_VERBOSE` - console text indicator for process-phase.

`PRO_STRUCTURE` - project setup (templates).

`__reader` - reader API.

`__writer` - writer API.

`__config` - project setup in dict format.

methods

`__init__` - initial constructor.

`get_reader` - getter for reader object.

`get_writer` - getter for writer object.

`gen_module` - generate file python module.

`__str__` - dunder method for GenModule.

```
GEN_VERBOSE = 'GEN_PY_MODULE::MODULE::GEN_MODULE'
```

```
PRO_STRUCTURE = '././conf/project.yaml'
```

```
VERBOSE = 'ATS_UTILITIES'
```

gen_module (*module_name*, *verbose=False*)

Generate file python module.

Parameters

- **module_name** (*<str>*) – parameter module name.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type *<bool>*

Exceptions ATSTypeError | ATSBadCallError

get_reader ()

Getter for reader object.

Returns read template object.

Return type *<ReadTemplate>*

Exceptions None

get_writer ()

Getter for writer object.

Returns write template object.

Return type *<WriteTemplate>*

Exceptions None

1.1.2 Module contents

Module `gen_py_module.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `gen_py_module` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `gen_py_module` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `GenPyModule` with attribute(s) and method(s). Load a base info, create an CLI interface and run operation(s).

class `gen_py_module.GenPyModule` (*verbose=False*)

Bases: `ats_utilities.cli.cfg_cli.CfgCLI`

Defined class `GenPyModule` with attribute(s) and method(s). Load a base info, create an CLI interface and run operation(s). It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

CONFIG - tool info file path.

LOG - tool log file path.

LOGO - logo for splash screen.

OPS - list of tool options.

logger - logger object API.

methods

`__init__` - initial constructor.
`process` - process and generate module setup.py.
`__str__` - dunder method for GenPyModule.

```
CONFIG = '/conf/gen_py_module.cfg'
```

```
GEN_VERBOSE = 'GEN_PY_MODULE'
```

```
LOG = '/log/gen_py_module.log'
```

```
LOGO = '/conf/gen_py_module.logo'
```

```
OPS = ['-g', '--gen', '-v', '--verbose', '--version']
```

```
VERBOSE = 'ATS_UTILITIES'
```

`process` (*verbose=False*)

Process and run operation.

Parameters `verbose` (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type *<bool>*

Exceptions None

CHAPTER 2

Installation

Navigate to release [page](#) download and extract release archive.

To install **gen_py_module** type the following

```
tar xvzf gen_py_module-x.y.z.tar.gz
cd gen_py_module-x.y.z/
# python2
pip install -r requirements.txt
python setup.py install_lib
python setup.py install_data
python setup.py install_egg_info
# python3
pip3 install -r requirements.txt
python3 setup.py install_lib
python3 setup.py install_data
python3 setup.py install_egg_info
```

You can use Docker to create image/container, or You can use pip to install

```
# python2
pip install gen-py-module
# python3
pip3 install gen-py-module
```


CHAPTER 3

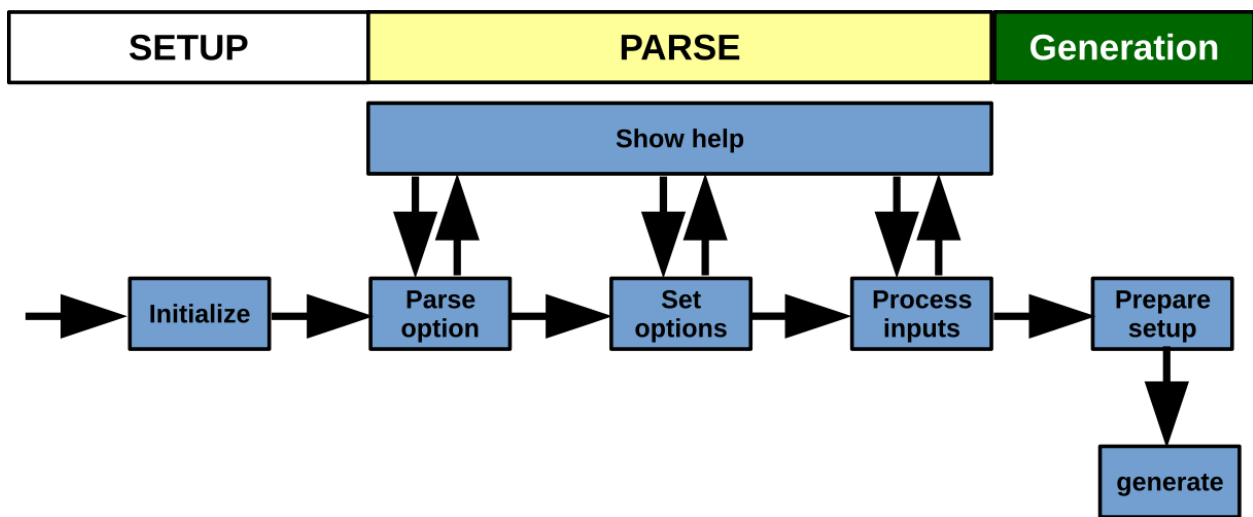
Dependencies

gen_py_module requires next modules and libraries

- [ats-utilities](#) - Python App/Tool/Script Utilities

Generation flow of py module

Base flow of generation process



gen_py_module is based on OOP

Code structure

```
gen_py_module/
├── conf/
│   ├── gen_py_module.logo
│   ├── gen_py_module.cfg
│   ├── gen_py_module_util.cfg
│   ├── project.yaml
│   └── template/
│       ├── abstract_abc_class.template
│       ├── abstract_base_class.template
│       ├── class.template
│       ├── empty.template
│       └── main.template
├── __init__.py
├── log/
│   └── gen_py_module.log
├── module/
│   ├── __init__.py
│   ├── module_selector.py
│   ├── read_template.py
│   └── write_template.py
└── run/
    └── gen_py_module_run.py
```


CHAPTER 6

Copyright and licence

Copyright (C) 2017 by vrontcevic.github.io/gen_py_module

gen_py_module is free software; you can redistribute it and/or modify it under the same terms as Python itself, either Python version 2.x/3.x or, at your option, any later version of Python 3 you may have available.

Lets help and support PSF.



CHAPTER 7

Indices and tables

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

g

`gen_py_module`, 7
`gen_py_module.module`, 6
`gen_py_module.module.module_selector`, 3
`gen_py_module.module.read_template`, 4
`gen_py_module.module.write_template`, 5

C

choose_module() (*gen_py_module.module.module_selector.ModuleSelector* *GenPyModule* attribute), 8
 class method), 4
 CONFIG (*gen_py_module.GenPyModule* attribute), 8

G

gen_module() (*gen_py_module.module.GenModule* method), 6
 gen_py_module(module), 7
 gen_py_module.module(module), 6
 gen_py_module.module.module_selector(module), 3
 gen_py_module.module.read_template(module), 4
 gen_py_module.module.write_template(module), 5
 GEN_VERBOSE (*gen_py_module.GenPyModule* attribute), 8
 GEN_VERBOSE (*gen_py_module.module.GenModule* attribute), 6
 GEN_VERBOSE (*gen_py_module.module.module_selector.ModuleSelector* attribute), 4
 GEN_VERBOSE (*gen_py_module.module.read_template.ReadTemplate* attribute), 4
 GEN_VERBOSE (*gen_py_module.module.write_template.WriteTemplate* attribute), 5
 GenModule (class in *gen_py_module.module*), 6
 GenPyModule (class in *gen_py_module*), 7
 get_check_setup() (*gen_py_module.module.write_template.WriteTemplate* method), 5
 get_reader() (*gen_py_module.module.GenModule* method), 7
 get_template() (*gen_py_module.module.read_template.ReadTemplate* method), 4
 get_writer() (*gen_py_module.module.GenModule* method), 7

L

LOGO (*gen_py_module.GenPyModule* attribute), 8
 LOGO (*gen_py_module.GenPyModule* attribute), 8

M

ModuleSelector (class in *gen_py_module.module.module_selector*), 3

O

OPS (*gen_py_module.GenPyModule* attribute), 8

P

PRO_STRUCTURE (*gen_py_module.module.GenModule* attribute), 6
 process() (*gen_py_module.GenPyModule* method), 8

R

read() (*gen_py_module.module.read_template.ReadTemplate* method), 5
 ReadTemplate (class in *gen_py_module.module.read_template*), 4

T

TEMPLATE_DIR (*gen_py_module.module.read_template.ReadTemplate* attribute), 4

V

VERBOSE (*gen_py_module.GenPyModule* attribute), 8
 VERBOSE (*gen_py_module.module.GenModule* attribute), 6
 VERBOSE (*gen_py_module.module.read_template.ReadTemplate* attribute), 4
 VERBOSE (*gen_py_module.module.write_template.WriteTemplate* attribute), 5

W

write() (*gen_py_module.module.write_template.WriteTemplate* method), 6

```
WriteTemplate          (class          in  
    gen_py_module.module.write_template),  
5
```