
gen_py_tool Documentation

Release

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

Vladimir Roncevic <elektron.ronca@gmail.com>

May 24, 2021

Contents:

1	gen_py_tool	3
1.1	gen_py_tool package	3
1.1.1	Subpackages	3
1.1.1.1	gen_py_tool.pro package	3
1.1.2	Module contents	34
2	Installation	37
3	Dependencies	39
4	Generation flow of py tool	41
5	Tool structure	43
6	Copyright and licence	47
7	Indices and tables	49
	Python Module Index	51
	Index	53

gen_py_tool is package for generation of Python tool.

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_tool package

1.1.1 Subpackages

1.1.1.1 gen_py_tool.pro package

Subpackages

gen_py_tool.pro.config package

Submodules

gen_py_tool.pro.config.pro_name module

Module pro_name.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProName with attribute(s) and method(s). Defined API for project name with preparations for generation.

class gen_py_tool.pro.config.pro_name.ProName (*verbose=False*)

Defined class ProName with attribute(s) and method(s). Defined API for project name with preparations for generation. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__verbose - enable/disable verbose option.
__pro_name - project name.

methods

__init__ - initial constructor.
pro_name - property methods for set/get operations.
is_pro_name_ok - checking is project name ok.
__str__ - dunder method for ProName.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::CONFIG::PRO_NAME'

is_pro_name_ok()

Checking is project name ok.

Returns boolean status, True (not None) | False.

Return type <bool>

Exceptions None

pro_name

Property method for getting project name.

Returns formatted project name | None.

Return type <str> | <NoneType>

Exceptions None

gen_py_tool.pro.config.pro_selector module

Module pro_selector.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProjectSelector with attribute(s) and method(s). Created API for project selector during generation process.

class gen_py_tool.pro.config.pro_selector.**ProjectSelector**

Defined class ProjectSelector with attribute(s) and method(s). Created API for project selector during generation process. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
TYPE_KEY - project type key.
SCHEMA_KEY - project schema key.
ELEMENT_KEY - project element key.
TEMPLATE_KEY - project template key.

methods

check_config_keys - checking configuration keys.
select_pro_type - select project type.
__str__ - dunder method for ProjectSelector.

ELEMENT_KEY = 'elements'

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::CONFIG::PROJECT_SELECTOR'

SCHEMA_KEY = 'schemas'

TEMPLATE_KEY = 'templates'

TYPE_KEY = 'types'

classmethod check_config_keys (*config*, *verbose=False*)
Checking configuration keys.

Parameters

- **config** (*<dict>*) – dictionary with configuration.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (all ok) | False.

Return type *<bool>*

Exceptions ATSTypeError | ATSBadCallError

classmethod select_pro_type (*config*, *verbose=False*)
Select project type.

Parameters

- **config** (*<dict>*) – dictionary with configuration.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns project type and project ID | None and None.

Return type *<str> <int> | <NoneType> <NoneType>*

Exceptions ATSTypeError | ATSBadCallError

gen_py_tool.pro.config.pro_type module

Module pro_type.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProType with attribute(s) and method(s). Defined API for project type with preparations for generation.

class gen_py_tool.pro.config.pro_type.**ProType** (*verbose=False*)

Defined class ProType with attribute(s) and method(s). Defined API for project type with preparations for generation. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__verbose - enable/disable verbose option.
__pro_type - project type.

methods

__init__ - initial constructor.
pro_type - property methods for set/get operations.
is_pro_type_ok - checking is project type ok.
__str__ - dunder method for ProType.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::CONFIG::PRO_TYPE'

is_pro_type_ok()

Checking is project type ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

pro_type

Property method for getting project type.

Returns formatted project type | None.

Return type <str> | <NoneType>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProConfig with attribute(s) and method(s). Defined project configuration container.

class gen_py_tool.pro.config.ProConfig(verbose=False)

Defined class ProConfig with attribute(s) and method(s). Defined project configuration container. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__verbose - enable/disable verbose option.
__config - configuration dictionary.

methods

__init__ - initial constructor.
pro_name - property methods for set/get operations.
is_config_ok - checking is project configuration ok.
__str__ - dunder method for ProConfig.

```
GEN_VERBOSE = 'GEN_PY_TOOL::PRO::CONFIG::PRO_CONFIG'
```

config

Property method for getting project configuration.

Returns formatted project configuration | None.

Return type <dict> | <NoneType>

Exceptions None

is_config_ok()

Checking is project configuration ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

gen_py_tool.pro.element package

Submodules

gen_py_tool.pro.element.element_container module

Module element_container.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ElementContainer with attribute(s) and method(s). Created API for element container.

class gen_py_tool.pro.element.element_container.**ElementContainer** (*verbose=False*)
Defined class ElementContainer with attribute(s) and method(s). Created API for element container. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

__verbose - enable/disable verbose option.

__element - element object.

methods

__init__ - initial constructor.

element - get/set property element object.

is_element_ok - checking is project element ok.

__str__ - dunder method for ElementContainer.

```
GEN_VERBOSE = 'GEN_PY_TOOL::PRO::ELEMENT::ELEMENT_CONTAINER'
```

element

Get element object property.

Returns element object | None.

Return type <dict> | <NoneType>

Exceptions None

is_element_ok()

Checking is project element ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

gen_py_tool.pro.element.element_keys module

Module element_keys.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ElementKeys with attribute(s). Created attributes for processing tool/gen element.

class gen_py_tool.pro.element.element_keys.ElementKeys

Defined class ElementKeys with attribute(s). Created attributes for processing tool/gen element. It defines:

attributes

ROOT_KEY - element key for elements.

TOOL_NAME_KEY - element key for TOOL_NAME.

TOOL_UPPER_KEY - element key for TOOL_UPPER.

TOOL_CLASS_KEY - element key for TOOL_CLASS.

TOOL_YEAR_KEY - element key for YEAR.

methods

None.

ROOT_KEY = 'elements'

TOOL_CLASS_KEY = 'TOOL_CLASS'

TOOL_NAME_KEY = 'TOOL_NAME'

TOOL_UPPER_KEY = 'TOOL_UPPER'

TOOL_YEAR_KEY = 'YEAR'

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ElementLoader with attribute(s) and method(s). Created API for element checking and loding.

class gen_py_tool.pro.element.**ElementLoader** (*element_file*, *verbose=False*)

Bases: `ats_utilities.config_io.base_check.FileChecking`, `gen_py_tool.pro.element.element_container.ElementContainer`, `gen_py_tool.pro.element.element_keys.ElementKeys`

Defined class ElementLoader with attribute(s) and method(s). Created API for element checking and loding. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

CONF_DIR - configuration directory.

methods

`__init__` - initial constructor.

`check_root_key` - check root key for element.

`process_element` - process element object.

`__str__` - dunder method for ElementLoader.

`CONF_DIR = '/../../conf/element/'`

`GEN_VERBOSE = 'GEN_PY_TOOL::PRO::ELEMENT::ELEMENT_LOADER'`

`VERBOSE = 'ATS_UTILITIES'`

`check_root_key` (*verbose=False*)

Check root key for element.

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

Returns boolean status, True (success) | False.

Return type *<bool>*

Exceptions None

`process_element` (*pro_name*, *verbose=False*)

Process element object.

Parameters

- **pro_name** (*<str>*) – project name.

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

Returns boolean status, True (success) | False.

Return type *<bool>*

Exceptions `ATSTypeError` | `ATSBadCallError`

gen_py_tool.pro.factory package

Subpackages

gen_py_tool.pro.factory.collectiner package

Subpackages

gen_py_tool.pro.factory.collectiner.gen package

Submodules

gen_py_tool.pro.factory.collectiner.gen.base module

Module base.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 BaseCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase.

```
class gen_py_tool.pro.factory.collectiner.gen.base.BaseCollectiner (verbose=False)
    Bases:      gen_py_tool.pro.config.pro_name.ProName,      gen_py_tool.pro.config.
                pro_type.ProType,      gen_py_tool.pro.schema.schema_container.SchemaContainer,
                gen_py_tool.pro.template.template_container.TemplateContainer
```

Defined class BaseCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase. It defines:

attributes

None.

methods

__init__ - initial constructor.

is_container_ok - checking is container ok.

__str__ - dunder method for BaseCollectiner.

is_container_ok ()

Checking is container ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool

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 ProCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase.

class `gen_py_tool.pro.factory.collectiner.gen.ProCollectiner` (*verbose=False*)
Bases: `gen_py_tool.pro.factory.collectiner.gen.base.BaseCollectiner`

Defined class ProCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase.
It defines:

attributes

None.

methods

`__init__` - initial constructor.
`update_gen_name` - peparing gen name for project.
`update_gen_class` - preparing gen class for project.
`update_gen_run` - preparing gen run for project.
`update_gen_pro_class` - preparing gen class for project.
`update_gen_conf` - preparing gen conf for project.
`update_gen_log` - preparing gen log for project.
`__str__` - dunder method for ProCollectiner.

update_gen_class()

Preparing generator project setup.

Exceptions None

update_gen_conf()

Preparing generator project setup.

Exceptions None

update_gen_log()

Preparing generator project setup.

Exceptions None

update_gen_name()

Preparing generator name for project schema.

Exceptions None

update_gen_pro_class()

Preparing generator project class setup.

Exceptions None

update_gen_run()

Preparing generator project setup.

Exceptions None

gen_py_tool.pro.factory.collectiner.tool package

Submodules

gen_py_tool.pro.factory.collectiner.tool.base module

Module base.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 BaseCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase.

class gen_py_tool.pro.factory.collectiner.tool.base.**BaseCollectiner** (*verbose=False*)
Bases: `gen_py_tool.pro.config.pro_name.ProName`, `gen_py_tool.pro.config.pro_type.ProType`, `gen_py_tool.pro.schema.schema_container.SchemaContainer`, `gen_py_tool.pro.template.template_container.TemplateContainer`

Defined class BaseCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase. It defines:

attributes

None.

methods

`__init__` - initial constructor.

`is_container_ok` - checking is container ok.

`__str__` - dunder method for BaseCollectiner.

is_container_ok ()

Checking is container ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase.

class `gen_py_tool.pro.factory.collectiner.tool.ProCollectiner` (*verbose=False*)
Bases: `gen_py_tool.pro.factory.collectiner.tool.base.BaseCollectiner`

Defined class ProCollectiner with attribute(s) and method(s). Defined project container for pre-processing phase. It defines:

attributes

None.

methods

`__init__` - initial constructor.

`update_tool_name` - preparing tool name for project.

`update_tool_class` - preparing tool class for project.

`update_tool_run` - preparing tool run for project.

`update_tool_conf` - preparing tool conf for project.

`update_tool_log` - preparing tool log for project.

`__str__` - dunder method for ProCollectiner.

`update_tool_class()`

Preparing tool class for project schema.

Exceptions None

`update_tool_conf()`

Preparing tool conf for project schema.

Exceptions None

`update_tool_log()`

Preparing tool log for project schema.

Exceptions None

`update_tool_name()`

Preparing tool name for project schema.

Exceptions None

`update_tool_run()`

Preparing tool run for project schema.

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 package `gen_py_tool.pro.factory.collectiner`.

gen_py_tool.pro.factory.extractiner package

Subpackages

gen_py_tool.pro.factory.extractiner.gen package

Submodules

gen_py_tool.pro.factory.extractiner.gen.base module

Module base.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 BaseExtractor with attribute(s) and method(s). Defined project container for post-processing phase.

class gen_py_tool.pro.factory.extractiner.gen.base.**BaseExtractor** (*verbose=False*)
Bases: *gen_py_tool.pro.schema.schema_container.SchemaContainer, gen_py_tool.pro.element.element_container.ElementContainer*

Defined class BaseExtractor with attribute(s) and method(s). Defined project container for post-processing phase. It defines:

attributes

None.

methods

`__init__` - initial constructor.

`is_container_ok` - checking is container ok.

`__str__` - dunder method for BaseExtractor.

is_container_ok()

Checking is container ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProExtractor with attribute(s) and method(s). Defined project container for post-processing phase.

class `gen_py_tool.pro.factory.extractiner.gen.ProExtractor(verbose=False)`

Bases: `gen_py_tool.pro.factory.extractiner.gen.base.BaseExtractor`

Defined class ProExtractor with attribute(s) and method(s). Defined project container for post-processing phase. It defines:

attributes

None.

methods

`__init__` - initial constructor.

`extract_gen_name` - extract gen name for project.

`extract_gen_class` - extract gen class for project.

`extract_gen_run` - extract gen run for project.

`extract_gen_conf` - extract gen conf for project.

`extract_gen_conf_util` - extract gen conf util for project.

`extract_gen_conf_template` - extract gen template for project.

`extract_gen_log` - extract gen log for project.

`extract_edit_config` - extract gen edit config for project.

`create_package_structure` - create package structure.

`create_package_structure` - create package structure.

`__str__` - dunder method for ProExtractor.

`create_package_structure()`

Creating generator package structure from project schema.

Exceptions None

`extract_edit_config()`

Extraction gen edit config from project schema.

Returns github edit configuration | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

`extract_gen_class()`

Extraction gen class from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

`extract_gen_conf()`

Extraction gen conf from project schema.

Returns ini configuraiton code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_gen_conf_template()
Extraction gen conf template from project schema.

Returns ini configuraiton code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_gen_conf_util()
Extraction gen conf util from project schema.

Returns ini configuraiton code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_gen_log()
Extraction gen log from project schema.

Returns empty log content | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_gen_name()
Extraction of gen name from project element.

Returns gen name | None.

Return type <str> | <NoneType>

Exceptions None

extract_gen_pro()
Extraction gen pro from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_gen_run()
Extraction gen run from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_read_template()
Extraction gen pro from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_write_template()
Extraction gen pro from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

gen_py_tool.pro.factory.extractiner.tool package

Submodules

gen_py_tool.pro.factory.extractiner.tool.base module

Module base.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 BaseExtractor with attribute(s) and method(s). Defined project container for post-processing phase.

class gen_py_tool.pro.factory.extractiner.tool.base.**BaseExtractor** (*verbose=False*)
Bases: *gen_py_tool.pro.schema.schema_container.SchemaContainer, gen_py_tool.pro.element.element_container.ElementContainer*

Defined class BaseExtractor with attribute(s) and method(s). Defined project container for post-processing phase. It defines:

attributes

None.

methods

`__init__` - initial constructor.

`is_container_ok` - checking is container ok.

`__str__` - dunder method for BaseExtractor.

is_container_ok ()

Checking is container ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ProExtractor with attribute(s) and method(s). Defined project container for post-processing phase.

class `gen_py_tool.pro.factory.extractiner.tool.ProExtractor(verbose=False)`
Bases: `gen_py_tool.pro.factory.extractiner.tool.base.BaseExtractor`

Defined class ProExtractor with attribute(s) and method(s). Defined project container for post-processing phase.
It defines:

attributes

None.

methods

`__init__` - initial constructor.
`extract_tool_name` - extract tool name for project.
`extract_tool_class` - extract tool class for project.
`extract_tool_run` - extract tool run for project.
`extract_tool_conf` - extract tool conf for project.
`extract_tool_conf_util` - extract tool conf util for project.
`extract_tool_log` - extract tool log for project.
`extract_edit_config` - extract tool edit config for project.
`create_package_structure` - create package structure.
`__str__` - dunder method for ProExtractor.

create_package_structure()
Creating tool package structure from project schema.

Exceptions None

extract_edit_config()
Extraction tool edit config from project schema.

Returns github edit configuration | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_tool_class()
Extraction tool class from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_tool_conf()
Extraction tool conf from project schema.

Returns ini configuraiton code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_tool_conf_util()
Extraction tool conf util from project schema.

Returns ini configuraiton code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_tool_log()

Extraction tool log from project schema.

Returns empty log content | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

extract_tool_name()

Extraction of tool name from project element.

Returns tool name | None.

Return type <str> | <NoneType>

Exceptions None

extract_tool_run()

Extraction tool run from project schema.

Returns python code | None.

Return type <str>, <str> | <NoneType>, <str>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 package gen_py_tool.pro.factory.extractiner.

gen_py_tool.pro.factory.gen package

Submodules

gen_py_tool.pro.factory.gen.deploy_gen module

Module `gen_extractor.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 DeployGen with attribute(s) and method(s). Created API for extracting elements from schema for generator.

```
class gen_py_tool.pro.factory.gen.deploy_gen.DeployGen(element, schema, verbose=False)
    Bases: gen_py_tool.pro.factory.extracter.gen.ProExtractor, gen_py_tool.pro.factory.gen.gen_elements.GenElements, gen_py_tool.pro.element.element_keys.ElementKeys
```

Defined class DeployGen with attribute(s) and method(s). Created API for extracting elements from schema for generator. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

__init__ - initial constructor.

deploy_gen - deployment of gen from project schema.

__str__ - dunder method for DeployGen.

```
GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY::GEN::DEPLOY_GEN'
```

```
deploy_modules()
```

Deployment of gen from project schema.

Returns list with gen name and modules.

Return type <list>

Exceptions None

gen_py_tool.pro.factory.gen.gen_elements module

Module gen_elements.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 GenElements with attribute(s). Created attributes for preparing project schema for generator.

```
class gen_py_tool.pro.factory.gen.gen_elements.GenElements
```

Defined class GenElements with attribute(s). Created attributes for preparing project schema for generator. It defines:

attributes

NAME - project type for generator.

MOD - project type for generator.

INIT - project type for generator.

RUN - project type for generator.

CLASS - project type for generator.

EDIT - project type for generator.

_EDIT - project type for generator.

CONF - project type for generator.

CFG - project type for generator.

CONFIG - project type for generator.

LOG - project type for generator.

UTIL - project type for generator.

methods

None.

```
CFG = 'cfg'
CLASS = 'generator_process_class'
CONF = 'conf'
CONFIG = 'configuration'
EDIT = 'editorconfig'
INIT = '__init__.py'
IO_CLASS = 'io_class'
LOG = 'log'
MOD = 'modules'
NAME = 'generator_name'
PRO = 'pro'
READ = 'read_template'
RUN = 'run'
TEMPLATE = 'template'
TEST = 'generator_test'
UTIL = 'util'
WRITE = 'write_template'
```

gen_py_tool.pro.factory.gen.prepare_gen module

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 PrepareGen with attribute(s) and method(s). Created API for preparing project schema for generator.

```
class gen_py_tool.pro.factory.gen.prepare_gen.PrepareGen (pro_property, template,
                                                         schema, verbose=False)
Bases: gen_py_tool.pro.factory.collectiner.gen.ProCollectiner
```

Defined class PrepareGen with attribute(s) and method(s). Created API for preparing project schema for generator. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

__init__ - initial constructor.

export - export formatted project schema.

__str__ - dunder method for PrepareGen.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY::GEN::PREPARE_GEN'

export (*verbose=False*)

Export generator project setup.

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

Returns schema content, boolean status (True) | None, False.

Return type <dict>, <bool> | <NoneType>, <bool>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 GenFactory with attribute(s) and method(s). Created API for preparing project schema for gen.

class gen_py_tool.pro.factory.gen.**GenFactory** (*pro_property, schema, template, element, verbose=False*)

Defined class GenFactory with attribute(s) and method(s). Created API for preparing project schema for tool. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

__init__ - initial constructor.

get_preparer - get preparer object for factory.

get_deployer - get deployer object for factory.

__str__ - dunder method for GenFactory.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY::GEN::GEN_FACTORY'

get_deployer ()

Get deployer object from GenFactory for Factory.

Returns object for deploying gen.

Return type <DeployGen>

Exceptions None

get_preparer()

Get preparer object from GenFactory for Factory.

Returns object for preparing gen.

Return type <PrepareGen>

Exceptions None

gen_py_tool.pro.factory.tool package

Submodules

gen_py_tool.pro.factory.tool.deploy_tool module

Module deploy_tool.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 DeployTool with attribute(s) and method(s). Created API for extracting elements from schema for tool.

class gen_py_tool.pro.factory.tool.deploy_tool.**DeployTool** (*element, schema, verbose=False*)
Bases: *gen_py_tool.pro.factory.extractiner.tool.ProExtractor*

Defined class DeployTool with attribute(s) and method(s). Created API for extracting elements from schema for tool. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

__init__ - initial constructor.

deploy_tool - deployment of tool from project schema.

__str__ - dunder method for DeployTool.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY::TOOL::DEPLOY_TOOL'

deploy_modules()

Deployment of tool from project schema.

Returns list with tool name and modules.

Return type <list>

Exceptions None

gen_py_tool.pro.factory.tool.prepare_tool module

Module prepare_tool.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 PrepareTool with attribute(s) and method(s). Created API for preparing project schema for tool.

```
class gen_py_tool.pro.factory.tool.prepare_tool.PrepareTool (pro_property, tem-  
plate, schema,  
verbose=False)
```

Bases: `gen_py_tool.pro.factory.collectiner.tool.ProCollectiner`

Defined class PrepareTool with attribute(s) and method(s). Created API for preparing project schema for tool. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

__init__ - initial constructor.

export - export formatted project schema.

__str__ - dunder method for PrepareTool.

```
GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY::TOOL::PREPARE_TOOL'
```

```
export (verbose=False)
```

Export tool project setup.

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

Returns schema content, boolean status (True) | None, False.

Return type <dict>, <bool> | <NoneType>, <bool>

Exceptions None

gen_py_tool.pro.factory.tool.tool_elements module

Module tool_elements.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ToolElements with attribute(s). Created attributes for preparing project schema for tool.

```
class gen_py_tool.pro.factory.tool.tool_elements.ToolElements
```

Defined class ToolElements with attribute(s). Created attributes for preparing project schema for tool. It defines:

attributes

NAME - project type for tool.

MOD - project type for generator.

INIT - project type for tool.
RUN - project type for generator.
CLASS - project type for tool.
EDIT - project type for generator.
_EDIT - project type for tool.
CONF - project type for generator.
CFG - project type for tool.
CONFIG - project type for generator.
LOG - project type for tool.
UTIL - project type for generator.

methods

None.

```
CFG = 'cfg'
CLASS = 'tool_name_class'
CONF = 'conf'
CONFIG = 'configuration'
EDIT = 'editorconfig'
INIT = '__init__.py'
LOG = 'log'
MOD = 'modules'
NAME = 'tool_name'
RUN = 'run'
UTIL = 'util'
```

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 ToolFactory with attribute(s) and method(s). Created API for preparing project schema for tool.

class `gen_py_tool.pro.factory.tool.ToolFactory` (*pro_property*, *schema*, *template*, *element*, *verbose=False*)

Defined class ToolFactory with attribute(s) and method(s). Created API for preparing project schema for tool. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.

methods

`__init__` - initial constructor.
`get_preparer` - get preparer object for factory.
`get_deployer` - get deployer object for factory.
`__str__` - dunder method for ToolFactory.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY::TOOL::TOOL_FACTORY'

get_deployer ()
Get deployer object from ToolFactory for Factory.

Returns object for deploying tool.

Return type <DeployTool>

Exceptions None

get_preparer ()
Get preparer object from ToolFactory for Factory.

Returns object for preparing tool.

Return type <PrepareTool>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 Factory with attribute(s) and method(s). Created API for project container factory.

class `gen_py_tool.pro.factory.Factory` (*pro_types*, *verbose=False*)
Defined class Factory with attribute(s) and method(s). Created API for project container factory. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__supported_pro_types - list with supported project types.

methods

`__init__` - initial constructor.
`check_pro_type` - checking project type is it supported.
`produce` - preparing project factory object.
`__str__` - dunder method for Factory.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::FACTORY'

check_pro_type (*pro_type*, *verbose=False*)
Checking project type is it supported.

Parameters

- **pro_type** (<str>) – project type.

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

Returns boolean status, True (supported) | False.

Return type <bool>

Exceptions ATSTypeError | ATSBadCallError

produce (*pro_property*, *schema*, *template*, *element*, *verbose=False*)
Preparing project factory object.

Parameters

- **pro_property** (<dict>) – project property (name and type).
- **template** (<dict>) – project tool/generator template.
- **schema** (<dict>) – project tool/generator schema.
- **verbose** (<bool>) – enable/disable verbose option.

Returns factory product object | None.

Return type <Python Object> | <NoneType>

Exceptions ATSTypeError | ATSBadCallError

gen_py_tool.pro.schema package

Submodules

gen_py_tool.pro.schema.schema_container module

Module schema_container.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 SchemaContainer with attribute(s) and method(s). Created API for schema container.

class gen_py_tool.pro.schema.schema_container.SchemaContainer (*verbose=False*)
Defined class SchemaContainer with attribute(s) and method(s). Created API for schema container. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__verbose - enable/disable verbose option.
__schema - schema object.

methods

__init__ - initial constructor.
schema - get/set property schema object.
is_schema_ok - checking is project schema ok.
__str__ - dunder method for SchemaContainer.

```
GEN_VERBOSE = 'GEN_PY_TOOL::PRO::SCHEMA::SCHEMA_CONTAINER'
```

```
is_schema_ok()
```

Checking is project schema ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

schema

Get schema object property.

Returns schema object | None.

Return type <dict> | <NoneType>

Exceptions None

gen_py_tool.pro.schema.schema_keys module

Module schema_keys.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 SchemaKeys with attribute(s). Created attributes for processing tool/gen schema.

class gen_py_tool.pro.schema.schema_keys.SchemaKeys

Defined class SchemaKeys with attribute(s). Created attributes for processing tool/gen schema. It defines:

attributes

TOOL_SCHEMA - schema tool key.

GEN_SCHEMA - schema generator key.

SUPPORTED_SCHEMAS - supportd schemas (checks by key).

methods

None.

```
GEN_SCHEMA = 'schema_generator'
```

```
SUPPORTED_SCHEMAS = ['schema_tool', 'schema_generator']
```

```
TOOL_SCHEMA = 'schema_tool'
```

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 SchemaLoader with attribute(s) and method(s). Created API for schema checking and loading.

```
class gen_py_tool.pro.schema.SchemaLoader (schema_file, verbose=False)
    Bases:          ats_utilities.config_io.base_check.FileChecking,      gen_py_tool.
    pro.schema.schema_container.SchemaContainer,      gen_py_tool.pro.schema.
    schema_keys.SchemaKeys
```

Defined class SchemaLoader with attribute(s) and method(s). Created API for schema checking and loading. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
CONF_DIR - configuration directory.

methods

__init__ - initial constructor.
check_root_key - check root key for schema.
__str__ - dunder method for SchemaLoader.

```
CONF_DIR = '/../../conf/schema/'
```

```
GEN_VERBOSE = 'GEN_PY_TOOL::PRO::SCHEMA::SCHEMA_LOADER'
```

```
VERBOSE = 'ATS_UTILITIES'
```

```
check_root_key (verbose=False)
    Check root key for schema.
```

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

Returns boolean status, True (success) | False.

Return type <bool>

Exceptions None

gen_py_tool.pro.template package

Submodules

gen_py_tool.pro.template.template_container module

Module template_container.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 TemplateContainer with attribute(s) and method(s). Created API for template container.

class `gen_py_tool.pro.template.template_container.TemplateContainer` (*verbose=False*)
Defined class TemplateContainer with attribute(s) and method(s). Created API for template container. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__verbose - enable/disable verbose option.
__template - template object.

methods

__init__ - initial constructor.
template - get/set property template object.
is_template_ok - checking is project template ok.
__str__ - dunder method for TemplateContainer.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::TEMPLATE::TEMPLATE_CONTAINER'

is_template_ok()

Checking is project template ok.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

template

Get template object property.

Returns template object | None.

Return type <dict> | <NoneType>

Exceptions None

gen_py_tool.pro.template.template_keys module

Module `template_keys.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 TemplateKeys with attribute(s). Created attributes for processing tool/gen template.

class `gen_py_tool.pro.template.template_keys.TemplateKeys`

Defined class TemplateKeys with attribute(s). Created attributes for processing tool/gen template. It defines:

attributes

TOOL_TEMPLATE - tool template key.
GEN_TEMPLATE - generator template key.
SUPPORTED_TEMPLATES - list of supported templates.

methods

None.

```
GEN_TEMPLATE = 'generator'
SUPPORTED_TEMPLATES = ['tool', 'generator']
TOOL_TEMPLATE = 'tool'
```

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 TemplateLoader with attribute(s) and method(s). Created API for template checking and loadig.

class `gen_py_tool.pro.template.TemplateLoader` (*template_file*, *verbose=False*)
Bases: `ats_utilities.config_io.base_check.FileChecking`, `gen_py_tool.pro.template.template_container.TemplateContainer`, `gen_py_tool.pro.template.template_keys.TemplateKeys`

Defined class TemplateLoader with attribute(s) and method(s). Created API for template checking and loadig.
It defines:

attributes

`GEN_VERBOSE` - console text indicator for process-phase.
`CONF_DIR` - configuration directory.

methods

`__init__` - initial constructor.
`check_root_key` - check root key for template.
`__str__` - dunder method for TemplateLoader.

`CONF_DIR` = `'/../../conf/template/'`

`GEN_VERBOSE` = `'GEN_PY_TOOL::PRO::TEMPLATE::TEMPLATE_LOADER'`

`VERBOSE` = `'ATS_UTILITIES'`

check_root_key (*verbose=False*)
Check root key for template.

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

Returns boolean status, True (success) | False.

Return type <bool>

Exceptions None

Submodules

gen_py_tool.pro.read_template module

Module read_template.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 project templates.

class gen_py_tool.pro.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 project templates. It defines:

attributes

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

methods

__init__ - initial constructor.
get_template_dir - get template dir path.
read - read a templates and return a content with status.
__str__ - dunder method for ReadTemplate.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::READ_TEMPLATE'

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

VERBOSE = 'ATS_UTILITIES'

get_template_dir()

Getter for template dir path.

Returns template dir path | None.

Return type <str> | <NoneType>

Exceptions None

read (*template_loader, pro_type, verbose=False*)

Read a templates and return a content with status.

Parameters

- **template_loader** (<dict>) – dict with template modules.
- **pro_type** (<str>) – project type (tool | generator).
- **verbose** (<bool>) – enable/disable verbose option.

Returns dict with templates and True | empty dict and False.

Return type <dict> <bool>

Exceptions ATSTypeError | ATSBadCallError

gen_py_tool.pro.write_template module

Module write_template.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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_tool.pro.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.

methods

__init__ - initial constructor.

write - write templates content with parameters to modules.

__str__ - dunder method for WriteTemplate.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::WRITE_TEMPLATE'

VERBOSE = 'ATS_UTILITIES'

write (*element, modules, verbose=False*)

Write templates content with parameters to modules.

Parameters

- **element** (<dict>) – processes element.
- **modules** (<list>) – modules for tool/generator.
- **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_tool 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_tool 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 GenPro with attribute(s) and method(s). Generate project by setup of parameters, templates, schemas.

```
class gen_py_tool.pro.GenPro (project_name, verbose=False)
    Bases:      ats_utilities.config_io.base_check.FileChecking,  gen_py_tool.pro.
    config.ProConfig, gen_py_tool.pro.config.pro_name.ProName, gen_py_tool.pro.
    config.pro_type.ProType
```

Defined class GenPro with attribute(s) and method(s). Generate project by setup of parameters, templates, schemas. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
PRO_STRUCTURE - project structure.

methods

__init__ - initial constructor.
is_pro_config_ok - checking is project configuration ok.
gen_pro - generate project structure.
__str__ - dunder method for GenPro.

GEN_VERBOSE = 'GEN_PY_TOOL::PRO::GEN_PRO'

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

VERBOSE = 'ATS_UTILITIES'

gen_pro (verbose=False)
Generate project structure.

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

Returns boolean status, True (success) | False.

Return type <bool>

Exceptions None

is_pro_config_ok (verbose=False)
Checking is project configuration ok.

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

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

1.1.2 Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> gen_py_tool 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_tool 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 GenPyTool with attribute(s) and method(s). Load a base info, create an CLI interface and run operation(s).

```
class gen_py_tool.GenPyTool (verbose=False)  
    Bases: ats_utilities.cli.cfg_cli.CfgCLI
```

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

attributes

- `__metaclass__` - setting cooperative metaclasses.
- `GEN_VERBOSE` - console text indicator for process-phase.
- `CONFIG` - tool info file path.
- `LOG` - tool log file path.
- `OPS` - list of tool options.
- `logger` - logger object API.

methods

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

```
CONFIG = '/conf/gen_py_tool.cfg'  
GEN_VERBOSE = 'GEN_PY_TOOL'  
LOG = '/log/gen_py_tool.log'  
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_tool** type the following:

```
tar xvzf gen_py_tool-x.y.z.tar.gz
cd gen_py_tool-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-tool
# python3
pip3 install gen-py-tool
```


CHAPTER 3

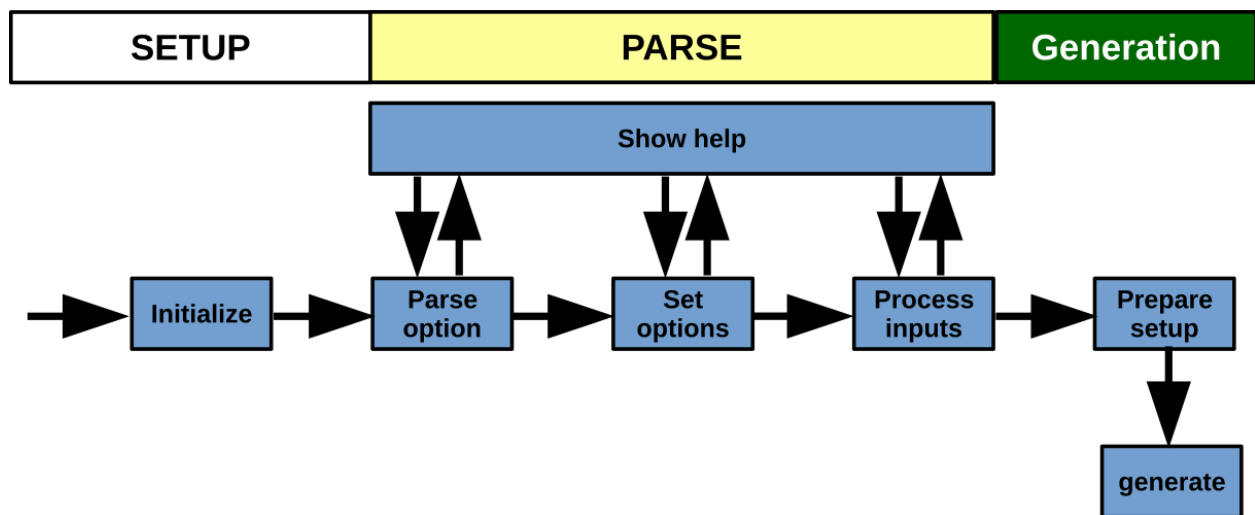
Dependencies

gen_py_tool requires next modules and libraries:

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

Generation flow of py tool

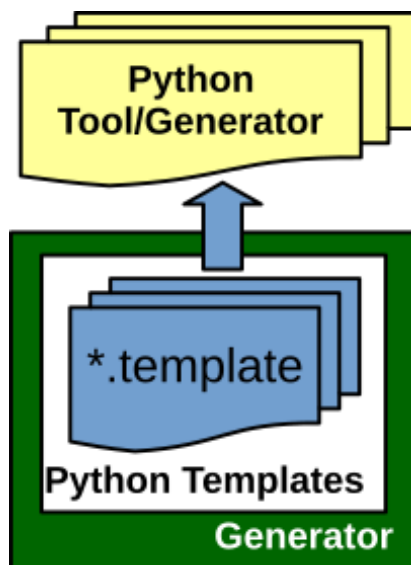
Base flow of generation process:



CHAPTER 5

Tool structure

`gen_py_tool` is based on OOP:



Code structure:

```
gen_py_tool/
├── conf/
│   ├── element/
│   │   ├── substitute_generator.yaml
│   │   └── substitute_tool.yaml
│   ├── gen_py_tool.cfg
│   ├── gen_py_tool_util.cfg
│   ├── project.yaml
│   └── schema/
│       ├── schema_generator.yaml
│       └── schema_tool.yaml
```

(continues on next page)

(continued from previous page)

```

└─ template/
    └─ generator/
        └─ editorconfig.template
        └─ generator_configuration.template
        └─ generator_configuration_util.template
        └─ generator_io_class.template
        └─ generator_process_class.template
        └─ generator_read_template.template
        └─ generator_test.template
        └─ generator_write_template.template
        └─ run_generator.template
    └─ template_generator.yaml
    └─ template_tool.yaml
    └─ tool/
        └─ editorconfig.template
        └─ run_tool.template
        └─ tool_configuration.template
        └─ tool_configuration_util.template
        └─ tool_name_class.template
__init__.py
log/
└─ gen_py_tool.log
pro/
└─ config/
    └─ __init__.py
    └─ pro_name.py
    └─ pro_selector.py
    └─ pro_type.py
└─ element/
    └─ element_container.py
    └─ element_keys.py
    └─ __init__.py
└─ factory/
    └─ collectiner/
        └─ gen/
            └─ base.py
            └─ __init__.py
        └─ __init__.py
        └─ tool/
            └─ base.py
            └─ __init__.py
    └─ extractiner/
        └─ gen/
            └─ base.py
            └─ __init__.py
        └─ __init__.py
        └─ tool/
            └─ base.py
            └─ __init__.py
    └─ gen/
        └─ deploy_gen.py
        └─ gen_elements.py
        └─ __init__.py
        └─ prepare_gen.py
    └─ __init__.py
    └─ tool/
        └─ deploy_tool.py

```

(continues on next page)

(continued from previous page)

```
├── ┌── __init__.py
│   ├── prepare_tool.py
│   └── tool_elements.py
├── __init__.py
├── read_template.py
├── schema/
│   ├── __init__.py
│   ├── schema_container.py
│   └── schema_keys.py
├── template/
│   ├── __init__.py
│   ├── template_container.py
│   └── template_keys.py
├── write_template.py
└── run/
    └── gen_py_tool_run.py
```


CHAPTER 6

Copyright and licence

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

gen_py_tool 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_tool, 34
gen_py_tool.pro, 33
gen_py_tool.pro.config, 6
gen_py_tool.pro.config.pro_name, 3
gen_py_tool.pro.config.pro_selector, 4
gen_py_tool.pro.config.pro_type, 5
gen_py_tool.pro.element, 8
gen_py_tool.pro.element.element_container, 7
gen_py_tool.pro.element.element_keys, 8
gen_py_tool.pro.factory, 26
gen_py_tool.pro.factory.collectiner, 13
gen_py_tool.pro.factory.collectiner.gen, 10
gen_py_tool.pro.factory.collectiner.gen.base, 10
gen_py_tool.pro.factory.collectiner.tool, 12
gen_py_tool.pro.factory.collectiner.tool.base, 12
gen_py_tool.pro.factory.extractiner, 19
gen_py_tool.pro.factory.extractiner.gen, 14
gen_py_tool.pro.factory.extractiner.gen.base, 14
gen_py_tool.pro.factory.extractiner.tool, 17
gen_py_tool.pro.factory.extractiner.tool.base, 17
gen_py_tool.pro.factory.gen, 22
gen_py_tool.pro.factory.gen.deploy_gen, 19
gen_py_tool.pro.factory.gen.gen_elements, 20
gen_py_tool.pro.factory.gen.prepare_gen, 21
gen_py_tool.pro.factory.tool, 25
gen_py_tool.pro.factory.tool.deploy_tool, 23
gen_py_tool.pro.factory.tool.prepare_tool, 23
gen_py_tool.pro.factory.tool.tool_elements, 24
gen_py_tool.pro.read_template, 32
gen_py_tool.pro.schema, 28
gen_py_tool.pro.schema.schema_container, 27
gen_py_tool.pro.schema.schema_keys, 28
gen_py_tool.pro.template, 31
gen_py_tool.pro.template.template_container, 29
gen_py_tool.pro.template.template_keys, 30
gen_py_tool.pro.write_template, 33

B

BaseCollectiner (class in *gen_py_tool.pro.factory.collectiner.gen.base*), 10

BaseCollectiner (class in *gen_py_tool.pro.factory.collectiner.tool.base*), 12

BaseExtractor (class in *gen_py_tool.pro.factory.extractiner.gen.base*), 14

BaseExtractor (class in *gen_py_tool.pro.factory.extractiner.tool.base*), 17

C

CFG (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21

CFG (*gen_py_tool.pro.factory.tool.tool_elements.ToolElements* attribute), 25

check_config_keys() (*gen_py_tool.pro.config.pro_selector.ProjectSelector* class method), 5

check_pro_type() (*gen_py_tool.pro.factory.Factory* method), 26

check_root_key() (*gen_py_tool.pro.element.ElementLoader* method), 9

check_root_key() (*gen_py_tool.pro.schema.SchemaLoader* method), 29

check_root_key() (*gen_py_tool.pro.template.TemplateLoader* method), 31

CLASS (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21

CLASS (*gen_py_tool.pro.factory.tool.tool_elements.ToolElements* attribute), 25

CONF (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21

CONF (*gen_py_tool.pro.factory.tool.tool_elements.ToolElements* attribute), 25

CONF_DIR (*gen_py_tool.pro.element.ElementLoader* attribute), 9

CONF_DIR (*gen_py_tool.pro.schema.SchemaLoader* attribute), 29

CONF_DIR (*gen_py_tool.pro.template.TemplateLoader* attribute), 31

CONFIG (*gen_py_tool.GenPyTool* attribute), 35

config (*gen_py_tool.pro.config.ProConfig* attribute), 7

CONFIG (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21

CONFIG (*gen_py_tool.pro.factory.tool.tool_elements.ToolElements* attribute), 25

create_package_structure() (*gen_py_tool.pro.factory.extractiner.gen.ProExtractor* method), 15

create_package_structure() (*gen_py_tool.pro.factory.extractiner.tool.ProExtractor* method), 18

D

deploy_modules() (*gen_py_tool.pro.factory.gen.deploy_gen.DeployGen* method), 20

deploy_modules() (*gen_py_tool.pro.factory.tool.deploy_tool.DeployTool* method), 23

DeployGen (class in *gen_py_tool.pro.factory.gen.deploy_gen*), 19

DeployTool (class in *gen_py_tool.pro.factory.tool.deploy_tool*), 23

E

EDIT (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21

EDIT (*gen_py_tool.pro.factory.tool.tool_elements.ToolElements* attribute), 25

element (*gen_py_tool.pro.element.element_container.ElementContainer* attribute), 7

ELEMENT_KEY (*gen_py_tool.pro.config.pro_selector.ProjectSelector* attribute), 5

ElementContainer (class in `gen_py_tool.pro.element.element_container`), 7
 ElementKeys (class in `gen_py_tool.pro.element.element_keys`), 8
 ElementLoader (class in `gen_py_tool.pro.element`), 9
 export() (`gen_py_tool.pro.factory.gen.prepare_gen.PrepareGen` method), 19
 export() (`gen_py_tool.pro.factory.tool.prepare_tool.PrepareTool` method), 24
 extract_edit_config() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 15
 extract_edit_config() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 18
 extract_gen_class() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 15
 extract_gen_conf() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 15
 extract_gen_conf_template() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 15
 extract_gen_conf_util() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 extract_gen_log() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 extract_gen_name() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 extract_gen_pro() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 extract_gen_run() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 extract_read_template() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 extract_tool_class() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 18
 extract_tool_conf() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 18
 extract_tool_conf_util() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 18
 extract_tool_log() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 19
 extract_tool_name() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 19
 extract_tool_run() (`gen_py_tool.pro.factory.extractiner.tool.ProExtractor` method), 19
 extract_write_template() (`gen_py_tool.pro.factory.extractiner.gen.ProExtractor` method), 16
 Factory (class in `gen_py_tool.pro.factory`), 26
 GenPro (`gen_py_tool.pro.GenPro` method), 34
 gen_py_tool (module), 34
 gen_py_tool.pro (module), 33
 gen_py_tool.pro.config (module), 6
 gen_py_tool.pro.config.pro_name (module), 3
 gen_py_tool.pro.config.pro_selector (module), 4
 gen_py_tool.pro.config.pro_type (module), 5
 gen_py_tool.pro.element (module), 8
 gen_py_tool.pro.element.element_container (module), 7
 gen_py_tool.pro.element.element_keys (module), 8
 gen_py_tool.pro.factory (module), 26
 gen_py_tool.pro.factory.collectiner (module), 13
 gen_py_tool.pro.factory.collectiner.gen (module), 10
 gen_py_tool.pro.factory.collectiner.gen.base (module), 10
 gen_py_tool.pro.factory.collectiner.tool (module), 12
 gen_py_tool.pro.factory.collectiner.tool.base (module), 12
 gen_py_tool.pro.factory.extractiner (module), 19
 gen_py_tool.pro.factory.extractiner.gen (module), 14
 gen_py_tool.pro.factory.extractiner.gen.base (module), 14
 gen_py_tool.pro.factory.extractiner.tool (module), 17
 gen_py_tool.pro.factory.extractiner.tool.base (module), 17
 gen_py_tool.pro.factory.gen (module), 22
 gen_py_tool.pro.factory.gen.deploy_gen (module), 19

`gen_py_tool.pro.factory.gen.gen_elements` (module), 20
`gen_py_tool.pro.factory.gen.prepare_gen` (module), 21
`gen_py_tool.pro.factory.tool` (module), 25
`gen_py_tool.pro.factory.tool.deploy_tool` (module), 23
`gen_py_tool.pro.factory.tool.prepare_tool` (module), 23
`gen_py_tool.pro.factory.tool.tool_elements` (module), 24
`gen_py_tool.pro.read_template` (module), 32
`gen_py_tool.pro.schema` (module), 28
`gen_py_tool.pro.schema.schema_container` (module), 27
`gen_py_tool.pro.schema.schema_keys` (module), 28
`gen_py_tool.pro.template` (module), 31
`gen_py_tool.pro.template.template_container` (module), 29
`gen_py_tool.pro.template.template_keys` (module), 30
`gen_py_tool.pro.write_template` (module), 33
`GEN_SCHEMA` (`gen_py_tool.pro.schema.schema_keys.SchemaKeys` attribute), 28
`GEN_TEMPLATE` (`gen_py_tool.pro.template.template_keys.TemplateKeys` attribute), 30
`GEN_VERBOSE` (`gen_py_tool.GenPyTool` attribute), 35
`GEN_VERBOSE` (`gen_py_tool.pro.config.pro_name.ProName` attribute), 4
`GEN_VERBOSE` (`gen_py_tool.pro.config.pro_selector.ProjectSelector` attribute), 5
`GEN_VERBOSE` (`gen_py_tool.pro.config.pro_type.ProType` attribute), 6
`GEN_VERBOSE` (`gen_py_tool.pro.config.ProConfig` attribute), 6
`GEN_VERBOSE` (`gen_py_tool.pro.element.element_container.ElementContainer` attribute), 7
`GEN_VERBOSE` (`gen_py_tool.pro.element.ElementLoader` attribute), 9
`GEN_VERBOSE` (`gen_py_tool.pro.factory.Factory` attribute), 26
`GEN_VERBOSE` (`gen_py_tool.pro.factory.gen.deploy_gen.DeployGen` attribute), 20
`GEN_VERBOSE` (`gen_py_tool.pro.factory.gen.GenFactory` attribute), 22
`GEN_VERBOSE` (`gen_py_tool.pro.factory.gen.prepare_gen.PrepareGen` attribute), 22
`GEN_VERBOSE` (`gen_py_tool.pro.factory.tool.deploy_tool.DeployTool` attribute), 23
`GEN_VERBOSE` (`gen_py_tool.pro.factory.tool.prepare_tool.PrepareTool` attribute), 24
`GEN_VERBOSE` (`gen_py_tool.pro.factory.tool.ToolFactory` attribute), 26
`GEN_VERBOSE` (`gen_py_tool.pro.gen_pro.GenPro` attribute), 34
`GEN_VERBOSE` (`gen_py_tool.pro.read_template.ReadTemplate` attribute), 32
`GEN_VERBOSE` (`gen_py_tool.pro.schema.schema_container.SchemaContainer` attribute), 27
`GEN_VERBOSE` (`gen_py_tool.pro.schema.SchemaLoader` attribute), 29
`GEN_VERBOSE` (`gen_py_tool.pro.template.template_container.TemplateContainer` attribute), 30
`GEN_VERBOSE` (`gen_py_tool.pro.template.TemplateLoader` attribute), 31
`GEN_VERBOSE` (`gen_py_tool.pro.write_template.WriteTemplate` attribute), 33
`GenElements` (class in `gen_py_tool.pro.factory.gen.gen_elements`), 20
`GenFactory` (class in `gen_py_tool.pro.factory.gen`), 22
`GenPro` (class in `gen_py_tool.pro`), 34
`GenPyTool` (class in `gen_py_tool`), 35
`get_deployer()` (`gen_py_tool.pro.factory.gen.GenFactory` method), 22
`get_deployer()` (`gen_py_tool.pro.factory.tool.ToolFactory` method), 26
`get_preparer()` (`gen_py_tool.pro.factory.gen.GenFactory` method), 22
`get_preparer()` (`gen_py_tool.pro.factory.tool.ToolFactory` method), 26
`get_template_dir()` (`gen_py_tool.pro.read_template.ReadTemplate` method), 32
`INIT` (`gen_py_tool.pro.factory.gen.gen_elements.GenElements` attribute), 21
`INIT` (`gen_py_tool.pro.factory.tool.tool_elements.ToolElements` attribute), 25
`IO_CLASS` (`gen_py_tool.pro.factory.gen.gen_elements.GenElements` attribute), 21
`is_config_ok()` (`gen_py_tool.pro.config.ProConfig` method), 7
`is_container_ok()` (`gen_py_tool.pro.factory.collectiner.gen.base.BaseCollectiner` method), 10
`is_container_ok()` (`gen_py_tool.pro.factory.collectiner.tool.base.BaseCollectiner` method), 12
`is_container_ok()` (`gen_py_tool.pro.factory.extractiner.gen.base.BaseExtractor` method), 14
`is_container_ok()` (`gen_py_tool.pro.factory.extractiner.tool.base.BaseExtractor` method), 17
`is_element_ok()` (`gen_py_tool.pro.element.element_container.ElementContainer` method), 8

is_pro_config_ok() (gen_py_tool.pro.GenPro ProCollectiner (class in
 method), 34 gen_py_tool.pro.factory.collectiner.gen),
 is_pro_name_ok() (gen_py_tool.pro.config.pro_name.ProName 11
 method), 4 ProCollectiner (class in
 is_pro_type_ok() (gen_py_tool.pro.config.pro_type.ProType gen_py_tool.pro.factory.collectiner.tool),
 method), 6 12
 is_schema_ok() (gen_py_tool.pro.schema.schema_container.SchemaContainer (class in gen_py_tool.pro.config), 6
 method), 28 produce() (gen_py_tool.pro.factory.Factory method),
 is_template_ok() (gen_py_tool.pro.template.template_container.TemplateContainer 2
 method), 30 ProExtractor (class in
 gen_py_tool.pro.factory.extractiner.gen),
 15
L
 LOG (gen_py_tool.GenPyTool attribute), 35 ProExtractor (class in
 LOG (gen_py_tool.pro.factory.gen.gen_elements.GenElements gen_py_tool.pro.factory.extractiner.tool),
 attribute), 21 18
 LOG (gen_py_tool.pro.factory.tool.tool_elements.ToolElements ProjectSelector (class in
 attribute), 25 gen_py_tool.pro.config.pro_selector), 4
 ProName (class in gen_py_tool.pro.config.pro_name), 3
 ProType (class in gen_py_tool.pro.config.pro_type), 5
M
 MOD (gen_py_tool.pro.factory.gen.gen_elements.GenElements
 attribute), 21 **R**
 MOD (gen_py_tool.pro.factory.tool.tool_elements.ToolElements READ (gen_py_tool.pro.factory.gen.gen_elements.GenElements
 attribute), 25 attribute), 21
 read() (gen_py_tool.pro.read_template.ReadTemplate
 method), 32
N
 NAME (gen_py_tool.pro.factory.gen.gen_elements.GenElements ReadTemplate (class in
 attribute), 21 gen_py_tool.pro.read_template), 32
 NAME (gen_py_tool.pro.factory.tool.tool_elements.ToolElements BOOT_KEY (gen_py_tool.pro.element.element_keys.ElementKeys
 attribute), 25 attribute), 8
 RUN (gen_py_tool.pro.factory.gen.gen_elements.GenElements
 attribute), 21
 RUN (gen_py_tool.pro.factory.tool.tool_elements.ToolElements
 attribute), 25
O
 OPS (gen_py_tool.GenPyTool attribute), 35
P
 PrepareGen (class in
 gen_py_tool.pro.factory.gen.prepare_gen),
 21
 PrepareTool (class in
 gen_py_tool.pro.factory.tool.prepare_tool),
 24
 PRO (gen_py_tool.pro.factory.gen.gen_elements.GenElements
 attribute), 21
 pro_name (gen_py_tool.pro.config.pro_name.ProName
 attribute), 4
 PRO_STRUCTURE (gen_py_tool.pro.GenPro attribute),
 34
 pro_type (gen_py_tool.pro.config.pro_type.ProType
 attribute), 6
 process() (gen_py_tool.GenPyTool method), 35
 process_element() (gen_py_tool.pro.element.ElementLoader
 method), 9
S
 schema (gen_py_tool.pro.schema.schema_container.SchemaContainer
 attribute), 28
 SCHEMA_KEY (gen_py_tool.pro.config.pro_selector.ProjectSelector
 attribute), 5
 SchemaContainer (class in
 gen_py_tool.pro.schema.schema_container),
 27
 SchemaKeys (class in
 gen_py_tool.pro.schema.schema_keys), 28
 SchemaLoader (class in gen_py_tool.pro.schema), 29
 select_pro_type() (gen_py_tool.pro.config.pro_selector.ProjectSelector
 class method), 5
 SUPPORTED_SCHEMAS
 (gen_py_tool.pro.schema.schema_keys.SchemaKeys
 attribute), 28
 SUPPORTED_TEMPLATES
 (gen_py_tool.pro.template.template_keys.TemplateKeys
 attribute), 31

T

TEMPLATE (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21
 template (*gen_py_tool.pro.template.template_container.TemplateContainer* attribute), 30
 TEMPLATE_DIR (*gen_py_tool.pro.read_template.ReadTemplate* attribute), 32
 TEMPLATE_KEY (*gen_py_tool.pro.config.pro_selector.ProjectSelector* attribute), 5
 TemplateContainer (class in *gen_py_tool.pro.template.template_container*), 29
 TemplateKeys (class in *gen_py_tool.pro.template.template_keys*), 30
 TemplateLoader (class in *gen_py_tool.pro.template*), 31
 TEST (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21
 TOOL_CLASS_KEY (*gen_py_tool.pro.element.element_keys.ElementKeys* attribute), 8
 TOOL_NAME_KEY (*gen_py_tool.pro.element.element_keys.ElementKeys* attribute), 8
 TOOL_SCHEMA (*gen_py_tool.pro.schema.schema_keys.SchemaKeys* attribute), 28
 TOOL_TEMPLATE (*gen_py_tool.pro.template.template_keys.TemplateKeys* attribute), 31
 TOOL_UPPER_KEY (*gen_py_tool.pro.element.element_keys.ElementKeys* attribute), 8
 TOOL_YEAR_KEY (*gen_py_tool.pro.element.element_keys.ElementKeys* attribute), 8
 ToolElements (class in *gen_py_tool.pro.factory.tool.tool_elements*), 24
 ToolFactory (class in *gen_py_tool.pro.factory.tool*), 25
 TYPE_KEY (*gen_py_tool.pro.config.pro_selector.ProjectSelector* attribute), 5

U

update_gen_class () (*gen_py_tool.pro.factory.collectiner.gen.ProCollector* method), 11
 update_gen_conf () (*gen_py_tool.pro.factory.collectiner.gen.ProCollector* method), 11
 update_gen_log () (*gen_py_tool.pro.factory.collectiner.gen.ProCollector* method), 11
 update_gen_name () (*gen_py_tool.pro.factory.collectiner.gen.ProCollector* method), 11
 update_gen_pro_class () (*gen_py_tool.pro.factory.collectiner.gen.ProCollector* method), 11

update_gen_run () (*gen_py_tool.pro.factory.collectiner.gen.ProCollector* method), 11
 update_tool_class () (*gen_py_tool.pro.factory.collectiner.tool.ProCollector* method), 13
 update_tool_conf () (*gen_py_tool.pro.factory.collectiner.tool.ProCollector* method), 13
 update_tool_log () (*gen_py_tool.pro.factory.collectiner.tool.ProCollector* method), 13
 update_tool_name () (*gen_py_tool.pro.factory.collectiner.tool.ProCollector* method), 13
 update_tool_run () (*gen_py_tool.pro.factory.collectiner.tool.ProCollector* method), 13
 UTIL (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21
 UTIL (*gen_py_tool.pro.factory.tool.tool_elements.ToolElements* attribute), 25
 VERBOSE (*gen_py_tool.GenPyTool* attribute), 35
 VERBOSE (*gen_py_tool.pro.element.ElementLoader* attribute), 9
 VERBOSE (*gen_py_tool.pro.GenPro* attribute), 34
 VERBOSE (*gen_py_tool.pro.read_template.ReadTemplate* attribute), 32
 VERBOSE (*gen_py_tool.pro.schema.SchemaLoader* attribute), 29
 VERBOSE (*gen_py_tool.pro.template.TemplateLoader* attribute), 31
 VERBOSE (*gen_py_tool.pro.write_template.WriteTemplate* attribute), 33

W

WRITE (*gen_py_tool.pro.factory.gen.gen_elements.GenElements* attribute), 21
 write () (*gen_py_tool.pro.write_template.WriteTemplate* method), 33
 WriteTemplate (class in *gen_py_tool.pro.write_template*), 33