
docstr_coverage Documentation

Release 2.3.0

Hunter McGushion

Apr 13, 2023

Contents

1	Docstr-Coverage API Essentials	1
1.1	get_docstring_coverage	1
1.2	Indices and tables	2
2	Complete Docstr-Coverage API	3
3	Indices and tables	5

Docstr-Coverage API Essentials

This section exposes the API for all the docstr-coverage functionality that will be necessary for most users.

1.1 get_docstring_coverage

```
docstr_coverage.coverage.get_docstring_coverage (filenames: list, skip_magic: bool
                                                = False, skip_file_docstring: bool
                                                = False, skip_init: bool = False,
                                                skip_class_def: bool = False,
                                                skip_private: bool = False, verbose:
                                                int = 0, ignore_names: Tuple[List[str],
                                                ...] = ()) → Tuple[Dict, Dict]
```

Checks contents of *filenames* for missing docstrings, and produces a report detailing docstring status.

Note: For a method with a more expressive return type, you may want to try the experimental *docstr_coverage.analyze* function.

Parameters

- **filenames** (*List*) – List of filename strings that are absolute or relative paths
- **skip_magic** (*Boolean, default=False*) – If True, skips all magic methods (double-underscore-prefixed), except ‘__init__’ and does not include them in the report
- **skip_file_docstring** (*Boolean, default=False*) – If True, skips check for a module-level docstring
- **skip_init** (*Boolean, default=False*) – If True, skips methods named ‘__init__’ and does not include them in the report
- **skip_class_def** (*Boolean, default=False*) – If True, skips class definitions and does not include them in the report. If this is True, the class’s methods will still be checked
- **skip_private** (*Boolean, default=False*) – If True, skips function definitions beginning with a single underscore and does not include them in the report.

- **verbose** (*Int in [0, 1, 2, 3], default=0*) –
 - 0) No printing.
 - 1) Print total stats only.
 - 2) Print stats for all files.
 - 3) Print missing docstrings for all files.
- **ignore_names** (*Tuple[List[str], ...], default=()*) – Patterns to ignore when checking documentation. Each list in *ignore_names* defines a different pattern to be ignored. The first element in each list is the regular expression for matching filenames. All remaining arguments in each list are regexes for matching names of functions/classes. A node is ignored if it matches the filename regex and at least one of the remaining regexes

Returns

- *Dict* – Links filename keys to a dict of stats for that filename. Example:

```
>>> {  
...     '<filename>': {  
...         'missing': ['<method_or_class_name>', '...'],  
...         'module_doc': '<Boolean>',  
...         'missing_count': '<missing_count int>',  
...         'needed_count': '<needed_docstrings_count int>',  
...         'coverage': '<percent_of_coverage float>',  
...         'empty': '<Boolean>'  
...     }, ...  
... }
```

- *Dict* – Total summary stats for all files analyzed. Example:

```
>>> {  
...     'missing_count': '<total_missing_count int>',  
...     'needed_count': '<total_needed_docstrings_count int>',  
...     'coverage': '<total_percent_of_coverage float>'  
... }
```

1.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Complete Docstr-Coverage API

This section exposes the complete docstr-coverage API.

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

CHAPTER 3

Indices and tables

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