Cartorio

Release 2.0.5

Humberto STEIN SHIROMOTO

Oct 22, 2021

CONTENTS:

CHAPTER

ONE

API

cartorio.log.**config_logger**(*log_config_file: pathlib.Path = PosixPath('/home/docs/checkouts/readthedocs.org/user_builds/cartorio/checkouts/v2.0.5/cartorio/conf/log*
→ logging.getLogger

Configure logger object

Parameters **log_config_file** (*Path, optional*) – Path where the log config file is. Defaults to PROJECT_ROOT / "cartorio" / "conf" / "logging.conf".

Returns Logger object

Return type (logging.getLogger)

Example

```
>>> _ = config_logger()
```

cartorio.log.**fun**(*func*)

Log a callable

Parameters **func** (*callable*) – Callable to be logged

Returns Callable outputs

Return type Callable

References

[1] https://dev.to/aldo/implementing-logging-in-python-via-decorators-1gje [2] https://stackoverflow.com/questions/6810999/how-to-determine-file-function-and-line-number

cartorio.log.**log**(*filename: str*, *logs_path: pathlib.Path*, *log_config_file=PosixPath('/home/docs/checkouts/readthedocs.org/user_builds/cartorio/checkouts/v2.0.5/cartorio*

Instantiate logger object

Parameters

- **filename** (*str*) – Log file
- **path** (*Path*) – Path where the log file is saved
- **test** (*bool*) – Return filename
- **log_config_file** (*Path, optional*) – Path contaning the log config file. Defaults to PROJECT_ROOT / "conf" / "logging.conf"

Returns Logging object

Return type (logging.getLogger())

Example

```
>>> logs_path = Path(__file__).resolve().parent
>>> logger = log("test.log", logs_path)
```

References

[1] https://realpython.com/python-logging/

cartorio.log.**make_path**(*path: pathlib.Path = PosixPath('/home/docs/checkouts/readthedocs.org/user_builds/cartorio/checkouts/v2.0.5/cartorio/logs')*) → pathlib.Path

Create logs directory if it doesn't exist

Parameters

- **path** (*Path, optional*) – Path where the log file is saved. Defaults to PROJECT_ROOT/logs/.
- **test** (*bool*) – Return filename

Returns Path to logs directory

Return type (Path)

Example

```
>>> _ = make_path()
```

cartorio.log.**set_handler**(*filename: str*, *log_format: str*, *logs_path: pathlib.Path*) → logging.FileHandler

Set file handler for logger object

Parameters

- **filename** (*str*) – File to be logged
- **log_format** (*str*) – Log format
- **logs_path** (*Path*) – Path where the log is saved.

Raises **IOError** – If folder doesn't exist

Returns File handler object

Return type (logging.FileHandler)

Example

```
>>> format_filename = f"{Path(__file__).stem}.log"
>>> log_format = logging.Formatter('%(asctime)-16s || %(name)s || %(process)d ||
↪%(levelname)s || %(message)s')
>>> logs_path = Path(__file__).resolve().parent
>>> _ = set_handler(__file__, log_format, logs_path=logs_path)
```

CHAPTER

TWO

INTALLATION

```
pip install cartorio
```

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

INDEX