telescopy Documentation

Release 0.0.dev9

Brett M. Morris

Jan 20, 2019

Contents

I telescopy Documentation **3**

1 Reference/API **7**

Python Module Index **13**

This is the documentation for telescopy.

Part I

telescopy Documentation

This is the documentation for telescopy.

CHAPTER 1

Reference/API

1.1 telescopy Package

1.1.1 Functions

`test`(**kwargs)	Run the tests for the package.

test

`telescopy.`**`test`**(***kwargs*)

Run the tests for the package.

This method builds arguments for and then calls `pytest.main`.

Parameters

package
[str, optional] The name of a specific package to test, e.g. 'io.fits' or 'utils'. Accepts comma separated string to specify multiple packages. If nothing is specified all default tests are run.

args
[str, optional] Additional arguments to be passed to `pytest.main` in the `args` keyword argument.

docs_path
[str, optional] The path to the documentation .rst files.

open_files
[bool, optional] Fail when any tests leave files open. Off by default, because this adds extra run time to the test suite. Requires the `psutil` package.

parallel
[int or 'auto', optional] When provided, run the tests in parallel on the specified number

of CPUs. If parallel is `'auto'`, it will use the all the cores on the machine. Requires the `pytest-xdist` plugin.

pastebin
[('failed', 'all', None), optional] Convenience option for turning on py.test pastebin output. Set to 'failed' to upload info for failed tests, or 'all' to upload info for all tests.

pdb
[bool, optional] Turn on PDB post-mortem analysis for failing tests. Same as specifying `--pdb` in `args`.

pep8
[bool, optional] Turn on PEP8 checking via the pytest-pep8 plugin and disable normal tests. Same as specifying `--pep8 -k pep8` in `args`.

plugins
[list, optional] Plugins to be passed to `pytest.main` in the `plugins` keyword argument.

remote_data
[{'none', 'astropy', 'any'}, optional] Controls whether to run tests marked with @pytest.mark.remote_data. This can be set to run no tests with remote data (`none`), only ones that use data from http://data.astropy.org (`astropy`), or all tests that use remote data (`any`). The default is `none`.

repeat
[`int`, optional] If set, specifies how many times each test should be run. This is useful for diagnosing sporadic failures.

skip_docs
[`bool`, optional] When `True`, skips running the doctests in the .rst files.

test_path
[str, optional] Specify location to test by path. May be a single file or directory. Must be specified absolutely or relative to the calling directory.

verbose
[bool, optional] Convenience option to turn on verbose output from py.test. Passing True is the same as specifying `-v` in `args`.

1.1.2 Classes

`Filter`([wavelength, transmissivity, path, name])	Transmissivity curve for a filter.
`Imager`([plate_scale, seeing, binning, . . .])	Container for an imager.
`Target`([magnitude, filter])	Container for target metadata.
`Telescope`([aperture_diameter, throughput])	Container for information about a telescope.
`UnsupportedPythonError`	

Filter

class telescopy.**Filter**(*wavelength=None*, *transmissivity=None*, *path=None*, *name=None*)

Bases: `object`

Transmissivity curve for a filter.

Parameters

wavelength

[Quantity] Wavelength array for the transmissivity curve.

transmissivity
[ndarray] Transmissivity of the filter

path
[str] Path to text transmissivity file

name
[str] Name of the filter

Methods Summary

`available_filters()`	Available filters stored in telescopy.
`from_name(name)`	Transmissivity of a built-in filter.
`plot([ax])`	Plot the transmissivity curve of a filter

Methods Documentation

static available_filters()
Available filters stored in telescopy.

Returns

filters
[list] List of available filter names

classmethod from_name(*name*)
Transmissivity of a built-in filter.

Parameters

name
[str] Must be one of the filters returned by `available_filters()`.

plot(*ax=None*)
Plot the transmissivity curve of a filter

Parameters

ax
[Axes or None] Axis object.

Returns

ax
[Axes] Plot with transmissivity curve

Imager

class telescopy.Imager(*plate_scale=None*, *seeing=None*, *binning=None*, *quantum_efficiency=None*, *gain=None*)

Bases: `object`

Container for an imager.

Parameters

plate_scale
[Quantity] Plate scale of detector in units of arcsec (per pixel)

seeing
[Quantity] Astronomical seeing at time of observation

binning
[int] Pixel binning factor

quantum_efficiency
[float] Quantum efficiency of the detector

gain
[float] Gain of the detector (e-'s per ADU)

Methods Summary

counts(telescope, target, exposure_duration)	Number of ADU detected by the detector.
image(telescope, target, exposure_duration)	Generate an image of target observed by telescope.

Methods Documentation

counts(*telescope*, *target*, *exposure_duration*)
Number of ADU detected by the detector.

Parameters

telescope
[Telescope] Telescope object

target
[Target] Target object

exposure_duration
[Quantity] Exposure duration (s, or compatible unit)

Returns

total_counts
[float] Number of counts estimated in the exposure.

image(*telescope*, *target*, *exposure_duration*, *n=20*)
Generate an image of target observed by telescope.

Parameters

telescope
[Telescope] Telescope object

target
[Target] Target object

exposure_duration
[Quantity] Exposure duration (s, or compatible unit)

n
[int] Pixel length of a side of the image returned

Returns

image
[ndarray] Simulated image of target.

Target

class telescopy.**Target**(*magnitude=None*, *filter=None*)

Bases: object

Container for target metadata.

Parameters

magnitude
[float] Target magnitude in filter

filter
[Filter] Filter object.

Telescope

class telescopy.**Telescope**(*aperture_diameter=None*, *throughput=None*)

Bases: object

Container for information about a telescope.

Parameters

aperture_diameter
[Quantity] (Effective) Diameter of the primary mirror

throughput
[float] Telescope throughput

Methods Summary

photons(target, exposure_duration)	Compute the number of photons detected in an exposure_duration exposure of target.

Methods Documentation

photons(*target*, *exposure_duration*)

Compute the number of photons detected in an exposure_duration exposure of target.

Parameters

target
[Target] Target object

exposure_duration
[Quantity] Length of the exposure

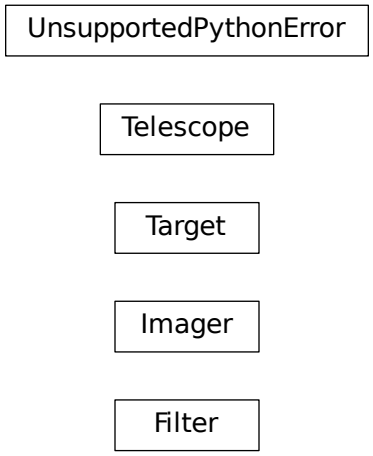
Returns

n_photons
[int] Number of photons observed.

UnsupportedPythonError

exception telescopy.**UnsupportedPythonError**

1.1.3 Class Inheritance Diagram



Python Module Index

t

telescopy, 7

Index

A

`available_filters()` (*telescopy.Filter static method*), 9

C

`counts()` (*telescopy.Imager method*), 10

F

`Filter` (*class in telescopy*), 8
`from_name()` (*telescopy.Filter class method*), 9

I

`image()` (*telescopy.Imager method*), 10
`Imager` (*class in telescopy*), 9

P

`photons()` (*telescopy.Telescope method*), 11
`plot()` (*telescopy.Filter method*), 9

T

`Target` (*class in telescopy*), 11
`Telescope` (*class in telescopy*), 11
`telescopy` (*module*), 7
`test()` (*in module telescopy*), 7

U

`UnsupportedPythonError`, 12