pathfinder Documentation

Release 0.6.2

John Keyes

2020-04-26 22:32:49.459061

Contents

The simplest way to install pathfinder is with pip:

```
pip install pathfinder
```

CHAPTER 1

Dependencies

pathfinder has no required external runtime dependencies.

However, image filtering requires *Pillow <https://pillow.readthedocs.io/en/stable/>*.

CHAPTER 2

Development Dependencies

The following packages are used in the development of pathfinder:

- nose: makes unit testing easier.
- coverage: code coverage.
- pylint: source code analyzer.
- Sphinx: documentation decorator.
- Pygments: Python syntax highlighting for documentation.
- docutils: reStructuredText support.
- Jinja: templating language.

CHAPTER 3

Pydoc

pathfinder package.

pathfinder.**find_paths**(*directory_path*, *just_dirs=None*, *just_files=None*, *regex=None*, *fnmatch=None*, *filter=None*, *ignore=None*, *abspath=None*, *depth=None*)
Find paths in the tree rooted at filepath.

pathfinder.**walk_and_filter**(*filepath*, *pathfilter*, *ignore=None*, *abspath=None*, *depth=None*)
Walk the file tree and filter it's contents.

pathfinder.**walk_and_filter_generator**(*filepath*, *pathfilter*, *ignore=None*, *abspath=None*, *depth=None*)
Walk the file tree and filter it's contents.

To ignore any paths an specify an ignore filter.

To return absolute paths pass True for the abspath parameter.

To limit how deep into the tree you travel, specify the depth parameter.

pathfinder - making it easy to find paths.

class pathfinder.filters.**AlwaysAcceptFilter**
Accept every path.

accepts(_)
Always returns True.

class pathfinder.filters.**AndFilter**(**args*)
Accept paths if all of it's filters accept the path.

accepts(*filepath*)
Returns True if all of the filters in this filter return True.

class pathfinder.filters.**ColorImageFilter**

class pathfinder.filters.**DirectoryFilter**
Accept directory paths.

accepts(*filepath*)
Returns True if filepath represents a directory.

class pathfinder.filters.**DotDirectoryFilter**
Do not accept a path for a directory that begins with a period.

class pathfinder.filters.**FileFilter**
Accept file paths.

accepts(*filepath*)
Returns True if filepath represents a file.

class pathfinder.filters.**FnmatchFilter**(*pattern*)
Accept paths if they match the specifed fnmatch pattern.

accepts(*filepath*)
Returns True if the fnmatch pattern matches the filepath.

class pathfinder.filters.**GreyscaleImageFilter**

class pathfinder.filters.**ImageDimensionFilter**(*max_width=None*, *max_height=None*, *min_width=None*, *min_height=None*)
Accept paths for Image files.

class pathfinder.filters.**ImageFilter**
Accept paths for Image files.

class pathfinder.filters.**NotFilter**(*pathfilter*)
Negate the accept of the specified filter.

accepts(*filepath*)
Returns True of the sub-filter returns False.

class pathfinder.filters.**OrFilter**(**args*)
Accept paths if any of it's filters accept the path.

accepts(*filepath*)
Returns True if any of the filters in this filter return True.

class pathfinder.filters.**RegexFilter**(*regex*)
Accept paths if they match the specified regular expression.

accepts(*filepath*)
Returns True if the regular expression matches the filepath.

class pathfinder.filters.**SizeFilter**(*max_bytes=None*, *min_bytes=None*)

accepts(*filepath*)
Returns True if filepath represents a file.

pathfinder.filters.**is_color_palette**(*palette*)
Return whether the palette has color.

pathfinder.filters.**is_greyscale_palette**(*palette*)
Return whether the palette is greyscale only.

pathfinder.filters.**stdv**(*band_means*)
Calculate the standard deviation of the image bands.

CHAPTER 4

Changelog

4.1 0.6.2

- raise an exception of the filepath parameter does not exist
- upgrade the version of Pillow

4.2 0.6.1

- fixed example in README.rst

4.3 0.6.0

- removed deprecated find and pathfind functions

4.4 0.5.4

- resolving security alert regarding the version of jinja2

4.5 0.5.3

- do not *chdir* in *walk_and_filter_generator*. #3. (https://github.com/rubik)

4.6 0.5.2

- Silly error in MANIFEST.in resolved.

4.7 0.5.1

- Added README.rst to MANIFEST.in to prevent install error from pip.

4.8 0.5

- new find_paths function returns a generator
- using any and all in OrFiter and AndFilter

4.9 0.4.1

- Fixed install error in setup.py

4.10 0.4

- File size filter
- Image filter
- Image dimensions filter
- Color image filter
- Greyscale image filter
- moved code to pathfinder package
- use nose for testing
- changed license from BSD to MIT
- override __or__ and __and__ for easy compound filter creation
- new Filter.find method
- ignore now works for filepaths

4.11 0.3.1

- Removed hard-coded file separators
- Added docstrings and comments

4.12 0.3

- Added depth parameter to walk_and_filter

4.13 0.2

- Added setup.py
- Fixed bug in NotFilter
- Tided pathfind function.

4.14 0.1

- First Cut

CHAPTER 5

pathfinder

pathfinder – a simpler *os.walk*

CHAPTER 6

Installation

Stable releases of pathfinder can be installed with pip or you may download a *.tgz* source archive from pypi. See the *Installation* page for more detailed instructions.

If you want to use the latest code, you can grab it from the Git repository, or fork it.

CHAPTER 7

Usage

7.1 Basic find

```
from pathfinder import find_paths

# all files and directories
paths = find_paths(".")

# all files
paths = find_paths(".", just_files=True)

# all directories
paths = find_paths(".", just_dirs=True)
```

By default *find_paths* prepends the path you search for to the results. If you want you can ensure the results only contain absolute paths:

```
paths = find_paths(".", abspath=True)
```

7.2 Filtering the results

Having a full listing is useful but wouldn't it be great if we could filter the results.

There a are a number of ways we can do this. Let's start with the Unix shell-style pattern approach:

```
# all PDF files
paths = find_paths(".", fnmatch="*.pdf")
```

fnmatching provides some power, but for more flexibility lets have a look at the regular expression support:

```
# all PDF files
paths = find_paths(".", regex=".*\.pdf")

# all PDF files with four letter base names
paths = find_paths(pwd, regex=".*/.{4}\.pdf")
```

pathfinder provides the ability to ignore certain paths too:

```
# create your ignore filter to ignore all PDF files
# from the files with three character extensions
from pathfinder import FnmatchFilter
ignore = FnmatchFilter("*.pdf")
find_paths(".", regex=".*/.*\..{3}$", ignore=ignore)

# ignore all files and directories that begin with .
ignore = RegexFilter("\..*")
find_paths(".", ignore=ignore)
```

7.3 Controlling the depth of the search

You may want to limit how to deep to search into a directory tree:

```
# only search down two levels
find_paths(".", depth=2)
```

7.4 Extra support for images

Let's find some images in the directory:

```
# all of the images
from pathfinder import ImageFilter
find_paths(".", filter=ImageFilter())
```

That is just a shortcut for matching multiple file extensions, but we can also filter the results based on the dimensions of the image:

```
# only images less than 20 pixels tall
from pathfinder import ImageDimensionFilter
find_paths(".", filter=ImageDimensionFilter(max_height=20))

# only images less than 10 pixels tall and wide
from pathfinder import ImageDimensionFilter
find_paths(".", filter=ImageDimensionFilter(max_height=10, min_height=10))
```

And we can also search for images based on their color paletter:

```
# only color images
from pathfinder import ColorImageFilter
find_paths(".", filter=ColorImageFilter())

# only greyscale images
from pathfinder import GreyscaleImageFilter
find_paths(".", filter=GreyscaleImageFilter())
```

7.5 Combining filters

Filters can also be combined to create even more complex filters (just in case you need them). pathfinder supports AND, OR and NOT functions.

```
# color images AND greater than 400 bytes
from pathfinder import ColorImageFilter
from pathfinder import SizeFilter
color = ColorImageFilter()
size = SizeFilter(max_bytes=400)
find_paths(".", filter=color & size)

# pdf OR txt files
from pathfinder import FnmatchFilter
txt = FnmatchFilter("*.txt")
pdf = FnmatchFilter("*.pdf")
find_paths(".", filter=txt | pdf)

# txt files, but NOT ones begining with a
from pathfinder import NotFilter
from pathfinder import SizeFilter
from pathfinder import FnmatchFilter
txt = FnmatchFilter("*.txt")
afiles = NotFilter(FnmatchFilter("*/a*"))
find_paths(".", filter=txt & afiles)
```

7.6 find shortcut

You can also run a find directly from a filter:

```
from pathfinder import SizeFilter
SizeFilter(max_bytes=1024).find(".")
```

CHAPTER 8

Changelog

The *Changelog* keeps track of changes per release.

Python Module Index

p

Index