

---

# urlchecker

Nov 15, 2022



---

## Contents:

---

<b>1</b>	<b>Documentation</b>	<b>3</b>
1.1	urlchecker.check . . . . .	3
1.2	urlchecker.core.fileproc . . . . .	3
1.3	urlchecker.core.urlproc . . . . .	4
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



*urlchecker* : A GitHub action to collect and check URLs in a project (code and documentation). The action aims at detecting and reporting broken links.



### 1.1 urlchecker.check

### 1.2 urlchecker.core.fileproc

Copyright (c) 2020-2022 Ayoub Malek and Vanessa Sochat

This source code is licensed under the terms of the MIT license. For a copy, see <<https://opensource.org/licenses/MIT>>.

`urlchecker.core.fileproc.check_file_type` (*file\_path: str, file\_types: List[str]*) → bool

Check file type to assert that only file with certain predefined extensions are checked. We currently support an extension verbatim, or regular expression to match the filename. For example, `.*` matches all hidden files, and `*.html` matches an html file.

**Args:**

- `file_path` (str) : path to file.
- `file_types` (list) : list of file extensions to accept.

**Returns:** (bool) true if file type is supported else false.

`urlchecker.core.fileproc.collect_links_from_file` (*file\_path: str, unique: bool = True*)  
→ List[str]

Collect all links in a file.

**Args:**

- `file_path` (str) : path to file.
- `unique` (bool) : specify whether to filter out duplicate links.

**Returns:** (list) list of links/ urls in a file.

`urlchecker.core.fileproc.get_file_paths` (*base\_path: str, file\_types: List[str], exclude\_files: List[str] = None, include\_patterns: List[str] = None*) → List[str]

Get path to all files under a give directory and its subfolders.

**Args:**

- `base_path` (str) : base path.
- `file_types` (list) : list of file extensions to accept.
- `include_patterns` (list) : list of files and patterns to include.
- `exclude_files` (list) : list of files or patterns to exclude

**Returns:** (list) list of file paths.

`urlchecker.core.fileproc.include_file` (*file\_path: str, exclude\_patterns: List[str] = None, include\_patterns: List[str] = None*) → bool

Check a file path for inclusion based on an OR regular expression. The user is currently not notified if a file is marked for removal.

**Args:**

- `file_path` (str) : a file path to check if should be included.
- `exclude_patterns` (list) : list of patterns to exclude.
- `include_patterns` (list) : list of patterns to include.

**Returns:** (bool) boolean indicating if the URL should be excluded (not tested).

`urlchecker.core.fileproc.remove_empty` (*file\_list: List[str]*) → List[str]

Given a file list, return only those that aren't empty string or None.

**Args:**

- `file_list` (list): a list of files to remove None or empty string from.

**Returns:** (list) list of (non None or empty string) contents.

## 1.3 urlchecker.core.urlproc



## CHAPTER 2

---

### Indices and tables

---

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



**u**

`urlchecker.core.fileproc`, 3



### C

`check_file_type()` (*in module `urlchecker.core.fileproc`*), 3  
`collect_links_from_file()` (*in module `urlchecker.core.fileproc`*), 3

### G

`get_file_paths()` (*in module `urlchecker.core.fileproc`*), 3

### I

`include_file()` (*in module `urlchecker.core.fileproc`*), 4

### R

`remove_empty()` (*in module `urlchecker.core.fileproc`*), 4

### U

`urlchecker.core.fileproc` (*module*), 3