

Package: urlchecker (via r-universe)

October 13, 2026

Title Run CRAN URL Checks from Older R Versions

Version 2.0.0.9000

Description Provide the URL checking tools available in R 4.1+ as a package for earlier versions of R. Also uses concurrent requests so can be much faster than the serial versions.

License GPL-3

URL <https://github.com/r-lib/urlchecker>, <https://urlchecker.r-lib.org/>

BugReports <https://github.com/r-lib/urlchecker/issues>

Depends R (>= 4.1)

Imports cli, curl, gitcreds, rlang (>= 1.1.0), tools, utils, xml2

Suggests covr, testthat (>= 3.0.0), webfakes, withr

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first url_check, qmd, url_update, print

Config/usethis/last-upkeep 2025-05-07

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libxml2-dev libssl-dev

Repository <https://r-lib.r-universe.dev>

Date/Publication 2026-07-09 06:07:42 UTC

RemoteUrl <https://github.com/r-lib/urlchecker>

RemoteRef HEAD

RemoteSha 0c0eb9c487f506443743387274e7997d2f949a61

Contents

url_check	2
url_update	3
Index	5

url_check	<i>Check urls in a package or project</i>
-----------	---

Description

For an R package, runs the `url_db_from_package_source` function in the `tools` package along with functions to check URLs in un-rendered Rmarkdown (`.Rmd`) and Quarto (`.qmd`) vignettes and in BibTeX (`.bib`) bibliographies. For non-package projects, URLs are extracted from all supported files found in the given directories.

Usage

```
url_check(  
  path = ".",  
  db = NULL,  
  parallel = TRUE,  
  pool = curl::new_pool(),  
  progress = TRUE,  
  fail = TRUE  
)
```

Arguments

path	Path(s) to check. Each element may be: • A package’s (development) source directory tree, a directory holding an unpacked source package, or a source package tarball (<code>.tar.gz</code>). A tarball is unpacked into a temporary directory (kept for the rest of the session, so the printed report can point into the sources). • A directory that is not an R package. All supported files found within (recursively) are scanned for URLs. Supported files are HTML, PDF, Rd, Markdown (<code>.md</code>, <code>.markdown</code>), R Markdown (<code>.Rmd</code>), Quarto (<code>.qmd</code>) and BibTeX (<code>.bib</code>). • A single file of one of the supported types above. path may be a character vector mixing any of these.
db	A url database
parallel	If TRUE, check the URLs in parallel
pool	A multi handle created by <code>curl::new_pool()</code> . If NULL use a global pool.
progress	Whether to show the progress bar for parallel checks
fail	If TRUE (the default), throw an error when one or more URLs are flagged, after printing the report. This yields a non-zero exit status, which is useful in CI/CD workflows. Set to FALSE to return the results instead of failing.

Value

A `url_checker_db` object (invisibly). This is a `check_url_db` object with an added class with a custom print method.

Ignoring URLs with .urlignore

Some URLs cannot be checked automatically, e.g. a link to a private repository, or a page behind a login or captcha. To stop `url_check()` from flagging (and even requesting) such URLs, list them in a `.urlignore` file. It is read from two locations relative to the checked root, and the patterns found are combined:

- `.urlignore` in the root directory (like `.gitignore`), and
- `tools/.urlignore` (handy for packages that keep it under `tools/`).

Each non-empty line is a glob pattern (blank lines and lines starting with `#` are ignored), matched against the whole URL. For example `https://github.com/acme/secret` matches that URL exactly, while `https://github.com/acme/*` matches every URL under that path. Matching URLs are dropped before checking, so they are never requested.

Note that CRAN's own URL checks do not read `.urlignore`, so an ignored URL may still be flagged when the package is submitted to CRAN.

Examples

```
## Not run:
url_check("my_pkg")
url_check(c("README.md", "docs"))

## End(Not run)
```

url_update

Update URLs in a package

Description

First uses [url_check](#) to check and then updates any URLs which are permanent (301) redirects.

Usage

```
url_update(path = ".", results = url_check(path, fail = FALSE))
```

Arguments

<code>path</code>	Path to the package
<code>results</code>	results from url_check .

Value

The results from `url_check(path)`, invisibly.

Examples

```
## Not run:  
url_update("my_pkg")  
  
## End(Not run)
```

Index

`curl::new_pool()`, [2](#)

`url_check`, [2](#), [3](#)

`url_update`, [3](#)