

# Package: urlchecker (via r-universe)

February 1, 2025

**Title** Run CRAN URL Checks from Older R Versions

**Version** 1.0.1.9000

**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 (>= 3.4)

**Imports** cli, curl, tools, xml2

**Suggests** covr

**Config/Needs/website** tidyverse/tidytemplate

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Config/pak/sysreqs** libssl-dev

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

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

**RemoteRef** HEAD

**RemoteSha** ac38ea46aa468dc4f6e3ae991e27ab754146314e

## Contents

url_check . . . . .	2
url_update . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

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

---

### Description

Runs the `url_db_from_package_source` function in the `tools` package along with a function to check URLs in un-rendered Rmarkdown vignettes.

### Usage

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

### Arguments

<code>path</code>	Path to the package
<code>db</code>	A url database
<code>parallel</code>	If TRUE, check the URLs in parallel
<code>pool</code>	A multi handle created by <code>curl::new_pool()</code> . If NULL use a global pool.
<code>progress</code>	Whether to show the progress bar for parallel checks

### Value

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

### Examples

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

---

url_update	<i>Update URLs in a package</i>
------------	---------------------------------

---

**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))
```

**Arguments**

path	Path to the package
results	results from <a href="#">url_check</a> .

**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