Package 'rurl'

May 1, 2025

Type Package

Title Lightweight URL Parsing and Cleaning Tools

Version 0.1.3

Language en-US

Description A lightweight toolkit for extracting structured information from URLs.
Includes functions for parsing, normalizing protocols, extracting domains, and constructing clean URLs.
The package includes a processed copy of the Public Suffix List from https://publicsuffix.org for domain extraction.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports curl

URL https://github.com/bart-turczynski/rurl, https://publicsuffix.org

BugReports https://github.com/bart-turczynski/rurl/issues

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

Contents

get_clean_url *Get cleaned URLs*

Description

This function returns the cleaned version of the URLs by ensuring that the URLs are valid and, if necessary, prepends "http://" when the protocol is missing.

Usage

```
get_clean_url(url, protocol_handling = "keep")
```

Arguments

url	A character vector containing URLs to be parsed. This can include URLs without a scheme (e.g., "example.com") or URLs with a scheme (e.g., "http://example.com").
protocol_handling	A character string specifying how to handle protocols. Can be one of "keep", "none", "strip", "http", "https". The protocol is preserved if it exists, and "http://" is added if missing. If "none", no protocol is added. If "http://" or "https://" the given protocol is added or changed to the one indicated.

Value

A character vector of cleaned URLs.

Examples

```
get_clean_url("example.com")
get_clean_url("http://example.com")
get_clean_url("https://example.com")
get_clean_url("ftp://example.com")
```

get_domain *Get domain names*

Description

This function extracts the domain name from a given URL. It returns only the domain part of the URL (e.g., "example.com" from "http://example.com"). #' Note the domain is determined based on Public Suffix List at https://publicsuffix.org/list/public_suffix_list.dat Which may not give intuitive results sometimes. For example, blogspot.com is treated as a TLD but wordpress.org is not.

Usage

```
get_domain(url, protocol_handling = "keep")
```

Arguments

`url`	A character vector containing URLs from which to extract the domain.
`protocol_handling`	A character string specifying how to handle protocols. Can be one of "keep", "none", "strip", "http", "https". The protocol is preserved if it exists, and "http://" is added if missing. If "none", no protocol is added. If "http://" or "https://" the given protocol is added or changed to the one indicated.

Details

For example:

get_domain("https://test.wordpress.org") https://test.wordpress.org "wordpress.org"

But:

get_domain("https://test.blogspot.com") https://test.blogspot.com "test.blogspot.com"

Deciding what is a "proper" domain name is an ambitious yet futile task. I gave up and decided to use something that already exists and is respected.

Value

A character vector with domain names extracted from the given URLs.

Examples

```
get_domain("http://example.com/path")
get_domain("https://sub.domain.org/")
get_domain("ftp://ftp.example.com")
```

`get_host` *Get URL hosts*

Description

This function extracts the host (domain) of a given URL. It returns the host name (e.g., "example.com") of the URL.

Usage

```
get_host(url, protocol_handling = "keep")
```

Arguments

`url`	A character vector containing URLs from which to extract the host.
`protocol_handling`	A character string specifying how to handle protocols. Can be one of "keep", "none", "strip", "http", "https". The protocol is preserved if it exists, and "http://" is added if missing. If "none", no protocol is added. If "http://" or "https://" the given protocol is added or changed to the one indicated.

Value

A character vector with the host of each URL. In layman's terms, the host being the part of the address "between the protocol and first slash / end of the string if no slash is present, e.g., test.wordpress.org, www.r-project.org. Note the host and the domain may be the same thing but for different reasons.

Examples

```
get_host("http://example.com")
get_host("ftp://example.com")
get_host("https://sub.domain.com")
```

---

get_parse_status *Get the parse status of URLs*

Description

Get the parse status of URLs

Usage

```
get_parse_status(url, protocol_handling = "keep")
```

Arguments

`url`	A character vector of URLs to be parsed.
`protocol_handling`	A character string specifying how to handle protocols. Can be one of "keep", "none", "strip", "http", "https". The protocol is preserved if it exists, and "http://" is added if missing. If "none", no protocol is added. If "http://" or "https://" the given protocol is added or changed to the one indicated.

Value

A character vector with the parse status of each URL:

- "ok" for http(s) URLs.
- "ok-ftp" for ftp and ftps URLs.
- "error" for unsupported schemes (mailto, file, etc.) or invalid URLs.

Examples

```
get_parse_status(c("http://example.com", "ftp://example.com", "mailto:user@example.com"))
get_parse_status(c("http://example.com", "not-a-url"))
```

get_path *Get URL paths*

Description

This function extracts the path component of a given URL. The path refers to the part of the URL that follows the domain, such as "/path/to/resource".

Usage

```
get_path(url, protocol_handling = "keep")
```

Arguments

url	A character vector containing URLs from which to extract the path.
protocol_handling	A character string specifying how to handle protocols. Can be one of "keep", "none", "strip", "http", "https". The protocol is preserved if it exists, and "http://" is added if missing. If "none", no protocol is added. If "http://" or "https://" the given protocol is added or changed to the one indicated.

Value

A character vector with the path of each URL. If no path exists, it will return an empty string.

Examples

```
get_path("http://example.com/path/to/resource")
get_path("ftp://example.com/another/path")
get_path("https://example.com")
```

get_scheme *Get URL schemes*

Description

This function extracts the scheme (protocol) of a given URL. It returns the scheme (e.g., "http", "https", "ftp", etc.) of the URL.

Usage

```
get_scheme(url, protocol_handling = "keep")
```

Arguments

url	A character vector containing URLs from which to extract the scheme.
protocol_handling	A character string specifying how to handle protocols. Can be one of "keep", "none", "strip", "http", "https". The protocol is preserved if it exists, and "http://" is added if missing. If "none", no protocol is added. If "http://" or "https://" the given protocol is added or changed to the one indicated.

Value

A character vector with the scheme (e.g., "http", "https", "ftp") of each URL.

Examples

```
get_scheme("http://example.com")
get_scheme("ftp://example.com")
get_scheme("https://example.com")
```