Package 'urlparse'

April 14, 2025

Type Package

Title Fast Simple URL Parser

Version 0.2.1

Description A fast and simple 'URL' parser package for 'R'. This package provides functions to parse 'URLs' into their components, such as scheme, user, password, host, port, path, query, and fragment.

License MIT + file LICENSE

URL https://github.com/dyfanjones/urlparse, https://dyfanjones.r-universe.dev/urlparse

BugReports https://github.com/dyfanjones/urlparse/issues

Encoding UTF-8

LinkingTo Rcpp

Imports Rcpp

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

Author Dyfan Jones [aut, cre]

Maintainer Dyfan Jones <dyfan.r.jones@gmail.com>

Repository CRAN

Date/Publication 2025-04-14 15:00:02 UTC

Contents

encoding *Escape characters for use in URLs.*

Description

This function encodes a character vector for use in URLs, escaping all special characters except for those specified in the `safe` parameter.

Usage

```
url_encoder(urls, safe = "")

url_decoder(urls)
```

Arguments

`urls`	A character vector to be encoded/decoded.
`safe`	A character vector of extra characters that should not be encoded.

Value

A character vector with the encoded URLs.

Examples

```
library(urlparse)

# Example 1:
url_encoder("foo = bar + 5")

# Example 2:
# prevent special characters being encoded:
url <- "https://example.com/path?query= 1+2"
url_encoder(url, ":/?=")

# Example 3:
url_decoder(url_encoder("foo = bar + 5"))
```

`url_build` *Builds a URL string from its components.*

Description

Builds a URL string from its components.

Usage

```
url_build(url_components)
```

Arguments

`url_components` A list containing the components of the URL: scheme, host, port, path, query, and fragment.

- **scheme** A character string for the new scheme (e.g., "http" or "https") or `NULL` to keep it unchanged.
- **host** A character string for the new host or `NULL` to keep it unchanged.
- **port** A character string for the new port or `NULL` to keep it unchanged.
- **path** A character string for the new path or `NULL` to keep it unchanged.
- **query** A list or character of new query parameters or `NULL` to keep it unchanged.
- **fragment** A character string for the new fragment or `NULL` to keep it unchanged.

Value

A URL string constructed from the provided components

Examples

```
library(urlparse)
url_build(list(
  scheme = "https",
  user = "",
  password = "",
  host = "host.com",
  port = 8000,
  path = "/path",
  query = "query",
  fragment = "fragment"
))
```

`url_modify` *Modifies a URL string by updating its components.*

Description

This function modifies a URL string by updating its components such as scheme, user, password, host, port, query, raw query, and fragment. If any of these components are not provided (i.e., `NULL`), the existing components of the URL are retained.

Usage

```
url_modify(
  url,
  scheme = NULL,
  user = NULL,
  password = NULL,
  host = NULL,
  port = NULL,
  path = NULL,
  query = NULL,
  fragment = NULL
)

set_scheme(url, scheme)

set_user(url, user)

set_password(url, password)

set_host(url, host)

set_port(url, port)

set_path(url, path)

set_query(url, query)

set_fragment(url, fragment)
```

Arguments

`url`	A character string representing the original URL.
`scheme`	A character string for the new scheme (e.g., "http" or "https") or `NULL` to keep it unchanged.
`user`	A character string for the username or `NULL` to keep it unchanged.
`password`	A character string for the new password or `NULL` to keep it unchanged.

host	A character string for the new host or NULL to keep it unchanged.
port	A character string for the new port or NULL to keep it unchanged.
path	A character string for the new path or NULL to keep it unchanged.
query	A list or character of new query parameters or NULL to keep it unchanged.
fragment	A character string for the new fragment or NULL to keep it unchanged.

Value

A character string representing the modified URL.

Examples

```
library(urlparse)

# Example 1: Modify the scheme and host of a URL
url_modify(
  "https://user:pass@host.com/path?query#fragment",
  scheme = "http",
  host = "example.com"
)

# Example 2: Add a query parameter to a URL
url_modify(
  "https://host.com/path", query = list(key1 = "value1", key2 = "value2")
)

# Example 3: Change the fragment of a URL
url_modify("https://host.com/path#old_fragment", fragment = "new_fragment")
```

---

url_parse *Parses a URL string into its components.*

Description

Parses a URL string into its components.

Usage

```
url_parse(url)
```

Arguments

url	The URL string to parse.

Value

A list containing the components of the URL: scheme, user, password, host, path, raw_path, query, raw_query, and fragment.

Examples

```
library(urlparse)
url_parse("https://host.com/path?query#fragment")
```

`url_parse_v2` *Parses a vector URLs into a dataframe.*

Description

Parses a vector of URLs into their respective components. It returns a data.frame where each row represents a URL, and each column represents a specific component of the URL such as the scheme, user, password, host, port, path, raw path, raw query, and fragment.

Usage

```
url_parse_v2(url)
```

Arguments

`url`	A vector of strings, where each string is a URL to be parsed.

Value

A data frame with the following columns: - href: The original URL. - scheme: The scheme component of the URL (e.g., "http", "https"). - user: The user component of the URL. - password: The password component of the URL. - host: The host component of the URL. - port: The port component of the URL. - path: The decoded path component of the URL. - raw_path: The raw path component of the URL. - raw_query: The raw query component of the URL. - fragment: The fragment component of the URL.

Examples

```
library(urlparse)
urls <- c("https://user:password@www.example.com:8080/path/to/resource?query=example#fragment",
          "http://www.test.com")
url_parse_v2(urls)
```

Index