Package 'pingr'

December 12, 2024

Title Check if a Remote Computer is Up

Version 2.0.5

Description Check if a remote computer is up. It can either just call the system ping command, or check a specified TCP port.

License MIT + file LICENSE

URL https://r-lib.github.io/pingr/, https://github.com/r-lib/pingr

BugReports https://github.com/r-lib/pingr/issues

Depends R (>= 3.6)

Imports processx, utils

Suggests covr, ps, testthat (>= 3.0.0)

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

Biarch true

NeedsCompilation yes

Author Gábor Csárdi [aut, cre],
Posit Software, PBC [cph, fnd]

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Repository CRAN

Date/Publication 2024-12-12 10:10:02 UTC

Contents

`apple_captive_test` *Download Apple's captive portal test*

Description

If the test page, returns "Success" that means that the computer is connected to the Internet.

Usage

```
apple_captive_test()
```

Details

Note that this function will fail if the computer is offline. Use `is_online()` to check if the computer is online.

Examples

```
apple_captive_test()
```

`is_online` *Is the computer online?*

Description

Check if the computer is online. It does three tries:

- Retrieve Apple's Captive Portal test page, see `apple_captive_test()`.
- Queries myip.opendns.com on OpenDNS, see `my_ip()`.
- Retrieves icanhazip.com via HTTPS, see `my_ip()`. If any of these are successful, it returns `TRUE`.

Usage

```
is_online(timeout = 1)
```

Arguments

`timeout`	Timeout for the queries. (Note: it is currently not used for the DNS query.)

Value

Possible values:

- `TRUE` Yes, online.
- `FALSE` No, not online.

Examples

```
is_online()
```

`my_ip` *Query the computer's public IP address*

Description

It can use a DNS query to opendns.com, if `method == "dns"`, or an HTTPS query to icanhazip.com, see https://github.com/major/icanhaz. The DNS query is much faster, the HTTPS query is secure.

Usage

```
my_ip(method = c("dns", "https"))
```

Arguments

`method`	Whether to use a DNS or HTTPS query.

Value

Computer's public IP address as a string.

Examples

```
my_ip()
my_ip(method = "https")
```

`nsl` *DNS query*

Description

Perform a DNS query for a domain. It supports custom name servers, and querying DNS records of certain class and type.

Usage

```
nsl(domain, server = NULL, type = 1L, class = 1L)
```

Arguments

domain	Domain to query.
server	Custom name server IP address, to use. Note that this must be an IP address currently. E.g. 8.8.8.8 is Google's DNS server.
type	Record type to query, an integer scalar. 1L is an A record, 28L is an AAAA record, etc. See e.g. https://en.wikipedia.org/wiki/List_of_DNS_record_types for the record types.
class	Query class. This is usually 1L, i.e. "Internet". See e.g. https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-2 for all DNS classes.

Value

A list of two entries currently, additional entries might be added later:

- `answer`: a data frame of DNS records, with columns: `name`, `class`, `type`, `ttl`, `data`. `data` is a list column and contains the IP(6) address for A and AAAA records, but it contains other data, e.g. host name for CNAME, for other records. If pingr could not parse a record (it only parses the most common records types: A, AAAA, NA, PTR, CNAME, TXT, MX, SOA), then the data of the record is included as a raw vector.
- `flags`: a named logical vector of flags `aa`, `tc`, `rd`, `ra`, `ad`, `cd`. See the RFC (https://www.ietf.org/rfc/rfc1035.txt) for these. On Windows they are all set to NA currently.

Examples

```
nsl("r-project.org")
nsl("google.com", type = 28L)
```

---

ping *Ping a remote server, to see if it is alive*

Description

This is the classic ping, using ICMP packages. Only the system administrator can send ICMP packages, so we call out to the system's ping utility.

Usage

```
ping(
  destination,
  continuous = FALSE,
  verbose = continuous,
  count = 3L,
  timeout = 1
)
```

Arguments

destination	Host name or IP address.
continuous	Logical, whether to keep pinging until the user interrupts.
verbose	Whether to print progress on the screen while pinging.
count	Number of pings to perform.
timeout	Timeout for a ping response.

Value

Vector of response times. NA means no response, in milliseconds. Currently NAs are always at the end of the vector, and not in their correct position.

Examples

```
ping("8.8.8.8")
ping("r-project.org")
```

---

ping_port *Check if a port of a server is active, measure response time*

Description

Check if a port of a server is active, measure response time

is_up() checks if a web server is up.

Usage

```
ping_port(
  destination,
  port = 80L,
  continuous = FALSE,
  verbose = continuous,
  count = 3L,
  timeout = 1
)

is_up(
  destination,
  port = 80,
  timeout = 0.5,
  fail_on_dns_error = FALSE,
  check_online = TRUE
)
```

Arguments

destination	Host name or IP address.
port	Port.
continuous	Logical, whether to keep pinging until the user interrupts.
verbose	Whether to print progress on the screen while pinging.
count	Number of pings to perform.
timeout	Timeout, in seconds. How long to wait for a ping to succeed.
fail_on_dns_error	If TRUE then is_up() fails if the DNS resolution fails. Otherwise it will return FALSE.
check_online	Whether to check first if the computer is online. Otherwise it is possible that the computer is behind a proxy, that hijacks the HTTP connection to destination.

Value

Vector of response times, in milliseconds. NA means no response within the timeout.

Examples

```
ping_port("r-project.org")


is_up("google.com")
is_up("google.com", timeout = 0.01)
```

Index