Package ‘rhub’

November 1, 2017

Title Connect to 'R-hub'

Version 1.0.2

Author Gábor Csárdi

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

Description Run 'R CMD check' on any of the 'R-hub' architectures, from
the command line. The current architectures include 'Windows', 'macOS',
'Solaris' and various 'Linux' distributions.

License MIT + file LICENSE

LazyData true

URL https://github.com/r-hub/rhub#readme

BugReports https://github.com/r-hub/rhub/issues

RoxygenNote 6.0.1.9000

Roxygen list(markdown = TRUE)

Imports assertthat,
callr,
clisymbols,
crayon,
desc,
httr,
jsonlite,
parsedate,
prettyunits,
R6,
rappdirs,
rcmdcheck (>= 1.2.1),
rematch,
utils,
whoami,
withr

Suggests covr,
testthat

Encoding UTF-8

R topics documented:

check — *Check an R package on r-hub*

Description

Check an R package on r-hub

Usage

```
check(path = ".", platform = NULL, email = NULL, valgrind = FALSE,
  check_args = character(), env_vars = character(),
  show_status = interactive())
```

Arguments

`path`	Path to a directory containing an R package, or path to source R package tarball built with `R CMD build` or `devtools::build()`.
`platform`	Platform to build/check the package on. See `platforms()` for the available platforms. If this is `NULL`, and the R session is interactive, then a menu is shown. If it is `NULL`, and the session is not interactive, then the default r-hub platforms is used. Can take a vector of platforms which saves time by building one R package tarball that is used for all the platforms specified.
`email`	Email address to send notification to about the build. It must be a validated email address, see `validate_email()`. If `NULL`, then the email address of the maintainer is used, as defined in the `DESCRIPTION` file of the package.
`valgrind`	Whether to run the check in valgrind. Only supported on Linux currently, and ignored on other platforms.
`check_args`	Extra arguments for the `R CMD check` command.
`env_vars`	Environment variables to set on the builder machine before the check. A named character vector.
`show_status`	Whether to show the status of the build as it is happening.

Value

An rhub_check object.

Examples

```
## Not run:
check(".")
check("mypackage_1.0.0.tar.gz", platform = "fedora-clang-devel")

## End(Not run)
```

---

check_for_cran   *Check an R-package on R-hub, for a CRAN submission*

Description

This function calls check() with arguments and platforms, that are suggested for a CRAN submission.

Usage

```
check_for_cran(path = ".", email = NULL, check_args = "--as-cran",
  env_vars = c(`_R_CHECK_FORCE_SUGGESTS_` = "true",
  `_R_CHECK_CRAN_INCOMING_USE_ASPELL_` = "true"), platforms = NULL, ...)
```

Arguments

path	Path to a directory containing an R package, or path to source R package tarball built with `R CMD build` or `devtools::build()`.
email	Email address to send notification to about the build. It must be a validated email address, see `validate_email()`. If `NULL`, then the email address of the maintainer is used, as defined in the `DESCRIPTION` file of the package.
check_args	Arguments for `R CMD check`. By default `--as-cran` is used.
env_vars	Environment variables to set on the builder. By default `_R_CHECK_FORCE_SUGGESTS_=true` is set, to require all packages used. `_R_CHECK_CRAN_INCOMING_USE_ASPELL_=true` is also set, to use the spell checker.
platforms	Character vector of platform ids to use (see `platforms()`), or `NULL`. If `NULL`, then a set of default platforms will be selected, see below.
...	Additional arguments are passed to `check()`.

Details

In particular, if `platforms` is `NULL` (the default), then

- It checks the package on Windows, and Linux.
- It checks the package on R-release and R-devel.
- It uses the `--as-cran` argument to `R CMD build`.
- It requires all dependencies, including suggested ones.

Value

An rhub_check object.

Examples

```
## Not run:
ch <- check_for_cran("package", show_status = FALSE)
ch$update()
ch$livelog(3)

## End(Not run)
```

check_on_linux *Check an R package on an r-hub platform*

Description

These functions provide a quick easy to use interface to check a package on a platform with some particular aspect. Which platform they use might change over time.

Usage

```
check_on_linux(path = ".", ...)

check_on_windows(path = ".", ...)

check_on_macos(path = ".", ...)

check_on_debian(path = ".", ...)

check_on_ubuntu(path = ".", ...)

check_on_fedora(path = ".", ...)

check_on_centos(path = ".", ...)

check_with_roldrel(path = ".", ...)

check_with_rrelease(path = ".", ...)

check_with_rpatched(path = ".", ...)

check_with_rdevel(path = ".", ...)

check_with_valgrind(path = ".", ...)

check_with_sanitizers(path = ".", ...)
```

Arguments

`path`	Path to a directory containing an R package, or path to source R package tarball built with `R CMD build` or `devtools::build()`.
`...`	Additional arguments are passed to `check()`.

Value

An rhub_check object.

`last_check` *The last rhub check of this R session*

Description

`rhub` caches the id(s) of the last submission. This can be retrieved with `last_check`.

Usage

```
last_check()
```

Value

An rhub_check object.

Examples

```
## Not run:
check("packagedir")
last_check()
last_check()$livelog()

## End(Not run)
```

`list_my_checks` *List all checks for an email address*

Description

List all checks for an email address

Usage

```
list_my_checks(email = email_address(), package = NULL, howmany = 20)
```

Arguments

`email`	Email address. By default it is guessed with `whoami::email_address()`. The address must be validated, see `validate_email()`.
`package`	`NULL`, or a character scalar. Can be used to restrict the search for a single package.
`howmany`	How many checks to show. The current API limit is 20.

Value

An `rhub_check` object.

Examples

```
## Not run:
ch <- list_my_checks()
ch
ch$details()

## End(Not run)
```

---

list_package_checks *List checks of a package*

Description

List checks of a package

Usage

```
list_package_checks(package = ".", email = NULL, howmany = 20)
```

Arguments

package	Directory of an R package, or a package tarball.
email	Email address that was used for the check(s). If NULL, then the maintainer address is used.
howmany	How many checks to show. The current maximum of the API is 20.

Value

An rhub_check object.

Examples

```
## Not run:
ch <- list_my_checks()
ch
ch$details(1)

## End(Not run)
```

---

list_validated_emails *List validated email addresses*

Description

List validated email addresses

Usage

```
list_validated_emails()
```

Value

A data frame with two columns: `email` and `token`. If in interactive mode, and there are no validated email addresses, then a message is printed and the data frame is returned invisibly.

See Also

Other email validation: `validate_email`

`platforms` *List all r-hub platforms*

Description

The platforms are pretty-printed in a short format. Use `as.data.frame(platforms())` to get all available platform metadata.

Usage

```
platforms()
```

Examples

```
## Not run:
platforms()
as.data.frame(platforms())

## End(Not run)
```

`rhub_check` *An* `rhub_check` *object holds status and results of rhub checks*

Description

An `rhub_check` object holds status and results of rhub checks

This class is similar to rhub_check (it inherites from it), but it shows the results of the checks in a more consise format by default. To show the details, call the `$details()` method.

Usage

```
ch$update()
ch$print(...)
ch$web(which = NULL)
ch$livelog(which = 1)

chl$update()
chl$print(...)
chl$details()
chl$livelog(which = 1)
```

Arguments

- ch An rhub check object. It can be created using check(), list_my_checks() or list_package_checks(). See also last_check().
- ... Extra arguments are currently ignored.
- which Which check to show, if the object contains multiple checks. For web the default is all checks. For livelog the default is the first check. A check can be selected via its number or id.

Details

An rhub_check object can be created by check(), list_my_checks(), or list_package_checks(). last_check() returns the last check(s) submitted from the current R session.

ch$update() updates the status of the check. Printing the check status to the screen does not perform an update, unless the status of the check(s) is unknown.

ch$print() prints the status of the check(s) to the screen.

ch$web() opens a tab or window in the default web browser, that points to the detailed logs of the check.

ch$livelog() shows the live log of the check. The live log can be interrupted using the usual interruption keyboard shortcut, usually CTRL+c or ESC.

The list_my_checks() and list_package_checks() functions create rhub_check_list objects.

See rhub_check for the description of the inherited chl$update(), chl$print(), chl$livelog() methods.

chl$details() prints the details for all checks in the list.

Examples

```
## Not run:
check()
ch <- last_check()
ch$update()
ch$web()
ch$livelog()

## End(Not run)
## Not run:
chl <- list_my_checks()
chl
chl$details()

## End(Not run)
```

---

validate_email *Validate an email address on r-hub*

Description

To build and check R packages on r-hub, you need to validate your email address. This is because r-hub sends out emails about build results.

Usage

```
validate_email(email = NULL, token = NULL)
```

Arguments

`email`	The email address to validate.
`token`	Token obtained from `rhub`, to validate the email address.

Details

The `rhub` package stores validated email addresses in a user configuration file, at a platform-dependent location. On your current platform the file is at /Users/gaborcsardi/Library/Application Support/rhub/validated_emails.csv.

To validate a new email address, call this function from an interactive R session, without any arguments.

To add an email address that was validated before (probably on another machine), to the configuration file, call this function with the `email` and `token` arguments.

See Also

Other email validation: `list_validated_emails`

Index