Package 'pins'

October 1, 2019

Type Package

Title Pin, Discover and Share Resources

Version 0.2.0

Maintainer Javier Luraschi <javier@rstudio.com>

Description Pin remote resources into a local cache to work offline, improve speed and avoid recomputing; discover and share resources in local folders, 'GitHub', 'Kaggle' or 'RStudio Connect'. Resources can be anything from 'CSV', 'JSON', or image files to arbitrary R objects.

License Apache License 2.0

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 3.2.0)

Imports base64enc, httr, jsonlite, magrittr, openssl, rappdirs, withr, yaml, zip

Suggests knitr, rmarkdown, R6, testthat

VignetteBuilder knitr

URL https://github.com/rstudio/pins

BugReports https://github.com/rstudio/pins/issues

NeedsCompilation no

Author Javier Luraschi [aut, cre], RStudio [cph]

Repository CRAN

Date/Publication 2019-10-01 04:50:05 UTC

R topics documented:

`board_cache_path` *Retrieve Default Cache Path*

Description

Retrieves the default path used to cache boards and pins. Makes use of the `rappdirs` package to use cache folders defined by each OS.

Usage

```
board_cache_path()
```

Examples

```
# retrieve default cache path
board_cache_path()
```

`board_default` *Default Board*

Description

Retrieves the default board, which defaults to `"local"` but can also be configured with the `pins.board` option.

Usage

```
board_default()
```

Examples

```
library(pins)

# create temp board
board_register_local("temp", cache = tempfile())

# configure default board
options(pind.board = "temp")

# retrieve default board
board_default()
```

---

board_deregister *Deregister Board*

Description

Deregisters a board, useful to disable boards no longer in use.

Usage

```
board_deregister(name, ...)
```

Arguments

name	An optional name to identify this board, defaults to the board name.
...	Additional parameters required to deregister a particular board.

Examples

```
# create a new local board
board_register("local", "other_board", cache = tempfile())

# pin iris to new board
pin(iris, board = "other_board")

# deregister new board
board_deregister("other_board")
```

board_get *Get Board*

Description

Retrieves information about a particular board.

Usage

```
board_get(name)
```

Arguments

name	The name of the board to use

board_list *List Boards*

Description

Retrieves all available boards.

Usage

```
board_list()
```

board_pin_create *Custom Boards*

Description

Family of functions meant to be used to implement custom boards extensions, not to be used by users.

Usage

```
board_pin_create(board, path, name, metadata, ...)

board_initialize(board, ...)

board_pin_get(board, name, ...)

board_pin_remove(board, name, ...)

board_pin_find(board, text, ...)

board_browse(board, ...)
```

Arguments

board	The board to extend, retrieved with board_get().
path	The path to store as a pin.
name	The name of the pin.
metadata	A list of metadata associated with this pin.
...	Additional parameteres.
text	The text patteren to find a pin.

---

board_register *Register Board*

Description

Registers a board, useful to find resources with pin_find() or pin to additional boards with pin().

Usage

```
board_register(board, name = board, cache = board_cache_path(), ...)
```

Arguments

board	The name of the board to register.
name	An optional name to identify this board, defaults to the board name.
cache	The local folder to use as a cache, defaults to board_cache_path().
...	Additional parameters required to initialize a particular board.

Details

A board requires a local cache to avoid downloading files multiple times. It is recommended to not specify the cache parameter since it defaults to a well known rappdirs. However, you are welcome to specify any other location for this cache or even a temp folder with tempfile(). Notice that, when using a temp folder, pins will be cleared when your R session restarts. The cache parameter can be also set with the pins.path option.

See Also

board_register_local, board_register_github, board_register_kaggle, board_register_rsconnect and board_register_datatxt.

Examples

```
# create a new local board
board_register("local", "other_board", cache = tempfile())

# create a Website board
board_register("datatxt",
               name = "txtexample",
               url = "https://datatxt.org/data.txt",
               cache = tempfile())
```

`board_register_datatxt` *Register Data TXT Board*

Description

Wrapper with explicit parameters over `board_register()` to register as a board a website describing resources with a `data.txt` file.

Usage

```
board_register_datatxt(name, url, cache = board_cache_path())
```

Arguments

`name`	The name for this board, usually the domain name of the website.
`url`	Path to the `data.txt` file or path containing it.
`cache`	The local folder to use as a cache, defaults to `board_cache_path()`.

See Also

board_register

Examples

```
# register website board using datatxt file
board_register_datatxt(name = "txtexample",
                       url = "https://datatxt.org/data.txt",
                       cache = tempfile())

# find pins
pin_find(board = "txtexample")
```

board_register_github *Register GitHub Board*

Description

Wrapper with explicit parameters over `board_register()` to register a GitHub repo as a board.

Usage

```
board_register_github(name = "github", repo = NULL,
  branch = "master", token = NULL, path = "",
  cache = board_cache_path())
```

Arguments

`name`	Optional name for this board, defaults to 'github'.
`repo`	The GitHub repository formatted as 'owner/repo', can be `NULL` if the `GITHUB_PAT` environment variable is set.
`branch`	The branch to use when commiting pins.
`token`	Token to use when `GITHUB_PAT` is not specified.
`path`	The subdirectory in the repo where the pins will be stored.
`cache`	The local folder to use as a cache, defaults to `board_cache_path()`.

Details

This function requires a GitHub repo to be manually created; otherwise, registering a GitHub board will fail.

When a file upload exceeds 25MB, a GitHub release file will be used since they support up to 2GB file uploads. This threshold can be configured through the `pins.github.release` option which is specified in megabytes and defaults to 25.

See Also

board_register

Examples

```
## Not run:
# the following example requires a GitHub API key
board_register_github(repo = "owner/repo")

## End(Not run)
```

board_register_kaggle *Register Kaggle Board*

Description

Wrapper with explicit parameters over `board_register()` to register Kaggle as a board.

Usage

```
board_register_kaggle(name = "kaggle", token = NULL,
  overwrite = FALSE, cache = board_cache_path())
```

Arguments

`name`	Optional name for this board, defaults to 'kaggle'.
`token`	The Kaggle token as a path to the `kaggle.json` file, can be `NULL` if the `~/.kaggle/kaggle.json` file already exists.
`overwrite`	Should `~/.kaggle/kaggle.json` be overriden?
`cache`	The local folder to use as a cache, defaults to `board_cache_path()`.

See Also

board_register

Examples

```
## Not run:
# the following example requires a Kaggle API token
board_register_kaggle(token = "path/to/kaggle.json")

## End(Not run)
```

board_register_local *Register Local Board*

Description

Wrapper with explicit parameters over `board_register()` to register a local folder as a board.

Usage

```
board_register_local(name = "local", cache = board_cache_path())
```

Arguments

name	Optional name for this board, defaults to 'local'.
cache	The local folder to use as a cache, defaults to board_cache_path().

See Also

board_register

Examples

```
# register local board using a temp folder
board_register_local(cache = tempfile())
```

---

board_register_rsconnect *Register RStudio Connect Board*

Description

Wrapper with explicit parameters over board_register() to register RStudio Connecet as a board.

Usage

```
board_register_rsconnect(name = "rsconnect", server = NULL,
  account = NULL, key = NULL, output_files = FALSE,
  cache = board_cache_path())
```

Arguments

name	Optional name for this board, defaults to 'rsconnect'.
server	Optional address to RStudio Connect server.
account	Optional account name to use with RStudio Connect.
key	The RStudio Connect API key.
output_files	Should the output in an automated report create output files?
cache	The local folder to use as a cache, defaults to board_cache_path().

See Also

board_register

Examples

```
## Not run:
# the following examples require an RStudio Connect API key

# register from rstudio
board_register_rsconnect()

# register from rstudio with multiple servers
board_register_rsconnect(server = "https://rstudio-connect-server")

# register from rstudio with multiple account
board_register_rsconnect(account = "account-name")

# register automated report for rstudio connect
board_register_rsconnect(key = Sys.getenv("RSTUDIO_KEY"),
                         server = "https://rstudio-connect-server")

## End(Not run)
```

`pin` *Pin Resource*

Description

Pins the given resource locally or to the given board.

Usage

```
pin(x, name = NULL, description = NULL, board = NULL, ...)
```

Arguments

`x`	An object, local file or remote URL to pin.
`name`	The name for the dataset or object.
`description`	Optional description for this pin.
`board`	The board where this pin will be placed.
`...`	Additional parameters.

Details

`pin()` allows you to cache remote resources and intermediate results with ease. When caching remote resources, usually URLs, it will check for HTTP caching headers to avoid re-downloading when the remote result has not changed.

This makes it ideal to support reproducible research by requiring manual instruction to download resources before running your R script.

In addition, `pin()` still works when working offline or when the remote resource becomes unavailable; when this happens, a warning will be triggered but your code will continue to work.

Examples

```
library(pins)

# define local board
board_register_local(cache = tempfile())

# cache the mtcars dataset
pin(mtcars)

# cache computation oveer mtcars
mtcars[mtcars$mpg > 30,] %>%
  pin(name = "mtefficient")

# retrieve cached pin
pin_get("mtefficient")

# url to remote resource
resource <- file.path("https://raw.githubusercontent.com/facebook/prophet",
                      "master/examples/example_retail_sales.csv")

# cache remote resource
pin(resource, name = "example_retail_sales")

# load cached csv
pin_get("example_retail_sales") %>% read.csv()

# cache and read csv
read.csv(pin(resource))
```

---

pin_find *Find Pin*

Description

Find a pin in any board registered using `board_register()`.

Usage

```
pin_find(text = NULL, board = NULL, ...)
```

Arguments

`text`	The text to find in the pin description or name.
`board`	The board name used to find the pin.
`...`	Additional parameters.

Details

`pin_find()` allows you to discover new resources or retrieve pins you've previously created with `pin()`.

The `pins` package comes with a CRAN packages board which allows searching all CRAN packages; however, you can add additional boards to search from like Kaggle, Github and RStudio Connect.

For 'local' and 'packages' boards, the 'text' parameter searches the title and description of a pin using a regular expression. Other boards search in different ways, most of them are just partial matches, please refer to their documentation to understand how other boards search for pins.

Once you find a pin, you can retrieve with `pin_get("pin-name")`.

Examples

```
library(pins)

# retrieve pins
pin_find()

# search pins related to 'cars'
pin_find("cars")

# search pins related to 'seattle' in the 'packages' board
pin_find("seattle", board = "packages")

# search pins related to 'london' in the 'packages' board
pin_find("london", board = "packages")


# retrieve 'hpiR/seattle_sales' pin
pin_get("hpiR/seattle_sales")

# retrieve 'bsamGP/London.Mortality' pin
pin_get("bsamGP/London.Mortality")
```

`pin_get` *Retrieve Pin*

Description

Retrieves a pin by name from the local or given board.

Usage

```
pin_get(name, board = NULL, cache = TRUE, ...)
```

Arguments

name	The name of the pin.
board	The board where this pin will be retrieved from.
cache	Should the pin cache be used? Defaults to TRUE.
...	Additional parameters.

Details

pin_get() retrieves a pin by name and, by default, from the local board. You can use the board parameter to specify which board to retrieve a pin from. If a board is not specified, it will use pin_find() to find the pin across all boards and retrieve the one that matches by name.

Examples

```
library(pins)

# define local board
board_register_local(cache = tempfile())

# cache the mtcars dataset
pin(mtcars)

# retrieve the mtcars pin
pin_get("mtcars")

# retrieve mtcars pin from packages board
pin_get("easyalluvial/mtcars2", board = "packages")
```

---

pin_reactive *Reactive Pin*

Description

Creates a pin that reacts to changes in the given board by polling pin_get(), useful when used from the shiny package.

Usage

```
pin_reactive(name, board, interval = 5000, session = NULL)
```

Arguments

name	The name of the pin.
board	The board where this pin will be retrieved from.
interval	Approximate number of milliseconds to wait to retrieve updated pin. This can be a numeric value, or a function that returns a numeric value.

session	The user session to associate this file reader with, or NULL if none. If non-null, the reader will automatically stop when the session ends.

---

pin_remove *Remove Pin*

Description

Unpins the given named pin from the given board.

Usage

```
pin_remove(name, board)
```

Arguments

name	The name for the pin.
board	The board from where this pin will be removed.

Details

Notice that some boards do not support deleting pins, this is the case for the Kaggle board. For these boards, you would manually have to remote resources using the tools the board provides.

Examples

```
library(pins)

# define local board
board_register_local(cache = tempfile())

# create mtcars pin
pin(mtcars)

# remove mtcars pin
pin_remove(mtcars, board = "local")
```

Index