Package ‘settingsSync’

July 23, 2025

Type Package

Title 'Rstudio' Addin to Sync Settings and Keymaps

Version 3.0.2

Description Provides a 'Rstudio' addin to download, merge and upload 'Rstudio' settings and keymaps, essentially 'syncing them' at will. It uses 'Google Drive' as a cloud storage to keep the settings and keymaps files.

License Apache License (== 2)

URL https://github.com/notPlancha/settingsSync

BugReports https://github.com/notPlancha/settingsSync/issues

Depends R (>= 4.2)

Imports cli, dplyr, glue, googledrive, jsonlite, rappdirs, tibble, yesno

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.1

Suggests spelling

Language en-US

Author André Plancha [aut, cre]

Maintainer André Plancha <andre.plancha@hotmail.com>

Repository CRAN

Date/Publication 2024-06-14 07:40:02 UTC

Contents

`mimic-on-off` *Simulate the functions before truly running them*

Description

These functions are used to simulate the main functions before truly running them, to disable changes to the files and to the cloud.

Usage

```
mimic_on()

mimic_off()

is_mimic_on()
```

Details

These are used in most examples, so the user settings don’t change when running [utils::example()]. Internally, these change the options ‘ss.mimic‘, ‘ss.mimic.local‘ and ‘ss.mimic.cloud‘.

Value

nothing

Functions

- `mimic_on()`: Enable mimic and write mimic files
- `mimic_off()`: Disable mimic
- `is_mimic_on()`: Check if mimic is on

Examples

```
mimic_on()
if(interactive()) {
  sync()
}
mimic_off()
```

`pull` *Pull Rstudio Settings*

Description

Pulls Rstudio settings from Google Drive, without pushing. This is just a helper function for [sync()], the main function, but can be used alone, although this function will not do any checking, and will just override.

Usage

```
pull(
  all = FALSE,
  addins_gd = FALSE,
  editor_bindings_gd = FALSE,
  rstudio_bindings_gd = FALSE
)
```

Arguments

`all` boolean, if TRUE will pull all settings from gd, overwriting them. This param overrides the values of the other params. Default is FALSE.

`addins_gd`, `editor_bindings_gd`, `rstudio_bindings_gd`
character or boolean, the json string to be written to the respective file or TRUE/FALSE; if TRUE, will read from Google Drive; if FALSE to not write that specific file. Default is FALSE

Value

nothing

See Also

[read_from_gd()], [sync()], [push()]

Examples

```
mimic_on()
  pull()          # does nothing
  pull(all = TRUE) # will pull all settings from gd, overwriting them
  pull(addins_gd = '{"insertPipeOperator": "Shift+Tab"}',)
  # will write to addins.json the string
  pull(
    addins_gd = '{"insertPipeOperator": "Shift+Tab"}',
    editor_bindings_gd= TRUE
  )
  # will write to addins.json the string and pull editor_bindings from gd
mimic_off()
```

push *Push Rstudio Settings*

Description

Pushes Rstudio settings to Google Drive, without pulling. This is just a helper function for [sync()], the main function, but can be used alone, although this function will not do any checking, and will just override. Because of the way [googledrive::drive_put] is built, this function reads from the files directly.

Usage

```
push(
  do_all = FALSE,
  do_addins = FALSE,
  do_editor_bindings = FALSE,
  do_rstudio_bindings = FALSE,
  progBar = NULL
)
```

Arguments

`do_all`	boolean, if TRUE will push all settings to gd, overwriting them. This param overrides the values of the other params. Default is FALSE.
`do_addins, do_editor_bindings, do_rstudio_bindings`	booleans, if TRUE will push the respective file. Default is FALSE
`progBar`	function, designed to work with [progress_bar()]. Runs after each file is pushed.

Value

nothing

Examples

```
mimic_on()
  push()             # does nothing
  push(do_all = TRUE) # will push all settings to gd, overwriting them
  push(do_editor_bindings = FALSE, do_rstudio_bindings = FALSE)
  # will push only editor and rstudio bindings
mimic_off()
```

read_from_gd *Read from Google Drive a JSON file inside rstudio folder*

Description

This is a helper function for [pull()] and [sync()] that: * reads a json file from Google Drive inside the rstudio folder * converts it to a data frame (to resolve conflicts in the future)

Usage

```
read_from_gd(what, progBar = NULL)
```

Arguments

what	One of c("rstudio_bindings", "editor_bindings", "addins"), the file to read
progBar	function, designed to work with [progress_bar()]. Runs after the file is read.

Value

A data frame with the contents of the file, converted from json

See Also

[read_from_local()] [sync()] [jsonlite::fromJSON()] [googledrive::drive_read_string()]

Examples

```
mimic_on()
  read_from_gd("rstudio_bindings")
  read_from_gd("editor_bindings")
  read_from_gd("addins")
mimic_off()
```

read_from_local *Read from local a JSON file inside rstudio folder*

Description

This is a helper function for [sync()] that: * reads a json file from the local rstudio folder * converts it to a data frame (to resolve conflicts in the future)

Usage

```
read_from_local(what)
```

Arguments

what	One of c("rstudio_bindings", "editor_bindings", "addins"), the file to read

Value

A data frame with the contents of the file, converted from json

See Also

[read_from_gd()], [push()], [sync()], [jsonlite::read_json()]

Examples

```
mimic_on()
  read_from_local("rstudio_bindings")
  read_from_local("editor_bindings")
  read_from_local("addins")
mimic_off()
```

rstudio_path *Get rstudio local path*

Description

Get rstudio local path

Usage

```
rstudio_path()
```

Value

path to rstudio local

See Also

[rappdirs::user_config_dir()], [is_windows()], [usethis::is_windows()]

Examples

```
rstudio_path()
```

sync *Sync Rstudio Settings*

Description

Gets the settings from Google Drive and from the local files, and merges them. If there are conflicts, will ask the user to resolve them. Finally, will write the merged settings to the local files, and push them to Google Drive. Will first ask for confirmation if interactive. This function is what's called by the addin.

Usage

```
sync(write = NULL, useProgBar = TRUE)
```

Arguments

`write`	boolean, if TRUE will write the merged settings to the local files, and push them to Google Drive. FALSE essentially just makes conflict resolution, without changing any files (basically a dry run). If a value other than NULL is provided, this will skip confirmation.
`useProgBar`	boolean, if TRUE will show a progress bar. Default is TRUE.

Details

NOTE: if it's not interactive, it won't write to files because of CRAN policies.

Value

nothing

See Also

[push()], [pull()]

Examples

```
mimic_on()
if(interactive()) {
  sync(write = TRUE)   # will immediately try to sync all settings
  sync(write = FALSE)  # dry run, will not write to files or push to gd
  sync()               # will ask for confirmation, then sync all settings
}
mimic_off()
```

Index