

Package ‘languageserversetup’

October 13, 2022

Type Package

Title Automated Setup and Auto Run for R Language Server

Version 0.1.2

Maintainer Jozef Hajnala <jozef.hajnala@gmail.com>

Description Allows to install the R 'languageserver' with all dependencies into a separate library and use that independent installation automatically when R is instantiated as a language server process. Useful for making language server seamless to use without running into package version conflicts.

License AGPL-3

Suggests tinytest

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

SystemRequirements procps (linux), ps (macos), wmic (windows)

URL <https://github.com/jozefhajnala/languageserversetup>

BugReports <https://github.com/jozefhajnala/languageserversetup/issues>

Language en-US

NeedsCompilation no

Author Jozef Hajnala [aut, cre]

Repository CRAN

Date/Publication 2020-04-10 09:20:02 UTC

R topics documented:

languageserver_add_to_rprofile	2
languageserver_detect	2
languageserver_install	3
languageserver_remove_from_rprofile	4
languageserver_startup	5

Index**6**

languageserver_add_to_rprofile
Add language server initialization to .Rprofile

Description

Add language server initialization to .Rprofile

Usage

```
languageserver_add_to_rprofile(rlsLib = getOption("langserver_library"),
  rprofilePath = locate_rprofile(), confirmBeforeChanging = TRUE,
  code = append_code(rlsLib = rlsLib))
```

Arguments

rlsLib	character(1), path to the library where the languageserver package will be installed.
rprofilePath	character(1), path to the file where to add the initialization code, or NULL. By default, adds the code to a .Rprofile file in the home directory of the current user. Please refer to ?Startup for more details around .Rprofile files. Notably, if the R_PROFILE_USER environment variable is set, the .Rprofile located in the home directory is ignored, therefore we may want to place the initialization code into the file specified by that variable using the rprofilePath argument in that case.
confirmBeforeChanging	logical(1), if TRUE, asks for user confirmation before changing the file. For non-interactive use, FALSE will skip the confirmation.
code	character(), the code to be added to the file. Defaults to the value of append_code().

Value

side-effects

languageserver_detect *Detect whether a process relates to the R Language Server*

Description

Detect whether a process relates to the R Language Server

Usage

```
languageserver_detect(pid, os, langServerProcessPatt, checkParents = TRUE)
```

Arguments

pid	integer(1), id of the process to investigate, usually retrieved by <code>Sys.getpid</code>
os	character(1), name of the OS, usually retrieved as the "sysname" element of <code>Sys.info</code> , all lowercase.
langServerProcessPatt	character(1), pattern to recognize the process created by languageserver.
checkParents	logical(1), if TRUE, parent processes are also checked in case when pid is not the R Language Server process. This is needed as the linting processes are created with callr as sub-processes of the main Language Server process.

Value

logical(1), TRUE if the process with pid (or, optionally, any of its parents) is detected as the R Language Server process. Otherwise FALSE.

languageserver_install

Install the languageserver package to a separate library

Description

Install the languageserver package to a separate library

Usage

```
languageserver_install(rlsLib = getOption("langserver_library"),
  strictLibrary = TRUE, fullReinstall = TRUE, fromGitHub = TRUE,
  confirmBeforeInstall = TRUE, dryRun = FALSE, ...)
```

Arguments

rlsLib	character(1), path to the library where the languageserver package will be installed.
strictLibrary	logical(1), if TRUE, all the dependencies of languageserver will be installed into rlsLib, otherwise only those that are needed but not present in other libraries in <code>.libPaths()</code> will be installed.
fullReinstall	logical(1). If TRUE, rlsLib will be recursively removed to re-install all the packages cleanly.
fromGitHub	logical(1), if TRUE, will use <code>install-github.me</code> to install the current development version from GitHub. Otherwise <code>install.packages()</code> is used to install the languageserver package from CRAN.
confirmBeforeInstall	logical(1) if TRUE, will ask the user to confirm the steps before installation. For non-interactive use, FALSE will skip the confirmation.

`dryRun` logical(1), if TRUE, most actions will only be reported, not taken - nothing will be removed, created or installed.

... further arguments passed to `install.packages()` in case `fromGitHub` is set to FALSE.

Value

side-effects

See Also

[utils::install.packages\(\)](#)

`languageserver_remove_from_rprofile`

Remove language server initialization from .Rprofile

Description

Remove language server initialization from `.Rprofile`

Usage

```
languageserver_remove_from_rprofile(rlsLib = getOption("langserver_library"),
  rprofilePath = locate_rprofile(), code = append_code(rlsLib = rlsLib),
  confirmBeforeChanging = TRUE)
```

Arguments

`rlsLib` character(1), path to the library where the `languageserver` package will be installed.

`rprofilePath` character(1), path to the file where to add the initialization code, or NULL. By default, adds the code to a `.Rprofile` file in the home directory of the current user. Please refer to `?Startup` for more details around `.Rprofile` files. Notably, if the `R_PROFILE_USER` environment variable is set, the `.Rprofile` located in the home directory is ignored, therefore we may want to place the initialization code into the file specified by that variable using the `rprofilePath` argument in that case.

`code` character(), the code to be added to the file. Defaults to the value of `append_code()`.

`confirmBeforeChanging` logical(1), if TRUE, asks for user confirmation before changing the file. For non-interactive use, FALSE will skip the confirmation.

Value

side-effects

`languageserver_startup`*Prepare language server to be started*

Description

Prepare language server to be started

Usage

```
languageserver_startup(rlsLib = getOption("langserver_library"),
  langServerProcessPatt = getOption("langserver_processPatt"),
  strictLibrary = TRUE, os = tolower(Sys.info()[["sysname"]]),
  pid = Sys.getpid())
```

Arguments

<code>rlsLib</code>	character(1), path to the library where the languageserver package will be installed.
<code>langServerProcessPatt</code>	character(1), pattern to recognize the process created by languageserver.
<code>strictLibrary</code>	logical(1), if TRUE, all the dependencies of languageserver will be installed into <code>rlsLib</code> , otherwise only those that are needed but not present in other libraries in <code>.libPaths()</code> will be installed.
<code>os</code>	character(1), name of the OS, usually retrieved as the "sysname" element of <code>Sys.info</code> , all lowercase.
<code>pid</code>	integer(1), id of the process to investigate, usually retrieved by <code>Sys.getpid</code>

Value

side-effects

Index

languageserver_add_to_rprofile, [2](#)
languageserver_detect, [2](#)
languageserver_install, [3](#)
languageserver_remove_from_rprofile, [4](#)
languageserver_startup, [5](#)

utils::install.packages(), [4](#)