Package: rprofile (via r-universe)

August 10, 2024

Title Load Both User-Global and Project-Specific R Profile Configurations

Version 0.4.0.9000

Description Use rprofile::load() inside a project '.Rprofile' file to ensure that the user-global '.Rprofile' is loaded correctly regardless of its location, and that other common resources (in particular 'renv') are also set up correctly.

BugReports https://github.com/klmr/rprofile/issues

Depends R (>= 4.1)

Imports utils

Enhances pkgload, renv

License MIT + file LICENSE

Encoding UTF-8

Repository https://klmr.r-universe.dev

RemoteUrl https://github.com/klmr/rprofile

RemoteRef main

RemoteSha 56bfb3b41d8fd66fb9b15f3978c85b483b36cea9

Contents

load *Initialize an R session by loading the all relevant R profile configurations*

Description

`rprofile::load()` attempts to load the global R profile configuration and any other common project configuration.

Usage

```
rprofile::load(..., isolate = FALSE, renv = TRUE, dotenv = TRUE, dev = TRUE)
```

Arguments

`...`	ignored; forces named argument passing
`isolate`	[`logical(1)`] whether to isolate the project from the global configuration (default: `FALSE`)
`renv`	[`logical(1)`] whether to activate an **renv**, if present (default: `TRUE`)
`dotenv`	[`logical(1)`] whether to load environment variables defined in a local '`.env`' file (default: `TRUE`)
`dev`	[`logical(1)|call`] whether to run `pkgload::load_all()` if the current project is a development package (default: `TRUE`; see **Details**)

Details

This function should be the first thing that gets executed inside a project '`.Rprofile`' file, and it should usually be written exactly as follows: `try(rprofile::load())` (the `try()` is present to ensure that `R CMD` can still run in the current directory when **rprofile** is not installed).

Unless `isolate = TRUE` is set, the user R profile configuration is preferentially looked up in the `R_PROFILE_USER` environment variable. If that is unset, it is instead loaded from '`~/.Rprofile`'. It is loaded (mostly) *as-if* its code was directly copied into the project '`.Rprofile`' file. By contrast, if `isolate = TRUE` is set, no attempt to load any further '`.Rprofile`' files is made.

`rprofile::load()` will by default also activate the **renv** associated with the current project, if any, and will also load environment variables defined in a local '`.env`' file. These two actions will happen *before* the user profile is loaded. See the **Note** below.

Lastly, `rprofile::load()` will check if the code is being run from an interactive session. If so, and if the project contains a '`DESCRIPTION`' file, **rprofile** will attempt to load **pkgload** and then execute `pkgload::load_all(export_all = FALSE)`. To avoid disrupting the regular package load order, this action will be deferred until after all default packages (given by `getOption('defaultPackages')`) have been loaded and attached. Users can customize which code should be run by passing an unevaluated expression (instead of `TRUE`) in the `dev` argument. Since this code will be evaluated *after* the remaining '`.Rprofile`' code has been run, the argument may refer to functions defined afterwards (see **Examples**).

Value

`rprofile::load()` will invisibly return whether loading the user R profile file succeeded: in case of an error, it returns `FALSE` and converts the error into a warning.

Note

You need to ensure that **renv** is not loaded redundantly in your '`.Rprofile`' file. In other words, please make sure that the line `source("renv/activate.R")`, which **renv** adds automatically, is *not* present in the file. `rprofile::load()` prevents **renv** from subsequently adding this line to the project '`.Rprofile`' file.

Examples

```
# Each option is configurable; in the extreme case, the function does nothing:
rprofile::load(isolate = TRUE, renv = FALSE, dotenv = FALSE, dev = FALSE)

## Not run:
# In general, the following code should be the first line of a project's
# `.Rprofile` file (using `try()` ensures that R code can still run in the
# current directory even if 'rprofile' is not installed — this is important in
# particular to allow `R CMD CHECK` to run successfully on CI/CD without having
# to first install 'rprofile'):
try(rprofile::load())

## End(Not run)

## Not run:
# We can customize how to load development packages, .e.g.:
try(rprofile::load(dev = quote(reload())))

reload = function () {
  devtools::document()
  devtools::load_all(quiet = TRUE)
}

## End(Not run)
```

Index