Package: envDocument (via r-universe)

October 30, 2024

Type Package

Title Document the R Working Environment

Version 2.4.1.9002

Date 2019-08-29

Description Prints out information about the R working environment
(system, R version,loaded and attached packages and versions)
from a single function ``env_doc()''. Optionally adds
information on git repository, tags, commits and remotes (if
available).

License GPL (>= 3.0)

URL https://github.com/dgJacks0n/envDocument

Suggests knitr(>= 1.13), stringr(>= 1.2.0), testthat(>= 1.0.2)

RoxygenNote 6.1.1

VignetteBuilder knitr

Imports git2r(>= 0.22.1), utils, methods

Repository https://dgjacks0n.r-universe.dev

RemoteUrl https://github.com/dgjacks0n/envdocument

RemoteRef HEAD

RemoteSha b4e5f546e076b7cbcf7edb0bfa4c6e68f69bc023

Contents

---

envDocument-package *Document the environment under which an analysis was performed*

Description

Provides a function env_doc that collects information on the script, environment and system that performed an analysis.

Details

Returns a data frame with information on:

- R version

- Attacheded packages and version

- OS, version and user

- Path and modification time of the calling script

- Git commit and tag info (optional, requires git2r)

Results are collected in a standardized data frame that can be queried or pretty-printed using kable or similar methods.

Author(s)

Maintainer: Donald Jackson <donald.jackson@bms.com>

See Also

Useful links:

- https://github.com/dgJacks0n/envDocument

`env_doc` *Report the working environment in which an analysis was performed.*

Description

env_doc prints or returns a data frame with the following information: - System version (OS, version, user, working directory)

Usage

```
env_doc(output = c("return", "print", "table"), system = TRUE,
  version = TRUE, packages = TRUE, script = TRUE, git = TRUE,
  domino = c("auto", "on", "off"))
```

Arguments

`output` How should output be handled? Character, values: return - return as a data frame (default); print - print to stdout; table - pretty-print using knitr::kable (requires package knitr)

`system` Include OS info from `getSysInfo`? Boolean, default TRUE

`version` Include R version from `getRversion`? Boolean, default TRUE

`packages` Include attached packages with repository and version from `getPackageInfo`? Boolean, default TRUE

`script` Include script path and modification time from `getScriptInfo`? Boolean, default TRUE

`git` Include git repository information from `getGitInfo` (note: requires `git2r`)? Boolean, default TRUE

`domino` Include Domino Datalab run information from `getDominoInfo`? Character, values: auto - include if available; on - include with warning if not available; off - do not include evn if available

Details

- R version

- Package names and versions

- Top-level script name and modification time

- Git hash, status and tag (if any; requires package git2r)

Value

If output = return (default): A data frame with columns for information type, variable name and value. NULL for output = print or output = table

Examples

```
env_doc("print") # print information to stdout
info <- env_doc() # return information as a consolidated data frame
```

---

getDominoInfo *Get information about a Domino Datalab run from environment variables in the run environment*

Description

Get information about a Domino Datalab run from environment variables in the run environment

Usage

```
getDominoInfo(drop_vars = c("DOMINO_API_HOST",
  "DOMINO_EXECUTOR_HOSTNAME", "DOMINO_USER_API_KEY"))
```

Arguments

drop_vars	Variables that will not be included in results. Character vector; see https://support.dominodatalab.com/hc/en-us/articles/205536355-Domino-environment-variables for a list of variables.

Value

data frame of Domino environment variables (renamed) and their values

---

getGitInfo *Get git repository information for a script*

Description

getGitInfo locates the git repository for the calling script (if any) and retrieves relevant informaiton such as last commit, status (e.g. modified since last commit) and tag (if any).

Usage

```
getGitInfo(scriptpath = NA)
```

Arguments

scriptpath	Path to script (optional, defaults to calling script from get_scriptPath())

Details

Requires that git2r be installed.

Examples

```
git_info <- getGitInfo()
```

getPackageInfo *Get information on attached packages*

Description

getPackageInfo returns information on attached packages including name, version and source.

Usage

```
getPackageInfo()
```

Examples

```
packages <- getPackageInfo()
```

getRepo *Locate the git repository where a file is tracked*

Description

getRepo looks the up the repository for a file and makes sure path is tracked in repo. Repositories are searched from the file directory up using discover_repository

Usage

```
getRepo(testPath = NA)
```

Arguments

testPath	Path to a file to find/check repo

getScriptInfo *Get information about the calling script*

Description

`getScriptInfo` retrieves the file path and modification time for the calling script.

Usage

```
getScriptInfo()
```

Details

Note: Currently this works if the script was called via source(), knitr::spin(), Rstudio's _compile notebook_ (ctrl-shift-K), Rscript or R -f.

Examples

```
script_info <- getScriptInfo()
```

getScriptPath *Get the path of the calling script*

Description

`getScriptPath` returns the full path of the script that called this function (if any) or NULL if path is not available

Usage

```
getScriptPath(absolute = TRUE)
```

Arguments

`absolute`	Return absolute path (absolute = TRUE, default) or path relative to working directory (FALSE)

Examples

```
## Not run:
mypath <- getScriptPath()

## End(Not run)
```

`getSysInfo` *Get system information.*

Description

`getSysInfo` returns the OS, hostname, userid and working directory for the current analysis as a data frame

Usage

```
getSysInfo()
```

Examples

```
getSysInfo()
```

`get_Rversion` *Get information on R version*

Description

Get information on R version

Usage

```
get_Rversion()
```

Examples

```
Rversion <- getRversion()
```

`infoNotFound` *Return a consistent failure result when something is not available*

Description

Return a consistent failure result when something is not available

Usage

```
infoNotFound()
```

Index