

Package: dvs (via r-universe)

November 21, 2024

Title Large/Sensitive File Linker for Git Repositories

Version 0.0.1.9000

Description This package allows users to version large or sensitive files under Git without tracking the file directly.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/rextendr/version 0.3.1

Imports rlang

Suggests testthat (>= 3.0.0), fs, jsonlite, withr, stringr, yaml, processx

Config/testthat/edition 3

Repository <https://a2-ai.r-universe.dev>

RemoteUrl <https://github.com/a2-ai/dvs>

RemoteRef HEAD

RemoteSha 0806991658b8fbc621f89e101ed10a6e703f9555

Contents

dvs_add	2
dvs_get	3
dvs_init	4
dvs_status	4

Index	6
--------------	----------

dvs_add	<i>copy files to the storage directory</i>
---------	--

Description

copy files to the storage directory

Usage

```
dvs_add(files, message = NULL, split_output = FALSE)
```

Arguments

files	file paths or glob patterns to add to the storage directory
message	optional: a message associated with the file(s) for versioning context to appear in dvs metadata files
split_output	optional: when TRUE, a list of two data frames is returned: successes and failures. <ul style="list-style-type: none"> • rows in successes are successfully added file inputs • rows in failures are inputs that returned errors when FALSE, the output is a single data frame whose rows are the files attempted to add in the given operation, the successes or failures of which are indicated in the outcome column

Details

after initializing the storage directory with [dvs_init](#), this command copies files to storage directory for other collaborators to version and retrieve with [dvs_get](#) If an explicitly inputted file doesn't exist, the command returns an error .

For any other error retrieving a particular file, the function itself will indicate the error type and message in the data frame output; the function itself will not return an error.

Value

one or two data frames whose rows are the files attempted to add in the given operation.

Examples

```
## Not run:
# would add all csv files in data/derived to the initialized storage directory
dvs_add("data/derived/*.csv")

# would add model/nonmem/1001/1001.ext and all files in data/derived
# to the initialized storage directory (excluding dvs metadata and .gitignore files)
dvs_add(c("model/nonmem/1001/1001.ext", "data/derived/*"))

## End(Not run)
```

dvs_get	<i>get or update files if they are in shared storage to add to the local repo</i>
---------	---

Description

get or update files if they are in shared storage to add to the local repo

Usage

```
dvs_get(files, split_output = FALSE)
```

Arguments

files	file paths or glob patterns to get from the storage directory
split_output	when <code>split_output</code> is true, a list of two data frames - successes and failures - is returned. Rows in successes are files successfully retrieved, and rows in failures are inputs that returned errors. When <code>split_output</code> is false, the output is a single data frame with all files attempted to recopy to the project directory, and whose success or failure is indicated as such in the outcome column.

Details

retrieves files previously added with [dvs_add](#) to the storage directory (initialized by [dvs_init](#)). If a file is explicitly inputted without a corresponding dvs metadata file i.e. hasn't been added yet with [dvs_add](#), the command returns an error.

For any other error retrieving a particular file, the function itself will indicate the error type and message in the data frame output; the function itself will not return an error.

Value

one or two data frames whose rows are the files attempted to get in the given operation.

Examples

```
## Not run:  
# would get all previously added files in data/derived from  
# the initialized storage directory  
dvs_get("data/derived/*.dvs")  
  
# would get all files in data/derived (excluding  
# .gitignore files) and ~Projects/project_x/large_file.pdf  
# from the initialized storage directory  
dvs_get(c("data/derived/*", "~Projects/project_x/large_file.pdf"))  
  
## End(Not run)
```

dvs_init	<i>initialize devious to add/get versioned files from the storage directory</i>
----------	---

Description

initialize devious to add/get versioned files from the storage directory

Usage

```
dvs_init(storage_directory, permissions = NULL, group = NULL)
```

Arguments

storage_directory	path to the desired storage directory for versioned files
permissions	optional: linux file permissions to set for files added to the storage directory (in octal format)
group	optional: primary group to set for files added to the storage directory

Value

A data frame with the storage directory, permissions, and group

Examples

```
## Not run:
# would initialize the project's storage directory at /data/project_x
dvs_init("/data/project_x")

# would initialize the project's storage directory at /data/project_x and
# configure the linux permissions "777" and primary group "project_x_group" for all
# files added to the storage directory
dvs_init("/data/project_x", 777, "project_x_group")

## End(Not run)
```

dvs_status	<i>status report for added files</i>
------------	--------------------------------------

Description

status report for added files

Usage

```
dvs_status(files = c(""), split_output = FALSE)
```

Arguments

files	optional: when specified, returns data frame with only these specified file paths or glob patterns.
split_output	when split_output is true, a list of two data frames - successes and failures - is returned. Rows in successes are files with successfully curated statuses, and rows in failures are inputs that returned errors. When split_output is false, the output is a single data frame with all attempted file status reports, and whose success or failure is indicated as such in the outcome column.

Details

gives the statuses of previously added files (current, unsynced, absent, or error) to make users aware if files stored in the storage directory don't exist in their local repository or have been updated. If no file paths or glob patterns are inputted, dvs_status gives the status of all previously added files. If there an error getting the status of a particular file, the function itself will not return an error, rather, a given error will be indicated in the data frame output.

Value

a data frame with the statuses of previously added files

Examples

```
## Not run:  
# would give the status of all previously added files  
dvs_status()  
  
# would attempt to get the status of all files in data/derived  
# (except for .gitignore files)  
dvs_status("data/derived/*")  
  
## End(Not run)
```

Index

dvs_add, 2, 3
dvs_get, 2, 3
dvs_init, 2, 3, 4
dvs_status, 4