

Package ‘ledger’

July 22, 2025

Type Package

Title Utilities for Importing Data from Plain Text Accounting Files

Version 2.0.11

URL <https://github.com/trevorld/r-ledger>,
<https://trevorldavis.com/R/ledger/>

BugReports <https://github.com/trevorld/r-ledger/issues>

Description Utilities for querying plain text accounting files from 'Ledger', 'HLedger', and 'Beancount'.

Imports dplyr (>= 0.7.0), rlang, stringr, tidyr (>= 0.7.0), tibble,
tidyselect (>= 1.2.0), tools

Suggests rio, testthat

SystemRequirements ledger (>= 3.1), hledger (>= 1.2), beancount (>=
2.0)

License MIT + file LICENSE

RoxygenNote 7.3.1

Encoding UTF-8

NeedsCompilation no

Author Trevor L. Davis [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6341-4639>>),
Jenya Sovetkin [ctb],
Chris Lloyd [ctb]

Maintainer Trevor L. Davis <trevor.l.davis@gmail.com>

Repository CRAN

Date/Publication 2024-05-20 12:20:05 UTC

Contents

default_toolchain	2
net_worth	2
prune_coa	3
register	4

default_toolchain	<i>Determine default tool chain used for reading in register</i>
-------------------	--

Description

default_toolchain determines default tool chain used for reading in register.

Usage

default_toolchain(file)

Arguments

file	Filename for a ledger, hledger, or beancount file.
------	--

net_worth	<i>Compute net worth</i>
-----------	--------------------------

Description

Computes net worth for a vector of dates. Computes net worth at the beginning of the day before any transactions have occurred.

Usage

```
net_worth(  
  file,  
  date = Sys.Date() + 1,  
  include = c("^asset", "^liabilit", "^<revalued>"),  
  exclude = NULL,  
  flags = "-V",  
  toolchain = default_toolchain(file),  
  ignore_case = TRUE  
)
```

Arguments

file	Filename for a ledger, hledger, or beancount file.
date	Vector of dates to compute net worth for. For each only the transactions (and price statements) before that date are used in the net worth calculation.
include	Character vector of regular expressions of accounts to include in the net worth calculation. Use "."* to include everything.
exclude	Character vector of regular expressions of accounts to exclude from the net worth calculation. Use NULL to exclude nothing.

flags	Extra flags to pass to register. If using ledger may want to try something like "-X USD".
toolchain	Toolchain used to read in register. Either "ledger", "hledger", "bean-report_ledger", or "bean-report_hledger".
ignore_case	logical value of whether to ignore case in regular expressions or not.

Value

net_worth returns a tibble

Examples

```
## Not run:
example_beancount_file <- system.file("extdata", "example.beancount", package = "ledger")
net_worth(example_beancount_file)
net_worth(example_beancount_file, c("2016-01-01", "2017-01-01", "2018-01-01"))

## End(Not run)
```

prune_coa	<i>Prune plaintext "Chart of Accounts" names to a given maximum depth</i>
-----------	---

Description

prune_coa is a convenience function that modifies a data frame by either editing in place or making a new variable containing the plaintext "Chart of Accounts" pruned to a given maximum depth e.g. "Assets:Checking:Credit-Union:Account1" at a maximum depth of 2 will be converted to "Assets:Checking". prune_coa uses tidyverse non-standard evaluation (NSE). prune_coa_string is a convenience function which does the pruning operation on character vectors.

Usage

```
prune_coa(df, depth = 1, variable, name)

prune_coa_string(x, depth = 1)
```

Arguments

df	A data frame
depth	How deep should the account structure be.
variable	Which variable to make less deep (default is to use "account")
name	New variable name (default is to edit the variable argument in place)
x	Character vector

Examples

```
df <- tibble::tribble(~account, ~amount,
  "Assets:Checking:BankA", 1000,
  "Assets:Checking:BankB", 1000,
  "Assets:Savings:BankA", 1000,
  "Assets:Savings:BankC", 1000)

prune_coa(df)
prune_coa(df, 2)
prune_coa(df, 3)
prune_coa(df, 4)
prune_coa(df, 2, account, account2)
prune_coa(prune_coa(df, 2, account, account2), 3, account2, account3)
prune_coa_string(df$account, 2)
```

register

Import a ledger, hledger, or beancount register

Description

register imports the register from a ledger, hledger, or beancount file as a tibble.

Usage

```
register(file, ..., toolchain = default_toolchain(file), date = NULL)
```

```
register_beancount(file, date = NULL)
```

```
register_hledger(
  file,
  flags = "",
  date = NULL,
  add_mark = TRUE,
  add_cost = TRUE,
  add_value = TRUE
)
```

```
register_ledger(file, flags = "", date = NULL)
```

Arguments

file	Filename for a ledger, hledger, or beancount file.
...	Arguments passed on to either register_ledger, register_hledger, or register_beancount
toolchain	Toolchain used to read in register. Either "ledger", "hledger", "beancount", "bean-report_ledger", or "bean-report_hledger".
date	End date. Only transactions (and implicitly price statements) strictly before this date are used.

flags	Character vector of additional command line flags to pass to either ledger csv or hledger register.
add_mark	Whether to add a column with the mark information. Only relevant for hledger files.
add_cost	Whether to add historical cost columns. Only relevant for hledger files.
add_value	Whether to add market value columns. Only relevant for hledger files.

Value

register returns a tibble.

Examples

```
if (Sys.which("ledger") != "") {
  example_ledger_file <- system.file("extdata", "example.ledger", package = "ledger")
  dfl <- register(example_ledger_file)
  head(dfl)
}
if (Sys.which("hledger") != "") {
  example_hledger_file <- system.file("extdata", "example.hledger", package = "ledger")
  dfh <- register(example_hledger_file)
  head(dfh)
}
if (Sys.which("bean-query") != "") {
  example_beancount_file <- system.file("extdata", "example.beancount", package = "ledger")
  dfb <- register(example_beancount_file)
  head(dfb)
}
```

Index

`default_toolchain`, [2](#)

`net_worth`, [2](#)

`prune_coa`, [3](#)

`prune_coa_string (prune_coa)`, [3](#)

`register`, [4](#)

`register_beancount (register)`, [4](#)

`register_hledger (register)`, [4](#)

`register_ledger (register)`, [4](#)