Package ‘shinyInvoice’

April 1, 2024

Title Shiny App - Generate a Pdf Invoice with 'Rmarkdown'

Version 0.0.5

Date 2024-03-31

Description Generate an invoice containing a header with invoice number and businesses details. The invoice table contains
any of: salary, one-
liner costs, grouped costs. Under the table signature and bank account details appear.
Pages are numbered when more than one. Source .json and .Rmd files are editable in the app. A .csv file with raw data can be downloaded.
This package includes functions for getting exchange rates between currencies based on
'quantmod' (Ryan and Ulrich, 2023 <https://CRAN.R-project.org/package=quantmod>).

Depends R (>= 3.5)

Imports tibble, dplyr, quantmod, rlang, stringr, shiny, lubridate

Suggests fs, jsonlite, pillar, readr, rhino, rjson, shinyAce, tinytex,
rmarkdown, curl

License GPL (>= 2)

Encoding UTF-8

SystemRequirements pandoc (>= 2.0)

URL https://github.com/fernandoroa/invoice-public

BugReports https://github.com/fernandoroa/invoice-public/issues

RoxygenNote 7.2.3

NeedsCompilation no

Author Fernando Roa [aut, cre]

Maintainer Fernando Roa <froao@unal.edu.co>

Repository CRAN

Date/Publication 2024-04-01 13:10:02 UTC

R topics documented:

`continue_sequence` *Functions continue_sequence, duplicate_last_list_element and date_bump_month*

Description

continue_sequence: suffix increase, i.e. from 1 to 2, from a to b

duplicate_last_list_element: copies last element from list and bumps its name

date_bump_month: increases or decreases date by one month

Usage

```
continue_sequence(chr_vector, sep = ”_”, factor = 1)

duplicate_last_list_element(list)

date_bump_month(date, decrease = FALSE)
```

Arguments

`chr_vector`	character, vector
`sep`	character, separating prefix from suffix
`factor`	numeric, increase or decrease sequence
`list,`	list
`date,`	date
`decrease,`	boolean

Value

vector

data.frame

date

Examples

```
continue_sequence(c(”a_1”, ”a_2”))
duplicate_last_list_element(list(a_1 = ”1”, b_1 = ”random”))
date_bump_month(as.Date(”2024-02-29”))
```

GetExchangeRates *Functions get_exchange_rates_symbol, try_exchange_rates_direct_and_indirect and GetExchangeRates*

Description

get_exchange_rates_symbol: uses getSymbols

GetExchangeRates: uses getFX and getSymbols

Usage

```
GetExchangeRates(from_curr, to_curr, from_date, to_date = from_date)

get_exchange_rates_symbol(from_curr, to_curr, from_date, to_date = from_date)

try_exchange_rates_direct_and_indirect(date, from_curr, to_curr, tries = 8)
```

Arguments

`from_curr`	character, currency symbol
`to_curr`	character, currency symbol
`from_date`	character, date
`to_date`	character, date
`date`	character, desired date
`tries`	numeric try how many days before desired date

Value

data.frame

data.frame

numeric vector

References

Ryan JA, Ulrich JM (2023). *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4.25, https://CRAN.R-project.org/package=quantmod.

Examples

```
from_curr <- c("CAD", "JPY", "USD")
to_curr <- c("USD", "USD", "EUR")



# get_exchange_rates_symbol and try_exchange_rates_direct_and_indirect
# use quantmod::getSymbols, which requires a working curl, as in:
```

```
if (requireNamespace("curl", quietly = TRUE)) {
  con <- curl::curl(url = "https://hb.cran.dev/get", open = "r", handle = curl::new_handle())
  print(con)
  close(con)
}

# Success
recent_date <- as.character(Sys.Date() - 7)
GetExchangeRates(from_curr = from_curr, to_curr = to_curr, recent_date, recent_date)

# last date mismatch day
GetExchangeRates(from_curr = from_curr, to_curr = to_curr, "2023-10-27", "2023-10-30")

# weekend, warning, gets only FX, fails for getSymbols
# update this dates to less than 180 days!

GetExchangeRates(from_curr = from_curr, to_curr = to_curr, "2023-10-28", "2023-10-28")
GetExchangeRates(from_curr = from_curr, to_curr = to_curr, "2023-10-29", "2023-10-29")
GetExchangeRates(from_curr = from_curr, to_curr = to_curr, "2023-07-08", "2023-07-09")

# fails for FX, > 180 days
GetExchangeRates(from_curr = from_curr, to_curr = to_curr, "2023-04-03", "2023-04-05")

# failure for getSymbols, when none is USD
GetExchangeRates(from_curr = "BRL", to_curr = "COP", "2023-07-07")

# getSymbols success
get_exchange_rates_symbol(from_curr = from_curr, to_curr = to_curr, "2023-07-03", "2023-07-05")

# getSymbols > 180 days ok
get_exchange_rates_symbol(from_curr = from_curr, to_curr = to_curr, "2023-04-03", "2023-04-05")

# failure, weekend days
weekend_failure <- try(get_exchange_rates_symbol(
  from_curr = from_curr,
  to_curr = to_curr, "2023-07-08", "2023-07-09"
), silent = TRUE)
weekend_failure

# works
try_exchange_rates_direct_and_indirect("2023-07-08", from_curr, to_curr, tries = 8)
try_exchange_rates_direct_and_indirect("2023-07-08", "BRL", "USD", tries = 8)
try_exchange_rates_direct_and_indirect("2023-07-08", "USD", "BRL", tries = 8)

# works indirectly
recent_date <- as.character(Sys.Date() - 7)
try_exchange_rates_direct_and_indirect(recent_date, "COP", "BRL", tries = 8)

# works with FX only, provided, not greater than 180 days

GetExchangeRates(from_curr = "COP", to_curr = "BRL", recent_date)
```

`runInvoice` *Function runInvoice*

Description

runInvoice: run shinyApp

Usage

```
runInvoice(installAll = FALSE)
```

Arguments

`installAll`	boolean, when `TRUE` dependencies are installed without asking. Defaults to `FALSE`

Value

shiny

Index