Package 'kisopenapi'

January 24, 2024

Title Korea Investment & Securities (KIS) Open Trading API

Version 0.0.2

Description API Wrapper to use Korea Investment & Securities (KIS) trading
system that provides various financial services like stock price
check, orders and balance check
<https://apiportal.koreainvestment.com/>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports cli (>= 3.3.0), data.table (>= 1.14.2), httr2 (>= 0.2.2),
jsonlite (>= 1.8.0)

RoxygenNote 7.2.3

NeedsCompilation no

Author Seokhoon Joo [aut, cre]

Maintainer Seokhoon Joo <seokhoonj@gmail.com>

Repository CRAN

Date/Publication 2024-01-24 15:30:02 UTC

R topics documented:

---

change_trading_env   *change trading environment*

Description

if you wrote live trading and paper trading evironment variables both in your .Renviron file, you can change trading environment.

Usage

```
change_trading_env()
```

Value

"live" or "paper" string value

Examples

```
## Not run: change_trading_environment()
```

---

get_balance   *get account balance*

Description

Get your account balance.

Usage

```
get_balance(prdt_code, rt_cash_flag = FALSE)
```

Arguments

prdt_code   A string specifying account product code

rt_cash_flag   A boolean specifying total deposit or not

Value

balance data.frame

Examples

```
## get account balance
## Not run:
# detail
get_balance(rt_cash_flag = FALSE)
# total deposit
get_balance(rt_cash_flag = TRUE)
## End(Not run)
```

---

get_buyable_cash   *get buyable cash*

Description

Get buyable amount of cash of the account

Usage

```
get_buyable_cash(prdt_code)
```

Arguments

prdt_code   A string specifying account product code

Value

A numeric specifying buyable cash

Examples

```
# get buyable cash
## Not run:
get_buyable_cash()

## End(Not run)
```

`get_current_price` *get current price*

Description

Get current stock price.

Usage

```
get_current_price(stock_code)
```

Arguments

`stock_code`   A string specifying stock number (stock code)

Value

current stock price

Examples

```
# get current price
## Not run:
get_current_price("005930")

## End(Not run)
```

`get_orders` *get orders*

Description

Return a list of orders that can be revised or canceled.

Usage

```
get_orders()
```

Value

data.frame of orders

Examples

```
# get a list of orders
## Not run:
get_orders()

## End(Not run)
```

---

get_order_history   *get order history*

Description

Get order history from start date to end date.

Usage

```
get_order_history(sdt, edt, prdt_code, zip_flag = TRUE)
```

Arguments

sdt	A string specifying start date "YYYYMMDD"
edt	A string specifying end date "YYYYMMDD"
prdt_code	A string specifying account product code
zip_flag	A boolean specifying choosing important columns

Value

Order history data frame

Examples

```
# get order history
## Not run:
get_order_history("20")

## End(Not run)
```

---

get_stock_history   *get stock history*

Description

Get stock history.

Usage

```
get_stock_history(stock_code, unit = c("D", "W", "M"))
```

Arguments

stock_code   A string specifying stock code

unit   A string specifying day, week, month

Value

stock history data frame

Examples

```
# get stock history
## Not run:
get_stock_history("005930")

## End(Not run)
```

---

get_stock_history_by_ohlcv
                        get stock history by ohlcv

Description

Get stock history by open, high, low, close, volume.

Usage

```
get_stock_history_by_ohlcv(stock_code, unit = "D", add_var = FALSE)
```

Arguments

stock_code   A string specifying stock code

unit   A string specifying day, week, month

add_var   A boolean adding volitility and percentage change

Value

stock history by ohlcv data frame

Examples

```
# get stock history
## Not run:
get_stock_history_by_ohlcv("005930")

## End(Not run)
```

---

get_stock_investor   *get stock investor*

Description

Get stock investor data using KIS API.

Usage

```
get_stock_investor(stock_code)
```

Arguments

stock_code   A string specifying stock code

Value

stock investor data frame

Examples

```
# get stock investor
## Not run:
get_stock_investor("005930")

## End(Not run)
```

---

get_stock_quotes *get stock quotes*

Description

Get current stock qutoes.

Usage

```
get_stock_quotes(stock_code)
```

Arguments

stock_code   A string specifying stock code

Value

current stock quotes data frame

Examples

```
# get stock quotes
## Not run:
get_stock_quotes("005930")

## End(Not run)
```

---

kis_order *KIS order*

Description

Order stocks.

Usage

```
kis_order(
  stock_code,
  order_qty,
  order_price,
  prdt_code,
  order_type = "00",
  buy_flag = TRUE
)

kis_buy(stock_code, order_qty, order_price, prdt_code, order_type = "00")

kis_sell(stock_code, order_qty, order_price, prdt_code, order_type = "00")
```

Arguments

`stock_code`	A string specifying stock code
`order_qty`	A numeric or string specifying order quantity
`order_price`	A numeric or string specifying order price
`prdt_code`	A string specifying account product code
`order_type`	A string specifying order type
`buy_flag`	A boolean specifying flag

Value

A list contains rt_cd: return code, msg_cd: message code, msg1: message

An order result

Examples

```
## Not run:
## buy
kis_buy(
  stock_code = "stock code",
  order_qty = "your order quantity",
  order_price = "your order price"
)
## sell
kis_sell(
  stock_code = "stock code",
  order_qty = "your order quantity",
  order_price = "your order price"
)

## End(Not run)
```

---

kis_revise_cancel   *KIS revise and cancel orders*

Description

Revise and cancel orders.

Usage

```
kis_revise_cancel(
  order_no,
  order_branch,
  order_qty,
  order_price,
```

```
  prdt_code,
  order_dv = "00",
  cncl_dv = c("01", "02"),
  qty_all_yn = c("Y", "N")
)

kis_revise(
  order_no,
  order_branch,
  order_qty,
  order_price,
  prdt_code,
  order_dv = "00",
  cncl_dv = "01",
  qty_all_yn = c("Y", "N")
)

kis_cancel(
  order_no,
  order_branch,
  order_qty,
  order_price,
  prdt_code,
  order_dv = "00",
  cncl_dv = "02",
  qty_all_yn = c("Y", "N")
)

kis_cancel_all()
```

Arguments

`order_no`	A string specifying order number
`order_branch`	A string specifying branch code
`order_qty`	A numeric or string specifying order quantity
`order_price`	A numeric or string specifying order price
`prdt_code`	A string specifying account product code
`order_dv`	A string specifying limit order(00) or market order(01)
`cncl_dv`	A string specifying revise(01) or cancel(02)
`qty_all_yn`	A string specifying total order quantity or not

Value

A list contains rt_cd: return code, msg_cd: message code, msg1: message

response

Examples

```
## revise
## Not run:
kis_revise(
  order_no = "your order number",
  order_branch = "your order branch", order_qty = "your order quantity",
  order_price = "your order price"
)

## End(Not run)
## cancel
## Not run:
kis_cancel(
  order_no = "your order number",
  order_branch = "your order branch", order_qty = "your order quantity",
  order_price = "your order price"
)

## End(Not run)
## cancel all
## Not run:
kis_cancel_all()

## End(Not run)
```

---

print_acnt_prdt_cd *print account product code*

Description

Print your account product code in the current trading environemnt. it changes depending on the trading environment.

Usage

```
print_acnt_prdt_cd()
```

Value

account product code

Examples

```
## Not run: print_acnt_prdt_cd()
```

---

print_app_key *print app key*

Description

Print your app key in the current trading environemnt. it changes depending on the trading environment.

Usage

```
print_app_key()
```

Value

app key

Examples

```
## Not run: print_app_key()
```

---

print_app_secret *print app secret*

Description

Print your app secret in the current trading environemnt. it changes depending on the trading environment.

Usage

```
print_app_secret()
```

Value

app secret

Examples

```
## Not run: print_app_secret()
```

---

print_base_url *print base url*

Description

Print your base url in the current trading environemnt. it changes depending on the trading environment.

Usage

```
print_base_url()
```

Value

base url of current trading environment

Examples

```
## Not run: print_base_url()
```

---

print_cano *print account number*

Description

Print your account number in the current trading environemnt. it changes depending on the trading environment.

Usage

```
print_cano()
```

Value

account number

Examples

```
## Not run: print_cano()
```

---

print_trading_env   *print trading environment*

Description

Print the current trading environment.

Usage

```
print_trading_env()
```

Value

"live" or "paper" string value

Examples

```
## Not run: print_trading_env()
```

---

set_auth   *set access token*

Description

Set access token (hash key) using kis app key and kis app secret.

Usage

```
set_auth()

get_auth()

print_auth()
```

Value

An access token (access token is stored in environment variable at the same time)

Examples

```
# set authentification
## Not run:
set_auth()

## End(Not run)
```

| set_trading_env | *Set KIS account number, account product code, app key and app secret* |

Description

Save environment variables for the current session. To set it permanently,
please add the following line to your .Renvrion file:

Usage

```
set_trading_env(cano, acnt_prdt_cd, app_key, app_secret, is_paper = FALSE)
```

Arguments

cano	A string specifying KIS common account number
acnt_prdt_cd	A string specifying KIS account product code
app_key	A string specifying KIS app key
app_secret	A string specifying KIS app secret
is_paper	A string specifying paper trading or not, default FALSE

Value

No return value, called to set environment variables

For live trading

KIS_CANO="YOUR ACCOUNT NUMBER" (first 8 digits of your account)
KIS_ACNT_PRDT_CD="01" (last 2 digits of your account)
KIS_APP_KEY="YOUR APP KEY"
KIS_APP_SECRET="YOUR APP SECRET"

For paper trading

KIS_PAPER_CANO="YOUR PAPER ACCOUNT NUMBER" (first 8 digits of your paper account)
KIS_PAPER_ACNT_PRDT_CD="01" (last 2 digits of your paper account)
KIS_PAPER_APP_KEY="YOUR PAPER APP KEY"
KIS_PAPER_APP_SECRET="YOUR PAPER APP SECRET"

Examples

```
## Set app key for the current session
## Not run:
## Set trading environment only for the current session
set_trading_env(
  cano = "your account number",
```

```
  acnt_prdt_no = "your account product code",
  app_key = "your app key",
  app_secret = "your app secret"
)
## Check environment variables
print_cano()
print_acnt_prdt_cd()
print_app_key()
print_app_secret()
## End(Not run)
```

Index