Package: qkiosk (via r-universe)

May 25, 2026

Type Package

Title R Client for QUANTkiosk

Version 0.2.29

Imports curl, jsonlite, stringdist, zoo

Depends xts

Suggests fansi

LazyData yes

Date 2026-04-16

Author Jeffrey A Ryan

Maintainer Jeffrey A. Ryan <jeff@quantkiosk.com>

Description R package to interact with QUANTkiosk API.

License GPL-3 + file LICENSE

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-04-16 16:30:12 UTC

RemoteUrl https://github.com/quantkiosk/qkiosk-r

RemoteRef HEAD

RemoteSha 72dcca8dcda83cf265174d89da1aa9dbeb2b0650

Contents

`qkiosk-package` *R Client for QUANTkiosk*

Description

R package to interact with QUANTkiosk API.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Official QUANTkiosk R API

Author(s)

Jeffrey A Ryan

Maintainer: Jeffrey A. Ryan <jeff@quantkiosk.com>

References

http://quantkiosk.com

account *QUANTkiosk™ Account Access*

Description

Utilities for account usage and data downloads.

Usage

```
qk_account(browser=FALSE, files=FALSE)

qk_download(file = NULL,
            type = c("data", "univ", "trials"),
            n = 10, available=FALSE, retjson = TRUE)

qk_get_apikey(req=TRUE)
qk_set_apikey(apiKey,verbose=TRUE)
```

Arguments

browser	open account page in browser
files	include custom files (if any) in account
file	file to download
type	data to download - either data or universe files.
available	return table of files and last modified times
retjson	should json be returned as raw string
n	number of files to select from
req	should an error be issued if not present
apiKey	your QUANTkiosk™ api key.
verbose	be verbose

Details

QUANTkiosk™ accounts control access as well as provide for custom data access.

To set (or get) your API key the functions `qk_get_apikey` and `qk_set_apikey` are provided. These rely your environment variable, which may be set from with your shell or from your session.

`qk_account` provides details on your current usage - as well as access to any custom data files, universe files, or trial data you may have. To access the files for download, the `qk_download` function allows for browsing your remote data as well as downloading via an interactive menu or by specifying the name of the file.

Value

- AsOfTime of request
- QuotaCurrent daily quota in credits
- UsageCurrent usage in credits consumed
- DataFilesNNumber of custom data files available
- DataFilesList of data files
- UnivFilesNNumber of custom univ files available
- UnivFilesList of univ files

Author(s)

QUANTkiosk™ authors.

References

QUANTkiosk™ Account Page: https://quantkiosk.com/account

`crox` *Crocs Institutional Holders Details By Issuer*

Description

Institutional Holders API example that (tragically) never goes out of style.

Usage

```
crox
```

Format

A data frame with 214 observations on the following 17 variables.

`issuerName` name of company

`filing` canonical EDGAR® filing path

`submissionType` filing type: 13F-HR or 13F-HR/A (amended)

`reportPeriod` period of report

`filedDate` when the data was filed

`issuerTitleOfClass` reported title or class name

`filerCik` Central Index Key of holder

`filerName` name of holder

`issuerQkid` QKID™ symbol

`value` fair market value of position at end of period

`QOQValue` quarter over quarter change in value of holding (QK™ calculated)

`shrsOrPrnAmt` shares 'SH' or principal 'PRN'

`QOQSshPrnAmt` quarter over quarter change in shares or principal amount of holding (QK™ calculated)

`portWgt` portfolio weight (QK™ calculated)

`QOQPortWgt` quarter over quarter change in portfolio weight (QK™ calculated)

`qtrsHeld` quarters held (QK™ calculated)

`newOrDeleted` is position 'NEW' or 'DEL'eted. See details. (QK™ calculated)

Details

`crox` provides an aggregated view of holdings across all institutional filers. This is a cross-filer view and is derived from between 5000-10000 individual filings.

References

Ownership homepage: https://www.quantkiosk.com/ownership

Ownership Dashboard: https://api.quantkiosk.com/ownership

Ownership API: https://www.quantkiosk.com/api/ownership

SG Capital Filings https://www.sec.gov/edgar/browse/?CIK=0001510099

Form 13F-HR FAQ: https://www.sec.gov/divisions/investment/13ffaq

Official 13(f) Securities https://www.sec.gov/divisions/investment/13flists

See Also

R function: `qk_institutional` Aggregated Manager example: `sgcap` Non-aggregated Manager (Submanager) example: `deshaw`

Examples

```
str(crox)

## Not run:
qk_holders(qk_ticker("CROX"), yyyy=2024, qq=2)

## End(Not run)
```

deshaw *D.E. Shaw Institutional Ownership Details (Including Submanagers)*

Description

Institutional Ownership API example of large manager filing with submanager details.

Usage

```
deshaw
```

Format

A data frame with 6481 observations on the following 28 variables.

`filerName` name of filer (i.e. fund name)

`filing` canonical EDGAR® filing path

`submissionType` filing type: 13F-HR or 13F-HR/A (amended)

`reportPeriod` period of report

`filedDate` when the data was filed

`inclMgrs` number of submanagers reported for. See `otherManager`

`issuer` name of instrument

`titleOfClass` title or class name of instrument

`issuerSIC` official standardized industrial code

`issuerSector` two-character sector code from QUANTkiosk™

`issuerTicker` common exchange ticker

`issuerQkid` QKID™ symbol

`value` fair market value of position at end of report period

`shrsOrPrnAmt` number of shares held or principal amount

`putCall` one of 'PUT', 'CALL' or empty

`shrsOrPrnAmtType` shares 'SH' or principal 'PRN'

`invDiscretion` type of discretion over investment decision

`votingAuthSole` number of shares of 'SOLE' voting authority

`votingAuthShared` number of shares of 'SHARED' voting authority

`votingAuthNone` number of shares of 'NONE' voting authority

`portWgt` portfolio weight (QK™ calculated)

`hasOtherManager` logical indicating if record is for other manager

`otherManager` reference number of other manager(s) from filing

`otherManagerName` name of other manager(s) from filing

`otherManagerFileNumber` file number of other manager(s)

`QtrsHeld` quarters held (QK™ calculated)

`QOQSshPrnAmt` quarter over quarter change in shares or principal amount of holding (QK™ calculated)

`QOQValue` quarter over quarter change in value of holding (QK™ calculated)

`QOQPortWgt` quarter over quarter change in portfolio weight (QK™ calculated)

`newOrDel` is position 'NEW' or 'DEL'eted. See details. (QK™ calculated)

Details

Institutional Ownership consists of quarterly '13F-HR' and amended reports '13F-HR/A' made by investors who hold at least $100 million of securities declared by the U.S. Securities and Exchange Commission. Filers include banks, insurance companies, and hedge funds. The filings are due 45 days after the end of each calendar quarter, and are filed via the EDGAR® system. The securities reported must be from the official '13(f) securities' list published by the SEC. See references section.

QUANTkiosk™ provides access to these filings in two forms at present. The first is an aggregated view that includes one row per unique instrument held. This is described in full in the SG Capital example `sgcap`. The second version (ex. `deshaw`) contains additional submanagers details:

- inclMgrs
- invDiscretion
- votingAuthSole
- votingAuthShared
- votingAuthNone
- otherManager
- otherManagerFileNumber

Additional fields not present in the official '13F-HR' or '13F-HR/A' but provided in both aggregated and non-aggregated views:

- issuerSIC
- issuerSector
- issuerQkid
- issuerTicker

These fields are useful for tracking changes between quarters as well as intrafiling analysis. `otherManager` and `otherManagerName` provides the reported reference number as filed for each entry. Details of this requirement are beyond this help page, but reference the official SEC documentation and rules for clarity. QK™ does not merge submanagers and instead reports the combined submanagers (if reported). For example, a 1;2 would mean the managers 1 and 2 share some aspect of the position.

`otherManagerFileNumber` is included to help disambiguate the reported ordering from the actual submanagers reporting. The names for the submanagers can be had using the `meta=TRUE` option in `qk_institutional`.

Additional fields pre-calculated by QUANTkiosk™ across time:

- QOQSshPrnAmt

- QOQValue
- QOQPortWgt
- newOrDel
- QtrsHeld

QK™ calculates these fields as quarter over quarter change in absolute terms. `newOrDel` marks holdings that are 'NEW' this quarter or 'DEL'eleted. Deleted entries are not in the filing, but are provided to identify positions traded out of. `QtrsHeld` shows duration of a continuously held position in aggregate, or in the case of submanager API data, positions held continuously by the exact combination of submanagers. The latter is less reliable due to reporting considerations in the source data.

References

Ownership homepage: https://www.quantkiosk.com/ownership

Ownership Dashboard: https://api.quantkiosk.com/ownership

Ownership API: https://www.quantkiosk.com/api/ownership

D.E. Shaw Filings: https://www.sec.gov/edgar/browse/?CIK=0001009207

Form 13F-HR FAQ: https://www.sec.gov/divisions/investment/13ffaq

Official 13(f) Securities https://www.sec.gov/divisions/investment/13flists

See Also

R function: qk_institutional Aggregated example: sgcap

Examples

```
str(deshaw)

## Not run:

# code that downloads  D.E. Shaw Ownership data.
deshaw_qkid <- qk_search_mgr("de shaw") # 0001009207.0000.E0000LMPJ
qk_institutional(deshaw_qkid, yyyyqq=202401, agg=FALSE)

## End(Not run)
```

`institutional` *QUANTkiosk™ Ownership API Functions*

Description

Query all ownership across firms, invididuals and issuers.

Usage

```
qk_institutional(qkid,
                 yyyyqq,
                 qtrs=1,
                 agg = TRUE,
                 wait = 0,
                 quiet = TRUE)

qk_insider(qkid,
           yyyyqq,
           qtrs=1,
           form = c('insider', 'intent', 'sales'),
           wait = 1,
           quiet = TRUE)

qk_beneficial(qkid,
              yyyyqq,
              qtrs=1,
              form = c('13D13G','13G','13D'),
              wait = 1,
              quiet = TRUE)

qk_activist(qkid,
            yyyyqq,
            qtrs=1,
            form = c('13D','13D13G'),
            wait = 1,
            quiet = TRUE)

qk_holders(qkid,
           yyyyqq,
           qtrs=1,
           agg = TRUE,
           rollprev = TRUE,
           wait = 1,
           quiet = TRUE)
```

Arguments

qkid	a qkid object representing the 'QKID™' of reporting institution or filer. Use qk_cik, qk_ticker, or similar. See qkid for details.
yyyyqq	a number of the form 'YYYYQQ' where 'YYYY' is a 4-digit year, and 'QQ' is a 2-digit quarter [00,01,02,03,04]. Use in conjuction with qtrs argument to request additional prior quarters in one call. Four digit values will be treated as a full year (i.e. 'YYYY') and expanded to YYYY00.
qtrs	number of previous quarters to request (including current yyyyqq.

form	for qk_beneficial and qk_activist requests, one of '13D' (activist), '13G' (passive), '13D13G' (activist + passive). For qk_insider requests one of 'insider' (345), 'intent' or 'sales' (form 144).
agg	if TRUE, all other managers reporting are rolled up into one record, otherwise filing will contain other managers if filed for as distinct holdings for one or more of the same instruments.
rollprev	if TRUE and requested quarter is latest quarter to file, the previous completed quarter will be used to augment the in-progress filings. Any new filers will override the last known holdings positions, and yet-to-file updates will allow for last known positions to be seen. Setting to FALSE will disable this merge and will only return the (potentially incomplete) quarter being requested.
wait	time in seconds between calls.
quiet	make request silent.

Details

QUANTkiosk™ Ownership combines multiple sources of information in an custom formatted data.frame. At it's core, the data is used to understand which participants have control over an entity. These participants file various forms depending on who they are and what their intent may be.

Ownership reporting periods vary based on regulatory rules. Some are quarterly (e.g. Institutional and Funds), whereas many are filed based on events (e.g. Insider, Beneficial/Activist). The arrival of the updates is dependent on filer and SEC rules. All requests are specified for the quarter a filing was made.

To allow for consistent interface, yyyyqq is specified as the year 'yyyy' and the 2-digit quarter being requested. For example, the second quarter of 2023 would be yyyyqq=202302. To request a full year, one can use the special quarter 00. To request all years use yyyyqq=0 or yyyyqq=000000.

All usage is metered by data requested, e.g. four quarters or one year are equal in credits used.

For 'Institutional' (qk_institutional) aggregated (agg=TRUE) requests, we roll up other managers filed within the main filer's report to better see total positions and changes. If agg=FALSE, other managers are displayed as reported with quarter over quarter change columns reflecting changes in exact reporting lines.

Value

A 'data.frame' of class qkinstitutional containing holdings or holders. See example data for fields.

Institutional

Institutional data consists of holdings reported by asset managers meeting certain criteria. Covering more than 15,000 managers over the last decade, these are extracted in a point-in-time manner from forms '13F-HR', '13F-NT' and the amended versions of these forms.

Insider

Insider ownership is reported through forms '3', '4', and '5'. These are reports filed in duplicate for insiders of a company to describe their ownership rights, including acquisition and disposal of shares. These are filed after a change in ownership.

Insider Intent

Insider Intent is a newer form, only required to be electronically filed in the last few years. This is used to indicate the intention to buy or sell, *but not the actual purchase or sale*. These are filed by 'insiders' of a company using form '144'. This is split into 'intent' disclosures, as well as any 'sale' in the 3 months prior to the filing.

Holders

The holders data includes all owners of a particular company, allowing the cross sectional intitutional holdings in one call. This data is aggregated from all individual reporting firms by quarter and made available as new filings arrive. At present this only contains '13F-HR' reporting, updated as filings arrive, including amendments to previous positions.

Blockholders / Beneficial / Activist

Any owner with a position greater than 5% must file under SEC's 'Schedule 13G (SC 13G)' or 'Schedule 13D (SC 13D)'. These are used to indicate control at or exceeding this threshold by one or more parties sharing control of the position.

Funds

The largest of all assets under management comes from mutual and closed-end funds that report on form 'NPORT-P'. As opposed to 'Institutional' filers, these represent publicly traded instruments. Reported quarterly.

Author(s)

QUANTkiosk™

References

SEC's EDGAR® Website: https://www.sec.gov/search-filings

QK Ownership™: https://quantkiosk.com/ownership

EDGAR® Form SC 13D, SC 13G FAQ: https://www.sec.gov/divisions/investment/13ffaq

EDGAR® Form 13F-HR FAQ: https://www.sec.gov/divisions/investment/13ffaq

EDGAR® Form 3,4,5 FAQ: https://www.sec.gov/divisions/investment/13ffaq

See Also

Small filer example: sgcap

Large filer example with submanagers: deshaw

Holders example: crox

Benficial Owner example: pershing

Examples

```
## Not run:
citadel <- qk_search_mgr("citadel")
citadel_h<- qk_institutional(citadel, yyyyqq=202301)

# all qtrs of 2023
citadel_h<- qk_institutional(citadel, yyyyqq=202300)

# nvidia's institutional holders for 202501, 202404, 202403, 202402
nvda_h <- qk_holders(qk_ticker("NVDA"), yyyyqq=202501, qtrs=4)

# nvidia's intent to sell 202501, 202404, 202403, 202402
nvda_insider<- qk_insider(qk_ticker("NVDA"), yyyyqq=202501, qtrs=4, form="insider")

# nvidia's intent to sell 202501, 202404, 202403, 202402
nvda_intent <- qk_insider(qk_ticker("NVDA"), yyyyqq=202501, qtrs=4, form="intent")

# nvidia's intent to sell 202501, 202404, 202403, 202402
nvda_sales <- qk_insider(qk_ticker("NVDA"), yyyyqq=202501, qtrs=4, form="sales")

# Restaurant Brands (QSR) block holders for 202501, 202404, 202403, 202402
qsr_b <- qk_beneficial(qk_ticker("QSR"), yyyyqq=2025)

## End(Not run)
```

---

nke *Nike (NKE) Insider Ownership Data*

Description

Insider Ownership API example of large firm.

Usage

```
nke
```

Format

A data frame with 60 observations on the following 40 variables.

`acceptanceDatetime` EDGAR® processing time

`filing` canonical EDGAR® filing path

`form` filing type: 3, 3/A (amended), 4, 4/A or 5, 5/A

`is10b5One` indication if transaction was made pursuant to a contract, instruction or written plan

`rptPeriod` a character vector

`issuerCik` issuer Central Index Key

`issuerName` issuer name

`issuerTradingSymbol` issuer trading symbol

`issuerQkid` issuer QKID™ symbol

`ownerCik` insider Central Index Key

`ownerName` a character vector

`isDirector` is company director

`isOfficer` is company officer

`officerTitle` officer title

`isTenPct` is a ten percent insider

`secTitle` security title

`secTitleFootnote` footnote for security title

`conversionOrExercisePrice` price

`conversionFootnote` a character vector

`isTxn` is reporting line a transaction

`isDeriv` is security a derivative

`txnDate` transaction date

`execDate` a logical vector

`txnCode` transaction code pursuant to Rule 16b-3. See Details.

`txnDesc` transaction footnote

`txnAmount` transaction amount

`acquiredOrDisposed` acquired or disposed

`txnPrice` transaction price

`txnValue` transaction value

`exerciseDate` exercise date

`exerciseDateFootnote` exercise date footnote

`expirationDate` expiration date

`expirationFootnote` expiration footnote

`underlyingSecTitle` underlying security

`underlyingShs` number of shares of underlying

`postTxnShs` shares owned post transaction

`postTxnFootnote` post transaction footnote

`ownershipForm` form of ownership

`ownershipNature` nature of ownership

`ownershipFootnote` ownership footnote

Details

The 'Insider Ownership' data provided by QUANTkiosk™ are gathered from SEC Form 3,4 and 5. The forms are used for initial insider position, changes in position and exit of position, respectively.

These forms must be filed before the end of the second business day following the day on which a transaction resulting in a change in beneficial ownership has been executed (see references below).

Codes are as described below (direct from SEC)

General Transaction Codes

P - Open market or private purchase of non-derivative or derivative security

S - Open market or private sale of non-derivative or derivative security

V - Transaction voluntarily reported earlier than required

Rule 16b-3 Transaction Codes

A — Grant, award or other acquisition pursuant to Rule 16b-3(d)

D — Disposition to the issuer of issuer equity securities pursuant to Rule 16b-3(e)

F — Payment of exercise price or tax liability by delivering or withholding securities incident to the receipt, exercise or vesting of a security issued in accordance with Rule 16b-3

I — Discretionary transaction in accordance with Rule 16b-3(f) resulting in acquisition or disposition of issuer securities

M — Exercise or conversion of derivative security exempted pursuant to Rule 16b-3

Derivative Securities Codes (Except for transactions exempted pursuant to Rule 16b3)

C — Conversion of derivative security

E — Expiration of short derivative position

H — Expiration (or cancellation) of long derivative position with value received

O — Exercise of out-of-the-money derivative security

X — Exercise of in-the-money or at-the-money derivative security

Other Section 16(b) Exempt Transaction and Small Acquisition Codes (except for Rule 16b-3 codes above)

C — Conversion of derivative security

G — Bona fide gift

L — Small acquisition under Rule 16a-6

W — Acquisition or disposition by will or the laws of descent and distribution

Z — Deposit into or withdrawal from voting trust

Other Transaction Codes

C — Conversion of derivative security

J — Other acquisition or disposition (describe transaction)

K — Transaction in equity swap or instrument with similar characteristics

U — Disposition pursuant to a tender of shares in a change of control transaction

References

SEC Investor Bulletin: https://www.sec.gov/files/forms-3-4-5.pdf

Form 4 Instructions: https://www.sec.gov/about/forms/form4data.pdf

Examples

```
data(nke)
```

`nvda_intent` *Nvidia (NVDA) Restricted or Control Shares Intention to Sell*

Description

Corporate insider and affiliates intention to sell disclosure.

Usage

```
nvda_intent
```

Format

A data frame with 54 observations on the following 24 variables.

`acceptanceDatetime` EDGAR® processing time

`accn` internal EDGAR® accession number

`filedDate` date filed in form yyyymmdd

`filename` canonical EDGAR® filing path

`form` filing type: 144 or 144/A

`issuerCik` issuer Central Index Key

`issuerName` issuer name

`issuerQkid` issuer QKID™ symbol

`filerCik` seller Central Index Key

`filerName` seller name

`relationshipToIssuer` a character vector

`securitiesClassTitle` a character vector

`acquiredDateYYYYMMDD` the date to be acquired

`natureOfAcquisitionTransaction` nature of acquisition

`nameOfPersonfromWhomAcquired` entity or person acquired from

`amountOfSecuritiesAcquired` amount of securities acquired

`paymentDateYYYYMMDD` a numeric vector

`natureOfPayment` a character vector

`price` the price at acquisition

`noOfUnitsSold` a numeric vector

`aggregateMarketValue` a numeric vector

`noOfUnitsOutstanding` a numeric vector

`price2` the price of expected sale

`marketCap` market capitalization as reported in USD

Details

Form 144 is a requirement to disclose the intention to sell restricted shares. It is made up of two different sections:

Intent the proposed amount of shares expected to sell `nvda_intent`

Sales all sales made over the preceeding three months `nvda_sales`

The intent section of a form is unique across multiple forms filed by a person or entity. The "Sales" section may contain duplicate entries in the data provided by the API. Thsi is intentional as it represents the time(s) the information was disclosed.

From the SEC, the securities and people reporting are:

Restricted securities are securities acquired in unregistered, private sales from the issuing company or from an affiliate of the issuer. Investors typically receive restricted securities through private placement offerings, Regulation D offerings, employee stock benefit plans, as compensation for professional services, or in exchange for providing "seed money" or start-up capital to the company. Rule 144(a)(3) identifies what sales produce restricted securities.

Control securities are those held by an affiliate of the issuing company. An affiliate is a person, such as an executive officer, a director or large shareholder, in a relationship of control with the issuer. Control means the power to direct the management and policies of the company in question, whether through the ownership of voting securities, by contract, or otherwise. If you buy securities from a controlling person or "affiliate," you take restricted securities, even if they were not restricted in the affiliate's hands.

References

Form 144 Investor Information: https://www.sec.gov/about/reports-publications/investorpubsrule144

Examples

```
data(nvda_sales)
```

nvda_sales *Nvidia (NVDA) Restricted or Control Shares Sold*

Description

Corporate insider and affiliates intention to sell historical sales.

Usage

```
nvda_sales
```

Format

A data frame with 386 observations on the following 18 variables.

`acceptanceDatetime` EDGAR® processing time

`accn` internal EDGAR® accession number

`filedDate` date filed in form yyyymmdd

`filename` canonical EDGAR® filing path

`form` filing type: 144 or 144/A

`issuerCik` issuer Central Index Key

`issuerName` issuer name

`issuerQkid` issuer QKID™ symbol

`filerCik` seller Central Index Key

`filerName` seller name

`relationshipToIssuer` a character vector

`sellerName` name of seller

`securitiesClassTitle` description of security

`saleDateYYYYMMDD` date shares were sold

`amountOfSecuritiesSold` number of shares sold

`grossProceeds` total proceeds of sale in USD

`asOf` unused

`price` share price in USD

Details

Form 144 is a requirement to disclose the intention to sell restricted shares. It is made up of two different sections:

Intent the proposed amount of shares expected to sell `nvda_intent`

Sales all sales made over the preceeding three months `nvda_sales`

The intent section of a form is unique across multiple forms filed by a person or entity. The “Sales” section may contain duplicate entries in the data provided by the API. Thsi is intentional as it represents the time(s) the information was disclosed.

From the SEC, the securities and people reporting are:

Restricted securities are securities acquired in unregistered, private sales from the issuing company or from an affiliate of the issuer. Investors typically receive restricted securities through private placement offerings, Regulation D offerings, employee stock benefit plans, as compensation for professional services, or in exchange for providing "seed money" or start-up capital to the company. Rule 144(a)(3) identifies what sales produce restricted securities.

Control securities are those held by an affiliate of the issuing company. An affiliate is a person, such as an executive officer, a director or large shareholder, in a relationship of control with the issuer. Control means the power to direct the management and policies of the company in question, whether through the ownership of voting securities, by contract, or otherwise. If you buy securities from a controlling person or "affiliate," you take restricted securities, even if they were not restricted in the affiliate’s hands.

References

Form 144 Investor Information: https://www.sec.gov/about/reports-publications/investorpubsrule144

Examples

```
data(nvda_sales)
```

`pershing` *Pershing Square Beneficial and Activist Details*

Description

Beneficial and Activist API example of single manager block positions for 2019-2020

Usage

```
data("pershing")
```

Format

A data frame with 72 observations on the following 25 variables.

`acceptanceDatetime` acceptance time to SEC’s EDGAR® system

`filing` canonical EDGAR® filing path

`form` filing type: SC 13D, SC 13G, SC 13D/A or SC 13G/A

`filedAsOf` when the data was filed

`dateOfEvent` date reporting for

`issuerName` name of issuer in submission

`issuerCik` Central Index Key of issuer

`issuerQkid` QKID™ symbol

`issuerTicker` common exchange ticker

`filerName` name of reporting filer

`filerCik` Central Index Key of filer

`rule13d13g` rule being reported for

`namesOfRptPersons` individual or firm name who is reporting

`isGrp` a character vector

`srcOfFunds` a character vector

`srcOfFundsDesc` a character vector

`placeOfOrg` a character vector

`soleVoting` a numeric vector

`sharedVoting` a numeric vector

`soleDispositoryPwr` a numeric vector

`sharedDispositoryPwr` a numeric vector

`aggAmtOwned` a numeric vector

`pctOfClass` a numeric vector

`personType` a character vector

`personTypeDesc` a character vector

Details

Beneficial and Activist Ownership describe owners (reporting) who maintain a large share of a public company (issuer). The forms filed when a filer maintains or exceeds a 5% are 'SC 13G' and 'SC 13D', and the follow-on amendments as thos positions change.

The difference in form types is related to filer *intent*, specifically whether the 5% owner is passive, or is attempting to exert control over the company. The canonical example of an large investor who may file both is 'Pershing Square', run by 'Bill Ackman'. We include this dataset to document the fields. Additional history is available for all filings of this type as well as for all history.

Through 2024, all filings have come as unstructured text and as such are very difficult to systematically extract data. QUANTkiosk™ runs a complex automated extraction process to provide reliable data in a form that is made available in R via `qk_beneficial` and `qk_activist` functions.

References

Ownership homepage: https://www.quantkiosk.com/ownership

Pershing Square Filings: https://www.sec.gov/edgar/browse/?CIK=0001336528

Form SC 13D/G information: https://www.sec.gov/rules-regulations/staff-guidance/compliance-disclosure-interpretations/exchange-act-sections-13d-13g-regulation-13d-g-beneficial-ow

Examples

```
str(pershing)

## Not run:
pershing.cik <- "0001336528"

qk_beneficial(pershing.cik, yyyy=c(2019,2020), qq=0)

## End(Not run)
```

`pfe` *Pfizer (PFE) Revenue Data*

Description

QUANTkiosk™ Fundamentals API (As Of)

Usage

```
data("pfe")
```

Format

A data frame with 60 observations on the following 40 variables.

`cik` the Central Index Key of the issuer

`pik` the QKID™ Permanent Index Key of the issuer

`acceptance_time` EDGAR® processing time

`acceptance_utc` EDGAR® processing time in unix (seconds since 19700101)

`stmt` statement including the item

`item` the QK™ standardized item code

`unit` unit of measurement - as reported

`form` form filing type: 10-Q, 10-Q/A, 10-K, 10-K/A, 8-K, 8-K/A, 20-F, 20-F/A, 6-K, 6-K/A

`filename` canonical EDGAR® filing path

`accn` EDGAR® accession number

`fye` fiscal year end

`fyeq` fiscal yeat end quarter

`fpb` fiscal period beginning

`fpe` fiscal period end

`filed` date filed

`fpe2filed` days from fiscal period end to filed

`filed40` forty day mark from fpe

`ffiled` first filed - will be different in a restatment

`cyqtr` calendar year quarter available

`cyear` calendar year available

`cqtr` calendar quarter available

`rptq` value - fiscal quarter (reported)

`rptqd` reporting quarter days

`rpty` value - fiscal year to date (reported)

`rptyd` reporting year days

`fp` fiscal period - e.g. Q1, Q2, Q3, Q4, FY

`fqtr` fiscal quarter - e.g. 1, 2, 3, 4

`fqd` fiscal quarter days

`fq` value - calculated fiscal quarter

`fytd` value - calculated fiscal year to date

`ttm` value - calculated trailing twelve months

`rstmt` restated flag

`rstmt.q` restated flag - quarterly value

`rstmt.y` restated flag - yearly value

`iq` imputed quarterly

`asof` as of date

`concept_id` unique id for auditing

`fqpy` fiscal quarter - prior year

`fytdpy` fiscal year to day - prior year

`ann` annual value - only fye

Details

QUANTkiosk™ Fundamentals are extracted from original sources, using a systematic process. Data is extracted and normalized into QK™ concepts that align across time and firms using a set of rules. Most concepts are currently using 'US-GAAP' taxonomy.

Source

QUANTkiosk™ https://www.quantkiosk.com/fundamentals

Examples

```
to_df(pfe)
```

qk_audit *QUANTkiosk Data Auditing Tools*

Description

Methods to audit orginal data sources to verify data correctness and interpretation.

Usage

```
qk_audit(x, ...)

## Default S3 method:
qk_audit(x, row, accn, fpe, open = FALSE, ctx=-1, ...)
```

Arguments

x	data.frame to audit
row	row of data.frame to display source
accn	optional accession number (qk_fn)
fpe	optional fiscal period end (qk_fn)
open	should browser window be opened to orginal source page
ctx	limit display of matching data to a smaller context window. defaults to full record.
...	unused

Details

Non-market data is derived from myriad sources and contain billions of datum. To better prove correctness (and to track data issues), QUANTkiosk provides a user inteface to interrogate the original sources in a variety of ways.

The first application is for the fundamentals data we curate from billions of datum published by companies. The qk_audit interface allows the researcher to request the source data - *in context* from the original filings, if applicable.

For most of the standardized financial reporting line items (GAAP), we have developed item codes to standardize reporting views when possible. These data are coallesced from multiple places within filings including tabular data and the xbrl (extensible business reporting language) mandaded in the US and in other countries for many instances of reporting.

qk_audit is designed to be used regularly to inspect context as well as numerical correctness of data returned by QUANTkiosk's APIs by using extracted data from the original HTML filings. These original tables do not include the additional contextual details present in the qk_fn data - often only presenting a row label and a set of period to date values. We process and extract the additional 30+ fields from a complex internal process that includes validation steps as well as the very facility to inspect as presented in qk_audit.

Value

A displayed version of the text-formatted original html tables, including context we are assigning and a highlighted section of the html that matches the data in question.

Some values may we return in the API are actually imputed from the data that may actually only report data in a period-to-date format.

Note

This functionality is not all inclusive as locating data within a filings is not one for one. Often data can be presented outside of statements or tables and therefore has no easy to display data source. This may be possible in future version fo QK.

Author(s)

QUANTkiosk

References

www.sec.gov

See Also

See Also as qk_fn

Examples

```
## Not run:
jnj <- qk_fn(qk_ticker("JNJ"), "SALE")
jnj |> to_df() |> qk_audit(55)

## End(Not run)
```

---

qk_complete *Live Download EDGAR® Filings Raw & Processed Data*

Description

Provide access to QUANTkiosk™ *QK Complete*™ professional subscription. This function is intended to be used as part of an ETL pipeline and run at fixed intervals to provide local live capture of filings data.

Usage

```
qk_complete(dir,
            subscription = "raw",
            rollup = "5m",
            last = "5m",
            date = today(),
            fmt = c("zip", "tar.gz"),
            gzip = TRUE,
            hide = TRUE,
            quiet = FALSE,
            force = FALSE)
```

Arguments

dir	Directory to use for raw filings. Will be created if needed.
subscription	Subscription type varies with access. 'raw' retrieves all '.txt' filings submitted to the EDGAR® system.
rollup	Window of time for which new filings are provided.
last	Lookback window to query for rollup archives.
date	SEC date to query (America/New_York)
fmt	Return archive type. Default is 'zip' format which contains uncompressed ('gzip') raw text filings.
gzip	Should files be compressed for local storage.
hide	Hide API key in request if quiet=TRUE.
quiet	Hide request from output.
force	Overwrite files if already downloaded.

Details

qk_complete syncs live data subscriptions in near realtime. For most applications this is sufficient to maintain a full history locally while the filings are arriving and being processed by the SEC.

Value

Called for its side-effects, the function returns a list of filings that have been extracted from the archive. The primary result is the writing of the raw (or processed) filings to disk, as well as associated meta data created by the api.

Note

This endpoint requires both historical and live access. Consult your account page for details.

Author(s)

QUANTkiosk™

References

SEC's EDGAR® Website: https://www.sec.gov/search-filings

Examples

```
## Not run:
edgardir <- tempdir()
qk_complete(edgardir)

dir(edgardir)

## End(Not run)
```

---

qk_filter *QK Object Manipulation*

Description

'generic' functions to filter and sort data returned from the QK API. These are helper functions designed to facilitate exploration of specific data structures provided by the API.

Usage

```
qk_filter(x, ...)

## S3 method for class 'qk_df'
qk_filter(x, by = c("top", "incr", "decr", "new", "del"), ...)

qk_sort(x, ...)
## S3 method for class 'qk_df'
qk_sort(x, by = c("value", "shrsOrPrnAmtType", "portWgt",
                                  "QtrsHeld", "qtrsHeld", "QOQSshPrnAmt",
                                  "QOQValue", "QOQPortWgt"), incr = TRUE, ...)

qk_summary(x, ...)
## S3 method for class 'qk_df'
qk_summary(x, n = c(10, 5), ...)
```

Arguments

x	R object
by	for qk_sort the column to sort on. For qk_filter, selection of rows matching description. See Details.
incr	sort in numeric increasing order. Set FALSE to order largest negative values.
n	number of elements in tabular summary.
...	unused by current methods.

Details

qkiosk makes available multiple data sets from the *quantkiosk* platform. These are often information dense and require further investigation. To help facilitate and align the experience across supported tools (e.g. dashboard, R, python), a common set of basic filtering is provided for convenience.

For 'Institutional Ownership' data returned from `qk_institutional`

See `deshaw` and `sgcap` for examples of data.

Value

filtered and sorted versions of original data.

Author(s)

QUANTkiosk

References

See `https://api.quantkiosk.com/ownership` dashboard examples

See Also

See Also as `qk_institutional`, ~~~

Examples

```
## Not run:
# SG Capital CIK
sg.cik <- 1510099

# sgcap is available as a dataset in package
sgcap <- qk_institutional(sg.cik, yyyy=2024, qq=1)

## End(Not run)

head(qk_filter(sgcap, by='incr'))
head(qk_sort(sgcap, by='QtrsHeld'))
head(qk_sort(qk_filter(sgcap, by='decr'), by='QOQValue', incr=FALSE))

qk_summary(sgcap)
```

qk_fn *Fundamentals Data*

Description

Functions to get 'as-reported' and 'revised' fundamentals line item data from QK™/Financials API.

Usage

```
qk_fn(qkids,
      items,
      qkid_items = NULL,
      from = 20000101,
      to = today(),
      asof = today(),
      asfiled = FALSE,
      aspit = getOption("qk.aspit",FALSE),
      ticker = TRUE,
      hide = TRUE,
      quiet = TRUE,
      cache = TRUE)

qk_fncodes()

## S3 method for class 'qk_fn'
qk_select(x, id_item, id, item, i, j, ...)

qkiditems(qkids,items,using=TRUE)

pitAsOf(pit,dt = today(), qtrs=1)

pitAsFiled(pit, dt = today(), qtrs=1)
```

Arguments

qkids	a qkid vector representing the 'QKID™' of reporting institution or filer. Use qk_cik, qk_ticker, or similar. See qkid for details.
items	a vector or list of vectors containing QUANTkiosk™ item codes (e.g. SALE, NI, AT, ATC) corresponding to US-GAAP, DEI, or other line item(s). See qk_fncodes()
qkid_items	an object of class qkid_item created via qkiditems. if specified, will ignore qkids and items. If qkid_items=NULL (default) qkids and items will be converted into a corresponding qkid_object.
from	start date of returned series.
to	end date of returned series. expresses as an integer or yyyymmdd date object.

asof	a number in form of YYYYMMDD. return last known values (incl. restated) for all periods up to 'asof' date. See details for more information on subscription limits.
asfiled	return as-filed values from original report (no restatements).
aspit	requests full point-in-time table for historical dates. ONLY AVAILABLE TO ENTERPRISE. By default this is attempted once and if not enabled either the *latest* point-in-time or as-filed data returned depening on asof or asfiled argument values.
ticker	should column names use tickers
hide	hide API Key from verbose (quiet=FALSE) output.
quiet	control verbosity.
cache	defaults to TRUE
pit	qkid-item timeseries from qk_fn(..., aspit=TRUE) call.
dt	date (yyyymmdd) used for point-in-time calculation. This will be converted using as.yyyymmdd.
qtrs	number of quarters to extract
x	an object of qk_fn
...	additional arguments if applicable
id_item	a vector of id-items in form 'TICKER.ITEM' to extract data.frame from qk_fn object.
id	the ticker of the entity for extraction in qk_select. See id_item argument
item	a vector or list of vectors containing QUANTkiosk™ item codes (e.g. SALE, NI, AT, ATC) corresponding to US-GAAP, DEI, or other line item(s). See qk_fncodes()
using	internal flag to indicate input form.
i	extract rows from qk_select call.
j	extract colums from qk_select call.

Details

To view current QUANTkiosk™ item codes call qk_fncodes(). This will return a list organized by supported statements. This is constantly expanding.

For each item in items requested a request applied to every qkid in qkids. All responses are cached by default. This can be disabled using cache=FALSE.

For enterprise subscriptions, if view='pit' is requested the subsequent object may contain restated and as-filed values and times. Two functions are provided to resolve to unique observations at each date: pitAsFiled converts the series to only include the initial observation recorded, whereas pitAsOf reconciles restatements up to and including the date specified.

Value

A multivariate `xts` timeseries if `full=FALSE`, containing one column for each `qkid`, `item` pair. The column names are derived from the QKID metadata. See examples for more details.

If `full=TRUE` a nested-list of data.frames are returned with additional details for each item value. See `pfe.sale` sample data. The general object looks like:

- qkid (e.g. "0001045810.000.001S5TZJ6" [NVDA])
  - sale (xts)
  - eps (xts)
  - atc (xts)
- qkid (e.g. "0001564408.000.00441QMK5" [SNAP])
  - sale (xts)
  - eps (xts)
  - atc (xts)

Note

Depending on a user's subscription level, there may be restrictions on the universe of available companies and available line items. For `asof` queries, the the requests will resolve to the date of the query unless the subscription is for arbitrary point-in-time data. See https://quantkiosk.com/point-in-time for a full discussion of the details.

Author(s)

QUANTkiosk™ team.

References

SEC EDGAR® filing and reporting rules.

See Also

See Also qkid See Also qk_fncodes

Examples

```
## Not run:
# get list of all QK™ codes
fncodes <- qk_fncodes()

# Pfizer and J&J quarterly revenue (SALE) and net income (NI)
pfe <- qk_fn(qk_ticker(c("PFE","JNJ")), c("SALE","NI"))

tail( to_df(pfe) )
tail( qk_select(pfe, "JNJ.SALE") )
# single company, single item - Net Income
qkiditems(qk_ticker("AMD"), "NI")
```

```
# single company, multiple items - Revenue + Net Income
qkiditems(qk_ticker("AMD"), c("SALE","NI"))

# multiple companies, single item (used for each companies)
qkiditems(qk_ticker(c("AMD", "AAPL")), "NI")

# multiple companies, multiple items (used for each companies) - Assets + Assets Current
qkiditems(qk_ticker(c("AMD", "AAPL")), c("AT", "ATC"))

# multiple companies, multiple differnt items for each
qkiditems(qk_ticker(c("AMD", "AAPL")), list("NI",c("AT","ATC")))

## End(Not run)
```

---

qk_select *Convert and Display QUANTkiosk Objects*

Description

Generic functions to extract, transform and display QUANTkiosk api data to common R objects.

Usage

```
qk_select(x, ...)

to_df(x, ...)
as.qk_df(x, view="", showfull=FALSE, cls="qk_df", highlight=NULL)

to_ts(x, ...)
## S3 method for class 'qk_fn'
to_ts(x, i='fq', dt='fpe', ...)
## S3 method for class 'qk_df'
to_ts(x, i='fq', dt='fpe', ...)

full(x, showfull=TRUE, ...)
## S3 method for class 'qk_fn'
full(x, showfull=TRUE, ...)
## S3 method for class 'qk_df'
full(x, showfull=TRUE, ...)

## S3 method for class 'qk_fn'
subset(x, ...)
## S3 method for class 'qk_fn'
as.data.frame(x, ...)

## S3 method for class 'qk_df'
subset(x, ...)
## S3 method for class 'qk_df'
```

```
head(x, ...)
## S3 method for class 'qk_df'
tail(x, ...)

highlight(x, ...)
## S3 method for class 'data.frame'
highlight(x, i, color='yellow', bg, fg, ...)
```

Arguments

x	Object to transform. These are dispatched to object specific methods when appropriate.
view	internal view represented for use in pit functions
showfull	Display all hidden columns.
highlight	internal use
i	Row selection
dt	Column to use for indexing by time (to_ts)
color	background color for highlighting
bg	background color for highlighting. color names or ansi 256 colors.
fg	foreground color for highlighting. color names or ansi 256 colors.
cls	additional classes to add to conversion
...	Additional arguments depending on object class.

Details

Most data fetched from QUANTkiosk results in either a *QK* typed object or a `data.frame` of class `qk_df`.

Central to manipulation of these data, the package provides a variety of `s3` methods to interact with **base** or **stats** functionalty.

Examples

```
head(crox)

highlight(crox, 2:3)
crox |> highlight(2) |> highlight(3,'green')
```

qk_univ *Universe Function*

Description

Tools to Fetch Universe Data

Usage

```
qk_univ(univ, dt, src = "QK", cache=TRUE)
```

Arguments

univ	name of universe
dt	date for universe in form yyyymmdd. NOT IMPLEMENTED.
src	source of universe - 'QK' or local file path. NOT IMPLEMENTED.
cache	should data come from cache if available

Details

The QK universe function allows for named universe requests. Each universe is constructed using a set of rules.

Value

A qkid object containing the universe.

Note

Neither dt= or src= are implemented in this version.

Author(s)

QUANTkiosk™ team.

References

https://quantkiosk.com/qkid

See Also

See Also qkid

Examples

```
## Not run:
qk100 <- qk_univ("QK100")

# top 1000 widely firms
qk1k <- qk_univ("QK1000")

# next 2000 most widely held firms
qk2k <- qk_univ("QK2000")

## End(Not run)
```

---

qkid *QKID™ Tools*

Description

Functions to create, extract and convert entity and instrument identifiers within QUANTkiosk™'s QKID™ open symbology.

Usage

```
qk_search(x, type=c('issuer','manager','fund','person'), n=10)
qk_search_co(x, n=10)
qk_search_mgr(x, n=10)

qk_ticker(ticker, ...)
to_ticker(qkid, ...)

qk_figi(figi, type = c("figi", "shareClass", "composite"), ...)
to_figi(qkid, type = c("figi", "shareClass", "composite"), ...)

qk_cik(cik, ...)
to_cik(qkid, ...)

qk_permid(permid, type = c("org", "instrument", "quote"), ...)
to_permid(qkid, type = c("org", "instrument", "quote"), ...)

qk_name(name, ...)
to_name(qkid, ...)

qk_sic(sic, ...)
to_sic(qkid, ...)

qk_sector(sector, ...)
to_sector(qkid, ...)
```

```
# convert string version of QKID to class `qkid`
qkid(qkid, src="qkid", srcid=qkid, retrieved=yyyymmdd())

is.qkid(x)
as.qkid(x)

.qkidVersion(qkid)
classname(qkid)
detail(qkid)

entity(qkid)
cls(qkid)
instrument(qkid)
```

Arguments

x	object to convert or test, or name to search
qkid	qkid classed object or one that can be converted to.
cik	a string representing a CIK (Central Index Key)
permid	a string representing a PermID
ticker	a string representing a ticker
figi	a string representing a FIGI
name	a string representing a name
sic	Standard Industrial Classification (SIC) code
sector	QK™ Sector (see below)
...	unused
type	optional conversion specifics, or one of 'issuer' or 'fund' in qk_search_co or qk_search_mgr, respectively.
src	source used in construction (e.g. qk_ticker("AAPL") => 'ticker')
srcid	value used in construction (e.g. qk_ticker("AAPL") => 'AAPL')
retrieved	date of retrieved (or created) qkid
n	number of search matches to return

Details

R functions to allow for mapping and manipulation of QUANTkiosk™'s symbology as well as common identifiers using a special qkid class.

Value

QUANTkiosk™'s public identifier, known as 'QKID™' is the central identifier used by the QK™ API. It is freely licensed under CC-BY-SA and is designed to provide institutional quality symbology usable within professional asset management while being simple to use.

For a full discussion, please refer to official website below and the associated documentation. The 'QKID™' is a portmantaeu identifier, combining well-established entity and instrument identifiers into a linked symbology suitable for research and trading. There are three main components to a fixed-width 'QKID™':

'QKID™' can also be mapped to sectors and SIC codes.

- ENTITY.CLS.INSTRUMENT
  - ENTITY: 10 characters uniquely matching an entity
  - CLS: 4 character code indicating the instrument class
  - INSTRUMENT: a unique, immutable instrument ID

'cls' and 'detail' provide access to `qkid` object internals and details, returning broken-down identifiers or meta details about the id.

To facilitate discovery, **qkiosk** provides search functionality. `qk_search_co` offers fuzzy matching to public companies (e.g. Pfizer or McDonalds), returning a menu to select the match from. The return value is the matched 'QKID'. Similar functionality is available to find investment managers (e.g. D.E Shaw) via the `qk_search_mgr` function.

Note

Point-in-time QKID™ queries are currently not available outside of a subscription, this may change in the future.

Author(s)

QUANTkiosk™

References

Official QKID™ site: https://www.quantkiosk.com/qkid

Examples

```
# QKID™ string to class qkid
qkid("0000200406.0000.001S5SHQ9")

## Not run:
qk_search_co("tesla")
qk_search_mgr("pershing")

jnj.qkid <- qk_ticker("JNJ")
to_ticker(jnj.qkid)
to_cik(jnj.qkid)

detail(jnj.qkid)


faang <- qk_ticker(c("META","AAPL","AMZN","NFLX","GOOGL"))
entity(faang)
cls(faang)
```

```
instrument(faang)

cls(faang)

to_ticker(faang)
to_name(faang)
to_permid(faang)
to_figi(faang)
to_cik(faang)

## End(Not run)
```

---

sgcap *SG Capital Institutional Ownership Details (Aggregated)*

Description

Institutional Ownership API example of small manager filing.

Usage

```
sgcap
```

Format

A data frame with 76 observations on the following 22 variables.

`filerName` name of filer (i.e. fund name)

`filing` canonical EDGAR® filing path

`submissionType` filing type: 13F-HR or 13F-HR/A (amended)

`reportPeriod` period of report

`filedDate` when the data was filed

`issuer` name of instrument

`titleOfClass` title or class name of instrument

`issuerSIC` official standardized industrial code

`issuerSector` two-character sector code from QUANTkiosk™

`issuerTicker` common exchange ticker

`issuerQkid` QKID™ symbol

`value` fair market value of position at end of report period

`shrsOrPrnAmt` number of shares held or principal amount

`putCall` one of 'PUT', 'CALL' or empty

`shrsOrPrnAmtType` shares 'SH' or principal 'PRN'

`portWgt` portfolio weight (QK™ calculated)

`hasOtherManager` logical value indicating if submanagers are reported

`QtrsHeld` quarters held (QK™ calculated)

`QOQSshPrnAmt` quarter over quarter change in shares or principal amount of holding (QK™ calculated)

`QOQValue` quarter over quarter change in value of holding (QK™ calculated)

`QOQPortWgt` quarter over quarter change in portfolio weight (QK™ calculated)

`newOrDel` is position 'NEW' or 'DEL'eted. See details. (QK™ calculated)

Details

Institutional Ownership consists of quarterly '13F-HR' and amended reports '13F-HR/A' made by investors who hold at least $100 million of securities declared by the U.S. Securities and Exchange Commission. Filers include banks, insurance companies, and hedge funds. The filings are due 45 days after the end of each calendar quarter, and are filed via the EDGAR® system. The securities reported must be from the official '13(f) securities' list published by the SEC. See references section.

QUANTkiosk™ provides access to these filings in two forms at present. The first is an aggregated view that includes one row per unique instrument held. The second version (ex. `deshaw`) contains additional submanagers details. See the respective help page.

Additional fields not present in the official '13F-HR' or '13F-HR/A' but provided in both aggregated and non-aggregated views:

- issuerSIC
- issuerSector
- issuerQkid
- issuerTicker

These fields are useful for tracking changes between quarters as well as intrafiling analysis.

Note that `hasOtherManager` is set to `TRUE` or `FALSE` if the filing reports other manager holdings. See `deshaw` for more details on additional fields in non-aggregated views

Additional fields pre-calculated by QUANTkiosk™ across time:

- QOQSshPrnAmt
- QOQValue
- QOQPortWgt
- newOrDel
- QtrsHeld

QK™ calculates these fields as quarter over quarter change in absolute terms. `newOrDel` marks holdings that are 'NEW' this quarter or 'DEL'eleted. Deleted entries are not in the filing, but are provided to identify positions traded out of. `QtrsHeld` shows duration of a continuously held position in aggregate, or in the case of submanager API data, positions held continuously by the exact combination of submanagers. The latter is less reliable due to reporting considerations in the source data.

References

Ownership homepage: https://www.quantkiosk.com/ownership

Ownership Dashboard: https://one.quantkiosk.com/ownership

SG Capital Filings https://www.sec.gov/edgar/browse/?CIK=0001510099

Form 13F-HR FAQ: https://www.sec.gov/divisions/investment/13ffaq

Official 13(f) Securities https://www.sec.gov/divisions/investment/13flists

See Also

R function: qk_institutional Non-aggregated (Submanager) example: deshaw

Examples

```
str(sgcap)

## Not run:

# code that generates SG Capital Ownership data.
qk_search_mgr("sgcap")
sgcap_qkid <- "0001510099.0000.E0000WD77"
qk_institutional(sgcap_qkid, yyyyqq=202401, agg=TRUE)

## End(Not run)
```

---

yyyymmdd *Simple Dates*

Description

Simplified dates expressed as integers to facilitate easy API access.

Usage

```
yyyymmdd(x = Sys.time(), tz = Sys.getenv("TZ"))

today()

to_qq(yyyymmdd)
to_yyyy(yyyymmdd)
to_yyyyqq(yyyymmdd)

qtrs(n)
days(n)
```

```
## S3 method for class 'yyyymmdd'
seq(from, to, by, length.out = NULL, along.with = NULL, ...)

## S3 method for class 'yyyymmdd'
diff(x, lag = 1, differences = 1, ...)

is.yyyymmdd(x)

as.yyyymmdd(x, ...)

## S3 method for class 'Date'
as.yyyymmdd(x, ...)
## S3 method for class 'numeric'
as.yyyymmdd(x, ...)

## S3 method for class 'yyyymmdd'
as.Date(x, ...)
## S3 method for class 'yyyymmdd'
as.POSIXct(x, ...)
## S3 method for class 'yyyymmdd'
as.POSIXlt(x, ...)
```

Arguments

x	R object
yyyymmdd	an 8-digit date
tz	time zone to use internally.
n	integer value to offset - may be positive or negative
from	starting date. Required
to	end date. Optional.
by	increment of the sequence. Optional. See 'Details'.
length.out	integer, optional. Desired length of the sequence.
along.with	take the length from the length of this argument.
lag	an integer indicating which lag to use.
differences	an integer indicating the order of the difference.
...	arguments passed to or from other methdod.

Details

R has a weath of date functionality but lacks a coherent integer date representation that is capable of date math and proper timezone handling. This set of tools provides this.

Value

TBD

Author(s)

QUANTkiosk

References

Most date functionality is provided by conversion to R's `Date` internally and cast back to a `yyyymmdd` before returning. See `Date` and `seq.Date` for more details.

See Also

See also `today`

Examples

```
# today - no TZ
today()
is.yyyymmdd(today())

# same output as today() but possible to change TZ
yyyymmdd()

# sequence
seq(yyyymmdd(), yyyymmdd() + 10)

seq(yyyymmdd(), by='month', length.out=10)
seq(yyyymmdd(), by='month', along.with=1:5)

# difference between dates
diff(yyyymmdd()+c(1:5,14:20), lag = 1, differences = 1)

# add durations
yyyymmdd() + 10
yyyymmdd() + qtrs(2)

# subtract durations
yyyymmdd() + qtrs(-2)
yyyymmdd() - qtrs(2)

# convert to yyyymmdd
as.yyyymmdd(Sys.Date())
as.yyyymmdd(20240124)

# convert back to R classes
as.Date(yyyymmdd())
as.POSIXct(yyyymmdd())
```

Index