Package ‘AnVILBilling’

May 2, 2024

Title Provide functions to retrieve and report on usage expenses in NHGRI AnVIL (anvilproject.org).

Date 2020-09-30

Version 1.14.0

Description AnVILBilling helps monitor AnVIL-related costs in R, using queries to a BigQuery table to which costs are exported daily. Functions are defined to help categorize tasks and associated expenditures, and to visualize and explore expense profiles over time. This package will be expanded to help users estimate costs for specific task sets.

License Artistic-2.0

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports methods, DT, shiny, bigrquery, shinytoastr, DBI, magrittr, dplyr, lubridate, plotly, ggplot2

Suggests testthat, knitr, BiocStyle, rmarkdown

RoxygenNote 7.1.1

VignetteBuilder knitr

biocViews Infrastructure, Software

BugReports https://github.com/vjcitn/AnVILBilling/issues

git_url https://git.bioconductor.org/packages/AnVILBilling

git_branch RELEASE_3_19

git_last_commit 4f76977

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-02

Author BJ Stubbs [aut].
   Vince Carey [aut, cre]

Maintainer Vince Carey <stvjc@channing.harvard.edu>
Description

accessor for reckoning element

Usage

ab_reckoning(x)

Arguments

x  an instance of avReckoning

Value

a tibble with one row for each expense type by time slice

Examples

dim(ab_reckoning(demo_rec))
**browse_reck**

**Description**
 prototypical cost exploring app

**Usage**

```r
browse_reck()
```

**Value**

returns "NULL"

**Examples**

```r
if (interactive()) browse_reck()
```

---

**browse_reck2**

**alternate app for AnVIL where htmlwidgets misbehaves**

**Description**

alternate app for AnVIL where htmlwidgets misbehaves

**Usage**

```r
browse_reck2()
```

---

**demo_rec**

**a demonstration avReckoning object**

**Description**

a demonstration avReckoning object

**Usage**

```r
demo_rec
```

**Format**

avReckoning instance
getBilling

Note
This is a snapshot of cost data collected for a specific project.

Examples

demo_rec

getBilling

Description
request billing data

Usage

getBilling(
    startDate,  
    endDate,    
    bqProject,  
    bqDataset,  
    bqTable,    
    bqBilling_code,  
    page_size = 50000
)

Arguments

startDate character(1) date of start of reckoning
endDate character(1) date of end of reckoning
bqProject character(1) GCP project id
bqDataset character(1) GCP dataset id for billing data in BQ
bqTable character(1) GCP table for billing data in BQ
bqBilling_code character(1) GCP billing code
page_size numeric(1) passed to dbConnect

Value

tbl_df

Note
On 21 August 2020 VJC changed condition on endDate to <=
**getKeys**

**Examples**

```r
if (interactive()) {
  getBilling(startDate="2020-08-01",
              endDate="2020-08-15",
              bqProject="bjbilling",
              bqTable="gcp_billing_export_v1_015E39_38569D_3CC771",
              bqDataset="anvilbilling", bqBilling_code="landmarkanvil2")
}
```

---

**getKeys**

return keys

---

**Description**

return keys

**Usage**

getKeys(mybilling)

**Arguments**

mybilling tbl_df

**Value**

character()

---

**getSkus**

List the available GCP product skus

---

**Description**

List the available GCP product skus

**Usage**

getSkus(mybilling)

**Arguments**

mybilling tbl_df

**Value**

character()
getSubmissionCost  
*Calculate costs for a workflow submission by ID*

**Description**

Calculate costs for a workflow submission by ID

**Usage**

getSubmissionCost(mybilling, submissionID)

**Arguments**

- **mybilling**: tbl_df
- **submissionID**: character(1) Terra submission ID

**Value**

numeric()

**Examples**

data(demo_rec)  # makes rec
ev = getValues(demo_rec@reckoning, "terra-submission-id")[1]  # for instance
getSubmissionCost(demo_rec@reckoning, v)

---

getSubmissionRam  
*Calculate ram usage for a workflow submission by ID*

**Description**

Calculate ram usage for a workflow submission by ID

**Usage**

getSubmissionRam(mybilling, submissionID)

**Arguments**

- **mybilling**: tbl_df
- **submissionID**: character(1) Terra submission ID

**Value**

data.frame
**getValues**

**Examples**

\[
data(demo_rec) \# \text{ makes rec}
\]

\[
v = \text{getValues(demo_rec@reckoning, "terra-submission-id")}[1] \# \text{ for instance}
\]

\[
\text{getSubmissionRam(demo_rec@reckoning,v)}
\]

---

**getValues**

deal with nested tables in a reckoning

---

**Description**

deal with nested tables in a reckoning

**Usage**

\[
\text{getValues(mybilling, mykey)}
\]

**Arguments**

\[
\text{mybilling} \quad \text{tbl_df from reckon()}
\]

\[
\text{mykey} \quad \text{character(1) key}
\]

**Value**

character()

**Examples**

\[
\text{if (interactive()) getValues(reckoning(demo_rec), "security")}
\]

---

**reckon**

perform reckoning

---

**Description**

perform reckoning

**Usage**

\[
\text{reckon(obj)}
\]

**Arguments**

\[
\text{obj} \quad \text{instance of avReckoningRequest}
\]

**Value**

instance of avReckoning
Examples

```r
data(demo_rec)
if (interactive()) reckon(demo_rec)
```

### Description

generic for accessor for reckoning component

### Usage

```r
reckoning(x)
```

### Arguments

- `x` object inheriting from `avReckoning`

### Value

`tbl_df`

### Examples

```r
if (interactive()) reckoning(reckon(demo_rec))
```

---

### Description

accessor for reckoning component

### Usage

```r
# S4 method for signature 'avReckoning'
reckoning(x)
```

### Arguments

- `x` instance of `avReckoning`

### Value

`tbl_df`
setup_billing_request

Examples

    if (interactive()) reckoning(reckon(demo_rec))
**subsetByKeyValue**  
filter a reckoning by 'label' retaining records associated with a particular key-value pair

**Description**
filter a reckoning by 'label' retaining records associated with a particular key-value pair

**Usage**
subsetByKeyValue(mybilling, mykey, myvalue)

**Arguments**
- mybilling: instance of avReckoning  
- mykey: character(1)  
- myvalue: character(1)

**Value**
data.frame

**Examples**
```r
example(reckon) # makes rec
v = getValues(ab_reckoning(demo_rec), "terra-submission-id")[1] # for instance
nt = subsetByKeyValue(ab_reckoning(demo_rec), "terra-submission-id", v)
head(nt)
dim(nt)
```

**subsetBySku**  
subset a billing object by sku

**Description**
subset a billing object by sku

**Usage**
subsetBySku(mybilling, mysku)

**Arguments**
- mybilling: tbl_df  
- mysku: character(1) GCP product sku
subsetBySku

Value

data.frame
Index

* datasets
  demo_rec, 3
  ab_reckoning, 2
  browse_reck, 3
  browse_reck2, 3
  demo_rec, 3
  getBilling, 4
  getKeys, 5
  getSkus, 5
  getSubmissionCost, 6
  getSubmissionRam, 6
  getValues, 7
  reckon, 7
  reckoning, 8
  reckoning, avReckoning-method, 8
  setup_billing_request, 9
  subsetByKeyValue, 10
  subsetBySku, 10