

Package ‘AvenueAPI’

August 9, 2017

Type Package

Title Retrieve Data from 7Park Data's Avenue API

Version 0.2

Author 7Park Data

Maintainer Derek Darves <derek@7parkdata.com>

Description Provides R-methods to retrieve and download metrics from the 7Park Data Avenue API.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.1.0)

Imports httr,
rjson,
reshape2,
dplyr,
lubridate,
xts,
zoo,
methods

URL <https://github.com/sevenpark/AvenueAPI-R>

RoxygenNote 6.0.1

Collate 'Avenue_Auth_Methods.R'
'data_resources.R'

R topics documented:

cc_names	2
check_api_key	2
check_date	3
check_dom_source	3
check_rev_firm	4
check_rev_source	4
connect_avenue	5
extension_names	5
extract_metrics	6
fetch_app_series	6

fetch_data	7
fetch_revenue_series	7
fetch_traffic_series	8
m1_names	9
m2_names	9
transform_avenue_series	10
Index	11

cc_names	<i>Valid Firm Names for the Credit Card Data Series</i>
----------	---

Description

Valid Firm Names for the Credit Card Data Series

Usage

cc_names

Format

A character vector of length 13

check_api_key	<i>Function to check the validity of an Avenue API key</i>
---------------	--

Description

Function to check the validity of an Avenue API key

Usage

check_api_key(object)

Arguments

object	character; typically, this function is called to validate the api_key slot of an object of class AvenueApiClient. It may also be used to examine a length 1 character vector.
--------	---

Value

An un-altered api key or, in the case of warnings, message(s) collected from the evaluation process.

`check_date`*Function to check the validity of date inputs*

Description

Function to check the validity of date inputs

Usage

```
check_date(x, future_warning = FALSE)
```

Arguments

<code>x</code>	Date; a length 1 Date or a string expressed in a standard unambiguous format. Descriptive errors are produced for numeric inputs or invalid string formats.
<code>future_warning</code>	logical; When TRUE, a warning message is produced if the supplied date is greater than current date + 1 day.

Value

When no errors are triggered, `check_date` returns a valid Date object equal to the input.

Examples

```
## Not run:
check_date(3)
Error in check_date(3) :
  '3' is an invalid date selection of class 'numeric'.
  You must supply a valid date object or a character of form 'YYYY-MM-DD'!

## End(Not run)
```

`check_dom_source`*Function to check the validity of an Avenue traffic domain*

Description

Function to check the validity of an Avenue traffic domain

Usage

```
check_dom_source(x)
```

Arguments

<code>x</code>	character; a length 1 character vector of the domain you are requesting, e.g. 'google.com'
----------------	--

Value

When no errors are triggered, `check_dom_source` returns an appropriate domain name which is passed to the Avenue API.

check_rev_firm	<i>Check that a specified company name is contained within the 7Parkdata Avenue API for a given revenue data source</i>
----------------	---

Description

Check that a specified company name is contained within the 7Parkdata Avenue API for a given revenue data source

Usage

```
check_rev_firm(x, data_source = "", validate_name = NULL)
```

Arguments

x	character; a company name
data_source	character; a data source name, one of m1, m2, or cc
validate_name	logical; when TRUE (the default) TRUE is tested against a vector of known/valid names.

check_rev_source	<i>Function to check the validity of an Avenue Revenue series source input</i>
------------------	--

Description

Function to check the validity of an Avenue Revenue series source input

Usage

```
check_rev_source(x)
```

Arguments

x	character; a length 1 character vector and one of "cc" (credit card panel), "m1" (Merchant Intel data), or "m2" (Merchant Intel2 data).
---	---

Value

When no errors are triggered, check_rev_source returns an appropriate dataset name which is passed to the Avenue API.

connect_avenue	<i>Set credentials for a 7Parkdata Avenue API connection.</i>
----------------	---

Description

Set credentials for a 7Parkdata Avenue API connection.

Usage

```
connect_avenue(api_key = "")
```

Arguments

api_key	character; a valid Avenue API key. Must be a length 1 character vector. Contact your 7Park Sales Representative to obtain your key
---------	--

Examples

```
## Not run:  
connection <- connect_avenue(api_key=good_key)  
  
## End(Not run)
```

extension_names	<i>Valid Domain Names for the Avenue Extension Series</i>
-----------------	---

Description

Valid Domain Names for the Avenue Extension Series

Usage

```
extension_names
```

Format

A character vector of length 12,092.

extract_metrics	<i>Helper function to transform an AvenueAPI data object from list to data.frame format</i>
-----------------	---

Description

Helper function to transform an AvenueAPI data object from list to data.frame format

Usage

```
extract_metrics(metrics = metrics, data = NULL)
```

Arguments

metrics	character; the metric names of the supplied object
data	list; a data set extracted from the <code>.\$data</code> slot of an object of class "AvenueAPI"

Value

a "long" data.frame of the requested metrics

fetch_app_series	<i>fetch_app_series Methods</i>
------------------	---------------------------------

Description

fetch_app_series Methods

Usage

```
fetch_app_series(.Object, app, cadence, start_date = "", end_date = "",
  country_code = "", region = NULL)
```

```
## S4 method for signature 'AvenueApiClient'
```

```
fetch_app_series(.Object, app = NULL,
  cadence = NULL, start_date = "", end_date = "", country_code = "",
  region = NULL)
```

Arguments

.Object	A valid AvenueApiClient connection
app	character; an length-1 vector of the Android package name, e.g. "com.facebook.katana"
cadence	character; the cadence requested. Must be one of: 'daily' (daily app usage) or 'weekly' (weekly app usage)
start_date	string; a length-one character vector expressed as 'YYYY-MM-DD' (ISO 8601) representing the start date of the requested series
end_date	string; a length-one character vector expressed as 'YYYY-MM-DD' (ISO 8601) representing the end date of the requested series

country_code	character; the ISO 3166 alpha-2 country_code for which you are requesting app data. If left unset, the API returns data for all countries.
region	character; the UN region code for which you are requesting app data. An error is thrown if both region and country_code are set.

Examples

```
## Not run:
# Get app usage data for Facebook
fb <- fetch_app_series(gave, app="com.facebook.katana",
                      cadence="daily",
                      start_date='2015-01-01',
                      country_code = 'US')

## End(Not run)
```

fetch_data	<i>fetch_data Method.</i>
------------	---------------------------

Description

fetch_data Method.

Usage

```
fetch_data(.Object, ave_url, params = "")

## S4 method for signature 'AvenueApiClient'
fetch_data(.Object, ave_url, params = "")
```

Arguments

.Object	A valid AvenueApiClient connection
ave_url	Fully constructed Avenue API URL
params	Params passed to the API

fetch_revenue_series	<i>fetch_revenue_series Methods</i>
----------------------	-------------------------------------

Description

fetch_revenue_series Methods

Usage

```
fetch_revenue_series(.Object, firm, data_source, start_date = "",
                    end_date = "", validate_name = TRUE)

## S4 method for signature 'AvenueApiClient'
fetch_revenue_series(.Object, firm = NULL,
                    data_source = NULL, start_date = "", end_date = "",
                    validate_name = TRUE)
```

Arguments

.Object	A valid AvenueApiClient connection
firm	character; an length-1 vector of the company name, e.g. "Chipotle"
data_source	character; the source dataset. Must be one of: 'cc' (credit card panel), 'm1' (Merchant Intel), 'm2' (Merchant Intel2)
start_date	string; a length-one character vector expressed as 'YYYY-MM-DD' (ISO 8601) representing the start date of series
end_date	string; a length-one character vector expressed as 'YYYY-MM-DD' (ISO 8601) representing the end date of series
validate_name	logical; when TRUE, AvenueAPI checks the supplied firm name against a vector of known/valid names.

Examples

```
## Not run:
# Get credit card data for Chipotle
chip <- fetch_revenue_series(connection,
  firm="Chipotle",
  data_source="cc",
  start_date='2015-01-01')

## End(Not run)
```

fetch_traffic_series *fetch_traffic_series Methods*

Description

fetch_traffic_series Methods

Usage

```
fetch_traffic_series(.Object, domain = NULL, platform = "",
  dataseries = "", start_date = "", end_date = "", country_code = "",
  validate_name = TRUE)
```

```
## S4 method for signature 'AvenueApiClient'
fetch_traffic_series(.Object, domain = NULL,
  platform = "PC", dataseries = "extension", start_date = "",
  end_date = "", country_code = "US", validate_name = TRUE)
```

Arguments

.Object	A valid AvenueApiClient connection
domain	character; an length-1 vector of the requested domain name, e.g. "www.google.com"
platform	character; the user's computing platform. Must be one of: 'PC' (desktop browser data), 'mobile' (mobile web browser data), or 'ALL' (mobile + PC).
dataseries	character; currently, the only valid value for this parameter is "extension" (the default). This will be updated as new traffic series are added to the API.

start_date	string; a length-one character vector expressed as 'YYYY-MM-DD' (ISO 8601) representing the start date of the requested series
end_date	string; a length-one character vector expressed as 'YYYY-MM-DD' (ISO 8601) representing the end date of the requested series
country_code	character; the ISO 3166 alpha-2 country_code for which you are requesting traffic data. If left unset, the API returns data for all countries.
validate_name	logical; when TRUE, AvenueAPI checks the supplied domain name against a vector of known/valid names.

Examples

```
## Not run:
# Get domain traffic data for google.com, US
goog <- fetch_traffic_series(conection, domain = 'google.com',
                             platform = 'PC',
                             start_date='2014-01-01',
                             country_code = 'US',
                             validate_name = TRUE)

## End(Not run)
```

m1_names

Valid Firm Names for the Merchant Intel Data Series

Description

Valid Firm Names for the Merchant Intel Data Series

Usage

```
m1_names
```

Format

A character vector of length 1,311

m2_names

Valid Firm Names for the Merchant Intel2 Data Series

Description

Valid Firm Names for the Merchant Intel2 Data Series

Usage

```
m2_names
```

Format

A character vector of length 571

`transform_avenue_series`

Transform a 7Parkdata Avenue API return into a "long" or "wide" dataframe for analysis.

Description

Transform a 7Parkdata Avenue API return into a "long" or "wide" dataframe for analysis.

Usage

```
transform_avenue_series(x, wide = FALSE)
```

Arguments

<code>x</code>	list; a 7Park Data revenue object pulled with <code>fetch_*_series</code> method.
<code>wide</code>	logical; if TRUE, a "wide" data frame is returned using dcast

Index

*Topic **datasets**

- cc_names, [2](#)
- extension_names, [5](#)
- m1_names, [9](#)
- m2_names, [9](#)

- cc_names, [2](#)
- check_api_key, [2](#)
- check_date, [3](#)
- check_dom_source, [3](#)
- check_rev_firm, [4](#)
- check_rev_source, [4](#)
- connect_avenue, [5](#)

- dcast, [10](#)

- extension_names, [5](#)
- extract_metrics, [6](#)

- fetch_app_series, [6](#)
- fetch_app_series, AvenueApiClient-method
(fetch_app_series), [6](#)
- fetch_data, [7](#)
- fetch_data, AvenueApiClient-method
(fetch_data), [7](#)
- fetch_revenue_series, [7](#)
- fetch_revenue_series, AvenueApiClient-method
(fetch_revenue_series), [7](#)
- fetch_traffic_series, [8](#)
- fetch_traffic_series, AvenueApiClient-method
(fetch_traffic_series), [8](#)

- m1_names, [9](#)
- m2_names, [9](#)

- transform_avenue_series, [10](#)