Package 'rechonest'

October 14, 2022

Type Package

Title R Interface to Echo Nest API

Version 1.2

Date 2016-03-16

Author Mukul Chaware[aut,cre]

Maintainer Mukul Chaware <mukul.chaware13@gmail.com>

Description The 'Echo nest' <http://the.echonest.com> is the industry's leading music intelligence company, providing developer with deepest understanding of music content and music fans. This package can be used to access artist's data including songs, blogs, news, reviews etc. Song's data including audio summary, style, danceability, tempo etc can also be accessed.

URL https://github.com/mukul13/rechonest

License MIT + file LICENSE

LazyData TRUE

Imports httr,RCurl,jsonlite

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-18 00:00:16

R topics documented:

---

basic_playlist	*To return basic playlist*

Description

To return basic playlist

Usage

```
basic_playlist(api_key, type = NA, artist_id = NA, artist = NA,
  song_id = NA, genre = NA, track_id = NA, results = 15, partner = NA,
  tracks = F, limited_interactivity = NA)
```

Arguments

api_key	Echo Nest API key
type	the type of the playlist to be generated
artist_id	artist id
artist	artist name
song_id	song ID
genre	genre name
track_id	track ID
results	the number of results desired
partner	partner catalog
tracks	tracks info
limited_interactivity	
	interactivity limitation

Value

data frame giving basic playlist

Examples

```
## Not run:
data=basic_playlist(api_key,type="artist-radio",artist=c("coldplay","adele"))

## End(Not run)
```

---

extract_artist_names   *To extract artist names from text.*

Description

To extract artist names from text.

Usage

```
extract_artist_names(api_key, text, min_hotttnesss = NA,
  max_hotttnesss = NA, min_familiarity = NA, max_familiarity = NA,
  sort = NA, results = NA)
```

Arguments

api_key	Echo Nest API key
text	text that contains artist names
min_hotttnesss	the minimum hotttnesss for returned artists
max_hotttnesss	the maximum hotttnesss for returned artists
min_familiarity	
	the minimum familiarity for returned artists
max_familiarity	
	the maximum familiarity for returned artists
sort	specified the sort order of the results
results	the number of results desired

Value

data frame giving artist's names

Examples

```
## Not run:
data=extract_artist_names(api_key,text="I like adele and Maroon 5")

## End(Not run)
```

---

get_artist_biographies
                        To get a list of artist biographies

Description

To get a list of artist biographies

Usage

```
get_artist_biographies(api_key, name = NA, id = NA, start = NA,
  results = 15, license = "unknown")
```

Arguments

api_key	Echo Nest API key
name	artist name
id	Echo Nest ID
start	the desired index of the first result returned
results	the number of results desired
license	the desired licenses of the returned images

Value

data frame giving artist's biographies

Examples

```
## Not run:
data=get_artist_biographies(api_key,name="coldplay")

## End(Not run)
```

---

get_artist_blogs   *To get blogs about artist*

Description

To get blogs about artist

Usage

```
get_artist_blogs(api_key, name = NA, start = NA, id = NA, results = 15,
  high_relevance = F)
```

Arguments

`api_key`	Echo Nest API key
`name`	artist's name
`start`	the desired index of the first result returned
`id`	artist's id
`results`	maximum size
`high_relevance`	if true only items that are highly relevant for this artist will be returned

Value

data frame giving blogs about artist

Examples

```
## Not run:
data=get_artist_blogs(api_key,name="coldplay",results=35)

## End(Not run)
```

`get_artist_data`	*To get artist's data*

Description

To get artist's data

Usage

```
get_artist_data(api_key, name = NA, id = NA, hotttnesss = T, terms = F,
  blogs = F, news = F, familiarity = F, audio = F, images = F,
  songs = F, reviews = F, discovery = F, partner = NA,
  biographies = F, doc_counts = F, artist_location = F,
  years_active = F, urls = F)
```

Arguments

`api_key`	Echo Nest API key
`name`	artist's name
`id`	artist's id
`hotttnesss`	artist's hotttnesss
`terms`	artist's terms
`blogs`	blogs about artist
`news`	news articles about artist
`familiarity`	artist's familiarity

`audio`	artist's audio details
`images`	artist's images details
`songs`	artist's songs details
`reviews`	reviews about artist
`discovery`	artist's discovery details
`partner`	partner catalog
`biographies`	artist's biographies
`doc_counts`	artist's doc_counts
`artist_location`	
	artist location
`years_active`	years active
`urls`	urls of artist websites

Value

data frame giving artist's hotttnesss

Examples

```
## Not run:
data=get_artist_data(api_key,name="coldplay",terms=T,blogs=T)

## End(Not run)
```

`get_artist_familiarity`

To get artist's familiarity

Description

To get artist's familiarity

Usage

```
get_artist_familiarity(api_key, name = NA, id = NA)
```

Arguments

`api_key`	Echo Nest API key
`name`	artist's name
`id`	artist's id

Value

data frame giving artist's familiarity

Examples

```
## Not run:
data=get_artist_familiarity(api_key,name="coldplay")

## End(Not run)
```

---

get_artist_hotttnesss   *To get artist's hotttnesss*

Description

To get artist's hotttnesss

Usage

```
get_artist_hotttnesss(api_key, name = NA, id = NA)
```

Arguments

api_key	Echo Nest API key
name	artist's name
id	artist's id

Value

data frame giving artist's hotttnesss

Examples

```
## Not run:
data=get_artist_hotttnesss(api_key,name="coldplay")

## End(Not run)
```

---

get_artist_images   *To get artist's images*

Description

To get artist's images

Usage

```
get_artist_images(api_key, name = NA, id = NA, start = NA, results = 15,
  license = "unknown")
```

Arguments

`api_key`	Echo Nest API key
`name`	artist name
`id`	Echo Nest ID
`start`	the desired index of the first result returned
`results`	the number of results desired
`license`	the desired licenses of the returned images

Value

data frame giving artist's images

Examples

```
## Not run:
data=list_genres(api_key)

## End(Not run)
```

`get_artist_news`   *To get news about artist*

Description

To get news about artist

Usage

```
get_artist_news(api_key, name = NA, id = NA, start = NA, results = 15,
  high_relevance = F)
```

Arguments

`api_key`	Echo Nest API key
`name`	artist's name
`id`	artist's id
`start`	the desired index of the first result returned
`results`	maximum size
`high_relevance`	if true only items that are highly relevant for this artist will be returned

Value

data frame giving news about artist

Examples

```
## Not run:
data=get_artist_news(api_key,name="coldplay",results=35)

## End(Not run)
```

---

get_artist_reviews   *To get reviews about artist*

Description

To get reviews about artist

Usage

```
get_artist_reviews(api_key, name = NA, id = NA, start = NA,
  results = 15)
```

Arguments

api_key	Echo Nest API key
name	artist's name
id	artist's id
start	the desired index of the first result returned
results	maximum size

Value

data frame giving blogs about artist

Examples

```
## Not run:
data=get_artist_reviews(api_key,name="coldplay",results=35)

## End(Not run)
```

`get_artist_songs`   *To get artist's songs*

Description

To get artist's songs

Usage

```
get_artist_songs(api_key, name = NA, id = NA, start = NA, results = 15)
```

Arguments

api_key	Echo Nest API key
name	artist's name
id	artist's id
start	the desired index of the first result returned
results	maximum size

Value

data frame giving artist's songs

Examples

```
## Not run:
data=get_artist_songs(api_key,name="coldplay")

## End(Not run)
```

`get_artist_terms`   *To get artist's terms*

Description

To get artist's terms

Usage

```
get_artist_terms(api_key, name = NA, id = NA)
```

Arguments

api_key	Echo Nest API key
name	artist's name
id	artist's id

Value

data frame giving artist's terms

Examples

```
## Not run:
data=get_artist_terms(api_key,name="coldplay")

## End(Not run)
```

---

get_artist_videos   *To get a list of video documents found on the web related to an artist*

Description

To get a list of video documents found on the web related to an artist

Usage

```
get_artist_videos(api_key, name = NA, id = NA, start = NA, results = 15)
```

Arguments

api_key	Echo Nest API key
name	artist name
id	Echo Nest ID
start	the desired index of the first result returned
results	the number of results desired

Value

data frame giving artist's videos

Examples

```
## Not run:
data=get_artist_videos(api_key,name="coldplay")

## End(Not run)
```

---

get_genre_info   *To get basic information about a genre*

Description

To get basic information about a genre

Usage

```
get_genre_info(api_key, genre, description = T, urls = T)
```

Arguments

api_key	Echo Nest API key
genre	the genre name
description	genre's description
urls	genre's urls

Value

data frame giving basic info about a genre

Examples

```
## Not run:
data=get_genre_info(api_key,genre="post rock")

## End(Not run)
```

---

get_top_genre_artists *To Return the top artists for the given genre*

Description

To Return the top artists for the given genre

Usage

```
get_top_genre_artists(api_key, genre)
```

Arguments

api_key	Echo Nest API key
genre	the genre name

Value

data frame top artist of the given genre

Examples

```
## Not run:
data=get_top_genre_artists(api_key,genre="pop")

## End(Not run)
```

---

get_top_hottt   *To return a list of the top hottt artists*

Description

To return a list of the top hottt artists

Usage

```
get_top_hottt(api_key, genre = NA, start = NA, results = 15)
```

Arguments

api_key	Echo Nest API key
genre	the set of genres of interest
start	the desired index of the first result returned
results	the number of results desired

Value

data frame giving top hottt artists

Examples

```
## Not run:
data=get_top_hottt(api_key)

## End(Not run)
```

get_top_terms   *To returns a list of the overall top terms*

Description

To returns a list of the overall top terms

Usage

```
get_top_terms(api_key, results = NA)
```

Arguments

api_key   Echo Nest API key

results   the number of results desired

Value

data frame giving top terms

Examples

```
## Not run:
data=get_top_terms(api_key)

## End(Not run)
```

get_twitter_handle   *To get the twitter handle for an artist*

Description

To get the twitter handle for an artist

Usage

```
get_twitter_handle(api_key, name = NA, id = NA)
```

Arguments

api_key   Echo Nest API key

name   artist name

id   Echo Nest ID

Value

data frame giving twitter handle

Examples

```
## Not run:
data=get_twitter_handle(api_key,name="coldplay")

## End(Not run)
```

---

list_genres   *To get genre's list*

Description

To get genre's list

Usage

```
list_genres(api_key)
```

Arguments

api_key   Echo Nest API key

Value

data frame giving genre's list

Examples

```
## Not run:
data=list_genres(api_key)

## End(Not run)
```

---

list_terms   *To get a list of the best typed descriptive terms*

Description

To get a list of the best typed descriptive terms

Usage

```
list_terms(api_key, type = "style")
```

Arguments

api_key	Echo Nest API key
type	term type

Value

data frame giving best typed descriptive terms

Examples

```
## Not run:
data=list_terms(api_key)

## End(Not run)
```

---

search_artist   *To search artist by using name*

Description

To search artist by using name

Usage

```
search_artist(api_key, name = NA, style = NA, hotttnesss = T,
  description = NA, start = NA, results = 15, sort = NA, partner = NA,
  artist_location = NA, genre = NA, mood = NA, rank_type = "relevance",
  fuzzy_match = F, max_familiarity = NA, min_familiarity = NA,
  max_hotttnesss = NA, min_hotttnesss = NA, artist_start_year_before = NA,
  artist_start_year_after = NA, artist_end_year_before = NA,
  artist_end_year_after = NA)
```

Arguments

`api_key`	Echo Nest API key
`name`	artist's name
`style`	artist's style
`hotttnesss`	artist's hotttnesss (Default is true)
`description`	artist's description
`start`	the desired index of the first result returned
`results`	maximum size
`sort`	to sort ascending or descending
`partner`	partner catalog
`artist_location`	
	artist location
`genre`	genre name
`mood`	mood like happy or sad
`rank_type`	For search by description, style or mood indicates whether results should be ranked by query relevance or by artist familiarity
`fuzzy_match`	if true, a fuzzy search is performed
`max_familiarity`	
	maximum familiarity
`min_familiarity`	
	minimum familiarity
`max_hotttnesss`	maximum hotttnesss
`min_hotttnesss`	minimum hotttnesss
`artist_start_year_before`	
	Matches artists that have an earliest start year before the given value
`artist_start_year_after`	
	Matches artists that have an earliest start year after the given value
`artist_end_year_before`	
	Matches artists that have a latest end year before the given value
`artist_end_year_after`	
	Matches artists that have a latest end year after the given value

Value

data frame giving artist's data

Examples

```
## Not run:
data=search_artist(api_key,"coldplay",sort="hotttnesss-desc",results=50)

## End(Not run)
```

---

search_genre   *To search for genres by name*

Description

To search for genres by name

Usage

```
search_genre(api_key, genre = NA, description = T, urls = T,
  results = 15)
```

Arguments

api_key	Echo Nest API key
genre	the genre name
description	genre's description
urls	genre's urls
results	the number of results desired

Value

data frame giving searched genres

Examples

```
## Not run:
data=search_genre(api_key,genre="rock")\

## End(Not run)
```

---

search_songs   *To search song*

Description

To search song

Usage

```
search_songs(api_key, artist = NA, artist_id = NA, title = NA,
  hotttnesss = T, style = NA, artist_location = T, combined = NA,
  sort = NA, audio_summary = F, partner = NA, min_name = NA,
  discovery = T, max_name = NA, min_val = NA, max_val = NA,
  start = NA, results = 15, mode = NA, key = NA, currency = T,
  description = NA, rank_type = "relevance", mood = NA, familiarity = T,
  song_type = NA, artist_start_year_before = NA,
  artist_start_year_after = NA, artist_end_year_before = NA,
  artist_end_year_after = NA)
```

Arguments

`api_key`	Echo Nest API key
`artist`	artist's name
`artist_id`	artist's id
`title`	song's title
`hotttnesss`	song's hotttnesss
`style`	artist's style
`artist_location`	
	artist location
`combined`	query both artist and title fields
`sort`	to sort ascending or descending
`audio_summary`	song's audio summary
`partner`	partner catalog
`min_name`	features' minimum value settings
`discovery`	artist's discovery measure
`max_name`	features' maximum value settings
`min_val`	features' minimum value settings
`max_val`	features' maximum value settings
`start`	the desired index of the first result returned
`results`	maximum size
`mode`	the mode of songs
`key`	the key of songs in the playlist
`currency`	song currency
`description`	song's description
`rank_type`	For search by description, style or mood indicates whether results should be ranked by query relevance or by artist familiarity
`mood`	a mood like happy or sad
`familiarity`	song's familiarity
`song_type`	controls the type of songs returned

artist_start_year_before

                  Matches artists that have an earliest start year before the given value

artist_start_year_after

                  Matches artists that have an earliest start year after the given value

artist_end_year_before

                  Matches artists that have a latest end year before the given value

artist_end_year_after

                  Matches artists that have a latest end year after the given value

Value

data frame giving artist's familiarity

Examples

```
## Not run:
data=search_songs(api_key,style="pop",results=31)

## End(Not run)
```

---

similar_artists   *To search similar artists by using names or IDs*

Description

To search similar artists by using names or IDs

Usage

```
similar_artists(api_key, name = NA, id = NA, seed_catalog = NA,
  hotttnesss = T, start = 0, results = 15, max_familiarity = NA,
  min_familiarity = NA, max_hotttnesss = NA, min_hotttnesss = NA,
  artist_start_year_before = NA, artist_start_year_after = NA,
  artist_end_year_before = NA, artist_end_year_after = NA)
```

Arguments

api_key	Echo Nest API key
name	artists' name (maximum upto 5 names)
id	Echo Nest IDs (maximum upto 5 IDs)
seed_catalog	seed catalog
hotttnesss	artist's hotttnesss
start	the desired index of the first result returned
results	maximum size
max_familiarity	
	maximum familiarity

min_familiarity
>   minimum familiarity

max_hotttnesss   maximum hotttnesss

min_hotttnesss   minimum hotttnesss

artist_start_year_before
>   Matches artists that have an earliest start year before the given value

artist_start_year_after
>   Matches artists that have an earliest start year after the given value

artist_end_year_before
>   Matches artists that have a latest end year before the given value

artist_end_year_after
>   Matches artists that have a latest end year after the given value

Value

data frame giving similar artists' data

Examples

```
## Not run:
data=similar_artists(api_key,name=c("coldplay","adele","maroon 5"),results=35 )

## End(Not run)
```

---

similar_genres   *To return similar genres to a given genre*

Description

To return similar genres to a given genre

Usage

```
similar_genres(api_key, genre = NA, description = T, urls = T,
  start = NA, results = 15)
```

Arguments

api_key	Echo Nest API key
genre	the genre name
description	genre's description
urls	genre's urls
start	the desired index of the first result returned
results	the number of results desired

Value

data frame giving similar genres

Examples

```
## Not run:
data=similar_genres(api_key,genre="rock")

## End(Not run)
```

---

standard_static_playlist

To return standard static playlist

Description

To return standard static playlist

Usage

```
standard_static_playlist(api_key, type = NA, artist_id = NA, artist = NA,
  song_id = NA, genre = NA, track_id = NA, results = 15, partner = NA,
  tracks = F, limited_interactivity = NA, song_selection = NA,
  variety = NA, distribution = NA, adventurousness = NA,
  seed_catalog = NA, sort = NA, song_type = NA)
```

Arguments

api_key	Echo Nest API key
type	the type of the playlist to be generated
artist_id	artist id
artist	artist name
song_id	song ID
genre	genre name
track_id	track ID
results	the number of results desired
partner	partner catalog
tracks	tracks info
limited_interactivity	
	interactivity limitation
song_selection	to determine how songs are selected from each artist in artist-type playlists
variety	the maximum variety of artists to be represented in the playlist
distribution	controls the distribution of artists in the playlist

    adventurousness

                   controls the trade between known music and unknown music

    seed_catalog   ID of seed catalog for the playlist

    sort   sorting parameter

    song_type   controls the type of songs returned

Value

data frame giving standard static playlist

Examples

```
## Not run:
data= standard_static_playlist(api_key,type="artist-radio",artist=c("coldplay","adele"))

## End(Not run)
```

---

suggest_artist_names   *To suggest artists based upon partial names*

Description

To suggest artists based upon partial names

Usage

```
suggest_artist_names(api_key, name, results = NA)
```

Arguments

    api_key   Echo Nest API key

    name   a partial artist name

    results   the number of results desired (maximum 15)

Value

data frame giving artist's names

Examples

```
## Not run:
data=suggest_artist_names(api_key,"cold")

## End(Not run)
```

Index