Package 'rtrek'

June 6, 2018

Version 0.1.0

Title Datasets and Functions Relating to Star Trek

Description Provides datasets related to the Star Trek fictional universe and functions for working with the data.
The package also provides access to real world datasets based on the televised series and other related licensed media productions.
It interfaces with Wikipedia (<https://www.wikipedia.org/>), the Star Trek API (STAPI) (<http://stapi.co/>), Memory Alpha (<http://memory-alpha.wikia.com/wiki/Portal:Main>), and Memory Beta (<http://memory-beta.wikia.com/wiki/Main_Page>)
to retrieve data, metadata and other information relating to Star Trek.
It also contains local datasets covering a variety of topics such as Star Trek universe species data, geopolitical data,
and summary datasets resulting from text mining analyses of Star Trek novels.
The package also provides functions for working with data from other Star Trek-related R data packages containing larger datasets not stored in 'rtrek'.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

ByteCompile true

URL https://github.com/leonawicz/rtrek

BugReports https://github.com/leonawicz/rtrek/issues

Depends R (>= 3.4.0)

Suggests testthat, knitr, rmarkdown, covr, leaflet, lintr, ggplot2, showtext, sysfonts, trekfont

Imports magrittr, dplyr, jsonlite, purrr

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Matthew Leonawicz [aut, cre]

Maintainer Matthew Leonawicz <mfleonawicz@alaska.edu>

Repository CRAN

Date/Publication 2018-06-06 11:28:00 UTC

R topics documented:

`rtrek` *rtrek: Star Trek datasets and related R functions.*

Description

The rtrek package contains a collection of Star Trek-themed datasets and R functions to assist with their use. The package interfaces with Wikipedia, Memory Alpha and Memory Beta to retrieve data, metadata and other information relating to the Star Trek fictional universe. It also contains local datasets covering a variety of topics such as Star Trek universe species data, geopolitical data, and datasets resulting from text mining analyses of Star Trek novels.

`stapi` *Retrieve Star Trek data from STAPI*

Description

Retrieve Star Trek data from the Star Trek API (STAPI).

Usage

```
stapi(id, page = 1, uid = NULL, page_count = FALSE)
```

Arguments

id	character, name of STAPI entity. See details.
page	integer vector, defaults to first page.
uid	NULL for search mode, character for extraction mode. See details.
page_count	logical, set to TRUE to do a preliminary check of the total number a pages of results available for a potential entity search. This will only have the impact of searching the first page.

Details

See stapiEntities for all the currently available API entities. These are the IDs for dataset collections or categories passed to id.

The universal ID uid can be supplied to retrieve a more specific subset of data. By default, uid = NULL and stapi operates in search mode. As part of a stepwise process, you can first use search mode. Then if the resulting data frame includes a uid column, you can make a second call to the function providing a specific uid. This puts stapi into extraction mode and will return satellite data associated with the unique entry from the original general sweep of the entity id.

rtrek employs anti-DDOS measures. It will not perform an API call to STAPI more than once per second. To be an even better neighbor, you can increase this wait time using options, e.g. options(rtrek_antiddos = 10) to increase the minimum time between API calls to ten seconds. Values less than one are ignored (defaulting back to one second) and a warning will be thrown when making any API call if this is the case.

Currently STAPI contains primarily real world data such as episode air dates, movie metadata, or production company information. Fictional world data is secondary and more limited.

Value

a data frame in search mode, a list in extraction mode, and nothing is returned in page count check mode but the result is printed to the console.

Examples

```
library(dplyr)
stapi("character", page_count = TRUE) # check first
stapi("character", page = 2) %>% select(1:2)
Q <- stapi("character", uid = "CHMA0000025118")
Q$episodes %>% select(uid, title, stardateFrom, stardateTo)
```

---

stapiEntities *Star Trek API entities.*

Description

A data frame with 40 rows and 4 columns listing the available STAPI entity IDs that can be passed to `stapi`, along with additional metadata regarding the content returned form an API call to each entity. This data frame helps you see what you will obtain from API calls beforehand. Every entity search returns a tibble data frame, with varying numbers of columns and different names depending on the entity content. There is also one nested column containing the column names of the data frame returned for each entity. This can be inspected directly for specific entities or `stapiEntities` can be unnested with a function like `tidyr::unnest`.

Usage

```
stapiEntities
```

Format

A data frame

See Also

`stapi`

---

stBooks *Star Trek novel metadata.*

Description

A data frame with 743 rows and 11 columns containing metadata on Star Trek novels and other books taken from original books. The data frame contains most of the novels but is not comprehensive and may be out of date temporarily whenever new novels are published.

Usage

```
stBooks
```

Format

A data frame

Details

There is considerable overlap in titles between `stBooksWP` and `stBooks`, but also a considerable number of unique entries. For example, the old novelizations by James Blish of the Original Series episodes are not found in `stBooks`. This dataset only goes back as far as 1979.

However, `stBooks` contains a number of additional columns providing metadata about each book that could only be parsed from books and not from the Wikipedia page that serves as the source for `stBooksWP`. These columns include the number of characters, words and chapters in a book. There may be some irregularities or erroneous entries based on the imperfect methods use to compile the metadata, but it is overall an accurate dataset.

See Also

st_book_series, stBooksWP

---

stBooksWP *Star Trek novel metadata from Wikipedia.*

Description

A data frame with 715 rows and 6 columns containing metadata on Star Trek novels and other books taken from the primary Wikipedia page: https://en.wikipedia.org/wiki/List_of_Star_Trek_novels. The data frame contains most of the novels but is not comprehensive, containing only the most easily scraped HTML table data, and may be out of date temporarily whenever new novels are published.

Usage

```
stBooksWP
```

Format

A data frame

Details

There is considerable overlap in titles between stBooksWP and stBooks, but also a considerable number of unique entries. For example, the old novelizations by James Blish of the Original Series episodes are not found in stBooks.

See Also

st_book_series, stBooks

---

stGeo *Raster grid location data for stellar cartographic map tile sets.*

Description

A data frame of with 18 rows and 4 columns. This data frame has an ID column for map tile set, a column of location names, and columns of respective column and row number of each location per map tile set.

Usage

```
stGeo
```

Format

A data frame

stSpecies *Species names and avatars, linked primarily from Memory Alpha.*

Description

A data frame with 9 rows and 2 columns.

Usage

```
stSpecies
```

Format

A data frame

stTiles *Available Star Trek map tile sets.*

Description

A data frame with 2 row and 8 columns.

Usage

```
stTiles
```

Format

A data frame

st_book_series *Go to Wikipedia entry for a specific book series*

Description

This function opens a browser tab to the main Wikipedia entry for all Star Trek novels automatically scrolled to the section pertaining to `id`. To see the available IDs, call `st_book_series` with no arguments. The Wikipedia page is the one that serves as the source for the `stBooksWP` dataset.

Usage

```
st_book_series(id)
```

Arguments

id	character, abbreviation for a series. See details.

Value

opens a browser tab, nothing is returned unless id is not provided, in which case a data frame is returned.

See Also

stBooksWP

Examples

```
st_book_series()
st_book_series("DS9")
```

---

st_datasets *Available datasets*

Description

List the available datasets in the rtrek package.

Usage

```
st_datasets()
```

Value

a character vector.

Examples

```
st_datasets()
```

st_font *Preview Star Trek fonts*

Description

This function produces a plot showing a preview of a Star Trek font from the `trekfont` package. It will return a message if any of `trekfont`, `showtext` or `ggplot2` are not installed. If `family` is missing, it will return a vector of all available font families.

Usage

```
st_font(family, size = 11)
```

Arguments

`family`	character, font family.
`size,`	numeric, font size passed to ggplot.

Details

In RStudio on Windows the font may not show in the RStudio graphics device. Try using the regular R GUI.

Value

a character vector, or a plot side effect. See details.

Examples

```
if(all(c("trekfont", "showtext", "ggplot2") %in% installed.packages())){
  st_font()
}
## Not run: st_font("Federation") # should be run in an interactive session
```

st_tiles *Return the url associated with a tile set*

Description

This function returns the url associated with a tile set matching `id`.

Usage

```
st_tiles(id)
```

Arguments

id	character, name of map tile set ID. See stTiles.

Details

Tile set data are stored in the stTiles dataset. See for available IDs.

Value

a character string.

See Also

stTiles, st_tiles_data

Examples

```
st_tiles("galaxy1")
```

---

st_tiles_data *Ancillary location data for map tiles*

Description

Obtain a table of ancillary data associated with various locations of interest, given a specific map tile set ID.

Usage

```
st_tiles_data(id)
```

Arguments

id	character, name of a map tile set.

Details

This function returns a small example data frame of location-specific data along with grid cell coordinates that are specific to the requested map tile set ID.

Value

a data frame

See Also

stTiles, st_tiles

Examples

```
st_tiles_data("galaxy2")
```

`tile_coords` *Simple CRS coordinates*

Description

Convert `(column, row)` numbers to `(x, y)` coordinates for a given tile set.

Usage

```
tile_coords(data, id)
```

Arguments

`data`	a data frame containing columns named `col` and `row`. These contain column-row number pairs defining matrix cells in tile set `id`. See details.
`id`	character, name of map tile set ID. See `stTiles`.

Details

This function converts column and row indices for an available map tile set matrix to coordinates that can be used in a Leaflet map. See `stTiles` for available tile sets.

`data` cannot contain columns named `x` or `y`, which are reserved for the column-appended output data frame.

Each tile set has a simple/non-geographical coordinate reference system (CRS). Respective coordinates are based on the dimensions of the source image used to generate each tile set. The same column and row pair will yield different map coordinates for different tile sets. Typical for matrices, columns are numbered increasing from left to right and rows increasing from top to bottom. The output of `tile_coords` is a typical Cartesian coordinate system, increasing from left to right and bottom to top.

Value

a data frame.

Examples

```
d <- data.frame(row = c(0, 3222, 6445), col = c(0, 4000, 8000))
tile_coords(d, "galaxy1")
```

`%>%` *Pipe*

Description

rtrek exports the the pipe operator, `%>%`, from magrittr.

Index