Package 'GNRS'

March 28, 2021

Title Access the 'Geographic Name Resolution Service'

Version 0.2.0

Description Provides tools for interacting with the 'geographic name resolution service' ('GNRS') API <https://github.com/ojalaquellueva/gnrs> and associated functionality. The 'GNRS' is a batch application for resolving & standardizing political division names against standard name in the geonames database <http://www.geonames.org/>. The 'GNRS' resolves political division names at three levels: country, state/province and county/parish. Resolution is performed in a series of steps, beginning with direct matching to standard names, followed by direct matching to alternate names in different languages, followed by direct matching to standard codes (such as ISO and FIPS codes). If direct matching fails, the 'GNRS' attempts to match to standard and then alternate names using fuzzy matching, but does not perform fuzzing matching of political division codes. The 'GNRS' works down the political division hierarchy, stopping at the current level if all matches fail. In other words, if a country cannot be matched, the 'GNRS' does not attempt to match state or county.

Depends R (>= 3.4.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports RCurl, jsonlite

Suggests knitr, rmarkdown, testthat, devtools

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Brad Boyle [aut],
Brian Maitner [aut, cre]

Maintainer Brian Maitner <bmaitner@gmail.com>

Repository CRAN

Date/Publication 2021-03-28 15:40:07 UTC

R topics documented:

GNRS *Standardize political division names*

Description

GNRS returns standardized political division names (according to geonames.org).

Usage

```
GNRS(political_division_dataframe, batches = NULL)
```

Arguments

`political_division_dataframe`	A properly formatted dataframe, see http://bien.nceas.ucsb.edu/bien/tools/gnrs/gnrs-api/
`batches`	NULL or Numeric. Optional number of batches to divide the request into for parallel processing.

Value

Dataframe containing GNRS results.

Note

To create an empty and properly formatted dataframe, use GNRS_template()

The fields the GNRS takes as input are titled "country", "state_province", and "county_parish" for simplicity, but these field actually refer to 0th-, 1st-, and 2nd-order political division, respectively. In the case of some exceptions (e.g. the UK) this distinction becomes important (e.g. Ireland is a 1st-order political division and should be treated as a "state_province" and cannot be matched as a country.)

Examples

```
## Not run:
results <- GNRS(political_division_dataframe = gnrs_testfile)



## End(Not run)
```

GNRS_get_counties *Get metadata on counties*

Description

Return metadata about counties, parishes, etc. used by the GNRS

Usage

```
GNRS_get_counties(state_province_id = "")
```

Arguments

state_province_id	A GNRS state_id, or a vector of state_ids.

Value

Dataframe containing information on counties/parishes (e.g. iso code, fips code, continent, standardized name)

Examples

```
## Not run:
states <- GNRS_get_states()
us_counties <- GNRS_get_counties(
state_province_id = states$state_province_id[
which(states$country_iso == "US")])

## End(Not run)
```

GNRS_get_countries *Get metadata on countries*

Description

Return metadata about countries used by the GNRS

Usage

```
GNRS_get_countries()
```

Value

Dataframe containing information on countries (e.g. iso code, fips code, continent, standardized name)

Examples

```
## Not run:
countries <- GNRS_get_countries()

## End(Not run)
```

GNRS_get_states *Get metadata on states*

Description

Return metadata about states used by the GNRS

Usage

```
GNRS_get_states(country_id = "")
```

Arguments

country_id	A GNRS country_id, or a vector of country_ids. If empty, will return metadata for all countries.

Value

Dataframe containing information on states/provinces (e.g. iso code, fips code, continent, standardized name)

Examples

```
## Not run:
states <- GNRS_get_states()

## End(Not run)
```

`GNRS_super_simple` *Standardize political division names*

Description

GNRS_super_simple returns standardized political division names (according to geonames.org).

Usage

```
GNRS_super_simple(
  country = NULL,
  state_province = NULL,
  county_parish = NULL,
  user_id = NULL
)
```

Arguments

`country`	A single country or a vector of countries. If a vector, length must equal length of species vector.
`state_province`	A single state/province or a vector of states. If a vector, length must equal length of species vector.
`county_parish`	A single county/parish or a vector of counties. If a vector, length must equal length of species vector.
`user_id`	A single identifier or vector of identifiers. This field is assigned if not provided and is used to maintain record order.

Value

Dataframe containing GNRS results.

Note

The fields the GNRS takes as input are titled "country", "state_province", and "county_parish" for simplicity, but these field actually refer to 0th-, 1st-, and 2nd-order political division, respectively. In the case of some exceptions (e.g. the UK) this distinction becomes important (e.g. Ireland is a 1st-order political division and should be treated as a "state_province" and cannot be matched as a country.)

Examples

```
## Not run:

results <- GNRS_super_simple(country = "United States of America")
results <- GNRS_super_simple(
            country = "United States",
            state_province = "Arizona",
            county_parish = "Pima County")


## End(Not run)
```

`GNRS_template` *Make a template for a GNRS query*

Description

GNRS_template builds a template that can be populated to submit a GNRS query.

Usage

```
GNRS_template(nrow = 1)
```

Arguments

`nrow`	The number of rows to include in the template

Value

Template data.frame that can be populated and then used in GNRS queries.

Examples

```
## Not run:

template<-GNRS_template(nrow = 2)
template$country<-c("United Stapes","Mexico")
template$state_province<-c("Arizona","Sinalo")
GNRS(political_division_dataframe = template)


## End(Not run)
```

gnrs_testfile *Names of 21 political divisions*

Description

A dataset containing the country, state/province, and country/parish names of 21 political divisions.

Usage

```
gnrs_testfile
```

Format

A data frame with 21 rows and 4 variables:

user_id Unique integer indentifying each row

country Country names, possibly containing errors

state_province State names, possibly containing errors

county_parish County names, possibly containing errors ...

Source

https://github.com/ojalaquellueva/gnrs

GNRS_version *Get metadata on current GNRS version*

Description

Return metadata about the current GNRS version

Usage

```
GNRS_version()
```

Value

Dataframe containing current GNRS version number, build date, and code version.

Examples

```
## Not run:
GNRS_version_metadata <- GNRS_version()

## End(Not run)
```

Index