Package ‘mnis’

October 13, 2022

Type Package

Title Easy Downloading Capabilities for the Members' Name Information Service

Version 0.3.1

Description An API package for the Members' Name Information Service operated by the UK parliament. Documentation for the API itself can be found here: <http://data.parliament.uk/membersdataplatform/default.aspx>.

URL https://docs.evanodell.com/mnis

BugReports https://github.com/EvanOdell/mnis/issues

License MIT + file LICENSE

LinkingTo Rcpp

Imports httr, jsonlite, utils, dplyr, tibble, stringi, Rcpp, janitor, purrr

RoxygenNote 7.1.1

Encoding UTF-8

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

NeedsCompilation yes

Author Evan Odell [aut, cre] (<https://orcid.org/0000-0003-1845-808X>)

Maintainer Evan Odell <evanodell91@gmail.com>

Repository CRAN

Date/Publication 2021-04-03 17:50:03 UTC

R topics documented:

constituency_results_tidy

constituency_results_tidy

Description

constituency_results_tidy

Usage

```
constituency_results_tidy(results, details)
```

Arguments

`results`	The tibble to tidy
`details`	The list to tidy

date_tidy — *A function that makes date variables returned from the API datable, ie by converting them to POSIXct. Does the same thing for datetimes.*

Description

A function that makes date variables returned from the API datable, ie by converting them to POSIXct. Does the same thing for datetimes.

Usage

```
date_tidy(df)
```

Arguments

`df`	The tibble with the undateable dates.

mnis — *mnis: Easy Downloading Capabilities for the Member Name Information Service*

Description

An API package for the Members' Name Information Service operated by the UK parliament. The package is intended to simplify pulling data from an API for users unfamiliar with APIs. Documentation for the API itself can be found here: http://data.parliament.uk/membersdataplatform/default.aspx.

Details

The package includes a built in function to remove a byte-order mark from the API data, and a parameter `tidy` with each function that converts variable names into an R friendly style, removing non-alphanumeric characters and converting to snake_case when equal to `TRUE`, its default value.

All functions requests data in JSON format and parse it to a tibble. The exception is `mnis_constituency_results()` which returns a single object containing a list (with constituency details) and a tibble (with election results).

None of the functions included in this package appear to run into the API rate limit, although there may be restrictions to custom requests, which allow a maximum of three parameters.

`mnis_all_members` *All members data*

Description

Returns a tibble with information on all members of both houses or a given house.

Usage

```
mnis_all_members(
  house = "all",
  party = NULL,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`house`	The house to which the member belongs. Accepts one of `'all'`, `'lords'` and `'commons'`. This parameter is not case sensitive, so `'commons'`, `'Commons'` and `'cOmMOnS'` will all return the same data. Defaults to `'all'`.
`party`	The party to which a member belongs. If `NULL`, all members are returned, subject to other parameters. Defaults to `NULL`.
`tidy`	Fix the variable names in the tibble to remove special characters and superfluous text, and converts the variable names to a consistent style. Defaults to `TRUE`.
`tidy_style`	The style to convert variable names to, if `tidy=TRUE`. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case"

Value

A tibble with information on all members of the House of Commons and/or the House of Lords that meet the criteria included in the function parameters.

Examples

```
## Not run:
x <- mnis_all_members(
  house = "all", party = NULL, tidy = TRUE,
  tidy_style = "snake_case"
)

## End(Not run)

## Not run:
x <- mnis_all_members(
  house = "all", party = NULL, tidy = TRUE,
  tidy_style = "snake_case"
)
```

```
## End(Not run)
```

mnis_all_reference *Returns a list of 39 tibbles, with the name and data returned from each respective reference function.*

Description

Returns a list of 39 tibbles, with the name and data returned from each respective reference function.

Usage

```
mnis_all_reference(tidy = TRUE, tidy_style = "snake_case")
```

Arguments

`tidy`	Fix the variable names in each tibble to remove special characters and superfluous text, and convert all the variable names to snake_case. Defaults to `TRUE`.
`tidy_style`	The style to convert variable names to, if `tidy=TRUE`. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case".

Value

Returns a list of 39 tibbles, with the name and data returned from each respective reference function.

Examples

```
## Not run:
z <- mnis_all_reference()

## End(Not run)
```

mnis_base *A generic function for the MNIS API*

Description

The function requests data in JSON format, but the type of object, and all URLs, paths and parameters are user-defined. `mnis_base` does not include the option to tidy variable names and data types.

Usage

```
mnis_base(request)
```

Arguments

`request` The request query being made to the MNIS URL.

Details

See the API documentation at http://data.parliament.uk/membersdataplatform/memberquery.aspx for details and limits on requests made to MNIS.

Examples

```
## Not run:

x <- mnis_base("House=Commons|IsEligible=true/")

## End(Not run)
```

---

mnis_basic_details *Additional member information*

Description

A series of basic function for the API lookup. Each of these functions accepts a member's ID and returns information; if no ID is given basic information on all members of both houses is returned. All functions return basic details about the member (name, date of birth, gender, constituency, party, IDs, current status, etc.), as well as any available additional information requested by the specific function.

Usage

```
mnis_basic_details(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_biography_entries(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_committees(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_addresses(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_constituencies(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_elections_contested(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_experiences(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_government_posts(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_honours(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_house_memberships(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_statuses(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_staff(ID = NULL, ref_dods = FALSE, tidy = TRUE, tidy_style = "snake_case")

mnis_interests(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_known_as(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_maiden_speeches(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_opposition_posts(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_other_parliaments(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

```
mnis_parliamentary_posts(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_parties(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)

mnis_preferred_names(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`ID` The member ID. If `NULL`, function calls `mnis_all_members()` and returns basic information on all members of both houses. Defaults to `NULL`.

`ref_dods` If `TRUE`, Request based on the DODS membership ID scheme. If `FALSE`, requests data based on the default membership ID scheme. Defaults to `FALSE`.

`tidy` If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.

`tidy_style` The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

A list with the data corresponding to the particular function called.

`mnis_additional` functions

- `mnis_basic_details` Basic biographical details on a given Member
- `mnis_biography_entries` Member biographical information (e.g. countries of interest, policy expertise etc...)
- `mnis_committees` Committees a Member sits or has sat on as well details on committee chairing
- `mnis_addresses` Member address information (e.g. website, twitter, constituency address etc...)
- `mnis_constituencies` Constituencies a Member has represented

- `mnis_elections_contested` Elections a Member has contested but not won
- `mnis_experiences` Non-parliamentary experience of a Member
- `mnis_government_posts` Government posts a Member has held
- `mnis_honours` Honours (e.g. MBE, OBE etc...) held by a Member
- `mnis_house_memberships` House membership list of a Member
- `mnis_statuses` Status history (e.g. suspensions and disqualifications) for a Member
- `mnis_staff` The staff employed by a Member
- `mnis_interests` Registered (financial) interests of a Member
- `mnis_known_as` Details of names a Member has chosen to be known as instead of their full title, only applicable to members of the House of Lords
- `mnis_maiden_speeches` Maiden speech dates for a Member
- `mnis_opposition_posts` Opposition posts a Member has held
- `mnis_other_parliaments` Other Parliaments that a Member has held a membership of (e.g. members of the Welsh Assembly)
- `mnis_parliamentary_posts` Parliamentary posts a Member has held
- `mnis_parties` Party affiliations of a Member
- `mnis_preferred_names` Full set of data about a Members' name (e.g. surname, forename, Honorary prefixes, full details of House of Lords title and rank if applicable, etc...)

See Also

`mnis_full_biog()`

Examples

```
## Not run:
x <- mnis_basic_details(172)

## End(Not run)
```

mnis_bom *Strip out BOM from JSON data*

Description

Strip out BOM from JSON data

Usage

```
mnis_bom(x)
```

Arguments

x	The GET return to strip BOM out of

mnis_constituency_results

Constituency election results

Description

Returns a list with details of the constituency and a tibble with election results.

Usage

```
mnis_constituency_results(
  constituency_id = NULL,
  election_id = 0,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

constituency_id	The ID of the constituency to return the data for. If NULL, no data is returned. Defaults to NULL.
election_id	The ID of the election to return the data for. Defaults to 0, which returns the result of all elections held in that constituency.
tidy	If TRUE, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to TRUE.
tidy_style	The style to convert variable names to, if tidy=TRUE. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case".

Value

A list with details of the constituency, labelled 'details' and a tibble with election results, labelled 'results'. The list and tibble are stored in a single object.

Examples

```
## Not run:
x <- mnis_constituency_results(constituency_id = 3709, election_id = 0)

## End(Not run)
```

mnis_department *mnis_department*

Description

Request the holders of posts in specific departments by department ID.

Usage

```
mnis_department(
  department_id = 0,
  bench = "Government",
  former = TRUE,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

department_id	The department to look up. `0` returns the cabinet/shadow cabinet, `-1` returns a list of all ministers.
bench	Flag to return either Government or Opposition information. This parameter is case insensitive, so both `'Opposition'` and `'opposition'` will return the same data. Defaults to `'Government'`.
former	Logical parameter to include both current and former ministers/shadow ministers. If `FALSE`, only includes current ministers/shadow ministers. Defaults to `TRUE`.
tidy	If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.
tidy_style	The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

A tibble with information on departments and ministers/shadow ministers.

Examples

```
## Not run:
x <- mnis_department(department_id = 0, bench = "Government", former = TRUE)

## End(Not run)
```

`mnis_eligible` *mnis_eligible*

Description

Returns all members who are able to sit in either house, or who are currently ineligible to sit. Members ineligible to sit include but are not necessarily limited to former MPs, members of the judiciary, who are recused from House of Lords duties.

Usage

```
mnis_eligible(
  eligible = TRUE,
  house = "all",
  party = NULL,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`eligible`	If the member is currently eligible to sit. Accepts `TRUE` or `FALSE`. Defaults to `TRUE`.
`house`	The house to which the member belongs. Accepts one of 'all', 'lords' and 'commons', defaults to 'all'. This parameter is not case sensitive, so 'commons', 'Commons' and 'cOmMOnS' will all return the same data.
`party`	The party to which a member belongs. Defaults to NULL. The party must be fully spelled out (e.g. 'green party'), the API does not accept searches on this parameter. For a tibble of parties, see `ref_parties()`. This parameter is not case sensitive.
`tidy`	Fix the variable names in the tibble to remove special characters and superfluous text, and converts the variable names to a consistent style. Defaults to `TRUE`.
`tidy_style`	The style to convert variable names to, if `tidy=TRUE`. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case"

Examples

```
## Not run:
x <- mnis_eligible(eligible = FALSE, house = "all", party = "labour")

y <- mnis_eligible(eligible = TRUE, house = "all", party = "green party")

z <- mnis_eligible(house = "commons")

## End(Not run)
```

mnis_full_biog *Full Biography*

Description

Requests all available biographical information for a given member, and returns it in a list

Usage

```
mnis_full_biog(
  ID = NULL,
  ref_dods = FALSE,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`ID` The ID number of the member, using the default MNIS scheme. If `ref_dods` is TRUE, accepts the Dods monitoring scheme instead. If left empty, returns the same data as `mnis_all_members()` with default parameters.

`ref_dods` Request based on the Dods monitoring member ID scheme. Defaults to FALSE. If FALSE, requests using the default MNIS identification scheme.

`tidy` Fix the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert variable names to a consistent style. Defaults to `TRUE`. Currently does not work.

`tidy_style` The style to convert variable names to, if `tidy==TRUE`.

See Also

`mnis_basic_details()`

Examples

```
## Not run:
df172 <- mnis_full_biog(172)

df500 <- mnis_full_biog(500)

## End(Not run)
```

mnis_general_election_results

Election results

Description

Returns an object containing list with details of the search parameter and a tibble with election results. Accepts queries on location type and name, and the start and end date to return general elections between. The API does not contain data for Norther Ireland.

Usage

```
mnis_general_election_results(
  location_type = "Country",
  location_name = "Great Britain",
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`location_type` The type of area to return information for. Accepts `'Country'`, `'Region'`, `'County'`, and `'Constituency'`. Defaults to `'Country'`.

`location_name` The location to return data for. It can be the name of any Country, Region, County or Constituency. Defaults to 'Great Britain'.

`start_date` Start date of search. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.

`end_date` End date of search. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. Defaults to current system date.

`tidy` If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.

`tidy_style` The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

Returns a list with details of the search parameter and a tibble with election results.

Examples

```
## Not run:
x <- mnis_general_election_results(
  location_type = "Country", location_name = "England",
  start_date = "2010-01-01", end_date = "2016-01-01"
)

## End(Not run)
```

`mnis_joined_between` *Members who joined between two dates.*

Description

Function returns all members who took their seats in the house between two given dates.

Usage

```
mnis_joined_between(
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  house = "all",
  party = NULL,
  eligible = "all",
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`start_date` The start date of the search, Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.

`end_date` The end date of the search. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. Defaults to the current system date.

`house` The house to which the member belongs. Accepts one of `'all'`, `'lords'` and `'commons'`. This parameter is not case sensitive. Defaults to `'all'`.

`party` All members from a given party who joined between the two dates. The party name must be fully spelled out (e.g. `'green party'`), the API does not accept searches on this parameter. For a tibble of parties, see `ref_parties()`. This parameter is not case sensitive. Defaults to `NULL`.

`eligible` If the member is currently eligible to sit. Accepts one of `'all'`, `'current'`, `'former'`. This parameter is not case sensitive. Defaults to `'all'`.

`tidy` If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.

`tidy_style` The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

A tibble with data on all members who joined between the two given dates.

Examples

```
## Not run:
x <- mnis_joined_between(
  start_date = "2015-01-01",
  end_date = "2017-01-01", party = "labour"
)

## End(Not run)
```

`mnis_lords_type` *Peers' party affiliations*

Description

Calls the API to return a tibble with details on the number of Lords and their party affiliations. Defaults to the current date, but can also return the number of Lords and their affiliations on a given date.

Usage

```
mnis_lords_type(date = Sys.Date(), tidy = TRUE, tidy_style = "snake_case")
```

Arguments

`date` Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. The API will return data on the composition of the House of Lords on that date. Defaults to the current system date.

`tidy` If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.

`tidy_style` The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

A tibble with information on the numbers of different types of Lords on a given date.

Examples

```
## Not run:
x <- mnis_lords_type()

## End(Not run)
```

`mnis_member_date` *Returns a tibble with a member's status on a given date.*

Description

Returns a tibble with a member's status on a given date.

Usage

```
mnis_member_date(
  ID = NULL,
  date = Sys.Date(),
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`ID` The ID of the member, or a vector with the IDs of multiple MPs. If given multiple IDs, the results are combined into a single tibble. Currently only accepts IDs from the default membership ID scheme. If `NULL` the function stops and no data is returned. Defaults to `NULL`.

`date` Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. Return details on the requested member's status on that date. Defaults to the current system date.

`tidy` If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.

`tidy_style` The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

Returns a tibble with the given member's status on the given date.

See Also

`mnis_mps_on_date()`

Examples

```
## Not run:
x <- mnis_member_date(172)

## End(Not run)
```

---

mnis_mps_on_date *All MPs between two dates*

Description

Requests data on all MPs who were members of the House of Commons on the date specified, (if only `date1` is included as a parameter), or on or between the two dates if both `date1` and `date2` are specified. Either `date1` or `date2` can be the latter of the two dates.

Usage

```
mnis_mps_on_date(
  date1 = Sys.Date(),
  date2 = NULL,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`date1`	The date to return the list of mps from. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. Defaults to current system date.
`date2`	An optional query parameter. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else than can be coerced to a date with `as.Date()`. If not `NULL`, the function returns a list of all MPs who were members between `date1` and `date2`. Defaults to `NULL`.
`tidy`	If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.
`tidy_style`	The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

A tibble with information on all MPs who were members of the House of Commons on the date specificed (if only `date1` is included as a parameter), or on or between the two dates if both `date1` and `date2` are specified.

See Also

mnis_party_state()

mnis_peers_on_date()

Examples

```
## Not run:
x <- mnis_mps_on_date(date1 = "2017-01-01", date2 = "2014-02-04")

## End(Not run)
```

mnis_party_state *Party standing on a given date*

Description

Returns data on the numbers and gender of MPs, by party, for the given date, in either the House of Commons or the House of Lords.

Usage

```
mnis_party_state(
  house = "Commons",
  date = Sys.Date(),
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

house	The house of parliament. Accepts either 'Lords' or 'Commons'. This parameter is not case sensitive, so both 'lords' and 'Lords' return the same result. Defaults to 'Commons'.
date	The date to query party standing on. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else than can be coerced to a date with as.Date(). Defaults to the current system date.
tidy	If TRUE, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to TRUE.
tidy_style	The style to convert variable names to, if tidy=TRUE. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case".

Value

A tibble with information on the total numbers and gender of MPs, by party, for the given date and house.

See Also

mnis_mps_on_date()

mnis_peers_on_date()

Examples

```
## Not run:
x <- mnis_party_state("2012-01-12")

## End(Not run)
```

mnis_peers_on_date *Peers on date*

Description

Requests data on all peers who were members of the House of Lords on the date specified, (if only date1 is included as a parameter), or on or between the two dates if both date1 and date2 are specified. Either date1 or date2 can be the latter of the two dates.

Usage

```
mnis_peers_on_date(
  date1 = Sys.Date(),
  date2 = NULL,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

date1	The date to return the list of peers from. Defaults to current system date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else than can be coerced to a date with as.Date().
date2	An optional query parameter. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else than can be coerced to a date with as.Date(). If not NULL, the function returns a list of all peers in the House of Lords between date1 and date2. Defaults to NULL.
tidy	If TRUE, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to TRUE.
tidy_style	The style to convert variable names to, if tidy=TRUE. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case".

Value

A tibble with information on all peers who were members of the House of Lords on the date specified, (if only `date1` is included as a parameter), or on or between the two dates if both `date1` and `date2` are specified.

See Also

`mnis_party_state()`

`mnis_peers_on_date()`

Examples

```
## Not run:
x <- mnis_peers_on_date(date1 = "2017-01-01")

## End(Not run)
```

`mnis_political_interests`

Political and Geographical Interests

Description

Returns a tibble with members with the specified interest or interests.

Usage

```
mnis_political_interests(
  interest,
  house = NULL,
  current = TRUE,
  tidy = TRUE,
  tidy_style = "snake_case"
)
```

Arguments

`interest` A single interest in the form of a string, or a character vector of interests. If a vector of multiple interests, returns all members who list all of those interests.

`house` The house the member belongs to. Accepts 'commons', 'lords' and `NULL`. This variable is not case sensitive. If `NULL`, returns members from both houses. Defaults to `NULL`.

`current` Logical. If `TRUE`, returns all current and former members of both houses. Defaults to `TRUE`.

`tidy`	If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.
`tidy_style`	The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, `"camelCase"` and `"period.case"`. Defaults to `"snake_case"`.

Value

A data frame of members with given interest(s).

Examples

```
## Not run:
x <- mnis_political_interests(c("Africa", "Jamaica"))

y <- mnis_political_interests(c("Africa", "Europe"))

z <- mnis_political_interests(c("Africa", "Education"))

## End(Not run)
```

`mnis_tidy` *Functions to tidy up the variable names returned from the API, and turn dates and datetimes to POSIXct.*

Description

Functions to tidy up the variable names returned from the API, and turn dates and datetimes to POSIXct.

member_tidy

ref_tidy

Usage

```
mnis_tidy(df, tidy_style = "snake_case")

member_tidy(df)

ref_tidy(df, tidy_style)
```

Arguments

`df`	The tibble to tidy.
`tidy_style`	The style to tidy the tibble with.

ref_address_types *Reference data*

Description

Reference data on various aspects of parliament. This data is useful for providing parameters for other function calls. The functions do not accept any arguments aside from the `tidy` and `tidy_style` parameters, which default to `TRUE` and `'snake_case'`, respectively. To return all a list with tibbles of all reference data, see `mnis_all_reference()`.

Usage

```
ref_address_types(tidy = TRUE, tidy_style = "snake_case")

ref_answering_bodies(tidy = TRUE, tidy_style = "snake_case")

ref_areas(tidy = TRUE, tidy_style = "snake_case")

ref_area_types(tidy = TRUE, tidy_style = "snake_case")

ref_biography_categories(tidy = TRUE, tidy_style = "snake_case")

ref_cabinets(tidy = TRUE, tidy_style = "snake_case")

ref_committees(tidy = TRUE, tidy_style = "snake_case")

ref_committee_types(tidy = TRUE, tidy_style = "snake_case")

ref_constituencies(tidy = TRUE, tidy_style = "snake_case")

ref_constituency_areas(tidy = TRUE, tidy_style = "snake_case")

ref_constituency_types(tidy = TRUE, tidy_style = "snake_case")

ref_countries(tidy = TRUE, tidy_style = "snake_case")

ref_departments(tidy = TRUE, tidy_style = "snake_case")

ref_disqualification_types(tidy = TRUE, tidy_style = "snake_case")

ref_elections(tidy = TRUE, tidy_style = "snake_case")

ref_election_types(tidy = TRUE, tidy_style = "snake_case")

ref_end_reasons(tidy = TRUE, tidy_style = "snake_case")

ref_experience_types(tidy = TRUE, tidy_style = "snake_case")
```

```
ref_government_post_departments(tidy = TRUE, tidy_style = "snake_case")

ref_government_posts(tidy = TRUE, tidy_style = "snake_case")

ref_government_ranks(tidy = TRUE, tidy_style = "snake_case")

ref_honourary_prefixes(tidy = TRUE, tidy_style = "snake_case")

ref_honour_lists(tidy = TRUE, tidy_style = "snake_case")

ref_honours(tidy = TRUE, tidy_style = "snake_case")

ref_interest_categories(tidy = TRUE, tidy_style = "snake_case")

ref_lords_membership_types(tidy = TRUE, tidy_style = "snake_case")

ref_lords_ranks(tidy = TRUE, tidy_style = "snake_case")

ref_opposition_post_departments(tidy = TRUE, tidy_style = "snake_case")

ref_opposition_posts(tidy = TRUE, tidy_style = "snake_case")

ref_opposition_ranks(tidy = TRUE, tidy_style = "snake_case")

ref_other_parliaments(tidy = TRUE, tidy_style = "snake_case")

ref_parliamentary_posts(tidy = TRUE, tidy_style = "snake_case")

ref_parliamentary_ranks(tidy = TRUE, tidy_style = "snake_case")

ref_parliament_types(tidy = TRUE, tidy_style = "snake_case")

ref_parties(tidy = TRUE, tidy_style = "snake_case")

ref_party_sub_types(tidy = TRUE, tidy_style = "snake_case")

ref_photo_outputs(tidy = TRUE, tidy_style = "snake_case")

ref_statuses(tidy = TRUE, tidy_style = "snake_case")

ref_titles(tidy = TRUE, tidy_style = "snake_case")
```

Arguments

`tidy` If `TRUE`, fixes the variable names in the tibble to remove non-alphanumeric characters and superfluous text, and convert to a consistent style. Defaults to `TRUE`.

`tidy_style` The style to convert variable names to, if `tidy=TRUE`. Accepts one of `"snake_case"`, "camelCase" and "period.case". Defaults to "snake_case".

mnis_reference **functions**

- ref_address_types The types of addresses available in member's contact details. Includes websites and social media, as well as physical addresses
- ref_answering_bodies The bodies that members' can address questions to
- ref_areas Geographic areas
- ref_area_types Identifiers for grouping areas (e.g. borough constituencies)
- ref_biography_categories Member biography categories
- ref_cabinets Connections that a member has to the cabinet or shadow cabinet
- ref_committees Identifier for parliamentary committees
- ref_committee_types Types of parliamentary committees
- ref_constituencies() All constituencies
- ref_constituency_areas() The links between constituencies and constituency areas
- ref_constituency_types() Constituency categories
- ref_countries() List of countries that could be listed as members' birthplace
- ref_departments() Government and opposition departments
- ref_disqualification_types() Types of ways members can be disqualified from sitting in the House
- ref_elections() Codes of general and by-elections
- ref_election_types() Election categories
- ref_end_reasons() Reasons a member may leave the House of Lords or the House of Commons
- ref_experience_types() Types of non-parliamentary experience members can list
- ref_government_post_departments() All deparments that can contain government posts
- ref_government_posts() All government posts
- ref_government_ranks() All government post ranks
- ref_honourary_prefixes() The types of honourary prefixes for members
- ref_honour_lists() The types of honour lists that a member may be honoured in
- ref_honours() The different honours available to members
- ref_interest_categories() The categories available for reporting financial interests
- ref_lords_membership_types() Different types of membership of the House of Lords
- ref_lords_ranks() Ranks that peers may hold
- ref_opposition_post_departments() The link between opposition posts and the government department they shadow
- ref_opposition_posts() Opposition posts
- ref_opposition_ranks() How opposition posts are ranked
- ref_other_parliaments() Other parliaments that a member may have sat in

- `ref_parliamentary_posts()` The different parliamentary posts available
- `ref_parliamentary_ranks()` How those parliamentary posts are ranked
- `ref_parliament_types()` Types of parliaments that parliamentary data may link to
- `ref_parties()` All parties that members can be affiliated with
- `ref_party_sub_types()` Sub-types of parties
- `ref_photo_outputs()` Outputs that a photo of a member may be linked to
- `ref_statuses()` A member's possible current status in the House
- `ref_titles()` Salutory titles

See Also

`mnis_all_reference()`

`tidy_bom` *A function to strip Byte Order Marks (BOM) out of JSON data returned from the API.*

Description

A function to strip Byte Order Marks (BOM) out of JSON data returned from the API.

Usage

```
tidy_bom(df)
```

Arguments

`df`	The GET returned from call to API.

Index