

Package ‘mnis’

February 7, 2017

Type Package

Title Easy Downloading Capabilities for the Members' Name Information Service

Version 0.2.1

Date 2017-01-31

Author Evan Odell [aut, cre]

Maintainer Evan Odell <evanodell191@gmail.com>

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/membersdataplatfrom/default.aspx>>.

URL <https://github.com/EvanOdell/mnis>

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

License MIT + file LICENSE

LazyData TRUE

Imports htr, jsonlite, utils, plyr

RoxygenNote 6.0.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-07 10:08:55

R topics documented:

| | |
|-------------------------------------|---|
| mnis | 2 |
| mnis_additional | 3 |
| mnis_all_members | 4 |
| mnis_base | 5 |
| mnis_constituency_results | 6 |

| | |
|-----------------------------------------|----|
| mnis_department | 7 |
| mnis_eligible | 7 |
| mnis_extra | 8 |
| mnis_full_biog | 10 |
| mnis_general_election_results | 10 |
| mnis_joined_between | 11 |
| mnis_lords_type | 12 |
| mnis_member_date | 13 |
| mnis_party_state | 13 |
| mnis_reference | 14 |
| mnis_tidy | 18 |

| | |
|--------------|-----------|
| Index | 20 |
|--------------|-----------|

| | |
|------|------------------------------------------------------------------------------------|
| mnis | <i>mnis: Easy Downloading Capabilities for the Member Name Information Service</i> |
|------|------------------------------------------------------------------------------------|

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/membersdataplatfrom/default.aspx>.

Details

Note that on some systems there may be a warning message stating 'JSON string contains (illegal) UTF8 byte-order-mark!'. This does not appear to have any impact on the data itself, but is due to the encoding used by the API.

mnis functions

```
mnis_additional
mnis_base
mnis_constituency_results
mnis_department
mnis_eligible
mnis_extra
mnis_full_biog
mnis_general_election_results
mnis_joined_between
mnis_lords_type
mnis_member_date
mnis_party_state
mnis_reference
```

| | |
|-----------------|------------------------|
| mnis_additional | <i>mnis_additional</i> |
|-----------------|------------------------|

Description

Basic function for the MNIS API lookup. The function requests data in JSON format and parses it to a data frame. Variable descriptions are taken from the mnis website. The API provides lengthy and complicated variable names, and the functions make some attempts to tidy up those names. To disable this feature, include `tidy = TRUE` in the function.

Usage

```
mnis_additional()
```

```
mnis_basic_details(ID = NULL, mem_id = TRUE, refDods = FALSE,  
  tidy = TRUE)
```

```
mnis_biography_entries(ID = NULL, mem_id = TRUE, refDods = FALSE,  
  tidy = TRUE)
```

```
mnis_committees(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_addresses(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_constituencies(ID = NULL, mem_id = TRUE, refDods = FALSE,  
  tidy = TRUE)
```

```
mnis_elections_contested(ID = NULL, mem_id = TRUE, refDods = FALSE,  
  tidy = TRUE)
```

```
mnis_experiences(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_government_posts(ID = NULL, mem_id = TRUE, refDods = FALSE,  
  tidy = TRUE)
```

```
mnis_honours(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_house_memberships(ID = NULL, mem_id = TRUE, refDods = FALSE,  
  tidy = TRUE)
```

```
mnis_statuses(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_staff(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_interests(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```
mnis_known_as(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

```

mnis_maiden_speeches(ID = NULL, mem_id = TRUE, refDods = FALSE,
  tidy = TRUE)

mnis_opposition_posts(ID = NULL, mem_id = TRUE, refDods = FALSE,
  tidy = TRUE)

mnis_other_parliaments(ID = NULL, mem_id = TRUE, refDods = FALSE,
  tidy = TRUE)

mnis_parliamentary_posts(ID = NULL, mem_id = TRUE, refDods = FALSE,
  tidy = TRUE)

mnis_parties(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)

mnis_preferred_names(ID = NULL, mem_id = TRUE, refDods = FALSE,
  tidy = TRUE)

```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------|
| ID | The ID number of the member. Defaults to NULL. |
| mem_id | Request based on the default membership ID scheme. |
| refDods | Request based on the DODS membership ID scheme. Defaults to FALSE. |
| tidy | Fix the variable names in the data frame to remove '@' characters and superfluous text. Defaults to TRUE. |

See Also

[mnis_full_biog](#) [mnis_extra](#)

Examples

```

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

## End(Not run)

```

| | |
|-------------------------------|-------------------------|
| <code>mnis_all_members</code> | <i>mnis_all_members</i> |
|-------------------------------|-------------------------|

Description

`mnis_all_members`

Usage

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

Arguments

| | |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| house | The house to which the member belongs. Accepts one of 'all', 'lords' and 'commons', defaults to 'all'. |
| party | The party to which a member belongs. Defaults to NULL. |
| joined_since | All members who joined after a given date, in yyyy-mm-dd format. |
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |

Value

A data frame with all members of both houses

Examples

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

## End(Not run)
```

mnis_base

mnis_Base

Description

Basic function for the MNIS API lookup. The function requests data in JSON format and parses it to a data frame. All URLs, paths and parameters are user-defined.

Usage

```
mnis_base(request, tidy = TRUE)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| request | The request query being made to the MNIS URL |
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |

Examples

```
## Not run:  
x <- mnis_Base('House=Commons|IsEligible=true/')  
  
## End(Not run)
```

```
mnis_constituency_results  
      mnis_constituency_results
```

Description

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

Usage

```
mnis_constituency_results(constituencyId = NULL, electionId = 0,  
  tidy = TRUE)
```

Arguments

| | |
|----------------|-----------------------------------------------------------------------------------------------------------|
| constituencyId | The ID of the constituency to return the data for. |
| electionId | The ID of the election to return the data for. Defaults to 0, which calls the latest result. |
| tidy | Fix the variable names in the data frame to remove '@' characters and superfluous text. Defaults to TRUE. |

Value

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

Examples

```
## Not run:  
x <- mnis_constituency_results(constituencyId = 3709, electionId = 0)  
  
## End(Not run)
```

| | |
|-----------------|------------------------|
| mnis_department | <i>mnis_department</i> |
|-----------------|------------------------|

Description

mnis_department

Usage

```
mnis_department(departmentId = 0, bench = "Government", former = TRUE,  
tidy = TRUE)
```

Arguments

| | |
|--------------|-----------------------------------------------------------------------------------------------------------|
| departmentId | The department look up. 0 returns the cabinet/shadow cabinet, -1 returns a list of all ministers/ |
| bench | Flag to return either Government or Opposition information. Defaults to Government. |
| former | Flag to include both current and former ministers/shadow ministers. Defaults to TRUE. |
| tidy | Fix the variable names in the data frame to remove '@' characters and superfluous text. Defaults to TRUE. |

Value

A list with information on the outcome of the most recent election in a constituency

Examples

```
## Not run:  
x <- mnis_department(departmentId = 0, bench = 'Government', former=TRUE)  
  
## End(Not run)
```

| | |
|---------------|----------------------|
| mnis_eligible | <i>mnis_eligible</i> |
|---------------|----------------------|

Description

Function returns all members who took their seats in the house between two given dates. The function requests data in JSON format and parses it to a data frame.

Usage

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

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'. |
| party | The party to which a member belongs. Defaults to NULL. |
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |

Examples

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

## End(Not run)
```

mnis_extra

mnis_extra

Description

A wrapper for [mnis_additional](#) functions. It combines the various options of [mnis_additional](#) into one dataframe, and the default is similar to [mnis_full_biog](#). Variable descriptions are taken from the mnis website: <<http://data.parliament.uk/membersdataplatfrom/memberquery.aspx>>

Usage

```
mnis_extra(ID, mem_id = TRUE, refDods = FALSE, Addresses = TRUE,
  BasicDetails = TRUE, BiographyEntries = TRUE, Committees = TRUE,
  Constituencies = TRUE, ElectionsContested = TRUE, Experiences = TRUE,
  GovernmentPosts = TRUE, Honours = TRUE, HouseMemberships = TRUE,
  Interests = TRUE, KnownAs = TRUE, MaidenSpeeches = TRUE,
  OppositionPosts = TRUE, OtherParliaments = TRUE,
  ParliamentaryPosts = TRUE, Parties = TRUE, PreferredNames = TRUE,
  Staff = TRUE, Statuses = TRUE, tidy = TRUE)
```


Arguments

| | |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | The ID number of the member. Defaults to NULL. |
| mem_id | Request based on the default membership ID scheme. |
| refDods | Request based on the DODS membership ID scheme. Defaults to FALSE. |
| Addresses | Member address information (e.g. website, twitter, constituency address etc...). |
| BasicDetails | Basic information about the Member (e.g. given name, HoL membership type, oaths etc...) |
| BiographyEntries | Member biographical information (e.g. countries of interest, policy expertise etc...) |
| Committees | Committees a Member sits or has sat on as well details on committee chairing. |
| Constituencies | Constituencies a Member has represented. |
| ElectionsContested | Elections a Member has contested but not won. |
| Experiences | Non-parliamentary experience of a Member. |
| GovernmentPosts | Government posts a Member currently holds. |
| Honours | Honours (e.g. MBE, OBE etc...) held by a Member. |
| HouseMemberships | House membership list of a Member. |
| Interests | Registered interests (financial) of a Member. |
| KnownAs | Details of names a Member has chosen to be known as instead of their full title (HoL members only). |
| MaidenSpeeches | Maiden speech dates for a Member. |
| OppositionPosts | Opposition posts a Member has held. |
| OtherParliaments | Other Parliaments that a Member has held a membership of. |
| ParliamentaryPosts | Parliamentary posts a Member has held. |
| Parties | Party affiliations of a Member. |
| PreferredNames | Full set of data about a Members' name (e.g. surname, forename, Honorary prefixes, full details of HoL title and rank etc...). |
| Staff | Staff employed by a Member. |
| Statuses | Status history (e.g. suspensions and disqualifications) for a Member. |
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |

See Also

[mnis_full_biog](#) [mnis_basic_details](#) [mnis_additional](#)

Examples

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

## End(Not run)
```

| | |
|----------------|----------------------|
| mnis_full_biog | <i>mnis_FullBiog</i> |
|----------------|----------------------|

Description

Requests all available biographical information for a given member.

Usage

```
mnis_full_biog(ID = NULL, mem_id = TRUE, refDods = FALSE, tidy = TRUE)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------|
| ID | The ID number of the member. Defaults to NULL. |
| mem_id | Request based on the default membership ID scheme. Defaults to TRUE. |
| refDods | Request based on the DODS membership ID scheme. Defaults to FALSE. |
| tidy | Fix the variable names in the data frame to remove '@' characters and superfluous text. Defaults to TRUE. |

Examples

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

## End(Not run)
```

| | |
|-------------------------------|--------------------------------------|
| mnis_general_election_results | <i>mnis_general_election_results</i> |
|-------------------------------|--------------------------------------|

Description

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

Usage

```
mnis_general_election_results(locationType = "Country",
  locationName = "Great Britain", startDate = "1900-01-01",
  endDate = Sys.Date())
```

Arguments

| | |
|--------------|----------------------------------------------------------------------------------------------------------------------------------|
| locationType | The type of area to return information for. Accepts 'Country', 'Region', 'County', and 'Constituency'. Defaults to 'Country'. |
| locationName | The location to return data for. It can be the name of any Country, Region, County or Constituency. Defaults to 'Great Britain'. |
| startDate | Start date of search. In yyyy-mm-dd format. Defaults to '1900-01-01' if no date is selected. |
| endDate | End date of search. In yyyy-mm-dd format. Defaults to current date if no date is selected. |

Value

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

Examples

```
## Not run:
x <- mnis_general_election_results(locationType = 'Country', locationName = 'England',
  startDate = '2010-01-01', endDate = '2016-01-01')

## End(Not run)
```

mnis_joined_between *mnis_joined_between*

Description

Function returns all members who took their seats in the house between two given dates. The function requests data in JSON format and parses it to a data frame. The

Usage

```
mnis_joined_between(startDate = "1900-01-01", endDate = Sys.Date(),
  house = "all", party = NULL, eligible = "all", tidy = TRUE)
```

Arguments

| | |
|------------------------|------------------------------------------------------------------------------------------------------------|
| <code>startDate</code> | The start date of the search, in yyyy-mm-dd format. Defaults to '1900-01-01'. |
| <code>endDate</code> | The end date of the search, in yyyy-mm-dd format. Defaults to the current date. |
| <code>house</code> | The house to which the member belongs. Accepts one of 'all', 'lords' and 'commons', defaults to 'all'. |
| <code>party</code> | The party to which a member belongs. Defaults to NULL. |
| <code>eligible</code> | If the member is currently eligible to sit. Accepts one of 'all', 'current', 'former', defaults to 'all'. |
| <code>tidy</code> | Fix the variable names in the data frame to remove '@' characters and superfluous text. Defaults to FALSE. |

Examples

```
## Not run:
x <- mnis_joined_between(startDate = '2015-01-01', endDate = '2017-01-01',
                          house='all', party='labour', eligible='all')

## End(Not run)
```

| | |
|------------------------------|------------------------|
| <code>mnis_lords_type</code> | <i>mnis_lords_type</i> |
|------------------------------|------------------------|

Description

Calls the API to return a data frame with details on the number of Lords and their affiliations.

Usage

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

Arguments

| | |
|-------------------|-----------------------------------------------------------------------------------------------------------|
| <code>date</code> | A date in yyyy-mm-dd format. Defaults to the current system date. |
| <code>tidy</code> | Fix the variable names in the data frame to remove '@' characters and superfluous text. Defaults to TRUE. |

Value

A data frame 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 | <i>mnis_member_date</i> |
|------------------|-------------------------|

Description

Returns a data frame with a members status on a given date.

Usage

```
mnis_member_date(ID = NULL, Date = Sys.Date(), tidy = TRUE)
```

Arguments

| | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | The ID of the member. |
| Date | A date in yyyy-mm-dd format. Defaults to the current system date. |
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |

Value

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

Examples

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

| | |
|------------------|-------------------------|
| mnis_party_state | <i>mnis_party_state</i> |
|------------------|-------------------------|

Description

A data frame with information on the numbers and gender of MPs, by party, for the given date.

Usage

```
mnis_party_state(House = "Commons", Date = Sys.Date(), tidy = TRUE)
```

Arguments

| | |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| House | The house of parliament. Defaults to 'Commons'. |
| Date | A date in yyyy-mm-dd format. Defaults to the current system date. |
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |

Value

A data frame with information on the numbers and gender of MPs, by party, by party, for the given date.

Examples

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

## End(Not run)
```

| | |
|----------------|-----------------------|
| mnis_reference | <i>mnis_reference</i> |
|----------------|-----------------------|

Description

A series of functions to return reference data. This data is useful for providing parameters for other function calls. The functions do not accept any arguments.

Usage

```
mnis_reference()

ref_address_types(tidy = TRUE)

ref_answering_bodies(tidy = TRUE)

ref_areas(tidy = TRUE)

ref_area_types(tidy = TRUE)

ref_biography_categories(tidy = TRUE)

ref_cabinets(tidy = TRUE)

ref_committees(tidy = TRUE)

ref_committee_types(tidy = TRUE)
```

ref_constituencies(tidy = TRUE)
ref_constituency_areas(tidy = TRUE)
ref_constituency_types(tidy = TRUE)
ref_countries(tidy = TRUE)
ref_departments(tidy = TRUE)
ref_disqualification_types(tidy = TRUE)
ref_elections(tidy = TRUE)
ref_election_types(tidy = TRUE)
ref_end_reasons(tidy = TRUE)
ref_experience_types(tidy = TRUE)
ref_government_post_departments(tidy = TRUE)
ref_government_posts(tidy = TRUE)
ref_government_ranks(tidy = TRUE)
ref_honourary_prefixes(tidy = TRUE)
ref_honour_lists(tidy = TRUE)
ref_honours(tidy = TRUE)
ref_interest_categories(tidy = TRUE)
ref_lords_membership_types(tidy = TRUE)
ref_lords_ranks(tidy = TRUE)
ref_opposition_post_departments(tidy = TRUE)
ref_opposition_posts(tidy = TRUE)
ref_opposition_ranks(tidy = TRUE)
ref_other_parliaments(tidy = TRUE)
ref_parliamentary_posts(tidy = TRUE)

```
ref_parliamentary_ranks(tidy = TRUE)
ref_parliament_types(tidy = TRUE)
ref_parties(tidy = TRUE)
ref_party_sub_types(tidy = TRUE)
ref_photo_outputs(tidy = TRUE)
ref_statuses(tidy = TRUE)
ref_titles(tidy = TRUE)
```

Arguments

| | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tidy | Fix the variable names in the data frame to remove special characters and superfluous text, and converts the variable names to all lower case with underscores between each word. Defaults to TRUE. |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Examples

```
## Not run:
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 departments 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()
# Salutary titles.

mnis_reference()
# Returns a list of all possible reference functions.

## End(Not run)
```

mnis_tidy

mnis_tidy

Description

Internal tidying function

Usage

`mnis_tidy(x)`

`ref_tidy(x)`

`constituency_results_tidy(results, details)`

Arguments

| | |
|----------------------|------------------------|
| <code>x</code> | The dataframe to tidy |
| <code>results</code> | The data frame to tidy |
| <code>details</code> | The list to tidy |

Index

*Topic **mnis**

- mnis_additional, 3
 - mnis_all_members, 4
 - mnis_base, 5
 - mnis_constituency_results, 6
 - mnis_department, 7
 - mnis_eligible, 7
 - mnis_extra, 8
 - mnis_full_biog, 10
 - mnis_general_election_results, 10
 - mnis_joined_between, 11
 - mnis_lords_type, 12
 - mnis_member_date, 13
 - mnis_party_state, 13
 - mnis_reference, 14
- constituency_results_tidy (mnis_tidy),
18
- mnis, 2
- mnis-package (mnis), 2
- mnis_additional, 2, 3, 8, 9
- mnis_Addresses (mnis_additional), 3
- mnis_addresses (mnis_additional), 3
- mnis_all_members, 4
- mnis_base, 2, 5
- mnis_basic_details, 9
- mnis_basic_details (mnis_additional), 3
- mnis_BasicDetails (mnis_additional), 3
- mnis_biography_entries
(mnis_additional), 3
- mnis_BiographyEntries
(mnis_additional), 3
- mnis_Committees (mnis_additional), 3
- mnis_committees (mnis_additional), 3
- mnis_Constituencies (mnis_additional), 3
- mnis_constituencies (mnis_additional), 3
- mnis_constituency_results, 2, 6
- mnis_ConstituencyResults
(mnis_constituency_results), 6
- mnis_department, 2, 7
- mnis_elections_contested
(mnis_additional), 3
- mnis_ElectionsContested
(mnis_additional), 3
- mnis_eligible, 2, 7
- mnis_Experiences (mnis_additional), 3
- mnis_experiences (mnis_additional), 3
- mnis_extra, 2, 4, 8
- mnis_full_biog, 2, 4, 8, 9, 10
- mnis_general_election_results, 2, 10
- mnis_GeneralElectionResults
(mnis_general_election_results),
10
- mnis_government_posts
(mnis_additional), 3
- mnis_GovernmentPosts (mnis_additional),
3
- mnis_Honours (mnis_additional), 3
- mnis_honours (mnis_additional), 3
- mnis_house_memberships
(mnis_additional), 3
- mnis_HouseMemberships
(mnis_additional), 3
- mnis_Interests (mnis_additional), 3
- mnis_interests (mnis_additional), 3
- mnis_joined_between, 2, 11
- mnis_known_as (mnis_additional), 3
- mnis_KnownAs (mnis_additional), 3
- mnis_lords_type, 2, 12
- mnis_maiden_speeches (mnis_additional),
3
- mnis_MaidenSpeeches (mnis_additional), 3
- mnis_member_date, 2, 13
- mnis_MemberDate (mnis_member_date), 13
- mnis_opposition_posts
(mnis_additional), 3
- mnis_OppositionPosts (mnis_additional),
3

- mnis_other_parliaments
(mnis_additional), 3
- mnis_OtherParliaments
(mnis_additional), 3
- mnis_parliamentary_posts
(mnis_additional), 3
- mnis_ParliamentaryPosts
(mnis_additional), 3
- mnis_Parties (mnis_additional), 3
- mnis_parties (mnis_additional), 3
- mnis_party_state, 2, 13
- mnis_preferred_names (mnis_additional),
3
- mnis_PreferredNames (mnis_additional), 3
- mnis_Reference (mnis_reference), 14
- mnis_reference, 2, 14
- mnis_Staff (mnis_additional), 3
- mnis_staff (mnis_additional), 3
- mnis_Statuses (mnis_additional), 3
- mnis_statuses (mnis_additional), 3
- mnis_tidy, 18

- ref_address_types (mnis_reference), 14
- ref_AddressTypes (mnis_reference), 14
- ref_answering_bodies (mnis_reference),
14
- ref_AnsweringBodies (mnis_reference), 14
- ref_area_types (mnis_reference), 14
- ref_Areas (mnis_reference), 14
- ref_areas (mnis_reference), 14
- ref_AreaTypes (mnis_reference), 14
- ref_biography_categories
(mnis_reference), 14
- ref_BiographyCategories
(mnis_reference), 14
- ref_Cabinets (mnis_reference), 14
- ref_cabinets (mnis_reference), 14
- ref_committee_types (mnis_reference), 14
- ref_Committees (mnis_reference), 14
- ref_committees (mnis_reference), 14
- ref_CommitteeTypes (mnis_reference), 14
- ref_Constituencies (mnis_reference), 14
- ref_constituencies (mnis_reference), 14
- ref_constituency_areas
(mnis_reference), 14
- ref_constituency_types
(mnis_reference), 14
- ref_ConstituencyAreas (mnis_reference),
14
- ref_ConstituencyTypes (mnis_reference),
14
- ref_Countries (mnis_reference), 14
- ref_countries (mnis_reference), 14
- ref_Departments (mnis_reference), 14
- ref_departments (mnis_reference), 14
- ref_disqualification_types
(mnis_reference), 14
- ref_DisqualificationTypes
(mnis_reference), 14
- ref_election_types (mnis_reference), 14
- ref_Elections (mnis_reference), 14
- ref_elections (mnis_reference), 14
- ref_ElectionTypes (mnis_reference), 14
- ref_end_reasons (mnis_reference), 14
- ref_EndReasons (mnis_reference), 14
- ref_experience_types (mnis_reference),
14
- ref_ExperienceTypes (mnis_reference), 14
- ref_government_post_departments
(mnis_reference), 14
- ref_government_posts (mnis_reference),
14
- ref_government_ranks (mnis_reference),
14
- ref_GovernmentPostDepartments
(mnis_reference), 14
- ref_GovernmentPosts (mnis_reference), 14
- ref_GovernmentRanks (mnis_reference), 14
- ref_honour_lists (mnis_reference), 14
- ref_honourary_prefixes
(mnis_reference), 14
- ref_HonouraryPrefixes (mnis_reference),
14
- ref_HonourLists (mnis_reference), 14
- ref_Honours (mnis_reference), 14
- ref_honours (mnis_reference), 14
- ref_interest_categories
(mnis_reference), 14
- ref_InterestCategories
(mnis_reference), 14
- ref_lords_membership_types
(mnis_reference), 14
- ref_lords_ranks (mnis_reference), 14
- ref_LordsMembershipTypes
(mnis_reference), 14
- ref_LordsRanks (mnis_reference), 14
- ref_opposition_post_departments

(mnis_reference), 14
ref_opposition_posts (mnis_reference),
14
ref_opposition_ranks (mnis_reference),
14
ref_OppositionPostDepartments
(mnis_reference), 14
ref_OppositionPosts (mnis_reference), 14
ref_OppositionRanks (mnis_reference), 14
ref_other_parliaments (mnis_reference),
14
ref_OtherParliaments (mnis_reference),
14
ref_parliament_types (mnis_reference),
14
ref_parliamentary_posts
(mnis_reference), 14
ref_parliamentary_ranks
(mnis_reference), 14
ref_ParliamentaryPosts
(mnis_reference), 14
ref_ParliamentaryRanks
(mnis_reference), 14
ref_ParliamentTypes (mnis_reference), 14
ref_Parties (mnis_reference), 14
ref_parties (mnis_reference), 14
ref_party_sub_types (mnis_reference), 14
ref_PartySubTypes (mnis_reference), 14
ref_photo_outputs (mnis_reference), 14
ref_PhotoOutputs (mnis_reference), 14
ref_Statuses (mnis_reference), 14
ref_statuses (mnis_reference), 14
ref_tidy (mnis_tidy), 18
ref_Titles (mnis_reference), 14
ref_titles (mnis_reference), 14