Package: hansard (via r-universe)

January 30, 2025

Type Package

Title Provides Easy Downloading Capabilities for the UK Parliament API

Version 0.8.0.9000

Maintainer Evan Odell <evanodell91@gmail.com>

Description Provides functions to download data from the
<https://explore.data.parliament.uk/> APIs. Because of the
structure of the API, there is a named function for each type
of available data for ease of use, as well as some functions
designed to retrieve specific pieces of commonly used data.
Functions for each new API will be added as and when they
become available.

License MIT + file LICENSE

URL https://docs.evanodell.com/hansard,
https://ropengov.github.io/hansard/,
https://github.com/rOpenGov/hansard

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

Depends R (>= 3.2.0)

Imports dplyr, httr, jsonlite, lubridate, snakecase, tibble, tidyr

Suggests covr, knitr, methods, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData TRUE

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

X-schema.org-isPartOf http://ropengov.org/

X-schema.org-keywords ropengov

Config/pak/sysreqs libicu-dev libssl-dev

Repository https://ropengov.r-universe.dev

RemoteUrl https://github.com/rOpenGov/hansard

RemoteRef HEAD

RemoteSha e9802c39f14e445e746d91fe2cf1651f2969510c

Contents

---

all_answered_questions

All answered parliamentary questions

Description

Imports data on all answered parliamentary questions in the House of Commons and/or in the House of Lords. The mp_id, tabling_mp_id and answering_body parameters accept a single ID or department names, or a list of IDs or department names, respectively.

This is the most flexible of the various functions that look up questions, as it queries all types of questions in both houses with a wide selection of parameters: The member who asks the question, the member who answers it and the relevant department can all be used to query the API. mp_id, tabling_mp_id and answering_body all accept lists of multiple relevant search parameters. This can be in the form of a list, a data.frame column, a character vector, etc.

Usage

```
all_answered_questions(
  mp_id = NULL,
  tabling_mp_id = NULL,
  house = NULL,
  answering_body = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_all_answered_questions(
  mp_id = NULL,
  tabling_mp_id = NULL,
  house = NULL,
  answering_body = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

mp_id   Accepts a member ID or vector of member IDs, and returns a tibble with all
               available questions answered by that member. Includes both oral and written

questions, and includes members of the House of Commons and the House of Lords. If NULL, returns a tibble with all available answered questions, subject to other parameters. Defaults to NULL.

tabling_mp_id   Accepts a member ID or vector of member IDs, and returns a tibble with all available questions asked by that member, subject to all other parameters. Includes both oral and written questions, and includes members of the House of Commons and the House of Lords. If NULL, returns a tibble with all available answered questions, subject to other parameters. Defaults to NULL.

house   The house to return questions from. Accepts either the short name of the legislature (e.g. 'commons' or 'lords') or the ID of the legislature (1 for the House of Commons, 2 for the House of Lords). The short names are not case sensitive. If NULL, returns answers from both houses, subject to other parameters. Defaults to NULL.

answering_body   The name of the government department that answers the question, or a vector of government deparment names. Accepts either the short name name of a department (e.g. 'Education' for the Department for Education, 'Digital, Culture, Media and Sport' for the Department for Digital, Culture, Media and Sport), or the ID of a particular department (e.g. 60 for the Department for Education). If NULL, returns answers from all departments, subject to other parameters. Defaults to NULL.

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

end_date   The latest date to include in the tibble. Defaults to '1900-01-01'. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.

extra_args   Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.

tidy   Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.

verbose   If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on all answered questions in the House of Commons and the House of Lords.

See Also

commons_answered_questions()

commons_oral_questions()

commons_oral_question_times()

commons_written_questions()

lords_written_questions()

mp_questions()

Examples

```
## Not run:
# All questions answered by Nicola Blackwood from 1 January 2017 onwards
x <- all_answered_questions(4019, start_date = "2017-01-01")

# All questions answered by Nicola Blackwood from 1 January 2017 onwards
# returns variables in camelCase style
y <- all_answered_questions(4019,
  start_date = "2017-01-01",
  tidy_style = "small_camel"
)

# All questions asked by Andrew Dismore from 1 January 2017 onwards
z <- hansard_all_answered_questions(
  tabling_mp_id = 179,
  start_date = "2017-01-01"
)

# Return all questions asked in the House of Lords
# answered by the Department for Education.
a <- hansard_all_answered_questions(house = "lords", answering_body = 60)

# Returns all questions asked in the House of Lords
# answered by the Department for Education.
b <- hansard_all_answered_questions(house = 2, answering_body = "Education")

# Accepts multiple inputs for mp_id, tabling_mp_id and answering_body
w <- hansard_all_answered_questions(
  mp_id = c(4019, 3980),
  tabling_mp_id = c(338, 172),
  answering_body = c("health", "justice"),
  start_date = "2016-12-18",
  end_date = "2017-03-12"
)

## End(Not run)
```

---

bills   *Bill data*

Description

Imports data on House of Commons and House of Lords bills.

Usage

```
bills(
  ID = NULL,
  amendments = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_bills(
  ID = NULL,
  amendments = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

ID	The ID of a given bill to return data on. If NULL, returns all bills, subject to other parameters. Defaults to NULL.
amendments	If TRUE, returns all bills with amendments, subject to other parameters. Defaults to FALSE.
start_date	Only includes bills introduced on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.
end_date	Only includes bills introduced on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on bills before the House of Lords and the House of Commons.

See Also

bill_stage_types()

bill_publications()

Examples

```
## Not run:
# Download data on all bills
x <- bills()

# Download data on all bill amendments
x <- bills(amendments = TRUE)

# Download data on a specific bills
x <- bills(1719)

# Download data on all bills introduced after a given date
x <- bills(start_date = "2016-01-01")

## End(Not run)
```

---

bill_publications   *Bill Publications*

Description

Returns details of all publications associated with a specific bill or all bills.

Usage

```
bill_publications(
  ID = NULL,
  publication_type = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = FALSE
)

hansard_bill_publications(
  ID = NULL,
```

```
    publication_type = NULL,
    start_date = "1900-01-01",
    end_date = Sys.Date(),
    extra_args = NULL,
    tidy = TRUE,
    tidy_style = "snake",
    verbose = FALSE
)
```

Arguments

ID	The ID of a specific bill to return publications for. If NULL, returns all bill publications subject to other parameters. Defaults to NULL.
publication_type	
	The type of bill publication to return, in the form of a string. For a character vector of bill publication types, see [bill_publication_types()](). If NULL, returns all publications of all types, subject to other parameters. Defaults to NULL.
start_date	Only includes bill publications on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.
end_date	Only includes bill publicationson or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation]() or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](). Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on bill publications.

See Also

[bill_stage_types()]()

[bills()]()

[bill_publication_types()]()

Examples

```
## Not run:
# Requesting a specific publication
x <- bill_publications(ID = 752025)

# Requesting all publications after a given date
y <- bill_publications(start_date = "2018-01-01")

## End(Not run)
```

---

bill_publication_types

> *A character vector of possible types of bill publications. For use with*
> bill_publications()

Description

A character vector of possible types of bill publications. For use with bill_publications()

Usage

```
bill_publication_types
```

Format

A character vector containing 38 elements

See Also

bill_publications()

bills()

bill_stage_types()

---

bill_stage_types   *Bill Stage Types*

Description

Returns a tibble with all possible bill stage types.

Usage

```
bill_stage_types(tidy = TRUE, tidy_style = "snake", verbose = TRUE)

hansard_bill_stage_types(tidy = TRUE, tidy_style = "snake", verbose = TRUE)
```

Arguments

`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by [to_any_case](#). Defaults to `'snake'`.
`verbose`	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble of bill stage types.

See Also

[bills()](#)

[bill_publications()](#)

Examples

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

## End(Not run)
```

---

commons_answered_questions   *House of Commons answered questions*

Description

Imports data on House of Commons answered questions. If all parameters are left empty, imports all available answered questions in a tibble.

If `answering_department` and/or `answered_by` are given a vector with multiple deparments/IDs, all possible combination of those criteria are returned.

Usage

```
commons_answered_questions(
  answering_department = NULL,
  answered_by = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
```

```
  tidy_style = "snake",
  verbose = TRUE
)

hansard_commons_answered_questions(
  answering_department = NULL,
  answered_by = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

answering_department

Accepts the name of a department or a list of department names. Returns a tibble with all answered questions in the House of Commons from the given department. Defaults to NULL.

answered_by   Accepts the ID of an MP, or a list of IDs. Returns a tibble with all answered questions in the House of Commons by the given MP(s). Defaults to NULL.

start_date   Only includes questions answered introduced on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.

end_date   Only includes questions answered on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.

extra_args   Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.

tidy   Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.

verbose   If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on all answered questions in the House of Commons.

See Also

all_answered_questions()

commons_oral_questions()

commons_oral_question_times()

commons_written_questions()

lords_written_questions()

mp_questions()

Examples

```
## Not run:
x <- commons_answered_questions(
  answering_department = c("health", "education"),
  answered_by = c("4019", "1542", "111"),
  start_date = "2017-01-01"
)

y <- commons_answered_questions(
  start_date = "2017-03-26",
  end_date = "2017-04-01"
)

## End(Not run)
```

---

commons_divisions   *House of Commons divisions*

Description

Imports data on House of Commons divisions (votes), either full details on how each member voted, or a summary of vote totals.

Usage

```
commons_divisions(
  division_id = NULL,
  division_uin = NULL,
  summary = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

```
hansard_commons_divisions(
  division_id = NULL,
  division_uin = NULL,
  summary = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

division_id	The id of a particular vote. If empty, returns a tibble with information on all commons divisions, subject to all other parameters. Defaults to `NULL`. Only accepts a single division ID at a time, but to return details on a list of division IDs use with `lapply`. The `division_id` takes priority over the `division_uid` parameter.
division_uin	The UIN of a particular vote. If empty, returns a tibble with information on all commons divisions, subject to all other parameters. Defaults to `NULL`. Only accepts a single division UIN at a time, but to return details on a list of division UINs use with `lapply`.
summary	If `TRUE`, returns a small tibble summarising a division outcome. Otherwise returns a tibble with details on how each MP voted. Has no effect if `division_id` is empty. Defaults to `FALSE`.
start_date	Only includes divisions on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
end_date	Only includes divisions on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to `NULL`.
tidy	Logical parameter. If `TRUE`, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
verbose	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with the results of divisions in the House of Commons.

Examples

```
## Not run:

## All commons divisions
x <- commons_divisions()

## Vote breakdown of specific commons division
y <- commons_divisions(division_id = 694163, summary = FALSE)

## End(Not run)
```

---

commons_division_date   *House of Commons Division Dates*

Description

Returns a tibble with the divisions (votes) in the House of Commons on a given date.

Usage

```
commons_division_date(
  date = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_commons_division_date(
  date = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

date   Returns all divisions on a given date. Defaults to NULL.

extra_args   Additional parameters to pass to API. Defaults to NULL.

tidy   Logical parameter. If TRUE, fixes 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 any style ac-
              cepted by [to_any_case](). Defaults to 'snake'.

verbose   If TRUE, displayes messages on the console on the progress of the API request.
              Defaults to TRUE.

Value

A tibble with the dates of divisions in the House of Commons.

Examples

```
## Not run:
# all commons divisions on 19 April 2017
x <- commons_division_date("2017-04-19")

## End(Not run)
```

---

commons_oral_questions

House of Commons Oral Questions

Description

Imports data on House of Commons oral questions, based on the asking MP, the answering department and the date. The `mp_id` and `answering_department` parameters accept a single ID or department names, or a list of IDs or department names, respectively.

Usage

```
commons_oral_questions(
  mp_id = NULL,
  answering_department = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_commons_oral_questions(
  mp_id = NULL,
  answering_department = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`mp_id`	The ID of a given MP asking an oral question, or a list of MP Ids. Defaults to `NULL`.
`answering_department`	
	The name of a department, or a list of departments. Defaults to `NULL`.
`start_date`	Only includes questions answered on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`.] Defaults to `'1900-01-01'`.
`end_date`	Only includes questions answered on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to `NULL`.
`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
`verbose`	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with details on all oral questions in the House of Commons.

See Also

all_answered_questions()

commons_answered_questions()

commons_oral_question_times()

commons_written_questions()

lords_written_questions()

mp_questions()

Examples

```
## Not run:

# Oral questions from a single MP to a single department
x <- commons_oral_questions(mp_id = 4019, answering_department = "education")


## Questions from multiple MPs and to multiple departments
```

```
y <- commons_oral_questions(
  mp_id = c(4019, 4051, 4588),
  answering_department = c("education", "health")
)


## End(Not run)
```

commons_oral_question_times

Commons oral question times

Description

Imports data on House of Commons oral question times. Query with parameters for the parliamentary session or the question ID. If tidy=TRUE, datetime variables are converted to POSIXct class.

Usage

```
commons_oral_question_times(
  session = NULL,
  question_id = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_commons_oral_question_times(
  session = NULL,
  question_id = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

session	Accepts a session in format yyyy/yy (e.g. "2016/17") and returns a tibble of all oral question times from that session. Defaults to NULL.
question_id	Accepts a question time ID, and returns a tibble of that question time.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.

tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with information on oral question times in the House of Commons.

See Also

all_answered_questions()

commons_answered_questions()

commons_oral_questions()

commons_written_questions()

lords_written_questions()

mp_questions()

Examples

```
## Not run:
x <- commons_oral_question_times(session = "2016/17", question_id = "685697")

## End(Not run)
```

---

commons_terms   *Parliamentary Thesaurus*

Description

Imports the parliamentary thesaurus. The API is rate limited to 5500 requests at a time, so some use of parameters is required.

Usage

```
commons_terms(
  search = NULL,
  class = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
```

```
)

hansard_commons_terms(
  search = NULL,
  class = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

search	A string to search the parliamentary thesaurus for.
class	The class of definition to be returned Accepts one of `'ID'`, `'ORG'`, `'SIT'`, `'NAME'`, `'LEG'`, `'CTP'`, `'PBT'` and `'TPG'`. Defaults to `NULL`.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to `NULL`.
tidy	Logical parameter. If `TRUE`, fixes 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 any style accepted by [to_any_case](#). Defaults to `'snake'`.
verbose	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with results from the parliamentary thesaurus.

Examples

```
## Not run:
x <- commons_terms(search = "estate")

x <- commons_terms(search = "estate", class = "ORG")

## End(Not run)
```

---

commons_written_questions

House of Commons Written Questions

Description

Imports data in a tibble on House of Commons written questions.

Usage

```
commons_written_questions(
  mp_id = NULL,
  answering_department = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_commons_written_questions(
  mp_id = NULL,
  answering_department = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

mp_id   Accepts a member ID or a list of member IDs and returns a tibble with all written
                      questions asked by that MP or list of MPs. If NULL, mp_id is not included as a
                      query parameter. Defaults to NULL.

answering_department
                      Accepts a string with a department name or partial name, or a list of such strings.
                      The query acts as a search, so passing 'health' will return all questions an-
                      swered by the Department of Health. If NULL, answering_department is not
                      included as a query parameter. Defaults to NULL.

start_date   Only includes questions tabled on or after this date. Accepts character values
                      in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt
                      or anything else that can be coerced to a date with as.Date(). Defaults to
                      '1900-01-01'.

end_date	Only includes questions tabled on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on written questions in the House of Commons.

Examples

```
## Not run:
# Returns a tibble with written questions from Jon Trickett,
# answered by the Cabinet Office.
x <- commons_written_questions(
  mp_id = 410,
  answering_department = "cabinet office"
)

# Returns a tibble with written questions from Jon Trickett or Diane Abbott,
# and answered by the Cabinet Office or the Home Office.
x <- commons_written_questions(
  mp_id = c(410, 172),
  answering_department = c("cabinet", "home")
)

## End(Not run)
```

---

constituencies	*House of Commons constituencies*

Description

Imports data on House of Commons constituencies, returning a tibble of all current and/or former Westminster constituencies, subject to parameters.

Usage

```
constituencies(
  current = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_constituencies(
  current = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

current	If TRUE, returns only current constituencies. If FALSE, returns only former constituencies. If NULL, returns all current and former constituencies. Defaults to NULL.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details of Westminster constituencies.

Examples

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

y <- constituencies(current = FALSE)

## End(Not run)
```

---

early_day_motions   *Early day motion data*

Description

Return data on the content, signatories, and sponsors of early day motions (EDMS).

Usage

```
early_day_motions(
  edm_id = NULL,
  session = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  signatures = 1,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_early_day_motions(
  edm_id = NULL,
  session = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  signatures = 1,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

edm_id	Accepts the ID number of an early day motion, and returns data on that motion. If NULL, returns all available Early Day Motions. Note that there, are as of 2017-06-15, 43,330 early day motions on listed in the API, so requesting all early day motions without other parameters is slow and very demanding on the API itself. Defaults to NULL.
session	Accepts a parliamentary session, in 'yyyy/yy' format. Defaults to NULL.
start_date	Only includes early day motions tabled on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.

end_date   Only includes early day motions tabled on or before this date. Accepts character
                  values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`,
                  `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults
                  to the current system date.

signatures   The minimum number of signatures required for inclusion in the tibble. Defaults
                  to 1.

extra_args   Additional parameters and queries to pass to API. These queries must be strings
                  and start with "&". See the [API documentation](#) or the package vignette for more
                  details. Defaults to `NULL`.

tidy   Logical parameter. If `TRUE`, fixes 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 any style ac-
                  cepted by [to_any_case](#). Defaults to `'snake'`.

verbose   If `TRUE`, displays messages on the console on the progress of the API request.
                  Defaults to `TRUE`.

Details

Early Day Motion IDs reset for each parliamentary session, so not including a query for `session`
but including an `edm_id` will return multiple early day motions with the same ID code from different
parliamentary sessions.

Value

A tibble with details on the content, signatories and sponsors of all or a specified early day motions.

See Also

[mp_edms()](#)

Examples

```
## Not run:

# Returns all EDMs with a given ID
x <- early_day_motions(edm_id = 1073)

# Return a specific early day motion by ID
x <- early_day_motions(edm_id = 1073, session = "2017/19")

## End(Not run)
```

---

edm_text   *Early Day Motion Text*

Description

A quick and dirty function for a specific use case, use with caution.

Usage

```
edm_text(id, tidy = TRUE, tidy_style = "snake", verbose = TRUE)
```

Arguments

id	The ID of an individual Early Day Motion, or a vector of IDs, as found in the about column of returns from early_day_motions()
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble of containing the EDM text and its ID.

Examples

```
## Not run:
y <- edm_text(c("811291", "811292", "811293"))

## End(Not run)
```

---

elections   *General and By-Elections*

Description

Imports data on general and by-elections from the 2010 General Election onwards.

If both ID and type are used to query the API, ID takes precedence and type is ignored.

Usage

```
elections(
  ID = NULL,
  type = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  label = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_elections(
  ID = NULL,
  type = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  label = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

ID	Accepts an ID for a general or by-election from the 2010 General Election onwards, and returns the date and type of the elction. If NULL, returns the date and type of all available elections, subject to other parameters. Defaults to NULL.
type	Accepts 'General Election' or 'By-election' as arguments if ID is NULL, and returns all General Elections or all By-elections, as specified, subject to other parameters. Defaults to NULL.
start_date	Only includes elections held on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.
end_date	Only includes elections held on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.
label	Label of the election. By-elections are in 'dd-mmm-yyyy By-election' format; e.g. '23-Feb-2017 By-election', and general elections use 'YYYY General Election' format. The parameter cannot search, so check your formatting, spelling and make sure there were actually elections with the label specified. If NULL, returns all Elections/By-elections subject to other parameters. Defaults to NULL.

extra_args   Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to NULL.

tidy   Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](#). Defaults to 'snake'.

verbose   If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on all elections from the 2010 General Election onwards, subject to function parameters. Includes the election ID, the date, and the type of election(s).

See Also

[election_results()](#)

[election_candidates()](#)

Examples

```
## Not run:
x <- elections(517994)

## End(Not run)
```

---

election_candidates   *Election candidate details*

Description

Returns the name and party of all candidates standing in an election, for each constituency. Note that for general elections this can return a very large tibble with hundreds of variables.

Usage

```
election_candidates(
  ID = NULL,
  constit_details = FALSE,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

```
hansard_election_candidates(
  ID = NULL,
  constit_details = FALSE,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`ID`	Accepts an ID for a general or by-election from the 2010 General Election onwards, and returns the results. If `NULL`, returns all available election results. Defaults to `NULL`.
`constit_details`	
	If `TRUE`, returns additional details on each constituency, including its GSS (Government Statistical Service) code. Defaults to `FALSE`.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to `NULL`.
`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by [to_any_case](#). Defaults to `'snake'`.
`verbose`	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

See Also

[elections()](#)

[election_results()](#)

Examples

```
## Not run:
x <- election_candidates(ID = 382037)

y <- election_candidates()

z <- election_candidates(constit_details = TRUE)

## End(Not run)
```

---

election_results   *General and By-Election Results*

Description

Imports results from general and by-elections from the 2010 General Election onwards.

Usage

```
election_results(
  ID = NULL,
  all_data = FALSE,
  calculate_percent = FALSE,
  constit_details = FALSE,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_election_results(
  ID = NULL,
  all_data = FALSE,
  calculate_percent = FALSE,
  constit_details = FALSE,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

ID
: Accepts an ID for a general or by-election from the 2010 General Election onwards, and returns the results. If NULL, returns all available election results. Defaults to NULL.

all_data
: If TRUE, returns vote share for all parties standing in any constituency in the election/elections returned. Defaults to FALSE. Note that aside from shorthand for the Conservatives, Labour, Liberal Democrat and Independent (Con, Lab, Lib and Ind, respectively) being converted to their full names, party names are not tidied, so will contain spaces in the case of parties with multiple words in their name, such as the Liberal Democrats. If a party did not stand in a constituency its vote count is listed as NA. There is a drawback to using this parameter, as multiple candidates from the same party in a constituency, or multiple independent candidates, have their vote totals combined.

calculate_percent

>   If TRUE, calculates the turnout percentage for each constituency in the tibble and
>   the majority of the winning candidate to one decimal place, and includes this
>   information in the tibble in additional columns labelled 'turnout_percentage'
>   and 'majority_percentage'. Defaults to FALSE.

constit_details

>   If TRUE, returns additional details on each constituency, including its GSS (Gov-
>   ernment Statistical Service) code. Defaults to FALSE.

extra_args   Additional parameters and queries to pass to API. These queries must be strings
>   and start with "&". See the API documentation or the package vignette for more
>   details. Defaults to NULL.

tidy   Logical parameter. If TRUE, fixes 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 any style ac-
>   cepted by to_any_case. Defaults to 'snake'.

verbose   If TRUE, displays messages on the console on the progress of the API request.
>   Defaults to TRUE.

Value

A tibble with the results of all general and by-elections, or of a specified general election or by-election.

See Also

elections()

election_candidates()

Examples

```
## Not run:
x <- election_results(ID = 382037)

y <- election_results()

z <- election_results(calculate_percent = TRUE, constit_details = TRUE)

w <- election_results(ID = 730039, all_data = TRUE)

## End(Not run)
```

---

epetition   *Individual epetitions*

Description

Imports data on a given epetition. For bulk epetion data, see [epetition_tibble()](#).

Usage

```
epetition(
  ID = NULL,
  by_constituency = FALSE,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_epetition(
  ID = NULL,
  by_constituency = FALSE,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

ID   The ID of a given petition. If `NULL`, returns all epetitions. Defaults to `NULL`.
                    See [epetition_tibble()](#) for a greater degree of flexibility when querying all
                    epetitions.

by_constituency

                    Accepts either `TRUE` or `FALSE`. If `TRUE`, provides a tibble with a breakdown of
                    signatures for each petition, by constituency. Defaults to `FALSE`.

extra_args   Additional parameters and queries to pass to API. These queries must be strings
                    and start with "&". See the [API documentation](#) or the package vignette for more
                    details. Defaults to `NULL`.

tidy   Logical parameter. If `TRUE`, fixes 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 any style ac-
                    cepted by [to_any_case](#). Defaults to `'snake'`.

verbose   If `TRUE`, displays messages on the console on the progress of the API request.
                    Defaults to `TRUE`.

Value

A tibble with details on electronic petitions submitted to parliament.

See Also

[epetition_tibble()](#)

Examples

```
## Not run:
x <- epetition(ID = 706964, by_constituency = TRUE)

## End(Not run)
```

---

epetition_tibble   *Bulk Epetition data*

Description

Get data on all epetitions submitted to parliament, with the label, sponsor, number of signatures, date created and epetition ID. For greater detail on indidivual epetitions, see [epetition()](#).

Usage

```
epetition_tibble(
  min_signatures = 1,
  max_signatures = NULL,
  status = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_epetition_tibble(
  min_signatures = 1,
  max_signatures = NULL,
  status = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`min_signatures`	The minimum number of signatures required for inclusion in the tibble. Defaults to 1.
`max_signatures`	The maximum number of signatures required for inclusion in the tibble. If `NULL`, there is no maximum number of signatures. Defaults to `NULL`.
`status`	The status of the petition, either `'open'` or `'closed'`. If `NULL`, returns all petitions both open and closed. Defaults to `NULL`.
`start_date`	Only includes epetitions created on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
`end_date`	Only includes epetitions created on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or a nything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to `NULL`.
`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
`verbose`	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with details on electronic petitions submitted to parliament.

See Also

epetition()

Examples

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

y <- epetition_tibble(max_signatures = 500)

z <- epetition_tibble(start_date = "2016-12-01", end_date = "2017-04-25")

## End(Not run)
```

| hansard | *hansard: Provides Easy Downloading Capabilities for the UK Parliament API* |

Description

Provides functions to request data from the data.parliament.uk APIs. Because of the structure of the data.parliament.uk API, there is a named function for each type of available data for ease of use. Functions for each new API will be added as and when they become available on <data.parliament.uk>. The API is rate limited to returning 5500 rows per request in some instances, though this has been inconsistent in testing.

Details

The API itself is still in beta, and only about half of the planned datasets are currently available. The package name is optimistic, as the actual contents of the Hansard are not yet available through this API.

In addition to the standard function names, each function in the hansard package has a wrapper where the name is prefixed with 'hansard_'. For example, both bills() and hansard_bills() will return the same result. This is because function names are taken from the specific API on <http://explore.data.parliament.uk/>, but they are often not very informative and could clash with functions in other packages (e.g. bills() is not a term unique to the British parliament).

For more details please see the [vignette](), or the API documentation on http://explore.data.parliament.uk/.

This package is in no way officially related to or endorsed by the UK Parliamentary Data Service.

| lords_amendments | *House of Lords Amendments.* |

Description

Returns a tibble with all available House of Lords amendments, subject to parameters.

Usage

```
lords_amendments(
  decision = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

```
hansard_lords_amendments(
  decision = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

decision	The decision on the amendments. Accepts one of `'Withdrawn'`, `'Agreed'`, `'Disagreed'`, `'Pending'`, `'NotMoved'`, `'Disposed'`. This parameter is not case sensitive. Defaults to `NULL`.
start_date	Only includes amendments to bills introduced on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
end_date	Only includes amendments to bills introduced on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to `NULL`.
tidy	Logical parameter. If `TRUE`, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
verbose	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with details on amendments proposed by the House of Lords.

Examples

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

x <- lords_amendments(decision = "Withdrawn")

## End(Not run)
```

---

lords_attendance_date   *House of Lords attendance by date*

Description

Imports data on House of Lords attendance on a given date.

Usage

```
lords_attendance_date(
  date = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_attendance_date(
  date = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

date	Accepts a date to return attendance data for. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](). Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Details

Please note that House of Lords attendance data is not as tidy as some of the others that are accessible through this API, and so additional work on the return from the API may be required.

Also note that this API does not appear to have been updated with data after 2017-01-31.

Value

A tibble with details on the lords who attended on a given date.

See Also

lords_attendance_session()

Examples

```
## Not run:
x <- lords_attendance_date(date = "2016-03-01")

## End(Not run)
```

---

lords_attendance_session

House of Lords attendance by session

Description

Imports data on House of Lords attendance from each parliamentary session. Please note that the attendance data is not as tidy as some of the others that are accessible through this API, and so additional work to prepare this data in a way that you want may be required.

Usage

```
lords_attendance_session(
  session_id = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_attendance_session(
  session_id = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

session_id	The ID of the House of Lords session. If NULL, returns a list of all sessions, subject to other parameters. Defaults to NULL.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.

tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Details

To return a tibble with all codes from available individual sessions, use lords_attendance_session(session_id=NULL), or use lords_sessions() to retrieve codes for a given date range. Attendance from multiple sessions can be accessed by using lapply with the output from lords_sessions().

Please note that House of Lords attendance data is not as tidy as some of the others that are accessible through this API, and so additional work on the return from the API may be required.

Also note that this API does not appear to have been updated with data after 2017-01-31.

Value

A tibble with details on the lords who attended a given session.

See Also

lords_attendance_date()

lords_sessions()

Examples

```
## Not run:
x <- lords_attendance_session(session_id = 706178)

# Returns a list of data frames with details of
# attendance for each day of a given vector of sessions.
u <- lords_sessions(start_date = "2017-01-01")
m <- lapply(u$about, lords_attendance_session)

## End(Not run)
```

---

lords_divisions *House of Lords divisions*

Description

Imports data on House of Lords divisions. Either a general query subject to parameters, or the results of a specific division. Individual divisions can be queried to return a short summary of the votes, or details on how each peer voted.

Usage

```
lords_divisions(
  division_id = NULL,
  summary = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_divisions(
  division_id = NULL,
  summary = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

division_id	The id of a particular vote. If empty, returns a tibble with information on all lords divisions. Defaults to NULL.
summary	If TRUE, returns a small tibble summarising a division outcome. Otherwise returns a tibble with details on how each peer voted. Has no effect if division_id is empty. Defaults to FALSE.
start_date	Only includes divisions on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.
end_date	Only includes divisions on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with the results of divisions in the House of Lords.

Examples

```
## Not run:
x <- lords_divisions(division_id = 705891, summary = TRUE)

x <- lords_divisions(division_id = 705891, summary = FALSE)

# Return all lords divisions in 2016
x <- lords_divisions(NULL, FALSE,
  start_date = "2016-01-01",
  end_date = "2016-12-31"
)

## End(Not run)
```

---

lords_interests   *Peers' registered interests*

Description

Registered financial interests of members of the House of Lords. If peer_id=NULL the actual details of registered interests are stored in a nested data frame.

Usage

```
lords_interests(
  peer_id = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_interests(
  peer_id = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`peer_id`	The ID of a member of the House of lords. If NULL, returns a tibble with all listed financial interests for all members. Defaults to NULL.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
`tidy`	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
`verbose`	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on the interests of peers in the House of Lords.

Examples

```
## Not run:
x <- lords_interests(4170)

y <- lords_interests()

## End(Not run)
```

---

lords_sessions   *Lords sessions*

Description

Returns the session code and other basic details for individual House of Lords sittings. Note that this API does not appear to have been updated with data after 2017-01-31.

Usage

```
lords_sessions(
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_sessions(
```

```
    start_date = "1900-01-01",
    end_date = Sys.Date(),
    tidy = TRUE,
    tidy_style = "snake",
    verbose = TRUE
)
```

Arguments

start_date   Only includes sessions starting on or after this date. Accepts character values
                in 'YYYY-MM-DD' format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt`
                or anything else that can be coerced to a date with `as.Date()`. Defaults to
                '1900-01-01'.

end_date   Only includes sessions ending on or before this date. Accepts character values
                in 'YYYY-MM-DD' format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt`
                or anything else that can be coerced to a date with `as.Date()`. Defaults to the
                current system date.

tidy   Logical parameter. If `TRUE`, fixes 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 any style ac-
                cepted by [to_any_case](). Defaults to `'snake'`.

verbose   If `TRUE`, displays messages on the console on the progress of the API request.
                Defaults to `TRUE`.

See Also

[lords_attendance_session()]()

[sessions_info()]()

Examples

```
## Not run:

a <- lords_sessions(start_date = "2017-01-01", end_date = "2017-01-31")

## End(Not run)
```

---

lords_written_questions

                              House of Lords written questions

---

Description

Imports data on House of Lords written questions asked by a given peer(s), and/or directed to a
given department.

Usage

```
lords_written_questions(
  peer_id = NULL,
  answering_department = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_written_questions(
  peer_id = NULL,
  answering_department = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

peer_id   Accepts a member ID or list of IDs, and returns a tibble with all written questions
                 asked by that member or members. If NULL, peer_id is not included in the query
                 to the API and so all written questions are returned subject to other function
                 parameters. Defaults to NULL.

answering_department
                 Accepts a string with a department name or partial name or a list of departmental
                 names, and returns all written questions by that department. The query acts
                 as a search, so entering 'health' will return all questions answered by the
                 Department of Health. If NULL, answering_department is not included as a query
                 parameter. Defaults to NULL.

start_date   Only includes questions tabled on or after this date. Accepts character values
                 in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt
                 or anything else that can be coerced to a date with as.Date(). Defaults to
                 '1900-01-01'.

end_date   Only includes questions tabled on or before this date. Accepts character values
                 in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt
                 or anything else that can be coerced to a date with as.Date(). Defaults to the
                 current system date.

extra_args   Additional parameters and queries to pass to API. These queries must be strings
                 and start with "&". See the API documentation or the package vignette for more
                 details. Defaults to NULL.

tidy	Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](). Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on written questions in the House of Lords.

See Also

[all_answered_questions()]()

[commons_answered_questions()]()

[commons_oral_questions()]()

[commons_oral_question_times()]()

[commons_written_questions()]()

[mp_questions()]()

Examples

```
## Not run:
x <- lords_written_questions() # Returns all written questions ever

x <- lords_written_questions(
  peer_id = c(3526, 4176),
  answering_department = c(
    "cabinet",
    "Transport"
  )
)

## End(Not run)
```

---

lord_vote_record   *Voting record of members of the House of Lords*

Description

Accepts an ID number for a member of the House of Commons, and returns a tibble of their votes.

Usage

```
lord_vote_record(
  peer_id = NULL,
  lobby = "all",
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lord_vote_record(
  peer_id = NULL,
  lobby = "all",
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

peer_id	The ID number of a member of the House of Lords. A value must be included for this parameter. Use the `lords_members()` to find IDs for members of the House of Lords. Defaults to NULL.
lobby	Accepts one of `'all'`, `'content'`, `'notcontent'`. `'content'` returns votes where the peer voted 'Content', `'notcontent'` returns votes where the peer voted 'Not Content' and `'all'` returns all available votes by the peer. This parameter is not case sensitive. Defaults to `'all'`.
start_date	Only includes divisions on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
end_date	Only includes divisions on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.

verbose	If TRUE, displayes messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on the voting record of a member of the House of Lords.

Examples

```
## Not run:
x <- lord_vote_record(530, lobby = "all")

x <- lord_vote_record(530, lobby = "content")

x <- lord_vote_record(530, lobby = "notcontent")

x <- lord_vote_record(530, lobby = "not-content")
# This will also work

## End(Not run)
```

---

members   *Members of both houses*

Description

Imports basic details on current and former Members of Parliament including the Lords and the Commons. For more details on a given member see mnis_full_biog from the mnis package.

Usage

```
members(
  ID = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_members(
  ID = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

```
commons_members(
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_commons_members(
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

lords_members(
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_lords_members(
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`ID`	The ID of a member of the House of Commons or the House of Lords to return data on. If `NULL`, returns a tibble of all members of both houses. Defaults to `NULL`.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to `NULL`.
`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by [to_any_case](#). Defaults to `'snake'`.
`verbose`	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with data on members of the House of Commons (`commons_members()`), the House of Lords, (`lords_members()`), or both (`members()`).

Member details functions

members Basic details on a given member from either house

commons_members MPs in the House of Commons

lords_members Peers in the House of Lords

See Also

members_search()

Examples

```
## Not run:
a <- members()

x <- members(172)

y <- commons_members()

z <- lords_members()

## End(Not run)
```

---

members_search   *Search for an MP or Peer by name and constituency*

Description

Note that there are problems with the remote endpoint for this API, and that correct search queries
may not return any results.

Usage

```
members_search(
  search = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_members_search(
  search = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

search	Accepts any lucene query string, using * as a multiple character wildcard, and ? as the single character wildcard. Searchs are not case sensitive. If NULL, returns a tibble with all members of both houses of parliament. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](). Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Details

Function searches for the string and returns a tibble with all matches from both houses of parliament. Returns all partial matches in the members' names, constituencies, twitter handle and webpage. The default search is NULL, which returns a tibble of all members of both houses, the same result as members().

Value

A tibble with the results of the search.

See Also

[members()]()

Examples

```
## Not run:
x <- members_search("*chris*")

x <- members_search(search = "*chris*")

## End(Not run)
```

---

mp_edms   *Early Day Motions by MP or Peer*

Description

Imports data on early day motions signed, sponsored or primarily sponsored by a given MP or Peer.

Usage

```
mp_edms(
  mp_id = NULL,
  primary_sponsor = TRUE,
  sponsor = TRUE,
  signatory = TRUE,
  full_data = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_mp_edms(
  mp_id = NULL,
  primary_sponsor = TRUE,
  sponsor = TRUE,
  signatory = TRUE,
  full_data = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

mp_id	The ID number of an MP or Peer. Required parameter, Defaults to NULL. Accepts single IDs in numerical or character format, or a list, data.frame column, etc. If given multiple IDs, the results are combined into a single tibble.
primary_sponsor	
	Includes all early day motions where the given member is the primary sponsor in the tibble. Defaults to TRUE.
sponsor	Includes all early day motions where the given member a sponsor (but not the primary sponsor) in the tibble. Defaults to TRUE.
signatory	Includes all early day motions signed (but not sponsored or primarily sponsored) by the given member in the tibble. Defaults to TRUE.
full_data	If TRUE, returns all available data on the EDMs signed or sponsored by a member. Defaults to FALSE. Note that this can be a very slow process compared to other hansard functions.
start_date	Only includes early day motions signed on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct,

	POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.
end_date	Only includes early day motions signed on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with information on the tibbles signed, sponsored and/or primarily sponsored by the given MP.

See Also

early_day_motions()

Examples

```
## Not run:

# All EDMs primarily sponsored, sponsored or signed by Mike Crockart
x <- mp_edms(
  mp_id = 3967, primary_sponsor = TRUE,
  sponsor = TRUE, signatory = TRUE
)

# Detailed data on all EDMs primarily sponsored by Mike Crockart
y <- mp_edms(
  mp_id = 3967, primary_sponsor = TRUE,
  sponsor = TRUE, signatory = FALSE, full_data = TRUE
)

## End(Not run)
```

---

mp_questions   *Questions asked by a given MP or MPs*

Description

Accepts an ID number for a member of the House of Commons, and returns a tibble of of all their oral and written questions.

Usage

```
mp_questions(
  mp_id = NULL,
  question_type = "all",
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_mp_questions(
  mp_id = NULL,
  question_type = "all",
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

mp_id	The ID number of a member of the House of Commons, or a vector of IDs. Defaults to `NULL`.
question_type	Accepts the arguments `'all'`, `'oral'` and `'written'`. This parameter is not case sensitive. Defaults to `'all'`.
start_date	Only includes questions answered on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
end_date	Only includes questions answered on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.

extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](#). Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on all questions asked by a member of the House of Commons.

See Also

[all_answered_questions()](#)

[commons_answered_questions()](#)

[commons_oral_questions()](#)

[commons_oral_question_times()](#)

[commons_written_questions()](#)

[lords_written_questions()](#)

Examples

```
## Not run:
x <- mp_questions(c(172, 3967), "all")

y <- mp_questions(mp_id = 172, question_type = "all")

z <- mp_questions(c(172, 3967), "written")

## End(Not run)
```

---

mp_vote_record   *Individual MP voting records*

Description

Accepts an ID number for a member of the House of Commons, and returns a tibble of their votes.

Usage

```
mp_vote_record(
  mp_id = NULL,
  lobby = "all",
  session = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_mp_vote_record(
  mp_id = NULL,
  lobby = "all",
  session = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`mp_id`	The ID number of a member of the House of Commons.
`lobby`	Accepts one of `'all'`, `'aye'` or `'no'`. `'aye'` returns votes where the MP voted `'aye'`, `'no'` returns votes where the MP voted `'no'`, `'all'` returns all available votes by the MP. This parameter is not case sensitive. Defaults to `'all'`.
`session`	The parliamentary session to return votes from, in `'YYYY/YY'` format. Defaults to `NULL`.
`start_date`	Only includes divisions on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
`end_date`	Only includes divisions on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to `NULL`.
`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by [to_any_case](). Defaults to `'snake'`. |
| verbose | If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE. |

Value

A tibble with details on the voting record of the given MP.

Examples

```
## Not run:
x <- mp_vote_record(172, lobby = "all")

x <- mp_vote_record(172, lobby = "aye")

x <- mp_vote_record(172, lobby = "no")

x <- mp_vote_record(172, session = "2016/17")

## End(Not run)
```

---

papers_laid   *Papers laid*

Description

Imports data on papers laid before the House.

Usage

```
papers_laid(
  withdrawn = FALSE,
  house = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_papers_laid(
  withdrawn = FALSE,
  house = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
```

```
    extra_args = NULL,
    tidy = TRUE,
    tidy_style = "snake",
    verbose = TRUE
)
```

Arguments

withdrawn	If TRUE, only returns withdrawn papers. Defaults to FALSE.
house	The house the paper was laid in. Accepts `'commons'` and `'lords'`. If NULL, returns both House of Commons and House of Lords. This parameter is case-insensitive. Defaults to NULL.
start_date	Only includes papers laid before the House on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
end_date	Only includes papers laid before the House on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
verbose	If TRUE, displayes messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details on papers laid before the given House.

Examples

```
## Not run:
x <- papers_laid(withdrawn = FALSE, house = "commons")

x <- papers_laid(withdrawn = TRUE, house = NULL)

## End(Not run)
```

---

publication_logs *House publications*

Description

Imports data on House of Commons and House of Lords publications.

Usage

```
publication_logs(
  ID = NULL,
  house = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_publication_logs(
  ID = NULL,
  house = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

ID	Publication ID. Defaults to NULL. If not NULL, requests a tibble with information on the given publication.
house	The house that produced the particular publication. Accepts 'commons' and 'lords'. If NULL or not 'commons' or 'lords', returns publications from both House of Commons and House of Lords. This parameter is case-insensitive. Defaults to NULL.
start_date	Only includes publications issued on or after this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to '1900-01-01'.
end_date	Only includes publications issued on or before this date. Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to the current system date.

extra_args	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to NULL.
tidy	Logical parameter. If TRUE, fixes 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 any style accepted by [to_any_case](#). Defaults to 'snake'.
verbose	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.

Value

A tibble with details from publications in the House of Commons and House of Lords

Examples

```
## Not run:
# All publications in the house of commons
x <- publication_logs(house = "commons")

# Returns a given publication
y <- publication_logs(683267)

## End(Not run)
```

---

research_briefings   *Parliamentary Research Briefings*

Description

Imports data on Parliamentary Research Briefings. To see a list of possible topics call `research_topics_list()` or `research_subtopics_list()` for both topics and subtopics. To see a list of briefing types, call `research_types_list()`. This function can return results with newlines in the text of the abstract or description of the research briefing, represented as '\\n'.

Usage

```
research_briefings(
  topic = NULL,
  subtopic = NULL,
  type = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
```

```
)

hansard_research_briefings(
  topic = NULL,
  subtopic = NULL,
  type = NULL,
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

`topic`	The topic of the parliamentary briefing. Defaults to `NULL`.
`subtopic`	The subtopic of the parliamentary briefing. Defaults to `NULL`.
`type`	The type of research briefing. Defaults to `NULL`.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the [API documentation](#) or the package vignette for more details. Defaults to `NULL`.
`tidy`	Logical parameter. If `TRUE`, fixes 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 any style accepted by [to_any_case](#). Defaults to `'snake'`.
`verbose`	If `TRUE`, displays messages on the console on the progress of the API request. Defaults to `TRUE`.

Value

A tibble with details on parliamentary research briefings on the given topic.

See Also

[research_subtopics_list()](#)

[research_types_list()](#)

[research_topics_list()](#)

Examples

```
## Not run:
x <- research_briefings("Housing and planning")

# Requests can be made using lists created using `research_topics_list`
# and `research_subtopics_list`

research_topics_list <- research_topics_list()
```

```
x <- research_briefings(topic = research_topics_list[[7]])

research_subtopics_list <- research_subtopics_list()

x <- research_briefings(subtopic = research_subtopics_list[[7]][10])

# Requests for certain briefing types can also be made using lists
# created with `research_types_list`.

research_types_list <- research_types_list()

x <- research_briefings(type = research_types_list[[3]])

## End(Not run)
```

---

research_topics_list   *Lists of research briefing topics, subtopics and types.*

Description

Returns lists of research briefing topics, subtopics and types. These functions do not accept any
arguments.

Usage

```
research_topics_list()

hansard_research_topics_list()

research_subtopics_list()

hansard_research_subtopics_list()

research_types_list()

hansard_research_types_list()
```

Value

A list with the different research topics/subtopics/types available.

Member details functions

research_topics_list A list with the different research topics available

research_subtopics_list A list of Parliamentary Research Briefings topics

research_types_list A list of types of Parliamentary Research Briefings

See Also

research_briefings()

Examples

```
## Not run:
research_topics_list <- research_topics_list()

research_subtopics_list <- research_subtopics_list()

research_types_list <- research_types_list()

## End(Not run)
```

---

sessions_info   *Parliamentary Session data*

Description

Imports data on Parliamentary Sessions. Note that due to the date format used by the API, if days==TRUE and the end_date and start_date parameters are not set to the default values, the function downloads all available data and then subsets the tibble between the two given dates.

Usage

```
sessions_info(
  days = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_sessions_info(
  days = FALSE,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)
```

Arguments

days   If TRUE, returns data for all available days. If FALSE, returns data on each parlia-
                     mentary session. If TRUE and non-default start_date and/or end_date parame-
                     ters are requested, the function must retrieve all days and subset based on the val-
                     ues passed to start_date and end_date. Not applicable to lords_sessions.
                     Defaults to FALSE.

start_date   Only includes sessions starting on or after this date. Accepts character values
                     in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt
                     or anything else that can be coerced to a date with as.Date(). Defaults to
                     '1900-01-01'.

end_date   Only includes sessions ending on or before this date. Accepts character values
                     in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt
                     or anything else that can be coerced to a date with as.Date(). Defaults to the
                     current system date.

extra_args   Additional parameters and queries to pass to API. These queries must be strings
                     and start with "&". See the [API documentation](#) or the package vignette for more
                     details. Defaults to NULL.

tidy   Logical parameter. If TRUE, fixes 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 any style ac-
                     cepted by [to_any_case](#). Defaults to 'snake'.

verbose   If TRUE, displays messages on the console on the progress of the API request.
                     Defaults to TRUE.

Value

A tibble with details on parliamentary sessions.

Examples

```
## Not run:
x <- sessions_info(days = TRUE)

y <- sessions_info(days = FALSE)

## End(Not run)
```

---

tv_programmes   *TV broadcast data*

Description

Imports data on TV broadcasts, clips of individual members and parliamentary TV channels.

Usage

```
tv_programmes(
  legislature = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_tv_programmes(
  legislature = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

tv_clips(
  mp_id = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

hansard_tv_clips(
  mp_id = NULL,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  extra_args = NULL,
  tidy = TRUE,
  tidy_style = "snake",
  verbose = TRUE
)

tv_channels(tidy = TRUE, tidy_style = "snake", verbose = TRUE)

hansard_tv_channels(tidy = TRUE, tidy_style = "snake", verbose = TRUE)
```

Arguments

`legislature`	Accepts one of either `'commons'` or `'lords'`. If NULL, returns all TV programmes for all chambers.
`start_date`	Only includes broadcasts on or after this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to `'1900-01-01'`.
`end_date`	Only includes broadcasts on or before this date. Accepts character values in `'YYYY-MM-DD'` format, and objects of class `Date`, `POSIXt`, `POSIXct`, `POSIXlt` or anything else that can be coerced to a date with `as.Date()`. Defaults to the current system date.
`extra_args`	Additional parameters and queries to pass to API. These queries must be strings and start with "&". See the API documentation or the package vignette for more details. Defaults to NULL.
`tidy`	Logical parameter. If TRUE, fixes 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 any style accepted by to_any_case. Defaults to `'snake'`.
`verbose`	If TRUE, displays messages on the console on the progress of the API request. Defaults to TRUE.
`mp_id`	Accepts the ID of an MP or peer, and returns all clips featuring that MP or peer. If NULL, returns data on all available clips. Defaults to NULL.

Value

A tibble with details on TV broadcasts.

A tibble with details on TV broadcasts featuring the given MP, or all available clips.

A tibble with details on the different broadcasting channels.

TV Programme functions

tv_programmes TV programmse broadcast, per legislature and date

tv_clips Clips of a given MP or Peer

tv_channels Details on the different parliamentary TV channels

Examples

```
## Not run:
x <- tv_programmes("commons",
  start_date = "2016-11-01",
  end_date = "2016-12-01"
)

## End(Not run)
```

```
## Not run:
x <- tv_clips(4591)

## End(Not run)
```

Index

66