Package ‘stortingscrape’
March 23, 2023

Type  Package
Title  Access Data from the Norwegian Parliament API
Version 0.1.3
Description Functions for retrieving general and specific data from the Norwegian Parliament, through the Norwegian Parliament API at <https://data.stortinget.no>.

URL  https://github.com/martigso/stortingscrape,
      https://martigso.github.io/stortingscrape/

BugReports https://github.com/martigso/stortingscrape/issues
Depends R (>= 4.2.0)
Encoding UTF-8
Imports rvest, httr, parallel, stringr, dplyr
Suggests imager, rmarkdown, knitr, pscl
License GPL (>= 3)
VignetteBuilder knitr
RoxygenNote 7.2.1
LazyData true
NeedsCompilation no
Author Martin Søyland [aut, cre]
Maintainer Martin Søyland <martin.soyland@stv.uio.no>
Repository CRAN
Date/Publication 2023-03-23 14:40:02 UTC

R topics documented:
cases ......................................................................................... 3
covid_relief .................................................................................. 3
covid_relief_result ........................................................................ 4
going_all_committees ....................................................................... 5
going_all_parties ............................................................................. 6
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>get_case</td>
<td>6</td>
</tr>
<tr>
<td>get_counties</td>
<td>9</td>
</tr>
<tr>
<td>get_decision_votes</td>
<td>10</td>
</tr>
<tr>
<td>get_hearing_input</td>
<td>11</td>
</tr>
<tr>
<td>get_hearing_program</td>
<td>12</td>
</tr>
<tr>
<td>get_meeting_agenda</td>
<td>13</td>
</tr>
<tr>
<td>get_mp</td>
<td>14</td>
</tr>
<tr>
<td>get_mp_bio</td>
<td>15</td>
</tr>
<tr>
<td>get_mp_pic</td>
<td>18</td>
</tr>
<tr>
<td>get_parlperiods</td>
<td>19</td>
</tr>
<tr>
<td>get_parlperiod_mps</td>
<td>20</td>
</tr>
<tr>
<td>get_parlperiod_presidency</td>
<td>21</td>
</tr>
<tr>
<td>get_parlsessions</td>
<td>22</td>
</tr>
<tr>
<td>get_proceedings</td>
<td>23</td>
</tr>
<tr>
<td>get_proposal_votes</td>
<td>24</td>
</tr>
<tr>
<td>get_publication</td>
<td>25</td>
</tr>
<tr>
<td>get_question</td>
<td>26</td>
</tr>
<tr>
<td>get_question_hour</td>
<td>28</td>
</tr>
<tr>
<td>get_result_vote</td>
<td>30</td>
</tr>
<tr>
<td>get_session_cases</td>
<td>31</td>
</tr>
<tr>
<td>get_session_committees</td>
<td>33</td>
</tr>
<tr>
<td>get_session_decisions</td>
<td>34</td>
</tr>
<tr>
<td>get_session_delegations</td>
<td>35</td>
</tr>
<tr>
<td>get_session_hearings</td>
<td>36</td>
</tr>
<tr>
<td>get_session_meetings</td>
<td>37</td>
</tr>
<tr>
<td>get_session_mp_speech_activity</td>
<td>38</td>
</tr>
<tr>
<td>get_session_parties</td>
<td>40</td>
</tr>
<tr>
<td>get_session_publications</td>
<td>41</td>
</tr>
<tr>
<td>get_session_questions</td>
<td>42</td>
</tr>
<tr>
<td>get_topics</td>
<td>43</td>
</tr>
<tr>
<td>get_vote</td>
<td>44</td>
</tr>
<tr>
<td>get_written_hearing_input</td>
<td>46</td>
</tr>
<tr>
<td>interp0203</td>
<td>47</td>
</tr>
<tr>
<td>mps4549</td>
<td>48</td>
</tr>
<tr>
<td>parl_periods</td>
<td>49</td>
</tr>
<tr>
<td>parl_sessions</td>
<td>49</td>
</tr>
<tr>
<td>read_obt</td>
<td>50</td>
</tr>
<tr>
<td>vote</td>
<td>50</td>
</tr>
<tr>
<td>vote_result</td>
<td>52</td>
</tr>
</tbody>
</table>

**Index**  

53
**Cases**

*Sorting cases in the 2019-2020 session*

### Description

A dataset containing all cases of the 2019-2020 parliamentary session in *Stortinget*

### Usage

cases

### Format

A list with four elements

- **$root** main data on the MP
- **$topics** named list by case id
- **$proposers** named list by case id
- **$spokespersons** named list by case id

Further description: get_session_cases

### Source


---

**Covid_relief**

*Vote id 85196*

### Description

A dataset containing all vote information on case id 85196

### Usage

covid_relief

### Format

A data frame with 22 columns and 71 rows

- **response_date** Date of data retrieval
- **version** Data version from the API
- **case_id** Case id up for vote
- **alternative_vote** Whether vote is an alternative vote
n_for  Number of votes for
n_absent Number of MPs absent
n_against Number of votes against
treatment_order Order of treated votes
agenda_case_number Case number on the agenda of the meeting
free_vote Logical indication of whether the vote is related to the case as a whole
comment Vote comment
meeting_map_number Number on the meeting map
personal_vote Logical indication of whether vote was recorded as roll call or not
president_id Id of president holding president chair at the time of voting
president_party_id Party of the sitting president
adopted Logical indication of whether the proposal voted on was adopted
vote_id Id of vote
vote_method Voting method
vote_result_type Result type (enstemmig_vedtatt = unanimously adopted)
vote_result_type_text See vote_result_type
vote_topic Description of the proposal voted upon
vote_datetime Date and time of vote

Source
https://data.stortinget.no/eksport/voteringer?sakid=85196

covid_relief_result  Vote id 85196 results

Description
A dataset containing vote matrix on vote id 17689

Usage
covid_relief_result

Format
A data frame with 8 columns and 169 rows

response_date Date of data retrieval
version Data version from the API
vote_id Id of vote
mp_id MP id
get_all_committees

- party_id: Party id
- vote: Vote: for, mot (against), ikke_tilstede (absent)
- permanent_sub_for: Id of the MP originally holding the seat, if the substitute is
- sub_for: Id of the MP originally holding the seat

Source

https://data.stortinget.no/eksport/voteringsresultat?voteringid=17689

get_all_committees  Parliamentary committees over all sessions

Description

Imports data on all committee names and ids over all sessions in the data.stortinget.no API.

Usage

get_all_committees()

Value

A data frame of committees, with the variables:

- response_date: Date of data retrieval
- version: Data version from the API
- id: Id of the committee
- name: Name of the committee
- sessionid: Id of session (empty)

See Also

get_session_committees

Examples

```r
## Not run:
coms <- get_all_committees()
head(coms)

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

**All parliamentary parties**

**Description**

A function for retrieving all Norwegian parliamentary parties in the data.stortinget.no API.

**Usage**

```r
get_all_parties()
```

**Value**

A data.frame of all parties, with the variables:

- `response_date`: Date of data retrieval
- `version`: Data version from the API
- `id`: Id of the party
- `name`: Name of the party
- `represented_party`: Whether the party is represented at the time of download
- `sessionid`: Id of session (empty)
- `period_id`: Id of parliamentary period (empty)

**See Also**

- `get_session_parties`

**Examples**

```r
## Not run:
parties <- get_all_parties()
head(parties)
## End(Not run)
```

### get_case

**Retrieve a parliamentary case**

**Description**

A function for retrieving single parliamentary case by id.
**Usage**

```r
get_case(caseid = NA, good_manners = 0)
```

**Arguments**

- **caseid** Character string indicating the id of the case to request
- **good_manners** Integer. Seconds delay between calls when making multiple calls to the same function

**Value**

A list with seven data frame elements:

1. **$root** (main data on the case)

   - `response_date` Date of data retrieval
   - `version` Data version from the API
   - `document_group` Case document group type
   - `finalized` Whether the case finalized
   - `reference` Relevant publication references
   - `id` Case id
   - `req_text` Recommendation (proposal) text
   - `committee_id` Id of committee handling the case
   - `title_short` Case short title
   - `decision_short` Case decision_short
   - `parenthesis_text` Case parenthesis text
   - `case_number` Case number
   - `session_id` Session id
   - `proceedings_id` Type of proceeding id
   - `proceedings_name` Type of proceeding name
   - `status` Status for case
   - `title` Case title (long)
   - `type` Case type
   - `decision_text` Decision text

2. **$topic** (the topics related to the case)

   - `is_main_topic` Is this (row) the main topic?
   - `main_topic_id` Id for main topic
   - `id` Topic id
   - `navn` Topic name

3. **$publication_references** (references for publications on the case)
get_case

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>export_id</td>
<td>Id for export of publication (used in <code>?get_publication</code>)</td>
</tr>
<tr>
<td>link_text</td>
<td>Publication title</td>
</tr>
<tr>
<td>link_url</td>
<td>URL to publication</td>
</tr>
<tr>
<td>type</td>
<td>Publication type</td>
</tr>
<tr>
<td>subtype</td>
<td>Publication subtype (chamber)</td>
</tr>
</tbody>
</table>

4. **$proposers** (MPs behind case proposal, when relevant)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp_id</td>
<td>MPs id</td>
</tr>
<tr>
<td>party_id</td>
<td>Party id of MPs</td>
</tr>
<tr>
<td>sub_mp</td>
<td>Whether MPs are substitutes</td>
</tr>
</tbody>
</table>

5. **$proceeding_steps** (case proceeding steps)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>step_name</td>
<td>Name of steps</td>
</tr>
<tr>
<td>step_number</td>
<td>Step order for case</td>
</tr>
<tr>
<td>outdated</td>
<td>Whether the step type is outdated</td>
</tr>
</tbody>
</table>

6. **$spokespersons** (all MPs that are spokespersons for the case)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp_id</td>
<td>MPs id</td>
</tr>
<tr>
<td>party_id</td>
<td>Party id of MPs</td>
</tr>
<tr>
<td>sub_mp</td>
<td>Whether MPs are substitutes</td>
</tr>
</tbody>
</table>

7. **$keywords** (all keywords associated with the case)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyword</td>
<td>Keywords for the case</td>
</tr>
</tbody>
</table>

See Also

- `get_session_cases`

Examples

```r
## Not run:

# Get one case
case <- get_case("30233")
```

get_counties

# Get multiple cases
cases <- lapply(c("30233", "30362", "30234", "30236"), get_case, good_manners = 2)
cases_root <- lapply(cases, function(x) x$root)
cases_root <- do.call(rbind, cases_root)
cases_root

cases_keywords <- lapply(1:nrow(cases_root), function(x){
  tmp <- cases[[x]]$keywords
  tmp$case_id <- cases_root$id[x]
  return(tmp)
})
cases_keywords <- do.call(rbind, cases_keywords)
cases_keywords

## End(Not run)

---

get_counties

Get list of MPs in a given parliamentary period

Description

A function for retrieving Norwegian MPs for a given parliamentary period from the parliament API.

Usage

get_counties(historical = FALSE)

Arguments

historical Logical. Whether or not to include historical counties.

Value

A data frame with the following variables:

- **response_date**: Date of data retrieval
- **version**: Data version from the API
- **historical_county**: Whether the county is historical (no longer exists)
- **id**: Id of the county
- **name**: Name of the county

Examples

## Not run:
# Request one MP by id
get_counts()  

# With historical counties  
get_counts(historical = TRUE)  

## End(Not run)  

---  

get_decision_votes  

Retrieve vote decision for a specified vote  

Description  
A function for retrieving vote decisions from a specific vote. Vote data are only available from the 2011-2012 session  

Usage  
get_decision_votes(voteid = NA, good_manners = 0)  

Arguments  
voteid  Character string indicating the id of the vote to request all votes from  
good_manners  Integer. Seconds delay between calls when making multiple calls to the same function  

Value  
A data.frame with the following variables:  

- response_date  Date of data retrieval  
- version  Data version from the API  
- vote_id  Id of the vote  
- decision_code  General code for decision  
- decision_comment  Comments for the decision  
- decision_number  Decision number  
- decision_reference  Reference for the decision  
- decision_text  Full text of the decision  

See Also  
get_session_decisions get_proposal_votes get_vote get_result_vote  

Examples  

## Not run:
get_hearing_input

 decision <- get_decision_votes(123)
 decision

 ## End(Not run)

---

### Description

A function for retrieving the hearing input for a specified hearing.

### Usage

```r
get_hearing_input(hearingid = NA, good_manners = 0)
```

### Arguments

- **hearingid** Character string indicating the id of the hearing to retrieve.
- **good_manners** Integer. Seconds delay between calls when making multiple calls to the same function

### Value

A data.frame the following variables:

- **response_date** Date of data retrieval
- **version** Data version from the API
- **hearing_id** Id of the hearing
- **hearing_type** Type of hearing
- **committee_id** Id of committee responsible for the hearing
- **hearing_input_date** Date of receiving input
- **hearing_input_id** Hearing input id
- **hearing_input_organization** Organization giving input
- **hearing_input_text** Full text of the hearing input
- **hearing_input_title** Title of the hearing input

### See Also

- `get_session_hearings`
- `get_hearing_program`
- `get_written_hearing_input`

### Examples

```r
## Not run:
```
get_hearing_program

get_hearing_program(hearingid = 10004166)

## End(Not run)

get_hearing_program

Retrieve the hearing program for a specified hearing

Description

A function for retrieving the hearing program for a specified hearing. The earlier periods (around 2005 and back) are less rich with data on some variables

Usage

get_hearing_program(hearingid = NA, good_manners = 0)

Arguments

hearingid  Character string indicating the id of the hearing to retrieve.

good_manners  Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

response_date  Date of data retrieval
version  Data version from the API
hearing_id  Id of the hearing
hearing_type  Type of hearing
committee_id  Id of committee responsible for the hearing
hearing_program_date  Date hearing program
hearing_program_footnote  Footnote for hearing program
order_number  Order number for the hearing program element
text  Description of the hearing participant
time_indication  Time stamp for participant hearing input
date  Date of participant input in hearing

See Also

get_session_hearings get_hearing_input get_written_hearing_input

Examples

## Not run:
get_meeting_agenda

Retreive agenda for a specified meeting

Description

A function for retrieving the agenda for a specific meeting.

Usage

get_meeting_agenda(meetingid = NA, good_manners = 0)

Arguments

- meetingid: Character string indicating the id of the meeting to request all votes from
- good_manners: Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

- response_date: Date of data retrieval
- version: Data version from the API
- agenda_number: The agenda number for the session
- meeting_date: Date of the meeting
- meeting_id: Meeting id
- meeting_place: Where the meeting was held
- agenda_case_reference: Reference for the case on the agenda
- agenda_case_number: Case number
- agenda_case_text: Case description
- agenda_case_type: Case type
- footnote: Footnote for the case
- proposition_id: If relevant, belonging proposition id
- committee_id: If relevant, id of the responsible committee
- loose_proposals: Whether there are loose proposals to the case
- case_id: Id of the case
- question_hour_type: If relevant, type of question hour
- question_id: If relevant, question id
get_mp

See Also

get_session_meetings get_case get_question get_question_hour

Examples

## Not run:

```
meetings0910 <- get_session_meetings("2009-2010")
meeting_agenda <- get_meeting_agenda(meetings0910$meeting_id[161])
meeting_agenda
```

## End(Not run)

---

**get_mp**

Extract information on specific MPs

**Description**

A function for retrieving information on Norwegian MPs from the parliament API

**Usage**

```
get_mp(mpid = NA, good_manners = 0)
```

**Arguments**

- `mpid`: Character string indicating the id of the MP to retrieve.
- `good_manners`: Integer. Seconds delay between calls when making multiple calls to the same function

**Value**

A data.frame with the following variables:

- `response_date`: Date of data retrieval
- `version`: Data version from the API
- `death`: MP date of death, if applicable
- `last_name`: MP last name
- `birth`: MP date of birth
- `first_name`: MP first name
- `id`: MP id
- `gender`: MP gender
get_mp_bio

See Also

get_mp_bio get_parlperiod_mps get_mp_pic get_session_mp_speech_activity

Examples

```r
## Not run:
# Request one MP by id
get_mp("AAMH")

# Request several MPs by id
ids <- c("AAMH", "AMSK", "MAAA")

mps <- lapply(ids, get_mp, good_manners = 2)

mps <- do.call(rbind, mps)
## End(Not run)
```

get_mp_bio  
Extract biography of specific MPs

Description

A function for retrieving biography of Norwegian MPs from the parliament API

Usage

get_mp_bio(mpid = NA, good_manners = 0)

Arguments

- **mpid**: Character string indicating the id of the MP to retrieve.
- **good_manners**: Integer. Seconds delay between calls when making multiple calls to the same function

Value

A list with ten data frames:

1. $root (main data on the MP)

<table>
<thead>
<tr>
<th>response_date</th>
<th>version</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of data retrieval</td>
<td>Data version from the API</td>
<td>Id of the MP</td>
</tr>
</tbody>
</table>
2. **Literature** (all literature the MP contributed to)

<table>
<thead>
<tr>
<th>Year</th>
<th>Year of entry publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of the publication</td>
</tr>
<tr>
<td>Last Name</td>
<td>MP's last name</td>
</tr>
<tr>
<td>More Years</td>
<td>Not described in the API</td>
</tr>
<tr>
<td>Publisher</td>
<td>Publisher</td>
</tr>
<tr>
<td>First Name</td>
<td>First name of the MP</td>
</tr>
<tr>
<td>Place</td>
<td>Place of publication</td>
</tr>
<tr>
<td>Title</td>
<td>Title of the publication</td>
</tr>
<tr>
<td>Type</td>
<td>MP’s role in publication (author etc)</td>
</tr>
</tbody>
</table>

3. **Leave_of_absence** (times the MP was on leave)

| From Date | Start date of leave |
| Reason | Reason for leave |
| To Date | End of leave |
| Type | Type of leave |
| Sub Last Name | Substitute MP last name (id not recorded) |
| Sub First Name | Substitute MP first name |

4. **Personalia** (the MP’s personalia)

| Seniority Aar | Number of years in parliament |
| Seniority Dager | Number of extra days (addition to years) |
| County of Birth | Birth county of the MP |
| Municipality of Birth | Birth municipality of the MP |
| Eulogy Date | Eulogy date of the MP, when applicable |

5. **Father** (the MP’s father personalia)

| Death Year | Father's year of death |
| Last Name | Father’s last name |
| Birth Year | Father’s year of birth |
| First Name | Father’s first name |
| Profession | Father’s profession |

6. **Mother** (the MP’s mother personalia)
7. **$parl_periods** (parliamentary periods the MP has held a seat)

- **from_date**: Date MP held seat from
- **county**: County the MP represented
- **party_id**: Party id for the MP’s party
- **rep_number**: Representative number (within the whole parliament)
- **parl_period_id**: Id of the parliamentary period
- **to_date**: Date MP held a seat to
- **type**: Type of representation

8. **$parl_positions** (parliamentary positions held by the MP)

- **from_year**: Year MP held position from
- **from_date**: Date MP held position from
- **committee_id**: Id of the position (in committee, cabinet, delegation, etc)
- **committee_name**: Position name
- **committee_type**: Position type
- **sorting**: *Not described in the API*
- **parl_period_id**: Parliamentary period the position was held (cabinet data missing)
- **to_year**: Year MP held position to
- **to_date**: Date MP held position to

9. **$vocation** (vocation and education of the MP outside of parliament)

- **several_periods_text**: Text description if the vocation was held for several periods
- **from_year**: Year MP held vocation from
- **from_year_unknown**: Logical indication for whether the start year is unknown
- **note**: Note for vocation
- **name**: Name of vocation
- **to_year**: Year MP held vocation to
- **to_year_unknown**: Logical indication for whether the end year is unknown
- **type**: Vocation type (10 = education, 20 = work)

10. **$other_positions** (other positions held outside parliament)
get_mp_pic

Retrieve picture of specific MPs

Description

A function for retrieving Norwegian MP pictures by id.

Usage

get_mp_pic(mpid = NA, size = "middels",
            destfile = NA, show_plot = FALSE,
            good_manners = 0)
get_parlperiods

Get Parliamentary Periods

Description

A function for retrieving dates of the parliamentary periods after WWII

Usage

get_parlperiods()

Value

A data.frame with the following variables:

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpid</td>
<td>Character string indicating the id of the MP to retrieve.</td>
</tr>
<tr>
<td>size</td>
<td>Character string size of the picture. Accepts values &quot;lite&quot; (small), &quot;middels&quot; (medium – default), and &quot;stort&quot; (big).</td>
</tr>
<tr>
<td>destfile</td>
<td>Character string specifying where to save the picture</td>
</tr>
<tr>
<td>show_plot</td>
<td>Logical. FALSE (default) if no plot should be produced and TRUE if plot should be produced. Requires the &quot;imager&quot; package.</td>
</tr>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same function</td>
</tr>
</tbody>
</table>

Value

Picture of the requested MP in the preferred size.

See Also

get_mp, get_parlperiod_mps, get_mp_bio

Examples

```r
## Not run:
# Request one MP by id
get_mp_pic(mpid = "AAMH", destfile = "~/Pictures/AAMH.jpeg", show_plot = TRUE, size = "stort")

# With good manners for multiple calls
lapply(c("AAMH", "CIH", "TKF"), function(x){
  get_mp_pic(mpid = x, destfile = paste0("~/Pictures/", x),
            show_plot = TRUE, size = "stort", good_manners = 2)
})
## End(Not run)
```
get_parlperiod_mps

response_date  Date of data retrieval
version        Data version from the API
from           Date session started
id             Id of for session (used for other functions)
to             Date session ended
years          From year to year in full format

See Also

get_parlsessions

Examples

## Not run:
parlper <- get_parlperiods()
parlper

## End(Not run)

get_parlperiod_mps  Get list of MPs in a given parliamentary period

Description

A function for retrieving Norwegian MPs for a given parliamentary period from the parliament API

Usage

get_parlperiod_mps(periodid = NA, substitute = FALSE, good_manners = 0)

Arguments

periodid          Character string indicating the id of the parliamentary period to retrieve.
substitute        Logical. Whether or not to include substitute MPs.
good_manners      Integer. Seconds delay between calls when making multiple calls to the same function
get_parlperiod_presidency

Value

A data.frame with the following variables:

- response_date: Date of data retrieval
- version: Data version from the API
- death: Date of death
- lastname: MP last name
- birth: Date of birth
- firstname: MP first name
- mp_id: MP id
- gender: MP gender
- county_id: Id of county MP represented
- party_id: Id of party MP represented
- substitute_mp: Logical for whether MP is a substitute
- period_id: Id of period represented in

See Also

get_mp_bio get_mp get_mp_pic get_session_mp_speech_activity

Examples

```r
## Not run:

# Request one MP by id
get_parlperiod_mps("2005-2009")

# Request MPs from several periods by id
mps <- lapply(ids, get_parlperiod_mps, good_manners = 2)
mps <- do.call(rbind, mps)

## End(Not run)
```

get_parlperiod_presidency

Get list of presidency in a given parliamentary period

Description

A function for retrieving the presidency for a given parliamentary period from the parliament API.
get_parlperiod_presidency(periodid = NA, good_manners = 0)

Arguments

periodid  Character string indicating the id of the parliamentary period to retrieve.
good_manners  Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

- response_date  Date of data retrieval
- version  Data version from the API
- last_name  Last name of presidency member
- first_name  First name of presidency member
- from_date  Presidency member from date
- party_id  Party affiliation of presidency member
- person_id  Id of the presidency member
- to_date  Presidency member to date
- position  Presidency position

See Also

get_mp get_mp_bio

Examples

## Not run:

# Request one MP by id
get_parlperiod_presidency("2005-2009")

## End(Not run)
get_proceedings

Usage

get_parlsessions()

Value

A data.frame with the following variables:

- **response_date**: Date of data retrieval
- **version**: Data version from the API
- **from**: Date session started
- **id**: Id of for session (used for other functions)
- **to**: Date session ended
- **years**: From year to year in full format

See Also

get_parlperiods

Examples

```r
## Not run:
parlses <- get_parlsessions()
parlses

## End(Not run)
```

get_proceedings

All parliamentary proceedings

Description

A function for retrieving all proceedings in Stortinget, both current and historical.

Usage

get_proceedings()

Value

A list with three dataframes:
1. $\textit{root}$ (only download meta data)

   - **response_date** Date of data retrieval
   - **version** Data version from the API

2. $\textit{proceedings}$ (description of main proceeding categories)

   - **id** Id of proceeding
   - **name** Name of proceeding

3. $\textit{proceedings\_steps}$ (description of proceeding steps within each main category)

   - **id** Id of proceeding step
   - **name** Name of proceeding step
   - **step\_number** Order of proceeding steps
   - **outdated** Whether the step is outdated
   - **main\_id** Id for proceeding type the step belongs to

**Examples**

```
## Not run:
get_proceedings()

## End(Not run)
```

---

**get\_proposal\_votes**

*Retreive all votes for a specified vote proposal*

**Description**

A function for retrieving all votes from a specific vote proposal. Vote data are only available from the 2011-2012 session

**Usage**

```
get\_proposal\_votes(voteid = NA, good\_manners = 0)
```
get_publication

Arguments

<table>
<thead>
<tr>
<th>voteid</th>
<th>Character string indicating the id of the vote to request all votes from</th>
</tr>
</thead>
<tbody>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same function</td>
</tr>
</tbody>
</table>

Value

A list with two elements:

1. $proposal_vote$ (main data on the vote proposal)
   - response_date  Date of data retrieval
   - version        Data version from the API
   - vote_id         Id of the vote

2. $proposal_by_parties$proposal_id (what parties (id) stood behind proposal(s))

See Also

get_vote get_decision_votes get_result_vote

Examples

```r
## Not run:

prop <- get_proposal_votes(7523)
prop

for(i in 1:length(prop$proposal_by_parties)){
  prop$proposal_vote$parties[i] <- paste0(prop$proposal_by_parties[i], collapse = " ", " )
}

## End(Not run)
```

get_publication

Retrieves a specific publication

Description

A function for retrieving a specific publication. Because these are formatted very differently in the API, the returning object is in a raw html_document format, best manipulated with html packages such as rvest.
get_publication

Usage

get_publication(publicationid = NA, good_manners = 0)

Arguments

publicationid Character string indicating the id of the publication to request all votes from

good_manners Integer. Seconds delay between calls when making multiple calls to the same function

Value

A raw html_document

See Also

get_question get_question_hour get_session_publications

Examples

## Not run:
pub <- get_publication("refs-201819-03-06")
(pub %>% html_elements("replikk")[1] %>% html_text())

## End(Not run)

get_publication

get_question

Parliamentary question

Description

A function for retrieving single parliamentary questions by id. For retrieving the whole debate over a question, the get_publication function has to be used

Usage

get_question(questionid = NA, good_manners = 0)

Arguments

questionid Character string indicating the id of the session to request interpellations from

good_manners Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:
response_date  Date of data retrieval
version  Data version from the API
justification  Justification for question
answ_by_id  Id for answering minister
answ_by_minister_id  Id for department of answering minister
answ_by_minister_title  Title for department of answering minister
answ_date  Date question was asked
answ_on_belhalf_of  Id of minister answered on behalf of, when relevant
answ_on_belhalf_of_minister_id  Id of department answered on behalf of, when relevant
answ_on_belhalf_of_minister_title  Title of department answered on behalf of, when relevant
agenda_number  Agenda number in meeting
moved_to  Date moved to
id  Question id
correct_person_id  Not documented in API
correct_person_minister_id  Not documented in API
correct_person_minister_title  Not documented in API
sendt_date  Date question was sent
session_id  Session id
question_text  Full question text
question_from_id  Id of MP asking the question
question_number  Question number
question_to_id  Id of minister the question was asked to
question_to_minister_id  Department id of minister the question was asked to
question_to_minister_title  Department title of minister the question was asked to
answer_text  Answer text (often empty)
title  Question title
type  Question type

See Also

get_question_hour get_publication

Examples

## Not run:
# An example of a possible workflow

## Retreive sessions
sessions <- get_parlsessions()

## Retreive all interpellations for a particular session
qsesh <- get_session_questions(sessions$id[9], q_type = "interpellasjoner")

## Retreve detailed information on all interpellations in that session
library(pbmcapply) # for progress bar. never use paralell on scraping
int1213 <- pbmclapply(qsesh$id, function(x){
  get_question(x, good_manners = 2)"
## get_question_hour

Get question hour details for a specified meeting

### Description
A function for retrieving detailed overview of the question hour for a specific meeting

### Usage
```r
get_question_hour(meetingid = NA, good_manners = 0)
```

### Arguments
- **meetingid**  
  Character string indicating the id of the meeting to request all votes from
- **good_manners**  
  Integer. Seconds delay between calls when making multiple calls to the same function

### Value
A list with ten data frames:

1. **$root** (download meta data)
   - **response_date**  
     Date of data retrieval
   - **version**  
     Data version from the API
   - **meetingid**  
     The called meeting id

2. **$question_hour_ministers** (id of ministers in parliament during question hour/time)
   - **id**  
     Id of ministers attending

3. **$question_time**
   - **question_justification**  
     Justification for question
   - **answer_by_id**  
     Id for answering minister
   - **answer_by_minister_id**  
     Id for department of answering minister
get_question_hour

answer_by_minister_title
answer_date
answer_on_behalf_of_id
answer_on_behalf_of_minister_id
answer_on_behalf_of_minister_title
agenda_case_number
date
moved_to
asked_by_other_id
question_id
correct_person
correct_person_minister_id
correct_person_minister_title
sent_date
session_id
question_text
question_from_id
question_number
question_to_id
question_to_minister_id
question_to_minister_title
status
answer
title
type

4. $publication_reference

export_id
link_text
link_url
type
sub_type

See Also

get_question get_session_questions get_publication

Examples

## Not run:
get_question_hour(10232)

## End(Not run)
get_result_vote

Retrieve vote results on MP level for a specified vote

Description

A function for retrieving vote results from a specific vote on MP level. Vote data are only available from the 2011-2012 session. Needs some preprocessing for use with rollcall packages, such as ideal.

Usage

get_result_vote(voteid = NA, good_manners = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>voteid</td>
<td>Character string indicating the id of the vote to request all votes from</td>
</tr>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same</td>
</tr>
<tr>
<td></td>
<td>function</td>
</tr>
</tbody>
</table>

Value

A data.frame with the following variables:

- response_date: Date of data retrieval
- version: Data version from the API
- vote_id: Id of vote
- mp_id: MP id
- party_id: Party id
- vote: Vote: for, mot (against), ikke_tilstede (absent)
- permanent_sub_for: Id of the MP originally holding the seat, if the substitute is permanent
- sub_for: Id of the MP originally holding the seat

See Also

get_decision_votes get_proposal_votes get_vote get_mp_bio

Examples

```r
## Not run:

v <- get_result_vote(12345)

table(v$vote)
```
p <- get_proposal_votes(12345)

stringr::str_replace_all(p$proposal_vote$proposal_text,
                       "\<(.*)\>|\r\n", "") ## End(Not run)

---

get_session_cases  
Cases in specified session

Description

A function for retrieving all cases treated in a specified parliamentary session.

Usage

get_session_cases(sessionid = NA, good_manners = 0, cores = 1)

Arguments

- **sessionid**: Character string indicating the id of the parliamentary session to retrieve.
- **good_manners**: Integer. Seconds delay between calls when making multiple calls to the same function.
- **cores**: Integer. Number of cores (1 by default) to use in structuring the data. More than 1 will not work on windows.

Value

A data.frame with the following variables:

1. **$root** (main data on the MP):
   - **response_date**: Date of data retrieval
   - **version**: Data version from the API
   - **treated_session_id**: Session the case was treated in
   - **document_group**: Document group the case belongs to
   - **reference**: Document reference
   - **id**: Case id
   - **com_req_id**: Committee recommendation id
   - **com_req_code**: Committee recommendation code
   - **title_short**: Short title of case
   - **caseFiled_id**: Id of filed case
   - **last_update_date**: Date of last update on case
   - **status**: Status of the case
   - **title**: Full title of the case
get_session_cases

<table>
<thead>
<tr>
<th>type</th>
<th>Type of case</th>
</tr>
</thead>
<tbody>
<tr>
<td>session_id</td>
<td>Session id of the case</td>
</tr>
<tr>
<td>committee_id</td>
<td>Responsible committee id</td>
</tr>
</tbody>
</table>

2. $topics (named list by case id)

<table>
<thead>
<tr>
<th>is_main_topic</th>
<th>Logical indication whether the topic is the main topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>main_topic_id</td>
<td>Id of the main topic for the case</td>
</tr>
<tr>
<td>id</td>
<td>Topic id</td>
</tr>
<tr>
<td>name</td>
<td>Topic name</td>
</tr>
</tbody>
</table>

3. $proposers (named list by case id)

<table>
<thead>
<tr>
<th>rep_id</th>
<th>Proposing MP id</th>
</tr>
</thead>
<tbody>
<tr>
<td>county_id</td>
<td>County id of proposing MP</td>
</tr>
<tr>
<td>party_id</td>
<td>Party id of proposing MP</td>
</tr>
<tr>
<td>rep_sub</td>
<td>Logical indicator for whether MP is a substitute</td>
</tr>
</tbody>
</table>

4. $spokespersons (named list by case id)

<table>
<thead>
<tr>
<th>rep_id</th>
<th>Spokesperson MP id</th>
</tr>
</thead>
<tbody>
<tr>
<td>county_id</td>
<td>County id of spokesperson MP</td>
</tr>
<tr>
<td>party_id</td>
<td>Party id of spokesperson MP</td>
</tr>
<tr>
<td>rep_sub</td>
<td>Logical indicator for whether MP is a substitute</td>
</tr>
</tbody>
</table>

See Also

get_case get_vote

Examples

```r
## Not run:
s0506 <- get_session_cases("2005-2006")
head(s0506)

## End(Not run)
```
Description

A function for retrieving Norwegian parliamentary committees for a specified parliamentary session

Usage

get_session_committees(sessionid = NA, good_manners = 0)

Arguments

sessionid Character string indicating the id of the parliamentary session to retrieve.
good_manners Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

- response_date Date of data retrieval
- version Data version from the API
- id Committee id
- name Committee name
- session_id Session id

See Also

get_all_committees get_mp_bio get_mp

Examples

```r
## Not run:
coms <- get_session_committees("2001-2002")
coms

## End(Not run)
```
get_session_decisions  Retrieve all decisions for a specified session

Description

A function for retrieving all decisions from a specific parliamentary session.

Usage

get_session_decisions(sessionid = NA, good_manners = 0)

Arguments

- **sessionid**: Character string indicating the id of the session to request all votes from
- **good_manners**: Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

- **response_date**: Date of data retrieval
- **version**: Data version from the API
- **session_id**: Session id
- **decision_id**: Decision id
- **case_id**: Case id
- **case_link_url**: URL for case to front end web page
- **decision_date**: Decision date
- **decision_link_url**: URL for decision to front end web page
- **decision_number**: Decision number within session
- **decision_text**: Decision text
- **decision_title**: Decision title
- **decision_type_id**: Decision type id
- **decision_type_name**: Decision type name

See Also

- get_decision_votes

Examples

```r
## Not run:

desci <- get_session_decisions("2004-2005")
head(desci)
```
get_session_delegations

Parliamentary delegations in specified session

Description
A function for retrieving delegations for a specified parliamentary session.

Usage
get_session_delegations(sessionid = NA, good_manners = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Character string indicating the id of the parliamentary session to retrieve.</td>
</tr>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same function</td>
</tr>
</tbody>
</table>

Value
A data.frame with the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response_date</td>
<td>Date of data retrieval</td>
</tr>
<tr>
<td>version</td>
<td>Data version from the API</td>
</tr>
<tr>
<td>id</td>
<td>Delegation id</td>
</tr>
<tr>
<td>name</td>
<td>Name of delegation</td>
</tr>
<tr>
<td>session_id</td>
<td>Session id</td>
</tr>
</tbody>
</table>

See Also
get_session_committees get_all_committees

Examples

## Not run:
degleg <- get_session_delegations("2015-2016")
degleg

## End(Not run)
get_session_hearings  Retrieve hearings in specified session

Description
A function for retrieving all hearings in a specified parliamentary session.

Usage
get_session_hearings(sessionid = NA, good_manners = 0, cores = 1)

Arguments
sessionid  Character string indicating the id of the parliamentary session to retrieve.
good_manners  Integer. Seconds delay between calls when making multiple calls to the same function
cores  Integer...

Value
A list with four elements:
1. $root (hearing meta data)
   - response_date  Date of data retrieval
   - version  Data version from the API
   - session_id  Session id

2. $hearing (main data on the hearing)
   - deadline_date  Deadline date for hearing
   - status  Data version from the API
   - hearing_id  Hearing id
   - input_deadline  Deadline date for input
   - written  Logical indication of whether the input was written
   - application_deadline  Deadline date for application to hearing
   - start_date  Start date for hearing
   - status_pub  Publication status for hearing
   - status_info_text  Status information text
   - type  Type of hearing
   - committee_id  Committee id for committee responsible for hearing
get_session_meetings

3. **$hearing_case_info** (named list by hearing id with information on the case(s) belonging to the hearing)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_reference</td>
<td>Text reference for case</td>
</tr>
<tr>
<td>case_id</td>
<td>Case id</td>
</tr>
<tr>
<td>case_short_title</td>
<td>Short title for case</td>
</tr>
<tr>
<td>case_publication</td>
<td>URL for front end web-page publication</td>
</tr>
<tr>
<td>case_title</td>
<td>Full title for case</td>
</tr>
</tbody>
</table>

4. **$hearing_date** (named list by hearing id with date(s) the hearing was held)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response_date</td>
<td>Date of data retrieval</td>
</tr>
<tr>
<td>version</td>
<td>Data version from the API</td>
</tr>
<tr>
<td>session_id</td>
<td>Session id</td>
</tr>
</tbody>
</table>

See Also

get_hearing_input, get_hearing_program, get_written_hearing_input

Examples

```r
## Not run:

hear <- get_session_hearings("2010-2011")
head(hear$hearing)

## End(Not run)
```

---

*get_session_meetings* Retrieve all meetings for a specified parliamentary session

Description

A function for retrieving meetings from a specific parliamentary session

Usage

```r
get_session_meetings(sessionid = NA, good_manners = 0)
```
get_session_mp_speech_activity

Arguments

- **sessionid**: Character string indicating the id of the session to request all votes from.
- **good_manners**: Integer. Seconds delay between calls when making multiple calls to the same function.

Value

A data.frame with the following variables:

- **response_date**: Date of data retrieval.
- **version**: Data version from the API.
- **session_id**: Session id.
- **agenda_number**: Agenda number within the session.
- **footnote**: Footnotes for the meeting.
- **meeting_id**: Meeting id.
- **no_meeting_text**: Description of why there was no meeting, if relevant.
- **evening_meeting**: Whether the meeting was an evening meeting or not.
- **note**: Note for the meeting.
- **meeting_date**: Date the meeting took place.
- **meeting_order**: Indicator for meeting order.
- **meeting_place**: Where the meeting took place.
- **transcript_id**: Id for transcript (usually empty).
- **additional_agenda**: Logical indicator for whether there was additional agenda to the meeting.

See Also

- **get_meeting_agenda**
- **get_question_hour**

Examples

```r
## Not run:

meet <- get_session_meetings("2013-2014")
head(meet)

## End(Not run)
```

get_session_mp_speech_activity

*Retrieve all speech activity from one MP for a given session*
Description

A function for retrieving all speech activity from an MP during a specific parliamentary session. Only available from the 2011-2012 session and onwards.

Usage

get_session_mp_speech_activity(sessionid = NA, mp_id = NA, good_manners = 0)

Arguments

- **sessionid**: Character string indicating the session to retrieve speeches from.
- **mp_id**: Character string for the MP to retrieve all speeches of in a given session.
- **good_manners**: Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

- **response_date**: Date of data retrieval
- **version**: Data version from the API
- **session_id**: Session id
- **agenda_case_number**: Number indicating the agenda number for the case
- **meeting_id**: Meeting id
- **speech_start_time**: Start time of speech
- **speech_type**: Type of speech
- **speech_length_secs**: Length of speech in seconds

See Also

get_mp get_mp_bio get_publication

Examples

```r
## Not run:
activ <- get_session_mp_speech_activity("2012-2013", "ALYS")
head(activ)
## End(Not run)
```
get_session_parties  Parliamentary parties in specified session

Description
A function for retrieving Norwegian parliamentary parties for a specified parliamentary session

Usage
get_session_parties(sessionid = NA, good_manners = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Character string indicating the id of the parliamentary session to retrieve.</td>
</tr>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same function</td>
</tr>
</tbody>
</table>

Value
A data.frame with the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response_date</td>
<td>Date of data retrieval</td>
</tr>
<tr>
<td>version</td>
<td>Data version from the API</td>
</tr>
<tr>
<td>id</td>
<td>Party id</td>
</tr>
<tr>
<td>name</td>
<td>Party name</td>
</tr>
<tr>
<td>represented_party</td>
<td>Logical indication of whether party was represented</td>
</tr>
<tr>
<td>session_id</td>
<td>Session id</td>
</tr>
<tr>
<td>period_id</td>
<td>Parliamentary period id</td>
</tr>
</tbody>
</table>

See Also

get_all_parties

Examples

```r
## Not run:

parties <- get_session_parties("2003-2004")
parties

## End(Not run)
```
get_session_publications

Retrieve publications of a type in a parliamentary session

Description
A function for retrieving one of several publication types within a parliamentary session.

Usage
get_session_publications(sessionid = NA, type = "referat", good_manners = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Character string indicating the id of the hearing to retrieve.</td>
</tr>
<tr>
<td>type</td>
<td>Character specifying type of publication to download. Available types are &quot;referat&quot; (minutes), &quot;innstilling&quot; (proposition), &quot;innbetegning&quot; (report), &quot;lovvedtak&quot; (law decision), &quot;lovanmerkning&quot; (law note), &quot;dok8&quot; (MP proposal) &quot;dok12&quot; (Constitutional proposal), and &quot;dokumentserie&quot; (document series). Defaults to &quot;referat&quot;.</td>
</tr>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same function</td>
</tr>
</tbody>
</table>

Value
A data.frame with the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response_date</td>
<td>Date of data retrieval</td>
</tr>
<tr>
<td>version</td>
<td>Data version from the API</td>
</tr>
<tr>
<td>session_id</td>
<td>Session id</td>
</tr>
<tr>
<td>publication_date</td>
<td>Date of publication</td>
</tr>
<tr>
<td>publication_id</td>
<td>Id of publication</td>
</tr>
<tr>
<td>publication_format</td>
<td>Publication format (XML)</td>
</tr>
<tr>
<td>publication_available_date</td>
<td>When the publication was made available</td>
</tr>
<tr>
<td>publication_title</td>
<td>Publication title</td>
</tr>
<tr>
<td>publication_type</td>
<td>Publication type</td>
</tr>
</tbody>
</table>

See Also
get_publication

Examples

```r
## Not run:
```
get_session_questions

Parliamentary questions in a session

Description
A function for retrieving all questions within a parliamentary session.

Usage
get_session_questions(sessionid = NA, q_type = NA, status = NA, good_manners = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionid</td>
<td>Character string indicating the id of the session to request interpellations from</td>
</tr>
<tr>
<td>q_type</td>
<td>Character string indicating type of question to retrieve. Options are &quot;interpellassjoner&quot; (interpellations), &quot;sporretimesporsmal&quot; (oral questions), or &quot;skriftligesporsmal&quot; (written questions).</td>
</tr>
<tr>
<td>status</td>
<td>Character string question status extraction. Possible values are NA (extract all questions), &quot;til_behandling&quot; (pending questions), &quot;trukket&quot; (withdrawn questions), &quot;bortfalt&quot; (lapsed questions), or &quot;alle&quot; (all questions)</td>
</tr>
<tr>
<td>good_manners</td>
<td>Integer. Seconds delay between calls when making multiple calls to the same function</td>
</tr>
</tbody>
</table>

Value
A data.frame with the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response_date</td>
<td>Date of data retrieval</td>
</tr>
<tr>
<td>version</td>
<td>Data version from the API</td>
</tr>
<tr>
<td>answ_by_id</td>
<td>Id of minister answering question</td>
</tr>
<tr>
<td>answ_by_minister_id</td>
<td>Department id of answering minister</td>
</tr>
<tr>
<td>answ_by_minister_title</td>
<td>Department title of answering minister</td>
</tr>
<tr>
<td>answ_date</td>
<td>Date answer was given</td>
</tr>
<tr>
<td>answ_on_belhalf_of</td>
<td>Answer given on behalf of</td>
</tr>
<tr>
<td>answ_on_belhalf_of_minister_id</td>
<td>Department id of minister given answer on behalf of</td>
</tr>
<tr>
<td>answ_on_belhalf_of_minister_title</td>
<td>Department title of minister given answer on behalf of</td>
</tr>
<tr>
<td>topic_ids</td>
<td>Id of relevant topics for question</td>
</tr>
<tr>
<td>moved_to</td>
<td>Question moved to</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><code>asked_by_other_id</code></td>
<td>MP id, if question was not asked by the questioning MP id</td>
</tr>
<tr>
<td><code>correct_person</code></td>
<td>Question id</td>
</tr>
<tr>
<td><code>correct_person_minister_id</code></td>
<td>Not documented in API</td>
</tr>
<tr>
<td><code>correct_person_minister_title</code></td>
<td>Not documented in API</td>
</tr>
<tr>
<td><code>sendt_date</code></td>
<td>Date the question was sent</td>
</tr>
<tr>
<td><code>session_id</code></td>
<td>Session id</td>
</tr>
<tr>
<td><code>question_from_id</code></td>
<td>Question from MP id</td>
</tr>
<tr>
<td><code>question_number</code></td>
<td>Question number within session</td>
</tr>
<tr>
<td><code>question_to_id</code></td>
<td>Question directed to minister id</td>
</tr>
<tr>
<td><code>question_to_minister_id</code></td>
<td>Question directed to minister department id</td>
</tr>
<tr>
<td><code>question_to_minister_title</code></td>
<td>Question directed to minister department title</td>
</tr>
<tr>
<td><code>status</code></td>
<td>Question status</td>
</tr>
<tr>
<td><code>title</code></td>
<td>Question title</td>
</tr>
<tr>
<td><code>type</code></td>
<td>Question type</td>
</tr>
</tbody>
</table>

**See Also**

- `get_question_hour`
- `get_question`

**Examples**

```r
## Not run:
interp <- get_session_questions(sessionid = "2013-2014",
    q_type = "interpellasjoner",
    status = "trukket")

test <- interp

## End(Not run)
```

---

**get_topics**

*Get list of topics and sub-topics for the Norwegian parliament*

**Description**

A function for retrieving topic keys used to label various data from the Norwegian parliament.

**Usage**

```r
get_topics(keep_sub_topics = TRUE)
```

**Arguments**

- `keep_sub_topics` Logical. Whether to keep sub-topics (default) for all main topics or not.
get_vote

Value

A list with two elements:

1. $topics (All topics)

   - response_date
   - version
   - is_main_topic
   - main_topic_id
   - id
   - name

2. $main_topics (exclusively main topics, if keep_sub_topics = TRUE)

   - response_date
   - version
   - is_main_topic
   - main_topic_id
   - id
   - name

Examples

```r
# Request the data
tops <- get_topics()

# Look at the first main topic
tops$main_topics[1, ]

# Extract all sub-topics for the first main topic
tops/topics[which(tops/topics$main_topic_id == 5), ]
```

---

get_vote 

Retrieve votes for a specific case

Description

A function for retrieving all votes from a case. Vote data are only available from the 2011-2012 session.

Usage

get_vote(caseid = NA, good_manners = 0)
get_vote

Arguments

caseid Character string indicating the id of the case to request all votes from
good_manners Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame with the following variables:

- `response_date` Date of data retrieval
- `version` Data version from the API
- `case_id` Case id up for vote
- `alternative_vote` Whether vote is an alternative vote
- `n_for` Number of votes for
- `n_absent` Number of MPs absent
- `n_against` Number of votes against
- `treatment_order` Order of treated votes
- `agenda_case_number` Case number on the agenda of the meeting
- `free_vote` Logical indication of whether the vote is related to the case as a whole
- `comment` Vote comment
- `meeting_map_number` Number on the meeting map
- `personal_vote` Logical indication of whether vote was recorded as roll call or not
- `president_id` Id of president holding president chair at the time of voting
- `president_party_id` Party of the sitting president
- `adopted` Logical indication of whether the proposal voted on was adopted
- `vote_id` Id of vote
- `vote_method` Voting method
- `vote_result_type` Result type (enstemmig_vedtatt = unanimously adopted)
- `vote_result_type_text` See `vote_result_type`
- `vote_topic` Description of the proposal voted upon
- `vote_datetime` Date and time of vote

See Also

get_decision_votes get_proposal_votes get_vote get_session_cases get_case

Examples

```r
## Not run:
get_vote(63033)
```

## End(Not run)
get_written_hearing_input

Retrieves written input for a specified hearing

Description

A function for retrieving written input for a specified hearing

Usage

get_written_hearing_input(hearingid = NA, good_manners = 0)

Arguments

hearingid  Character string indicating the id of the hearing to retrieve.
good_manners  Integer. Seconds delay between calls when making multiple calls to the same function

Value

A data.frame the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>response_date</td>
<td>Date of data retrieval</td>
</tr>
<tr>
<td>version</td>
<td>Data version from the API</td>
</tr>
<tr>
<td>hearing_id</td>
<td>Id of the hearing</td>
</tr>
<tr>
<td>hearing_type</td>
<td>Type of hearing</td>
</tr>
<tr>
<td>committee_id</td>
<td>Id of committee responsible for the hearing</td>
</tr>
<tr>
<td>hearing_input_date</td>
<td>Date of receiving input</td>
</tr>
<tr>
<td>hearing_input_id</td>
<td>Hearing input id</td>
</tr>
<tr>
<td>hearing_input_organization</td>
<td>Organization giving input</td>
</tr>
<tr>
<td>hearing_input_text</td>
<td>Full text of the hearing input</td>
</tr>
<tr>
<td>hearing_input_title</td>
<td>Title of the hearing input</td>
</tr>
</tbody>
</table>

See Also

get_hearing_input, get_hearing_program, get_session_hearings

Examples

## Not run:

get_written_hearing_input(hearingid = 10004243)

## End(Not run)
Interpellations from the 2002-2003

Description
A dataset containing all interpellations in the 2002-2003 parliamentary session in *Stortinget*

Usage
interp0203

Format
A data frame with 26 columns and 22 rows

- **response_date**: Date of data retrieval
- **version**: Data version from the API
- **answ_by_id**: Id of minister answering question
- **answ_by_minister_id**: Department id of answering minister
- **answ_by_minister_title**: Department title of answering minister
- **answ_date**: Date answer was given
- **answ_on_belhalf_of**: Answer given on behalf of
- **answ_on_belhalf_of_minister_id**: Department id of minister given answer on behalf of
- **answ_on_belhalf_of_minister_title**: Department title of minister given answer on behalf of
- **topic_ids**: Id of relevant topics for question
- **moved_to**: Question moved to
- **asked_by_other_id**: MP id, if question was not asked by the questioning MP
- **id**: Question id
- **correct_person**: Not documented in API
- **correct_person_minister_id**: Not documented in API
- **correct_person_minister_title**: Not documented in API
- **sendt_date**: Date the question was sent
- **session_id**: Session id
- **question_from_id**: Question from MP id
- **question_number**: Question number within session
- **question_to_id**: Question directed to minister id
- **question_to_minister_id**: Question directed to minister department id
- **question_to_minister_title**: Question directed to minister department title
- **type**: Question type
- **title**: Question title
- **status**: Question status
mps4549  

Members of parliament from the 1945-1949

Description

A dataset containing all MPs during the 1945-1949 parliamentary period in Stortinget

Usage

mps4549

Format

A data frame with 12 columns and 150 rows

response_date  Date of data retrieval  
version  Data version from the API  
death  Date of death  
lastname  MP last name  
birth  Date of birth  
firstname  MP first name  
mp_id  MP id  
gender  MP gender  
county_id  Id of county MP represented  
party_id  Id of party MP represented  
substitute_mp  Logical for whether MP is a substitute  
period_id  Id of period represented in

Source

https://data.stortinget.no/eksport/representanter?stortingsperiodeid=1945-49
parl_periods

Parliamentary periods

Description
A dataset containing all parliamentary periods in Stortinget

Usage
parl_periods

Format
A data frame with 12 columns and 150 rows

- response_date: Date of data retrieval
- version: Data version from the API
- from: Date session started
- id: Id of for session (used for other functions)
- to: Date session ended
- years: From year to year in full format

Source
https://data.stortinget.no/eksport/stortingsperioder

parl_sessions

Parliamentary sessions

Description
A dataset containing all parliamentary sessions in Stortinget

Usage
parl_sessions

Format
A data frame with 6 columns and 36 rows

- response_date: Date of data retrieval
- version: Data version from the API
- from: Date session started
- id: Id of for session (used for other functions)
- to: Date session ended
- years: From year to year in full format
**Source**

https://data.stortinget.no/eksport/sesjoner

---

**read_obt**

*Read Oslo-Bergen-Tagger processed files into R*

**Description**

A function reading OBT-tagged files

**Usage**

read_obt(file = NA)

**Arguments**

file character. Path to OBT-tagged file

**Value**

A data frame with the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentence</td>
<td>Sentence number</td>
</tr>
<tr>
<td>index</td>
<td>Token number in sentence</td>
</tr>
<tr>
<td>token</td>
<td>Raw token, as read by OBT originally</td>
</tr>
<tr>
<td>lwr</td>
<td>Lowercase raw token</td>
</tr>
<tr>
<td>lemma</td>
<td>Lemmatized token</td>
</tr>
<tr>
<td>pos</td>
<td>Part of Speech</td>
</tr>
<tr>
<td>morph</td>
<td>Morphological tags</td>
</tr>
</tbody>
</table>

**Examples**

```r
## Not run:
sample_text <- read_obt("./inst/extdata/obt_sample.txt")
head(sample_text)

## End(Not run)
```

---

**vote**

*Meta data on votes of case id 78686*

**Description**

A dataset containing vote information on case id 78686 in Stortinget
Usage

vote

Format

A list with three elements (votes)

- response_date: Date of data retrieval
- version: Data version from the API
- case_id: Case id up for vote
- alternative_vote: Whether vote is an alternative vote
- n_for: Number of votes for
- n_absent: Number of MPs absent
- n_against: Number of votes against
- treatment_order: Order of treated votes
- agenda_case_number: Case number on the agenda of the meeting
- free_vote: Logical indication of whether the vote is related to the case as a whole
- comment: Vote comment
- meeting_map_number: Number on the meeting map
- personal_vote: Logical indication of whether vote was recorded as roll call or not
- president_id: Id of president holding president chair at the time of voting
- president_party_id: Party of the sitting president
- adopted: Logical indication of whether the proposal voted on was adopted
- vote_id: Id of vote
- vote_method: Voting method
- vote_result_type: Result type (enstemmig_vedtatt = unanimously adopted)
- vote_result_type_text: See vote_result_type
- vote_topic: Description of the proposal voted upon
- vote_datetime: Date and time of vote

Source

https://data.stortinget.no/eksport/voteringsresultat?voteringid=15404
Roll call vote results for vote ids 15404, 15405, and 15406

Description
A dataset containing all personal votes for votes 15404, 15405, and 15406 in Stortinget

Usage
vote_result

Format
A list with one vote per element

response_date Date of data retrieval
version Data version from the API
vote_id Id of vote
mp_id MP id
party_id Party id
vote Vote: for, mot (against), ikke_tilstede (absent)
permanent_sub_for Id of the MP originally holding the seat, if the substitute is permanent
sub_for Id of the MP originally holding the seat

Source
Index

* datasets
  cases, 3
covid_relief, 3
covid_relief_result, 4
interp0203, 47
mps4549, 48
parl_periods, 49
parl_sessions, 49
vote, 50
vote_result, 52
cases, 3

get_all_committees, 5, 33, 35
get_all_parties, 6, 40
get_case, 6, 14, 32, 45
get_counties, 9
get_decision_votes, 10, 25, 30, 34, 45
get_hearing_input, 11, 12, 37, 46
get_hearing_program, 11, 12, 37, 46
get_meeting_agenda, 13, 38
get_mp, 14, 18, 19, 21, 22, 33, 39
get_mp_bio, 15, 15, 19, 21, 22, 30, 33, 39
get_mp_pic, 15, 18, 18, 21
get_parlperiod_mps, 15, 18, 19, 20
get_parlperiod_presidency, 21
get_parlperiods, 19, 23
get_parl_sessions, 20, 22
get_proceedings, 23
get_proposal_votes, 10, 24, 30, 45
get_publication, 25, 26, 27, 29, 39, 41
get_question, 14, 26, 29, 43
get_question_hour, 14, 26, 27, 28, 38, 43
get_result_vote, 10, 25, 30
get_session_cases, 3, 8, 31, 45
get_session_committees, 5, 33, 35
get_session_decisions, 10, 34
get_session_delegations, 35
get_session_hearings, 11, 12, 36, 46
get_session_meetings, 14, 37
get_session_mp_speech_activity, 15, 18, 21, 38
get_session_parties, 6, 40
get_session_publications, 26, 41
get_session_questions, 29, 42
get_topics, 43
get_vote, 10, 25, 30, 32, 44, 45
get_written_hearing_input, 11, 12, 37, 46
ideal, 30
interp0203, 47
mps4549, 48
parl_periods, 49
parl_sessions, 49
read_obt, 50
rvest, 25
vote, 50
vote_result, 52