

# Package ‘stortingscrape’

May 9, 2026

**Type** Package

**Title** Access Data from the Norwegian Parliament API

**Version** 0.4.1

**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, httr2, parallel, stringr

**Suggests** magick, rmarkdown, knitr, pscl

**License** GPL (>= 3)

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**LazyData** true

**NeedsCompilation** no

**Author** Martin Søyland [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-2463-8469>>)

**Maintainer** Martin Søyland <[martin.soyland@stv.uio.no](mailto:martin.soyland@stv.uio.no)>

**Repository** CRAN

**Date/Publication** 2025-04-07 08:00:02 UTC

## Contents

cases . . . . .	3
covid_relief . . . . .	3
covid_relief_result . . . . .	4
get_all_committees . . . . .	5

get_all_parties . . . . .	6
get_case . . . . .	6
get_counties . . . . .	9
get_decision_votes . . . . .	10
get_hearing_input . . . . .	11
get_hearing_program . . . . .	12
get_meeting_agenda . . . . .	13
get_mp . . . . .	14
get_mp_bio . . . . .	15
get_mp_pic . . . . .	18
get_parlperiods . . . . .	19
get_parlperiod_mps . . . . .	20
get_parlperiod_presidency . . . . .	21
get_parlsessions . . . . .	22
get_proceedings . . . . .	23
get_proposal_votes . . . . .	24
get_publication . . . . .	25
get_question . . . . .	26
get_question_hour . . . . .	28
get_result_vote . . . . .	30
get_session_cases . . . . .	31
get_session_committees . . . . .	33
get_session_decisions . . . . .	34
get_session_delegations . . . . .	35
get_session_hearings . . . . .	36
get_session_meetings . . . . .	37
get_session_mp_speech_activity . . . . .	38
get_session_parties . . . . .	40
get_session_publications . . . . .	41
get_session_questions . . . . .	42
get_topics . . . . .	43
get_vote . . . . .	45
get_written_hearing_input . . . . .	46
interp0203 . . . . .	47
mps4549 . . . . .	48
parl_periods . . . . .	49
parl_sessions . . . . .	49
read_obt . . . . .	50
st_party_colors . . . . .	51
vote . . . . .	52
vote_result . . . . .	53

---

cases	<i>Storting cases in the 2019-2020 session</i>
-------	--

---

**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**

<https://data.stortinget.no/eksport/saker?sesjonid=2019-2020>

---

covid_relief	<i>Vote id 85196</i>
--------------	----------------------

---

**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

**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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>id</b>	Id of the committee
<b>name</b>	Name of the committee
<b>sessionid</b>	Id of session (empty)

### See Also

[get\\_session\\_committees](#)

### Examples

```
## 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

```
get_all_parties()
```

### Value

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

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

### See Also

[get\\_session\\_parties](#)

### Examples

```
## 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**

```
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)

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

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

<b>is_main_topic</b>	Is this (row) the main topic?
<b>main_topic_id</b>	Id for main topic
<b>id</b>	Topic id
<b>navn</b>	Topic name

3. **\$publication\_references** (references for publications on the case)

<b>export_id</b>	Id for export of publication (used in ?get_publication)
<b>link_text</b>	Publication title
<b>link_url</b>	URL to publication
<b>type</b>	Publication type
<b>subtype</b>	Publication subtype (chamber)

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

<b>mp_id</b>	MPs id
<b>party_id</b>	Party id of MPs
<b>sub_mp</b>	Whether MPs are substitutes

5. **\$proceeding\_steps** (case proceeding steps)

<b>step_name</b>	Name of steps
<b>step_number</b>	Step order for case
<b>outdated</b>	Whether the step type is outdated

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

<b>mp_id</b>	MPs id
<b>party_id</b>	Party id of MPs
<b>sub_mp</b>	Whether MPs are substitutes

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

<b>keyword</b>	Keywords for the case
----------------	-----------------------

### See Also

[get\\_session\\_cases](#)

### Examples

```
## Not run:

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

```
# 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	<i>Get list of electoral districts</i>
--------------	--

---

## Description

A function for retrieving current and/or historical electoral districts (counties) for the Norwegian parliament.

## Usage

```
get_counties(historical = FALSE)
```

## Arguments

**historical** Logical. Whether or not to include historical counties.

## Value

A data frame with the following variables:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>historical_county</b>	Whether the county is historical (no longer exists)
<b>id</b>	Id of the county
<b>name</b>	Name of the county

**Examples**

```
## Not run:
# Request one MP by id
get_counties()

# With historical counties
get_counties(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:

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

**See Also**

[get\\_session\\_decisions](#) [get\\_proposal\\_votes](#) [get\\_vote](#) [get\\_result\\_vote](#)

**Examples**

```
## Not run:
decision <- get_decision_votes(123)
decision

## End(Not run)
```

---

get_hearing_input	<i>Retrieve the hearing input for a specified hearing</i>
-------------------	---

---

**Description**

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

**Usage**

```
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:

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

**See Also**

[get\\_session\\_hearings](#) [get\\_hearing\\_program](#) [get\\_written\\_hearing\\_input](#)

**Examples**

```
## Not run:
get_hearing_input(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**

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

**Value**

A data.frame with the following variables:

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

**See Also**

[get\\_session\\_hearings](#) [get\\_hearing\\_input](#) [get\\_written\\_hearing\\_input](#)

**Examples**

```
## Not run:
s0910 <- get_session_hearings("2009-2010")
hearing <- get_hearing_program(s0910$hearing$hearing_id[1])
head(hearing)

## End(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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>agenda_number</b>	The agenda number for the session
<b>meeting_date</b>	Date of the meeting
<b>meeting_id</b>	Meeting id
<b>meeting_place</b>	Where the meeting was held
<b>agenda_case_reference</b>	Reference for the case on the agenda
<b>agenda_case_number</b>	Case number
<b>agenda_case_text</b>	Case description
<b>agenda_case_type</b>	Case type
<b>footnote</b>	Footnote for the case
<b>proposition_id</b>	If relevant, belonging proposition id
<b>committee_id</b>	If relevant, id of the responsible committee
<b>legacy_question_id</b>	Legacy question id, only used for mathing with old data (pre 10.12.2024)
<b>loose_proposals</b>	Whether there are loose proposals to the case

<b>case_id</b>	Id of the case
<b>question_hour_type</b>	If relevant, type of question hour
<b>question_id</b>	If relevant, question id

**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**

<b>mpid</b>	Character string indicating the id of the MP to retrieve.
<b>good_manners</b>	Integer. Seconds delay between calls when making multiple calls to the same function

**Value**

A data.frame with the following variables:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>death</b>	MP date of death, if applicable
<b>last_name</b>	MP last name
<b>birth</b>	MP date of birth
<b>first_name</b>	MP first name

<b>id</b>	MP id
<b>gender</b>	MP gender

**See Also**

[get\\_mp\\_bio](#) [get\\_parlperiod\\_mps](#) [get\\_mp\\_pic](#) [get\\_session\\_mp\\_speech\\_activity](#)

**Examples**

```
## 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	<i>Extract biography of specific MPs</i>
------------	--

---

**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)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>id</b>	Id of the MP

2. **\$literature** (all literature the MP contributed to)

<b>year</b>	Year of entry publication
<b>description</b>	Description of the publication
<b>last_name</b>	MP's last name
<b>more_years</b>	<b>Not described in the API</b>
<b>publisher</b>	Publisher
<b>first_name</b>	First name of the MP
<b>place</b>	Place of publication
<b>title</b>	Title of the publication
<b>type</b>	MP's role in publication (author etc)

3. **\$leave\_of\_absence** (times the MP was on leave)

<b>from_date</b>	Start date of leave
<b>reason</b>	Reason for leave
<b>to_date</b>	End of leave
<b>type</b>	Type of leave
<b>sub_last_name</b>	Substitute MP last name (id not recorded)
<b>sub_first_name</b>	Substitute MP first name

4. **\$personalia** (the MP's personalia)

<b>seniority_aar</b>	Number of years in parliament
<b>seniority_dager</b>	Number of extra days (addition to years)
<b>county_of_birth</b>	Birth county of the MP
<b>municipality_of_birth</b>	Birth municipality of the MP
<b>eulogy_date</b>	Eulogy date of the MP, when applicable

5. **\$father** (the MP's father personalia)

<b>death_year</b>	Father's year of death
<b>last_name</b>	Father's last name
<b>birth_year</b>	Father's year of birth
<b>first_name</b>	Father's first name
<b>profession</b>	Father's profession

6. **\$mother** (the MP's mother personalia)

<b>death_year</b>	Mother's year of death
<b>last_name</b>	Mother's last name
<b>birth_year</b>	Mother's year of birth
<b>first_name</b>	Mother's first name
<b>profession</b>	Mother's profession

7. **\$parl\_periods** (parliamentary periods the MP has held a seat)

<b>from_date</b>	Date MP held seat from
<b>county</b>	County the MP represented
<b>party_id</b>	Party id for the MP's party
<b>rep_number</b>	Representative number (within the whol parliament)
<b>parl_period_id</b>	Id of the parliamentary period
<b>to_date</b>	Date MP held a seat to
<b>type</b>	Type of representation

8. **\$parl\_positions** (parliamentary positions held by the MP)

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

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

<b>several_periods_text</b>	Text description if the vocation was held for several periods
<b>from_year</b>	Year MP held vocation from
<b>from_year_unknown</b>	Logical indication for whether the start year is unknown
<b>note</b>	Note for vocation
<b>name</b>	Name of vocation
<b>to_year</b>	Year MP held vocation to
<b>to_year_unknown</b>	Logical indication for whether the end year is unknown

**type** Vocation type (10 = education, 20 = work)

#### 10. \$other\_positions (other positions held outside parliament)

<b>several_periods_text</b>	Text description if the vocation was held for several periods (removed from API)
<b>from_year</b>	Year MP held vocation from
<b>from_year_sorting</b>	<b>Not described in API</b> (removed from API)
<b>from_year_unknown</b>	Logical indication for whether the start year is unknown
<b>max_to_year</b>	The last possible time the MP held the position (removed from API)
<b>note</b>	Note for position
<b>min_to_year</b>	The earliest possible time the MP held the position (removed from API)
<b>level</b>	<b>Not described in API</b>
<b>organization</b>	Organization holding the position
<b>place</b>	Place of the position
<b>to_year</b>	Year MP held position to
<b>to_year_unknown</b>	Logical indication for whether the end year is unknown
<b>type</b>	Position type
<b>position</b>	Position name/description

#### See Also

[get\\_mp](#) [get\\_parlperiod\\_mps](#) [get\\_mp\\_pic](#) [get\\_session\\_mp\\_speech\\_activity](#)

#### Examples

```
## Not run:

# Request one MP by id
get_mp_bio("AAMH")

## End(Not run)
```

---

get_mp_pic	<i>Retrieve picture of specific MPs</i>
------------	---

---

#### 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)
```

**Arguments**

mpid	Character string indicating the id of the MP to retrieve.
size	Character string size of the picture. Accepts values "lite" (small), "middels" (medium – default), and "stort" (big).
destfile	Character string specifying where to save the picture
show_plot	Logical. FALSE (default) if no plot should be produced and TRUE if plot should be produced. Requires the "imager" package.
good_manners	Integer. Seconds delay between calls when making multiple calls to the same function

**Value**

Picture of the requested MP in the preferred size.

**See Also**

[get\\_mp](#) [get\\_parlperiod\\_mps](#) [get\\_mp\\_bio](#)

**Examples**

```
## 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\_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:

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

**See Also**

[get\\_parsessions](#)

**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

**Value**

A data.frame with the following variables:

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

**See Also**

[get\\_mp\\_bio](#) [get\\_mp](#) [get\\_mp\\_pic](#) [get\\_session\\_mp\\_speech\\_activity](#)

**Examples**

```
## Not run:

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

# Request MPs from several periods by id
ids <- c("1961-65", "1997-01", "2009-2013")
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.

**Usage**

```
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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>last_name</b>	Last name of presidency member
<b>first_name</b>	First name of presidency member
<b>from_date</b>	Presidency member from date
<b>party_id</b>	Party affiliation of presidency member
<b>person_id</b>	Id of the presidency member
<b>to_date</b>	Presidency member to date
<b>position</b>	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\_parlsessions      *Get Parliamentary Sessions*

---

**Description**

A function for retrieving dates of the parliamentary sessions after WWII

**Usage**

```
get_parlsessions()
```

**Value**

A data.frame with the following variables:

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

**See Also**

[get\\_parlperiods](#)

**Examples**

```
## 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. **\$root** (only download meta data)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API

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

<b>id</b>	Id of proceeding
<b>name</b>	Name of proceeding

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

<b>id</b>	Id of proceeding step
<b>name</b>	Name of proceeding step
<b>step_number</b>	Order of proceeding steps
<b>outdated</b>	Whether the step is outdated
<b>main_id</b>	Id for proceeding type the step belongs to

### Examples

```
## Not run:

get_proceedings()

## End(Not run)
```

---

get\_proposal\_votes      *Retrieve 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)
```

### 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 list with two elements:

1. **\$proposal\_vote** (main data on the vote proposal)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>vote_id</b>	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**

```
## 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      *Retrieve 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 node extraction functions such as `rvest::html_elements()`.

**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\_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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>justification</b>	Justification for question
<b>answ_by_id</b>	Id for answering minister
<b>answ_by_minister_id</b>	Id for department of answering minister
<b>answ_by_minister_title</b>	Title for department of answering minister
<b>answ_date</b>	Date question was asked
<b>answ_on_belhalf_of</b>	Id of minister answered on behalf of, when relevant
<b>answ_on_belhalf_of_minister_id</b>	Id of department answered on behalf of, when relevant
<b>answ_on_belhalf_of_minister_title</b>	Title of department answered on behalf of, when relevant
<b>agenda_number</b>	Agenda number in meeting
<b>moved_to</b>	Date moved to
<b>id</b>	Question id
<b>legacy_id</b>	Question legacy id (only use to match with data downloaded before 10.12.2024)
<b>correct_person_id</b>	Not documented in API
<b>correct_person_minister_id</b>	Not documented in API
<b>correct_person_minister_title</b>	Not documented in API
<b>sendt_date</b>	Date question was sent
<b>session_id</b>	Session id
<b>question_text</b>	Full question text
<b>question_from_id</b>	Id of MP asking the question
<b>question_number</b>	Question number
<b>question_to_id</b>	Id of minister the question was asked to
<b>question_to_minister_id</b>	Department id of minister the question was asked to
<b>question_to_minister_title</b>	Department title of minister the question was asked to
<b>answer_text</b>	Answer text (often empty)
<b>title</b>	Question title
<b>type</b>	Question type
<b>asked_by_other_id</b>	ID of person that asked the question on behalf of question_from_id

**See Also**

[get\\_question\\_hour](#) [get\\_publication](#) [get\\_meeting\\_agenda](#)

**Examples**

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

## Retrieve sessions
sessions <- get_parlsessions()

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

```
## Retrieve detailed information on all interpellations in that session
library(pbmclapply) # for progress bar. never use paralell on scraping
int1213 <- pbmclapply(qsesh$id, function(x){
  get_question(x, good_manners = 2)
}, mc.cores = 1)

quest1213 <- do.call(rbind, int1213)

## End(Not run)
```

---

get\_question\_hour      *Retrieve question hour details for a specified meeting*

---

### Description

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

### Usage

```
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)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>meetingid</b>	The called meeting id

2. **\$question\_hour\_ministers** (id of ministers in parliament during question hour/time)

<b>id</b>	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
answer_by_minister_title	Title for department of answering minister
answer_date	Date of receiving answer
answer_on_behalf_of_id	Id of minister answered on behalf of, when relevant
answer_on_behalf_of_minister_id	Id of department answered on behalf of, when relevant
answer_on_behalf_of_minister_title	Title of department answered on behalf of, when relevant
agenda_case_number	Case number on agenda
date	Date question hour was held
moved_to	Date moved to
asked_by_other_id	Id for minister asking on behalf of another, when relevant
question_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
sent_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
status	Question status
answer	Answer text (often empty)
title	Question title
type	Question type

### 4. \$publication\_reference

<b>export_id</b>	Export id for publication (see get_publication())
<b>link_text</b>	Description text for publication
<b>link_url</b>	URL for publication
<b>type</b>	Type of publication
<b>sub_type</b>	Subtype for publication (location)

#### See Also

[get\\_question](#) [get\\_session\\_questions](#) [get\\_publication](#)

#### Examples

## Not run:

```
get_question_hour(10232)

## End(Not run)
```

---

get_result_vote	<i>Retrieve vote results on MP level for a specified vote</i>
-----------------	---

---

### 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

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:

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

### See Also

[get\\_decision\\_votes](#) [get\\_proposal\\_votes](#) [get\\_vote](#) [get\\_mp\\_bio](#)

**Examples**

```
## 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", "") |>
  stringr::str_trim()

## 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)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>treated_session_id</b>	Session the case was treated in
<b>document_group</b>	Document group the case belongs to
<b>reference</b>	Document reference
<b>id</b>	Case id

<b>com_req_id</b>	Committee recommendation id
<b>com_req_code</b>	Committee recommendation code
<b>title_short</b>	Short title of case
<b>case_filed_id</b>	Id of filed case
<b>last_update_date</b>	Date of last update on case
<b>status</b>	Status of the case
<b>title</b>	Full title of the case
<b>type</b>	Type of case
<b>session_id</b>	Session id of the case
<b>committee_id</b>	Responsible committee id

## 2. \$topics (named list by case id)

<b>is_main_topic</b>	Logical indication whether the topic is the main topic
<b>main_topic_id</b>	Id of the main topic for the case
<b>id</b>	Topic id
<b>name</b>	Topic name

## 3. \$proposers (named list by case id)

<b>rep_id</b>	Proposing MP id
<b>county_id</b>	County id of proposing MP
<b>party_id</b>	Party id of proposing MP
<b>rep_sub</b>	Logical indicator for whether MP is a substitute

## 4. \$spokespersons (data frame by case id)

<b>case_id</b>	Case id
<b>rep_id</b>	Spokesperson(s) MP id for the case
<b>county_id</b>	County id of spokesperson MP
<b>party_id</b>	Party id of spokesperson MP
<b>rep_sub</b>	Logical indicator for whether MP is a substitute

### See Also

[get\\_case](#) [get\\_vote](#)

### Examples

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

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

---

```
get_session_committees
```

*Parliamentary committees in specified session*

---

## 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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>id</b>	Committee id
<b>name</b>	Committee name
<b>session_id</b>	Session id

## See Also

[get\\_all\\_committees](#) [get\\_mp\\_bio](#) [get\\_mp](#)

## Examples

```
## 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:

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

### See Also

[get\\_decision\\_votes](#)

### Examples

```
## Not run:

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

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

---

```
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

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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>id</b>	Delegation id
<b>name</b>	Name of delegation
<b>session_id</b>	Session id

### See Also

[get\\_session\\_committees](#) [get\\_all\\_committees](#)

### Examples

```
## Not run:

deleg <- get_session_delegations("2015-2016")
deleg

## 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)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>session_id</b>	Session id

2. **\$hearing** (main data on the hearing)

<b>deadline_date</b>	Deadline date for hearing
<b>status</b>	Data version from the API
<b>hearing_id</b>	Hearing id
<b>input_deadline</b>	Deadline date for input
<b>written</b>	Logical indication of whether the input was written
<b>application_deadline</b>	Deadline date for application to hearing
<b>start_date</b>	Start date for hearing
<b>status_pub</b>	Publication status for hearing
<b>status_info_text</b>	Status information text
<b>type</b>	Type of hearing
<b>committee_id</b>	Committee id for committee responsible for hearing

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

<b>hearing_id</b>	Hearing id
<b>case_reference</b>	Text reference for case
<b>case_id</b>	Case id
<b>case_short_title</b>	Short title for case
<b>case_publication</b>	URL for front end web-page publication
<b>case_title</b>	Full title for case

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

<b>hearing_id</b>	Hearing id
<b>date</b>	Date of hearing
<b>place</b>	Where the hearing was held

### See Also

[get\\_hearing\\_input](#) [get\\_hearing\\_program](#) [get\\_written\\_hearing\\_input](#)

### Examples

```
## 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

```
get_session_meetings(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:

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

**See Also**

[get\\_meeting\\_agenda](#) [get\\_question\\_hour](#)

**Examples**

```
## 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:

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

**See Also**

[get\\_mp](#) [get\\_mp\\_bio](#) [get\\_publication](#)

**Examples**

```
## 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

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:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>id</b>	Party id
<b>name</b>	Party name
<b>represented_party</b>	Logical indication of whether party was represented
<b>session_id</b>	Session id
<b>period_id</b>	Parliamentary period id

### See Also

[get\\_all\\_parties](#)

### Examples

```
## 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

sessionid	Character string indicating the id of the hearing to retrieve.
type	Character specifying type of publication to download. Available types are "referat" (minutes), "innstilling" (proposition), "innberetning" (report), "lovvedtak" (law decision), "lovanmerkning" (law note), "dok8" (MP proposal) "dok12" (Constitutional proposal), and "dokumentserie" (document series). Defaults to "referat".
good_manners	Integer. Seconds delay between calls when making multiple calls to the same function

### Value

A data.frame with the following variables:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>session_id</b>	Session id
<b>publication_date</b>	Date of publication
<b>publication_id</b>	Id of publication
<b>publication_format</b>	Publication format (XML)
<b>publication_available_date</b>	When the publication was made available
<b>publication_title</b>	Publication title
<b>publication_type</b>	Publication type

### See Also

[get\\_publication](#)

**Examples**

```
## Not run:

pub <- get_session_publications("1998-99")
head(pub)

## End(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**

sessionid	Character string indicating the id of the session to request interpellations from
q_type	Character string indicating type of question to retrieve. Options are "interpellasjoner" (interpellations), "sporretimesporsmal" (oral questions), or "skriftligesporsmal" (written questions).
status	Character string question status extraction. Possible values are NA (extract all questions), "til_behandling" (pending questions), "trukket" (withdrawn questions), "bortfalt" (lapsed questions), or "alle" (all questions)
good_manners	Integer. Seconds delay between calls when making multiple calls to the same function

**Value**

A data.frame with the following variables:

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>answ_by_id</b>	Id of minister answering question
<b>answ_by_minister_id</b>	Department id of answering minister
<b>answ_by_minister_title</b>	Department title of answering minister
<b>answ_date</b>	Date answer was given
<b>answ_on_belhalf_of</b>	Answer given on behalf of

<b>answ_on_belhalf_of_minister_id</b>	Department id of minister given answer on behalf of
<b>answ_on_belhalf_of_minister_title</b>	Department title of minister given answer on behalf of
<b>topic_ids</b>	Id of relevant topics for question
<b>moved_to</b>	Question moved to
<b>asked_by_other_id</b>	MP id, if question was not asked by the questioning MP
<b>id</b>	Question id
<b>correct_person</b>	Not documented in API
<b>correct_person_minister_id</b>	Not documented in API
<b>correct_person_minister_title</b>	Not documented in API
<b>sendt_date</b>	Date the question was sent
<b>session_id</b>	Session id
<b>question_from_id</b>	Question from MP id
<b>question_number</b>	Question number within session
<b>question_to_id</b>	Question directed to minister id
<b>question_to_minister_id</b>	Question directed to minister department id
<b>question_to_minister_title</b>	Question directed to minister department title
<b>status</b>	Question status
<b>title</b>	Question title
<b>type</b>	Question type

**See Also**

[get\\_question\\_hour](#) [get\\_question](#)

**Examples**

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

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**

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

**Arguments**

keep\_sub\_topics

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

**Value**

A list with two elements:

1. **\$topics** (All topics)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>is_main_topic</b>	Logical indicator for whether the topic is a main topic
<b>main_topic_id</b>	Id of main topic
<b>id</b>	Id of topic
<b>name</b>	Name of topic

2. **\$main\_topics** (exclusively main topics, if keep\_sub\_topics = TRUE)

<b>response_date</b>	Date of data retrieval
<b>version</b>	Data version from the API
<b>is_main_topic</b>	Logical indicator for whether the topic is a main topic
<b>main_topic_id</b>	Id of main topic
<b>id</b>	Id of topic
<b>name</b>	Name of topic

**Examples**

```
## Not run:
# 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), ]

## End(Not run)
```

---

get_vote	<i>Retreive votes for a specific case</i>
----------	---

---

### 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)
```

### 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:

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

**See Also**

[get\\_decision\\_votes](#) [get\\_proposal\\_votes](#) [get\\_vote](#) [get\\_session\\_cases](#) [get\\_case](#)

**Examples**

```
## Not run:

get_vote(63033)

## End(Not run)
```

---

```
get_written_hearing_input
```

*Retrieve 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:

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

**See Also**

[get\\_hearing\\_input](#) [get\\_hearing\\_program](#) [get\\_session\\_hearings](#)

**Examples**

```
## Not run:

get_written_hearing_input(hearingid = 10004243)

## End(Not run)
```

---

 interp0203

*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

### Source

<https://data.stortinget.no/eksport/interpellasjoner?sesjonid=2002-2003>

---

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	<i>Parliamentary periods</i>
--------------	------------------------------

---

**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	<i>Parliamentary sessions</i>
---------------	-------------------------------

---

**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	<i>Read Oslo-Bergen-Tagger processed files into R</i>
----------	---

---

**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:

<b>sentence</b>	Sentence number
<b>index</b>	Token number in sentence
<b>token</b>	Raw token, as read by OBT originally
<b>lwr</b>	Lowercase raw token
<b>lemma</b>	Lemmatized token
<b>pos</b>	Part of Speech
<b>morph</b>	Morphological tags

**Examples**

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

## End(Not run)
```

---

st_party_colors	<i>Color palette for parties in the Storting</i>
-----------------	--

---

**Description**

A color palette for all (current) parties in the Storting

**Usage**

```
st_party_colors
```

**Format**

A vector of party abbreviations and official hex colors

**Arbeiderpartiet (Labour Party)** <https://www.arbeiderpartiet.no/om/presse/profil/>

**Fremskrittspartiet (Progress Party)** <https://www.frp.no/files/Grafiske-retningslinjer/FrP-Profilmannual-2023.pdf>

**Høyre (Conservative Party)** <https://hoyre.no/design/farger/>

**Kristelig Folkeparti (Christian Democratic Party)** <https://krf.no/ressursbank/logoarkiv/>

**Miljøpartiet De Grønne (Green Party)** <https://mdg.no/partiet/organisasjon#logo>

**Pasientfokus (Patient Focus)** <https://no.wikipedia.org/wiki/Mal:Farge/Pasientfokus>

**Rødt (Red Party)** <https://roedt.no/grafisk-materiell>

**Senterpartiet (Centre Party)** <https://profil.senterpartiet.no/point/no/senterpartietbc/component/default/24406>

**Sosialistisk Venstreparti (Socialist Left Party)** <https://www.sv.no/ressursbanken/grafisk/grafisk-profil/>

**Venstre (Liberal Party)** <https://www.venstre.no/organisasjon/visuell-identitet/>

**Source**

See list of links above; there are several color alternatives for most parties.

**Examples**

```
## Not run:

barplot(table(get_parlperiod_mps(parl_periods$id[1])$party_id), col = st_party_colors)

## 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>

---

`vote_result`*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**

<https://data.stortinget.no/eksport/voteringsresultat?voteringid=15404>, <https://data.stortinget.no/eksport/voteringsresultat?voteringid=15405>, <https://data.stortinget.no/eksport/voteringsresultat?voteringid=15406>

# Index

## \* datasets

- cases, 3
- covid\_relief, 3
- covid\_relief\_result, 4
- interp0203, 47
- mps4549, 48
- parl\_periods, 49
- parl\_sessions, 49
- st\_party\_colors, 51
- vote, 52
- vote\_result, 53

cases, 3

covid\_relief, 3

covid\_relief\_result, 4

get\_all\_committees, 5, 33, 35

get\_all\_parties, 6, 40

get\_case, 6, 14, 32, 46

get\_counties, 9

get\_decision\_votes, 10, 25, 30, 34, 46

get\_hearing\_input, 11, 12, 37, 47

get\_hearing\_program, 11, 12, 37, 47

get\_meeting\_agenda, 13, 27, 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\_parlsessions, 20, 22

get\_proceedings, 23

get\_proposal\_votes, 10, 24, 30, 46

get\_publication, 25, 26, 27, 29, 39, 41

get\_question, 14, 26, 26, 29, 43

get\_question\_hour, 14, 26, 27, 28, 38, 43

get\_result\_vote, 10, 25, 30

get\_session\_cases, 3, 8, 31, 46

get\_session\_committees, 5, 33, 35

get\_session\_decisions, 10, 34

get\_session\_delegations, 35

get\_session\_hearings, 11, 12, 36, 47

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, 45, 46

get\_written\_hearing\_input, 11, 12, 37, 46

ideal, 30

interp0203, 47

mps4549, 48

parl\_periods, 49

parl\_sessions, 49

read\_obt, 50

rvest::html\_elements(), 25

st\_party\_colors, 51

vote, 52

vote\_result, 53