

# Package ‘twfy’

June 26, 2017

**Title** Drive the API for TheyWorkForYou

**Version** 0.1.0

**Description** An R wrapper around the API of TheyWorkForYou, a parliamentary monitoring site that scrapes and repackages Hansard (the UK's parliamentary record) and augments it with information from the Register of Members' Interests, election results, and voting records to provide a unified source of information about UK legislators and their activities. See <<http://www.theyworkforyou.com>> for details.

**URL** <https://conjugateprior.github.io/twfy>

**BugReports** <https://github.com/conjugateprior/twfy/issues>

**Depends** R (>= 3.0.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** httr, jsonlite

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Will Lowe [aut, cre]

**Maintainer** Will Lowe <[wlowe@princeton.edu](mailto:wlowe@princeton.edu)>

**Repository** CRAN

**Date/Publication** 2017-06-26 16:37:19 UTC

## R topics documented:

call_api . . . . .	2
convertURL . . . . .	3
getBoundary . . . . .	3

getComments . . . . .	4
getCommittee . . . . .	5
getConstituencies . . . . .	5
getConstituency . . . . .	6
getDebates . . . . .	6
getGeometry . . . . .	7
getHansard . . . . .	8
getLord . . . . .	9
getLords . . . . .	10
getMLA . . . . .	10
getMLAs . . . . .	11
getMP . . . . .	12
getMPInfo . . . . .	13
getMPs . . . . .	14
getMPsInfo . . . . .	15
getMSP . . . . .	15
getMSPs . . . . .	16
getPerson . . . . .	17
getWMS . . . . .	18
getWrans . . . . .	18
get_api_key . . . . .	20
set_api_key . . . . .	20
twfy . . . . .	21

<b>Index</b>	<b>22</b>
--------------	-----------

---

call_api	<i>Call the API directly</i>
----------	------------------------------

---

## Description

All the other functions call this one - it's exported only for debugging purposes - so use them instead.

## Usage

```
call_api(endpoint, ...)
```

## Arguments

endpoint	Function name e.g. 'getConstituencies'
...	endpoint's (often optional) named arguments

## Details

If you're really curious about implementation, read on. Each API function introspects to see what its function name is, bundles up its named arguments, and calls this function with them. Consequently, aside from a bit of argument checking and/or return value massaging, every function is implemented exactly the same way.

**Value**

Response content, as unserialized by `jsonlite::fromJSON`

---

 convertURL

*Convert URL*


---

**Description**

Converts a parliament.uk Hansard URL into a TheyWorkForYou one, if possible.

**Usage**

```
convertURL(url)
```

**Arguments**

`url`                      url you want to convert

**Value**

A one row `data.frame` with columns `gid` and `url`.

**Examples**

```
## Not run:

site <- "http://www.publications.parliament.uk"
page <- "/pa/cm201212/cmhansrd/cm120418/debtext/120418-0001.htm"
in_page <- "#12041847002086"
address <- paste0(site, page, in_page)

res <- convertURL(address)
names(res) # gid, url

## End(Not run)
```

---

 getBoundary

*Get constituency boundary information as KML*


---

**Description**

Get constituency boundary information as KML

**Usage**

```
getBoundary(name)
```

**Arguments**

name                    Name of constituency

**Value**

A character vector full of KML (an XML dialect)

---

getComments                    *Get comments left on TheyWorkForYou*

---

**Description**

Get comments left on TheyWorkForYou

**Usage**

```
getComments(start_date = NULL, end_date = NULL, search = NULL,
             pid = NULL, page = NULL, num = NULL)
```

**Arguments**

start\_date            Beginning date  
end\_date                End date  
search                 A search string  
pid                     A person identifier  
page                    which page of results to provide. Defaults to first page  
num                     Number of results to return

**Value**

A list containing data.frames with columns

- comment\_id Comment identifier
- user\_id User identifier
- eobject\_id
- body Text of comment
- posted date and time posted
- major
- gid
- firstname Commenter's first name
- lastname Commenter's last name
- url URL of the comment
- useurl URL endpoint for commenter

---

getCommittee	<i>Get members of a Parliamentary Select Committee</i>
--------------	--

---

**Description**

If name is not specified, all committees and their members are returned.

**Usage**

```
getCommittee(name = NULL, date = NULL)
```

**Arguments**

name	Name of the committee. Partial names will be matched.
date	Date for which the committee membership is required

**Details**

TheyWorkForYou notes that "We have no information since the 2010 general election, and information before may be inaccurate."

**Value**

A data.frame of committee members, and also committees if name is not specified.

---

getConstituencies	<i>Get constituency names</i>
-------------------	-------------------------------

---

**Description**

Get constituency names

**Usage**

```
getConstituencies(date = NULL, search = NULL)
```

**Arguments**

date	Date for which constituency names are required
search	Search string

**Value**

a data.frame with single column name containing constituency names

---

getConstituency	<i>Get information for a constituency</i>
-----------------	---

---

**Description**

Get information for a constituency

**Usage**

```
getConstituency(name = NULL, postcode = NULL)
```

**Arguments**

name	Name of constituency
postcode	A postcode
	One of name or postcode is required.

**Value**

A data.frame with columns

- nameConstituency name
- pa\_idConstituency identifier
- bbc\_constituency\_idBBC numeric identifier
- guardian\_election\_resultsURL for election results
- guardian\_idGuardian numeric identifier
- guardian\_nameGuardian name identifier

---

getDebates	<i>Get information about parliamentary debates</i>
------------	--

---

**Description**

Get information about parliamentary debates

**Usage**

```
getDebates(type = c("commons", "westminsterhall", "lords", "scotland",
  "northernireland"), date = NULL, search = NULL, person = NULL,
  gid = NULL, order = c("d", "r"), page = NULL, num = NULL)
```

**Arguments**

type	location of the debate. Defaults to the House of Commons
date	Date on which debates are required
search	A search string
person	A person identifier to specify which speaker's contributions are required
gid	A speech/debate identifier to restrict to a particular debate
order	whether to order results by date or relevance. Defaults to date
page	which page of results to provide. Defaults to first page
num	Number of results to return

**Value**

A complex data.frame of debate information. Documentation is somewhat lacking, and many columns contain data.frames so you'll have to dig around.

---

getGeometry	<i>Return geometry information for a constituency</i>
-------------	---

---

**Description**

From TheyWorkForYou: "This currently includes, for Great Britain, the latitude and longitude of the centre point of the bounding box of the constituency, its area in square meters, the bounding box itself and the number of parts in the polygon that makes up the constituency. For Northern Ireland, as we don't have any better data, it only returns an approximate (estimated by eye) latitude and longitude for the constituency's centroid."

**Usage**

```
getGeometry(name)
```

**Arguments**

name	Name of constituency
------	----------------------

**Value**

A data.frame with columns

- parts number of connected parts of the constituency
- area area in square meters
- min\_lat minimum latitude in the constituency
- centre\_lat latitude of the central point of the constituency
- max\_lat maximum latitude in the constituency
- min\_long minimum longitude in the constituency

- centre\_long longitude of the central point of the constituency
- max\_long maximum longitude in the constituency
- srid\_n
- min\_e
- centre\_e
- max\_e
- min\_n
- centre\_n
- max\_n

---

 getHansard

*Search Hansard*


---

### **Description**

This needs much more documentation.

### **Usage**

```
getHansard(search = NULL, person = NULL, order = c("d", "r"),
           page = NULL, num = NULL)
```

### **Arguments**

search	A search string
person	A person identifier
order	whether to order results by date or relevance. Defaults to date
page	which page of results to provide. Defaults to first page
num	Number of results to return

### **Value**

Search results



---

getLord	<i>Get information about a Lord</i>
---------	-------------------------------------

---

**Description**

Get information about a Lord

**Usage**

```
getLord(id)
```

**Arguments**

id	Person identifier
----	-------------------

**Value**

a data.frame with columns

- member\_id Member identifier
- house Which chamber 1: The House of Commons, 2: The House of Lords.
- constituency Constituency represented
- party Party
- entered\_house Date when entered the House of Lords
- left\_house Date when left the House of Lords or 9999-12-31 if still in office
- entered\_reason Why they entered
- left\_reason Why they left
- person\_id Person identifier
- lastupdate Date TheyWorkForYou updated this information
- title Title, e.g. 'Baroness'
- given\_name First names
- family\_name Family name
- lordofname Location associated with title
- full\_name First names then family name

---

getLords	<i>Get information about Lords</i>
----------	------------------------------------

---

**Description**

Get information about Lords

**Usage**

```
getLords(date = NULL, party = NULL, search = NULL)
```

**Arguments**

date	ISO-style date, e.g. "1990-01-02", to compile a list of Lords for
party	Restrict list to Lords in this party
search	A search term

**Value**

A data.frame with columns

- member\_id Member identifier, dependent on position
- person\_id Person identifier
- name Name of Lord
- party Party
- office NULL if no offices held, or a list with one element: a data.frame with column headings:
  - dept The committee or commission name
  - position Job role, e.g. Member or Leader,
  - from\_date Date tenure in office began
  - to\_date Date tenure in office ended, or 9999-12-31 if still in office

---

getMLA	<i>Get information on a Member of the Legislative Assembly (Northern Ireland)</i>
--------	---

---

**Description**

Get information on a Member of the Legislative Assembly (Northern Ireland)

**Usage**

```
getMLA(id = NULL, postcode = NULL, constituency = NULL,
        always_return = NULL)
```

**Arguments**

id	A MLA identifier
postcode	A postcode, which specifies a region represented by the MLA
constituency	Name of a constituency
always_return	whether to try to return an MP even if the seat is vacant or it before an election

**Value**

a data.frame with rows representing the MLA's spells in the Assembly and columns

- member\_id Member identifier for each spell in the Assembly
- house 1: House of Commons, 2: House of Lords, 3: Legislative Assembly (NI), 4: Scottish Parliament
- constituency Constituency represented
- party Party
- entered\_house Date MP entered Parliament
- left\_house Date MP left Parliament, or 9999-12-31 if still in place
- entered\_reason Reason MP entered, e.g. general\_election
- left\_reason Reason MP left, e.g. general\_election\_standing
- person\_id Person identifier
- lastupdate When TheyWorkForYou last updated this information
- title Title, if any
- given\_name First name
- family\_name Family name
- full\_name First name and family name
- url URL path relative to TheyWorkForYou's hostname
- image URL path to jpg relative to TheyWorkForYou's hostname
- image\_height Image height in pixels
- image\_width Image width in pixels

---

getMLAs	<i>Get information on Members of the Legislative Assembly (Northern Ireland)</i>
---------	--

---

**Description**

Get information on Members of the Legislative Assembly (Northern Ireland)

**Usage**

```
getMLAs(date = NULL, party = NULL, search = NULL)
```

**Arguments**

date	The date for which the MLAs are required
party	Restrict MLAs to those in a party
search	A search string

**Value**

A data.frame with columns

- member\_id Member identifier for each spell in the Legislative Assembly
- person\_id Person identifier
- name Full name of MLA
- party Party represented
- constituency Name of MLA's constituency

---

getMP

*Get information about a Member of Parliament*

---

**Description**

Returns

**Usage**

```
getMP(id = NULL, postcode = NULL, constituency = NULL,
      always_return = NULL)
```

**Arguments**

id	An MP identifier
postcode	A postcode, used to identify a constituency and thereby an MP
constituency	Name of a constituency
always_return	whether to try to return an MP even if the seat is vacant or it before an election

**Value**

a data.frame with rows representing the MP's spells in Parliament and columns

- member\_id Member identifier for each spell in parliament
- house 1: House of Commons, 2: House of Lords, 3: Legislative Assembly (NI), 4: Scottish Parliament
- constituency Constituency represented
- party Party
- entered\_house Date MP entered Parliament

- left\_house Date MP left Parliament
- entered\_reason Reason MP entered, e.g. general\_election
- left\_reason Reason MP left, e.g. general\_election\_standing
- person\_id Person identifier
- lastupdate When TheyWorkForYou last updated this information
- title Title, if any
- given\_name First name
- family\_name Family name
- full\_name First name and family name
- url URL path relative to TheyWorkForYou's hostname
- image URL path to jpg relative to TheyWorkForYou's hostname
- image\_height Image height in pixels
- image\_width Image width in pixels

---

getMPInfo

*Get more information about a Member of Parliament*


---

### Description

Unless you want all the four hundred or so fields available about an MP you should probably specify the ones you want in a comma-separated string to `fields`.

### Usage

```
getMPInfo(id, fields = NULL)
```

### Arguments

<code>id</code>	An MP identifier
<code>fields</code>	A comma separated character vector of field names

### Details

Note that unlike other functions in this package the result is a list

### Value

a list

---

`getMPs`*Get information about Members of Parliament*

---

**Description**

Get information about Members of Parliament

**Usage**

```
getMPs(date = NULL, party = NULL, search = NULL)
```

**Arguments**

<code>date</code>	Date for which the MP list is constructed
<code>party</code>	Restrict to MPs in this party
<code>search</code>	A search string

**Value**

a `data.frame` with columns

- `member_id` Member identifier
- `person_id` Person identifier
- `name` MP's first names then family name
- `party` Party
- `constituency` Constituency represented
- `office` NULL if no offices held, or a list with one element: a `data.frame` with column headings:
  - `dept` The committee or commission name
  - `position` Role, e.g. Member,
  - `from_date` Date tenure in office began
  - `to_date` Date tenure in office ended, or 9999-12-31 if still in office

or an empty list if parliament is dissolved.

---

`getMPsInfo`*Get more information about a Member of Parliament*

---

**Description**

Get more information about a Member of Parliament

**Usage**

```
getMPsInfo(ids, fields = NULL)
```

**Arguments**

<code>ids</code>	A character vector of comma-separated MP identifiers
<code>fields</code>	A comma separated character vector of field names

**Value**

A list of list of MP information

---

`getMSP`*Get information on a Member of the Scottish Parliament*

---

**Description**

Get information on a Member of the Scottish Parliament

**Usage**

```
getMSP(id = NULL, postcode = NULL, constituency = NULL,  
       always_return = NULL)
```

**Arguments**

<code>id</code>	Person identifier
<code>postcode</code>	A postcode, which specifies the constituency whose MSP is required
<code>constituency</code>	Name of a constituency
<code>always_return</code>	whether to try to return an MP even if the seat is vacant or it before an election

**Value**

a data.frame with rows representing the MP's spells in Parliament and columns

- member\_id Member identifier for each spell in parliament
- house 1: House of Commons, 2: House of Lords, 3: Legislative Assembly (NI), 4: Scottish Parliament
- constituency Constituency represented
- party Party
- entered\_house Date MP entered Parliament
- left\_house Date MP left Parliament, or 9999-12-31 if still in place
- entered\_reason Reason MP entered, e.g. general\_election
- left\_reason Reason MP left, e.g. general\_election\_standing
- person\_id Person identifier
- lastupdate When TheyWorkForYou last updated this information
- title Title, if any
- given\_name First name
- family\_name Family name
- full\_name First name and family name
- url URL path relative to TheyWorkForYou's hostname
- image URL path to jpg relative to TheyWorkForYou's hostname
- image\_height Image height in pixels
- image\_width Image width in pixels

---

getMSPs

*Get information about Members of the Scottish Parliament*

---

**Description**

Get information about Members of the Scottish Parliament

**Usage**

```
getMSPs(date = NULL, party = NULL, search = NULL)
```

**Arguments**

date	The date for which the MSPs are required
party	Restrict MSPs to those in a party
search	A search string



**Value**

A data.frame with columns

- member\_id Member identifier for each spell in the Scottish parliament
- person\_id Person identifier
- name Full name of MSP
- party Party represented
- constituency Name of MSP's constituency

---

getPerson

*Get information about a person*

---

**Description**

Get information about a person

**Usage**

```
getPerson(id)
```

**Arguments**

id                    A person identifier

**Value**

a data.frame with rows representing the person's spells in the whatever legislative body they are members of, with columns

- member\_id Member identifier for each spell in the Assembly
- house 1: House of Commons, 2: House of Lords, 3: Legislative Assembly (NI), 4: Scottish Parliament
- constituency Constituency represented
- party Party (in that spell)
- entered\_house Date MP entered Parliament
- left\_house Date MP left Parliament, or 9999-12-31 if still in place
- entered\_reason Reason MP entered, e.g. general\_election
- left\_reason Reason MP left, e.g. general\_election\_standing
- person\_id Person identifier
- lastupdate When TheyWorkForYou last updated this information
- title Title, if any
- given\_name First name
- family\_name Family name
- full\_name First name and family name

---

getWMS	<i>Get written ministerial responses</i>
--------	--

---

**Description**

Get written ministerial responses

**Usage**

```
getWMS(date = NULL, search = NULL, person = NULL, gid = NULL,
        order = c("d", "r"), page = NULL, num = NULL)
```

**Arguments**

date	Date for which responses are required
search	A search string
person	A person identifier to specify which minister provided the answers
gid	A response identifier to return
order	whether to order results by date or relevance. Defaults to date
page	which page of results to provide. Defaults to first page
num	Number of results to return

**Value**

A list of information about written ministerial responses

---

getWrans	<i>Get written answers to questions</i>
----------	---

---

**Description**

The output of this function needs documentation.

**Usage**

```
getWrans(date = NULL, search = NULL, person = NULL, gid = NULL,
          order = c("d", "r"), page = NULL, num = NULL)
```

**Arguments**

date	Date for which answers are required
search	A search string
person	A person identifier to specify who provided the answers
gid	A written question and answer identifier to return
order	whether to order results by date or relevance. Defaults to date
page	which page of results to provide. Defaults to first page
num	Number of results to return

**Value**

A data.frame with two columns

- entry a data.frame
- subs a data.frame

. The entry data.frame has columns

- ebject\_id
- htype
- gid
- hpos
- section\_id
- subsection\_id
- hdate
- htime
- source\_url
- major
- minor
- video\_status
- colnum
- body

. subs is a list containing data.frames with all the columns in entry and in addition

- excerpt
- listurl
- commentsurl
- totalcomments
- comment

---

get_api_key	<i>Get the API key</i>
-------------	------------------------

---

**Description**

Get the current API key or request it if not present. When you provide a key it will be stored as a local environment variable `TWFY_API_KEY` and also in your `~/.Renvi ron`, making it available for all subsequent R sessions.

**Usage**

```
get_api_key()
```

**Details**

API keys can be requested at <http://theyworkforyou.com/api/key>.

**Value**

The current API key

---

set_api_key	<i>Set the API key</i>
-------------	------------------------

---

**Description**

This function manually sets or resets the API key. It is mostly useful for updating your key as all API function calls will ask for a key and store it locally and in `~/.Renvi ron` any time they do not find one.

**Usage**

```
set_api_key(api_key)
```

**Arguments**

api_key	a new API key
---------	---------------

**Value**

Nothing

---

twfy

*twfy: A package for accessing TheyWorkForYou*

---

### **Description**

twfy is an R wrapper around the API of TheyWorkForYou, a parliamentary monitoring site that scrapes and repackages Hansard (the UK's parliamentary record) and augments it with information from the Register of Members' Interests, election results, and voting records to provide a unified source of information about UK legislators and their activities. <<http://www.theyworkforyou.com>> for details.

### **Details**

twfy functions currently mirror the structure and naming scheme of the API as documented at <http://theyworkforyou.com/api/>. While the package attempts where possible to provide more detail on structure of the API's return values, this is not guaranteed to be correct and the link above should be your primary source of documentation.

# Index

[call\\_api](#), 2  
[convertURL](#), 3

[get\\_api\\_key](#), 20  
[getBoundary](#), 3  
[getComments](#), 4  
[getCommittee](#), 5  
[getConstituencies](#), 5  
[getConstituency](#), 6  
[getDebates](#), 6  
[getGeometry](#), 7  
[getHansard](#), 8  
[getLord](#), 9  
[getLords](#), 10  
[getMLA](#), 10  
[getMLAs](#), 11  
[getMP](#), 12  
[getMPInfo](#), 13  
[getMPs](#), 14  
[getMPsInfo](#), 15  
[getMSP](#), 15  
[getMSPs](#), 16  
[getPerson](#), 17  
[getWMS](#), 18  
[getWrans](#), 18

[set\\_api\\_key](#), 20

[twfy](#), 21  
[twfy-package \(twfy\)](#), 21