Package 'ausPH'

November 16, 2026

Title Retrieve Data from the Parliamentary Handbook of the Commonwealth of Australia

Version 1.1.1

Description Retrieves data from the Australian Parliamentary Library's Parliamentary Handbook <https://handbook.aph.gov.au/> and returns it as tidy tables: members and senators, their parliamentary and party service, ministries, shadow ministries, presiding officers, committees, parties, electorates,
parliaments, sitting dates, elections and two candidate preferred results. Known errors in the Handbook are flagged, and can optionally be corrected from a documented errata table.

License MIT + file LICENSE

URL https://github.com/palesl/ausPH

BugReports https://github.com/palesl/ausPH/issues

Encoding UTF-8

Language en-AU

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

Imports rjson, dplyr, readr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Pat Leslie [aut, cre] (ORCID: <https://orcid.org/0000-0002-9218-8981>)

Maintainer Pat Leslie <patrick.leslie@anu.edu.au>

Repository CRAN

Date/Publication 2026-11-16 08:50:43 UTC

Contents

get2CP . . . 2
getCommitteeMembers . . . 3
getElections . . . 3
getElectorates . . . 4
getErrata . . . 5
getIndividuals . . . 5
getMinisters . . . 6
getMinistries . . . 6
getOccupations . . . 7
getParliaments . . . 7
getParlService . . . 8
getParties . . . 9
getPartyService . . . 9
getPresidingOfficers . . . 10
getQualifications . . . 11
getSchools . . . 12
getShadowMinisters . . . 12
getSittingDates . . . 13

Index **14**

`get2CP` *Data on two candidate preferred results since 1919*

Description

Results come from a copy of the Handbook stored in the package, supplemented with results the Handbook is missing (documented in the `Note` and `source` columns). A supplementary result is only used where the Handbook has none for that division and year. By default, results for any elections the Handbook has listed since the stored copy was made are added. The date of the stored copy is in the `scraped` attribute of the result, and the date of any update in the `updated` attribute.

Usage

```
get2CP(include_byelections = FALSE, update = TRUE)
```

Arguments

`include_byelections`
: Choose to include house by-elections. If include_byelections = TRUE, by-election results are added.

`update`
: if TRUE (the default), add results for elections listed in the Handbook since the stored copy was made (three requests, plus one per new election); if FALSE, use the stored copy only.

Value

a tibble. `pID` links a candidate's party to `getParties()` where the Handbook's party name matches a party exactly; otherwise it is NA.

Examples

```
get2CP(update=FALSE)
```

`getCommitteeMembers` *Data on committee memberships since 1902*

Description

Data on committee memberships since 1902

Usage

```
getCommitteeMembers()
```

Value

a tibble. `end_date` is NA for current memberships.

Examples

```
getCommitteeMembers()
```

`getElections` *Data on elections since 1901*

Description

All federal general elections, House by-elections and supplementary elections, excluding elections for casual vacancies in the Senate. The package stores a copy reconciled with the Handbook; by default, any elections the Handbook has listed since are added to it. The date of the stored copy is in the `scraped` attribute of the result, and the date of any update in the `updated` attribute.

Usage

```
getElections(
  chamber = c("all", "house", "senate"),
  byelections = FALSE,
  apply_errata = FALSE,
  update = TRUE
)
```

Arguments

chamber	Select from c("all","house","senate")
byelections	Choose to include house by-elections. If byelections = TRUE, then a list of elections with by-elections and their divisions is returned.
apply_errata	if TRUE, replace values listed in getErrata() with their corrected values. By default the Handbook's values are returned.
update	if TRUE (the default), add elections listed in the Handbook since the stored copy was made (two requests); if FALSE, use the stored copy only.

Value

a tibble. Id is the package's election identifier (used as ElectionID in get2CP()); HandbookId is the Handbook's own identifier, NA for elections the Handbook does not list. For elections added by an update, FullSenateElection is NA, because the Handbook's list of elections does not say whether an election was a double dissolution. handbook_flag marks elections not_in_handbook and issues listed in getErrata().

Examples

```
getElections(chamber="all", update=FALSE)
```

---

getElectorates *Data on electorates since 1901*

Description

Data on electorates since 1901

Usage

```
getElectorates()
```

Value

a tibble

Examples

```
getElectorates()
```

getErrata *Known errors in the Parliamentary Handbook*

Description

A table of values the package has found to be wrong in the Parliamentary Handbook, with the corrected value and the evidence for it. By default the get* functions return the Handbook's values and only mark these rows in their `handbook_flag` column; with `apply_errata = TRUE` they replace the value with `corrected_value`.

Usage

```
getErrata()
```

Details

An erratum applies only while the Handbook still has `handbook_value`, so once the Handbook is fixed the erratum stops having any effect.

Value

a tibble with one row per correction: the table and row it applies to (`table`, `key`), the `field`, the `handbook_value`, the `corrected_value`, a short `issue` code, the `evidence` and its `source`.

Examples

```
getErrata()
```

getIndividuals *Data on individuals and biographical information since 1901*

Description

Data on individuals and biographical information since 1901

Usage

```
getIndividuals()
```

Value

a tibble

Examples

```
getIndividuals()
```

getMinisters *Data on Ministers since 1901*

Description

Data on Ministers since 1901

Usage

```
getMinisters()
```

Value

a tibble with one row per ministerial role. `RDateStart` and `RDateEnd` are the dates of the role; `MDateStart` and `MDateEnd` are the dates of the ministry.

Examples

```
getMinisters()
```

getMinistries *Data on Ministries and Governing Parties since 1901*

Description

Data on Ministries and Governing Parties since 1901

Usage

```
getMinistries()
```

Value

a tibble with one row per ministry and governing party. Ministries for which the Handbook lists no party (e.g. the first Barton Ministry) are kept, with `Party` and `pID` set to NA.

Examples

```
getMinistries()
```

getOccupations *Data on all listed prior occupations held (both primary and secondary)*

Description

Data on all listed prior occupations held (both primary and secondary)

Usage

```
getOccupations()
```

Value

a tibble with one row per person and occupation

Examples

```
getOccupations()
```

getParliaments *Data on Parliamentary terms since 1901*

Description

Data on Parliamentary terms since 1901

Usage

```
getParliaments()
```

Value

a tibble

Examples

```
getParliaments()
```

`getParlService` *Data on parliamentary tenures of MPs and senators since 1901*

Description

By default the whole table is retrieved from the Handbook (one request). If the Handbook cannot be reached, or `update = FALSE`, the copy stored in the package is returned instead. The date the data was retrieved is in the `scraped` attribute of the result.

Usage

```
getParlService(
  chamber = c("all", "house", "senate"),
  apply_errata = FALSE,
  update = TRUE
)
```

Arguments

`chamber`	choose to include MPs ("house"), senators ("senate") or both ("all").
`apply_errata`	if TRUE, replace values listed in `getErrata()` with their corrected values. By default the Handbook's values are returned.
`update`	if TRUE (the default), retrieve the current data from the Handbook; if FALSE, use the stored copy.

Value

a tibble. `DateEnd` is NA for terms that have not ended. `handbook_flag` marks rows with a known problem in the Handbook: `overlapping_term` (the term starts before the member's previous term ends), `placeholder_end_date` (a term that is not the member's latest has no end date in the Handbook) and any issue listed in `getErrata()`.

Examples

```
getParlService(chamber="house")

# the copy stored in the package, without querying the Handbook
getParlService(chamber="house", update=FALSE)
```

getParties *Data on parties since 1901*

Description

Data on parties since 1901

Usage

```
getParties()
```

Value

a tibble. Note that the Handbook gives a party a new `pID` when it changes its name, and that abbreviations are not unique (three parties use "LIB"), so `Abbrev` should not be used as a key.

Examples

```
getParties()
```

getPartyService *Data on party service of MPs and senators since 1901*

Description

Returns the copy of the Handbook's party service records stored in the package. The Handbook has no single source for these records (each person's must be retrieved separately), so updating takes a few minutes and is optional: with `update = TRUE`, the records of everyone who has joined or served in parliament since the stored copy was taken are retrieved again. The date of the stored copy is in the `scraped` attribute of the result, and the date of any update in the `updated` attribute.

Usage

```
getPartyService(apply_errata = FALSE, update = FALSE)
```

Arguments

`apply_errata`	if TRUE, replace values listed in `getErrata()` with their corrected values. By default the Handbook's values are returned. No party service errata are listed yet, so this currently has no effect.
`update`	if TRUE, update the stored copy from the Handbook (about 250 requests, two to three minutes). Defaults to FALSE.

Value

a tibble. `handbook_flag` marks rows with a known problem in the Handbook: `same_day_party_record` (a one-day record that starts on the same day as another party record for the same person), `placeholder_end_date` (the Handbook has 1900-01-01, returned as NA) and any issue listed in `getErrata()`.

Examples

```
getPartyService()
```

`getPresidingOfficers` *Data on presiding officers and their deputies since 1901*

Description

Speakers of the House of Representatives, Presidents of the Senate, their deputies, Chairs of Committees, temporary chairs, members of the Speaker's panel and acting holders of these offices, with the dates each person held the office, from the Handbook's records of parliamentary positions.

Usage

```
getPresidingOfficers(
  chamber = c("all", "house", "senate"),
  office = NULL,
  apply_errata = FALSE,
  update = TRUE
)
```

Arguments

`chamber`	choose to include House offices ("house"), Senate offices ("senate") or both ("all").
`office`	the offices to include, as they are named in the Handbook's `Office` column (e.g. "Speaker of the House of Representatives", "President of the Senate"). The default, NULL, includes every office.
`apply_errata`	if TRUE, replace values listed in `getErrata()` with their corrected values. By default the Handbook's values are returned.
`update`	if TRUE (the default), retrieve the current data from the Handbook; if FALSE, use the stored copy.

Details

By default the whole table is retrieved from the Handbook (one request, plus one for `getParlService()`, used to place each office holder in a chamber). If the Handbook cannot be reached, or `update = FALSE`, the copy stored in the package is returned instead. The date the data was retrieved is in the `scraped` attribute of the result.

The Handbook records an office as held from one parliament to the next, so a Speaker or President re-elected after a general election has one row per parliament, with a gap between them.

Value

a tibble with one row per person and term in an office: `PHID`, `DisplayName`, `Office`, `Chamber` ("House" or "Senate"), `DateStart`, `DateEnd` (NA while the office is held), the Handbook's `Note` and `handbook_flag`. `Chamber` is the chamber the person sat in when the term began, since some office names (e.g. "Chairman of Committees") were used in both chambers. `handbook_flag` marks rows with a known problem in the Handbook: `placeholder_start_date` (no start date), `placeholder_end_date` (no end date, though the person has left parliament), `end_before_start`, `outside_parliamentary_service` (the term starts when the person was not in parliament, according to `getParlService()`) and any issue listed in `getErrata()`.

Examples

```
# Speakers, from the copy stored in the package
getPresidingOfficers(chamber = "house",
                     office = "Speaker of the House of Representatives",
                     update = FALSE)

getPresidingOfficers()
```

---

getQualifications *Data on all listed Qualifications*

Description

Data on all listed Qualifications

Usage

```
getQualifications()
```

Value

a tibble with one row per person and qualification

Examples

```
getQualifications()
```

getSchools *Data on all listed Secondary Schools*

Description

Data on all listed Secondary Schools

Usage

```
getSchools()
```

Value

a tibble with one row per person and school, as listed in the Handbook (some entries name more than one school)

Examples

```
getSchools()
```

getShadowMinisters *Data on Shadow Ministers since 1901*

Description

Data on Shadow Ministers since 1901

Usage

```
getShadowMinisters()
```

Value

a tibble with one row per shadow ministerial role. `RDateStart` and `RDateEnd` are the dates of the role; `MDateStart` and `MDateEnd` are the dates of the shadow ministry.

Examples

```
getShadowMinisters()
```

`getSittingDates` *Data on sitting dates in the federal parliament*

Description

Dates up to the end of the stored copy come from the package; by default, later years are retrieved from the Handbook (one request per year). The date of the stored copy is in the `scraped` attribute of the result, and the date of any update in the `updated` attribute.

Usage

```
getSittingDates(apply_errata = FALSE, update = TRUE)
```

Arguments

`apply_errata`	if TRUE, replace values listed in `getErrata()` with their corrected values. By default the Handbook's values are returned.
`update`	if TRUE (the default), add sitting dates after the stored copy from the Handbook; if FALSE, use the stored copy only.

Value

a tibble with one row per sitting date and whether each chamber sat. `handbook_flag` marks dates listed in `getErrata()`, e.g. `chamber_did_not_sit` or `continuation_of_previous_sitting`.

Examples

```
getSittingDates()
```

Index

get2CP, 2
get2CP(), *4*
getCommitteeMembers, 3
getElections, 3
getElectorates, 4
getErrata, 5
getErrata(), *4*, *8–11*, *13*
getIndividuals, 5
getMinisters, 6
getMinistries, 6
getOccupations, 7
getParliaments, 7
getParlService, 8
getParlService(), *10, 11*
getParties, 9
getParties(), *3*
getPartyService, 9
getPresidingOfficers, 10
getQualifications, 11
getSchools, 12
getShadowMinisters, 12
getSittingDates, 13