

Package ‘bunddev’

May 8, 2026

Title Discover and Call 'Bund.dev' APIs

Version 0.2.3

Depends R (>= 4.1.0)

Description Provides a registry of APIs listed on <<https://bund.dev>> and a core 'OpenAPI' client layer to explore specs and perform requests. Adapter helpers return tidy data frames for supported APIs, with optional response caching and rate limiting guidance.

License MIT + file LICENSE

URL <https://buecker.ms/bunddev/>, <https://github.com/mchlbckr/bunddev>

BugReports <https://github.com/mchlbckr/bunddev/issues>

Encoding UTF-8

RoxygenNote 7.3.3

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Imports cli, dplyr, httr2, jsonlite, purrr, rlang, stringr, tibble, tidy, tools, xml2, yaml

VignetteBuilder knitr, rmarkdown

NeedsCompilation no

Author Michael Buecker [aut, cre] (ORCID: <<https://orcid.org/0000-0003-0045-8460>>)

Maintainer Michael Buecker <michael.buecker@fh-muenster.de>

Repository CRAN

Date/Publication 2026-04-22 07:30:02 UTC

Contents

abfallnavi_fraktionen	6
abfallnavi_fraktionen_hausnummern	7
abfallnavi_fraktionen_strassen	8

abfallnavi_hausnummern	9
abfallnavi_kalender_download	10
abfallnavi_ort	11
abfallnavi_orte	12
abfallnavi_strassen	13
abfallnavi_termine_hausnummern	14
abfallnavi_termine_strassen	15
ausbildungssuche_details	16
ausbildungssuche_search	17
autobahn_charging_stations	19
autobahn_charging_station_details	21
autobahn_closures	22
autobahn_closure_details	23
autobahn_parking_lorries	24
autobahn_parking_lorry_details	26
autobahn_roads	27
autobahn_roadworks	28
autobahn_roadwork_details	29
autobahn_warnings	30
autobahn_warning_details	32
autobahn_webcams	33
autobahn_webcam_details	35
bewerberboerse_details	36
bewerberboerse_search	38
bunddev_auth_get	40
bunddev_auth_set	40
bunddev_cache_dir	42
bunddev_call	43
bunddev_call_tidy	45
bunddev_endpoints	46
bunddev_info	47
bunddev_list	48
bunddev_ms_to_posix	49
bunddev_parameters	49
bunddev_parameters_for	51
bunddev_parameter_values	52
bunddev_rate_limit_get	53
bunddev_rate_limit_set	54
bunddev_registry	55
bunddev_search	56
bunddev_spec	56
bunddev_spec_path	57
bunddev_timestamp_to_ms	58
bundeshaushalt_budget_data	59
bundesrat_aktuelles	61
bundesrat_mitglieder	62
bundesrat_plenum_aktuelle_sitzung	63
bundesrat_plenum_chronologisch	64

bundesrat_plenum_kompakt	65
bundesrat_plenum_naechste_sitzungen	66
bundesrat_praesidium	67
bundesrat_startlist	68
bundesrat_stimmverteilung	69
bundesrat_termine	70
bundestag_article	71
bundestag_ausschuesse	72
bundestag_ausschuss	73
bundestag_conferences	74
bundestag_lobbyregister_search	75
bundestag_mdb_bio	76
bundestag_mdb_index	77
bundestag_speaker	78
bundestag_video_feed	79
dashboard_deutschland_geo	80
dashboard_deutschland_get	82
dashboard_deutschland_indicators	84
ddb_institutions	85
ddb_institution_sectors	86
ddb_search	87
destatis_catalogue_cubes	89
destatis_catalogue_tables	90
destatis_data_cube	92
destatis_data_table	93
deutschlandatlas_query	95
diga_catalog_entries	96
diga_charge_item_definitions	98
diga_device_definitions	99
diga_organizations	100
diga_questionnaires	102
diga_questionnaire_responses	103
dip_bundestag_aktivitaet	105
dip_bundestag_aktivitaet_list	106
dip_bundestag_drucksache	108
dip_bundestag_drucksache_list	110
dip_bundestag_drucksache_text	112
dip_bundestag_drucksache_text_list	113
dip_bundestag_person	115
dip_bundestag_person_list	116
dip_bundestag_plenarprotokoll	118
dip_bundestag_plenarprotokoll_list	119
dip_bundestag_plenarprotokoll_text	121
dip_bundestag_plenarprotokoll_text_list	122
dip_bundestag_vorgang	124
dip_bundestag_vorgangsposition	126
dip_bundestag_vorgangsposition_list	128
dip_bundestag_vorgang_list	130

dwd_alpine_forecast_text	132
dwd_avalanche_warnings	133
dwd_coast_warnings	134
dwd_crowd_reports	135
dwd_municipality_warnings	137
dwd_sea_warning_text	138
dwd_station_overview	139
dwd_warnings_nowcast	140
eco_visio_counters	142
eco_visio_data	143
feiertage_list	144
handelsregister_search	146
hilfsmittel_nachweisschema	147
hilfsmittel_produk t	149
hilfsmittel_produk t art	150
hilfsmittel_produk te	152
hilfsmittel_produk t gruppe	154
hilfsmittel_tree	155
hilfsmittel_untergruppe	156
hochwasserzentralen_bundeslaender	158
hochwasserzentralen_bundesland_geojson	159
hochwasserzentralen_bundesland_info	160
hochwasserzentralen_lagepegel	161
hochwasserzentralen_pegel_info	162
jobsuche_logo	163
jobsuche_search	164
jobsuche_search_app	167
linikatlas_german_places	169
linikatlas_german_states	170
linikatlas_hospital_detail	171
linikatlas_icd_codes	172
linikatlas_locations	173
linikatlas_ops_codes	174
linikatlas_search	175
linikatlas_states	176
ladestationen_query	177
lebensmittelwarnung_warnings	179
luftqualitaet_airquality	180
luftqualitaet_airquality_limits	182
luftqualitaet_annualbalances	183
luftqualitaet_components	184
luftqualitaet_measures	185
luftqualitaet_measures_limits	186
luftqualitaet_meta	187
luftqualitaet_networks	188
luftqualitaet_scopes	189
luftqualitaet_stationsettings	190
luftqualitaet_stationtypes	191

luftqualitaet_thresholds	192
luftqualitaet_transgressions	193
luftqualitaet_transgressiontypes	194
marktstammdaten_filters_gaserzeugung	195
marktstammdaten_filters_gasverbrauch	196
marktstammdaten_filters_stromerzeugung	197
marktstammdaten_filters_stromverbrauch	199
marktstammdaten_gaserzeugung	200
marktstammdaten_gasverbrauch	201
marktstammdaten_stromerzeugung	203
marktstammdaten_stromverbrauch	204
mudab_parameters	206
mudab_parameters_biologie	207
mudab_parameters_biota	209
mudab_parameters_sediment	210
mudab_parameters_wasser	211
mudab_parameter_values	213
mudab_plc_measurements	214
mudab_plc_parameters	216
mudab_plc_stations	217
mudab_project_stations	218
mudab_stations	220
nina_archive_mowas	221
nina_archive_mowas_mapping	222
nina_covid_infos	222
nina_covid_map	223
nina_covid_rules	224
nina_covid_ticker	225
nina_covid_ticker_message	225
nina_dashboard	226
nina_event_code	227
nina_event_codes	228
nina_faqs	228
nina_logo	229
nina_logos	230
nina_mapdata	231
nina_mowas_rss	232
nina_notfalltipps	233
nina_version	233
nina_warning	234
nina_warnings	235
nina_warning_geojson	236
nina_warning_json	237
pegel_online_measurements	238
pegel_online_measurements_plot	240
pegel_online_station	241
pegel_online_stations	243
pegel_online_timeseries	244

pegel_online_waters	246
psm_anwendungen	248
psm_kultur_gruppen	250
psm_mittel	251
psm_schadorg_gruppen	252
psm_stand	253
psm_wirkstoffe	254
regionalatlas_query	255
smard_indices	257
smard_table	259
smard_timeseries	260
tagesschau_channels	261
tagesschau_homepage	262
tagesschau_news	264
tagesschau_search	265
travelwarning_healthcare	266
travelwarning_representatives_country	267
travelwarning_representatives_germany	269
travelwarning_state_names	270
travelwarning_warning	271
travelwarning_warnings	273
Index	275

abfallnavi_fraktionen *List waste fractions*

Description

List waste fractions

Usage

```
abfallnavi_fraktionen(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per waste fraction and columns:

- id** Integer. Fraction identifier.
- name** Character. Fraction name (e.g. "Restabfall").
- iconNr** Integer. Icon number for display.
- farbeRgb** Character. RGB colour hex code.

See Also

Other Abfallnavi: [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

abfallnavi_fraktionen_hausnummern

List waste fractions for a house number

Description

List waste fractions for a house number

Usage

```
abfallnavi_fraktionen_hausnummern(hausnummern_id, safe = TRUE, refresh = FALSE)
```

Arguments

- hausnummern_id House number id.
- safe Logical; if TRUE (default), apply rate-limiting and cache GET responses to `tools::R_user_dir("bundle", "cache")`.
- refresh Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per waste fraction and columns:

- id** Integer. Fraction identifier.
- name** Character. Fraction name (e.g. "Restabfall").
- iconNr** Integer. Icon number for display.
- farbeRgb** Character. RGB colour hex code.

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

abfallnavi_fraktionen_strassen
List waste fractions for a street

Description

List waste fractions for a street

Usage

```
abfallnavi_fraktionen_strassen(strassen_id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>strassen_id</code>	Street id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per waste fraction and columns:

id Integer. Fraction identifier.

name Character. Fraction name (e.g. "Restabfall").

iconNr Integer. Icon number for display.

farbeRgb Character. RGB colour hex code.

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

abfallnavi_hausnummern

List house numbers for a street

Description

List house numbers for a street

Usage

```
abfallnavi_hausnummern(strassen_id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>strassen_id</code>	Street id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per house number and columns:

id Integer. House number identifier.

nr Character. House number label.

plz Character. Postal code.

staticId Character. Static identifier (base64-encoded).

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

abfallnavi_kalender_download

Download calendar file

Description

Download calendar file

Usage

```
abfallnavi_kalender_download(  
  region,  
  format,  
  jahr,  
  ort,  
  strasse,  
  hnr,  
  fraktion,  
  safe = TRUE,  
  refresh = FALSE  
)
```

Arguments

region	Region code.
format	File format.
jahr	Year.
ort	Place name.
strasse	Street id.
hnr	House number id.
fraktion	Fraction ids.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Downloads a calendar file for the requested address and fraction.

Value

A one-row [tibble](#) with columns:

region Character. Region code passed to the request.

format Character. File format ("pdf" or "ics").

jahr Integer. Year passed to the request.

bytes List. Raw file bytes (raw vector).

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

abfallnavi_ort	<i>Get a place by id</i>
----------------	--------------------------

Description

Get a place by id

Usage

```
abfallnavi_ort(ort_id, safe = TRUE, refresh = FALSE)
```

Arguments

ort_id	Place id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row [tibble](#) with columns:

id Integer. Place identifier.

name Character. Place name.

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

abfallnavi_orte *List available places*

Description

List available places

Usage

```
abfallnavi_orte(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The Abfallnavi API provides waste collection data for supported regions. Start by listing places (Orte) to obtain an `ortId`. API documentation: <https://abfallnavi.api.bund.dev/>.

Value

A `tibble` with one row per place and columns:

id Integer. Place identifier.

name Character. Place name.

See Also

[abfallnavi_strassen\(\)](#) for streets in a place.

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

Examples

```
## Not run:  
abfallnavi_orte()  
  
## End(Not run)
```

abfallnavi_strassen *List streets for a place*

Description

List streets for a place

Usage

```
abfallnavi_strassen(ort_id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>ort_id</code>	Place id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per street and columns:

id Integer. Street identifier.
name Character. Street name.
staticId Character. Static identifier (base64-encoded).
hausNrList List. House numbers for this street.
plz Character. Postal code (may be NA).
ortsteilName Character. District name.
ort List. Nested place object.

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_termine_hausnummern\(\)](#), [abfallnavi_termine_strassen\(\)](#)

 abfallnavi_termine_hausnummern

List collection dates for a house number

Description

List collection dates for a house number

Usage

```
abfallnavi_termine_hausnummern(
  hausnummern_id,
  fraktion,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

hausnummern_id	House number id.
fraktion	Fraction ids.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per collection date and columns:

id Integer. Collection event identifier.

bezirk List. District information for this collection.

datum Character. Collection date (YYYY-MM-DD).

date_time POSIXct. Parsed collection date (Europe/Berlin).

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termine_strassen\(\)](#)

`abfallnavi_termin_e_strassen`*List collection dates for a street*

Description

List collection dates for a street

Usage

```
abfallnavi_termin_e_strassen(  
  strassen_id,  
  fraktion,  
  safe = TRUE,  
  refresh = FALSE  
)
```

Arguments

<code>strassen_id</code>	Street id.
<code>fraktion</code>	Fraction ids.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A [tibble](#) with one row per collection date and columns:

id Integer. Collection event identifier.

bezirk List. District information for this collection.

datum Character. Collection date (YYYY-MM-DD).

date_time POSIXct. Parsed collection date (Europe/Berlin).

See Also

Other Abfallnavi: [abfallnavi_fraktionen\(\)](#), [abfallnavi_fraktionen_hausnummern\(\)](#), [abfallnavi_fraktionen_strassen\(\)](#), [abfallnavi_hausnummern\(\)](#), [abfallnavi_kalender_download\(\)](#), [abfallnavi_ort\(\)](#), [abfallnavi_orte\(\)](#), [abfallnavi_strassen\(\)](#), [abfallnavi_termin_e_hausnummern\(\)](#)

`ausbildungssuche_details`*Get training offer details*

Description

Get training offer details

Usage

```
ausbildungssuche_details(offer_id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>offer_id</code>	Offer id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev_auth_set", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns detailed information for a single offer.

Value

A one-row tibble with:

offer_id Requested offer id.

data Raw detail payload for the offer (list-column).

See Also

[ausbildungssuche_search\(\)](#) to find offer ids.

Other Ausbildungssuche: [ausbildungssuche_search\(\)](#)

Examples

```
## Not run:
Sys.setenv(AUSBILDUNGSSUCHE_API_KEY = "infosysbub-absuche")
bunddev_auth_set("ausbildungssuche", type = "api_key", env_var = "AUSBILDUNGSSUCHE_API_KEY")
ausbildungssuche_details(12345)

## End(Not run)
```

 ausbildungssuche_search

Search training offers

Description

Search training offers

Usage

```
ausbildungssuche_search(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	<p>Named list of query parameters:</p> <ul style="list-style-type: none"> sty Training type filter (integer). ids Occupation id(s), comma-separated (integer/character). orte Location id(s), comma-separated (integer/character). page Page index, starting at 0 (integer). size Page size (integer, max defined by API). uk Radius filter, e.g. "25" or "Bundesweit" (character). re Region/Bundesland code (character). bart Training category filter (integer). ityp Integration type filter (integer). bt Start date/time window filter (integer code). ban Provider id(s), comma-separated (integer/character). bg Only entries with Bildungsgutschein support (TRUE/FALSE).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	<p>How to handle list columns when <code>flatten = TRUE</code>:</p> <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. <p>Warning: this can significantly increase the number of rows.</p>

Details

The Ausbildungssuche API provides training offer data from the Bundesagentur fuer Arbeit. Authentication is required via X-API-Key (clientId infosysbub-absuche, discoverable from <https://web.arbeitsagentur.de/weite>)
API documentation: <https://ausbildungssuche.api.bund.dev/>.

This adapter uses the X-API-Key header. Set it via `bunddev_auth_set()` and `AUSBILDUNGSSUCHE_API_KEY`.

Value

A tibble with one row per training offer:

id Offer id (integer).

unterrichtsform Teaching format metadata (list-column).

dauer Duration metadata (list-column).

anbieterbewertung Provider rating text (character).

angebot Offer metadata (list-column).

adresse Address metadata (list-column).

unterrichtszeiten Class schedule text (character).

kosten_wert Cost value (character).

kosten_waehrung Cost currency (character).

kosten_bemerkung Cost note (character).

foerderung Funding flag (TRUE/FALSE).

link Offer link (character).

bemerkung Additional note text (character).

beginn Start date text (character).

ende End date text (character).

individueller_einstieg Flexible start flag (TRUE/FALSE).

anmeldeschluss Registration deadline text (character).

bemerkung_zeit Timing note (character).

pruefende_stelle Examining body (character).

eigene_angebotsnummer Provider's own offer id (character).

teilnehmer_min Minimum participants (integer).

teilnehmer_max Maximum participants (integer).

aktualisierungsdatum Update timestamp in milliseconds (numeric).

aktualisierungsdatum_time Update timestamp as POSIXct in Europe/Berlin.

links Top-level links object (list-column).

aggregations Aggregation metadata (list-column).

page Paging metadata (list-column).

See Also

`bunddev_parameters()` to inspect available query parameters. `ausbildungssuche_details()` for a single offer and `bunddev_auth_set()` for authentication.

Other Ausbildungssuche: `ausbildungssuche_details()`

Examples

```
## Not run:
Sys.setenv(AUSBILDUNGSSUCHE_API_KEY = "infosysbub-absuche")
bunddev_auth_set("ausbildungssuche", type = "api_key", env_var = "AUSBILDUNGSSUCHE_API_KEY")
ausbildungssuche_search(params = list(size = 5))

## End(Not run)
```

autobahn_charging_stations

List Autobahn charging stations

Description

List Autobahn charging stations

Usage

```
autobahn_charging_stations(road_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>road_id</code>	Road identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

Returns charging stations for a specific Autobahn road id. API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per charging-station entry:

road_id Road identifier (character).
identifier Entry identifier (character).
title Title (character).
subtitle Subtitle (character).
display_type Display type (character).
icon Icon identifier (character).
is_blocked Whether the road is blocked (logical).
future Whether the entry is future-dated (logical).
start_timestamp Start timestamp string (character).
start_time Parsed start time (POSIXct).
point Point coordinate string (character).
extent Extent coordinate string (character).
coordinate_lat Latitude (character).
coordinate_long Longitude (character).
description Description text (character).
footer Footer text (character).
route_recommendation Route recommendations (list-column).
lorry_parking_feature_icons Lorry parking feature icons (list-column).

With `flatten = TRUE`, the two list-columns are transformed according to `flatten_mode`.

See Also

[autobahn_roads\(\)](#) and [autobahn_charging_station_details\(\)](#).

Other Autobahn: [autobahn_charging_station_details\(\)](#), [autobahn_closure_details\(\)](#), [autobahn_closures\(\)](#), [autobahn_parking_lorries\(\)](#), [autobahn_parking_lorry_details\(\)](#), [autobahn_roads\(\)](#), [autobahn_roadwork_details\(\)](#), [autobahn_roadworks\(\)](#), [autobahn_warning_details\(\)](#), [autobahn_warnings\(\)](#), [autobahn_webcam_details\(\)](#), [autobahn_webcams\(\)](#)

Examples

```
## Not run:
roads <- autobahn_roads()
autobahn_charging_stations(roads$road_id[[1]], flatten = TRUE)

## End(Not run)
```

`autobahn_charging_station_details`*Get Autobahn charging station details*

Description

Get Autobahn charging station details

Usage

```
autobahn_charging_station_details(  
  station_id,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

<code>station_id</code>	Charging station identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns full details for a single charging station entry. API documentation: <https://autobahn.api.bund.dev>.

Value

A one-row tibble for the selected charging station with the same columns as `autobahn_charging_stations()`. `road_id` is typically NA for detail calls.

See Also

`autobahn_charging_stations()` to list stations.

Other Autobahn: `autobahn_charging_stations()`, `autobahn_closure_details()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_parking_lorry_details()`, `autobahn_roads()`, `autobahn_roadwork_details()`, `autobahn_roadworks()`, `autobahn_warning_details()`, `autobahn_warnings()`, `autobahn_webcam_details()`, `autobahn_webcams()`

Examples

```
## Not run:
roads <- autobahn_roads()
stations <- autobahn_charging_stations(roads$road_id[[1]])
autobahn_charging_station_details(stations$identifier[[1]])

## End(Not run)
```

autobahn_closures *List Autobahn closures*

Description

List Autobahn closures

Usage

```
autobahn_closures(road_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

road_id	Road identifier.
flatten	Logical; if TRUE, simplify nested list columns according to flatten_mode. Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when flatten = TRUE: "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns current closures for a specific Autobahn road id. API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per closure entry:

road_id Road identifier (character).
identifier Entry identifier (character).
title Title (character).
subtitle Subtitle (character).
display_type Display type (character).

icon Icon identifier (character).

is_blocked Whether the road is blocked (logical).

future Whether the entry is future-dated (logical).

start_timestamp Start timestamp string (character).

start_time Parsed start time (POSIXct).

point Point coordinate string (character).

extent Extent coordinate string (character).

coordinate_lat Latitude (character).

coordinate_long Longitude (character).

description Description text (character).

footer Footer text (character).

route_recommendation Route recommendations (list-column).

lorry_parking_feature_icons Lorry parking feature icons (list-column).

With `flatten = TRUE`, the two list-columns are transformed according to `flatten_mode`.

See Also

[autobahn_roads\(\)](#) and [autobahn_closure_details\(\)](#).

Other Autobahn: [autobahn_charging_station_details\(\)](#), [autobahn_charging_stations\(\)](#), [autobahn_closure_details\(\)](#), [autobahn_parking_lorries\(\)](#), [autobahn_parking_lorry_details\(\)](#), [autobahn_roads\(\)](#), [autobahn_roadwork_details\(\)](#), [autobahn_roadworks\(\)](#), [autobahn_warning_details\(\)](#), [autobahn_warnings\(\)](#), [autobahn_webcam_details\(\)](#), [autobahn_webcams\(\)](#)

Examples

```
## Not run:
roads <- autobahn_roads()
autobahn_closures(roads$road_id[[1]], flatten = TRUE)

## End(Not run)
```

autobahn_closure_details

Get Autobahn closure details

Description

Get Autobahn closure details

Usage

```
autobahn_closure_details(closure_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>closure_id</code>	Closure identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns full details for a single closure entry. API documentation: <https://autobahn.api.bund.dev>.

Value

A one-row tibble for the selected closure with the same columns as `autobahn_closures()`. `road_id` is typically NA for detail calls.

See Also

`autobahn_closures()` to list closures.

Other Autobahn: `autobahn_charging_station_details()`, `autobahn_charging_stations()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_parking_lorry_details()`, `autobahn_roads()`, `autobahn_roadwork_details()`, `autobahn_roadworks()`, `autobahn_warning_details()`, `autobahn_warnings()`, `autobahn_webcam_details()`, `autobahn_webcams()`

Examples

```
## Not run:
roads <- autobahn_roads()
closures <- autobahn_closures(roads$road_id[[1]])
autobahn_closure_details(closures$identifier[[1]])

## End(Not run)
```

`autobahn_parking_lorries`

List Autobahn lorry parking areas

Description

List Autobahn lorry parking areas

Usage

```
autobahn_parking_lorries(road_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>road_id</code>	Road identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns lorry parking areas for a specific Autobahn road id. API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per lorry-parking entry:

road_id Road identifier (character).
identifier Entry identifier (character).
title Title (character).
subtitle Subtitle (character).
display_type Display type (character).
icon Icon identifier (character).
is_blocked Whether the road is blocked (logical).
future Whether the entry is future-dated (logical).
start_timestamp Start timestamp string (character).
start_time Parsed start time (POSIXct).
point Point coordinate string (character).
extent Extent coordinate string (character).
coordinate_lat Latitude (character).
coordinate_long Longitude (character).
description Description text (character).
footer Footer text (character).
route_recommendation Route recommendations (list-column).
lorry_parking_feature_icons Lorry parking feature icons (list-column).

With `flatten = TRUE`, the two list-columns are transformed according to `flatten_mode`.

See Also

[autobahn_roads\(\)](#) and [autobahn_parking_lorry_details\(\)](#).

Other Autobahn: [autobahn_charging_station_details\(\)](#), [autobahn_charging_stations\(\)](#), [autobahn_closure_details\(\)](#), [autobahn_closures\(\)](#), [autobahn_parking_lorry_details\(\)](#), [autobahn_roads\(\)](#), [autobahn_roadwork_details\(\)](#), [autobahn_roadworks\(\)](#), [autobahn_warning_details\(\)](#), [autobahn_warnings\(\)](#), [autobahn_webcam_details\(\)](#), [autobahn_webcams\(\)](#)

Examples

```
## Not run:
roads <- autobahn_roads()
autobahn_parking_lorries(roads$road_id[[1]], flatten = TRUE)

## End(Not run)
```

autobahn_parking_lorry_details

Get Autobahn lorry parking details

Description

Get Autobahn lorry parking details

Usage

```
autobahn_parking_lorry_details(
  lorry_id,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

lorry_id	Lorry parking identifier.
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via tidyr::unnest_longer(). Warning: this can significantly increase the number of rows.

Details

Returns full details for a single lorry parking entry. API documentation: <https://autobahn.api.bund.dev>.

Value

A one-row tibble for the selected lorry-parking entry with the same columns as `autobahn_parking_lorries()`. `road_id` is typically NA for detail calls.

See Also

`autobahn_parking_lorries()` to list parking areas.

Other Autobahn: `autobahn_charging_station_details()`, `autobahn_charging_stations()`, `autobahn_closure_details()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_roads()`, `autobahn_roadwork_details()`, `autobahn_roadworks()`, `autobahn_warning_details()`, `autobahn_warnings()`, `autobahn_webcam_details()`, `autobahn_webcams()`

Examples

```
## Not run:
roads <- autobahn_roads()
parking <- autobahn_parking_lorries(roads$road_id[[1]])
autobahn_parking_lorry_details(parking$identifier[[1]])

## End(Not run)
```

autobahn_roads	<i>List Autobahn road ids</i>
----------------	-------------------------------

Description

List Autobahn road ids

Usage

```
autobahn_roads()
```

Details

Lists Autobahn road ids from the Autobahn App API (Autobahn GmbH). API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per Autobahn id:

road_id Autobahn identifier (character).

See Also

`autobahn_roadworks()` and `autobahn_warnings()` for road-specific data.

Other Autobahn: `autobahn_charging_station_details()`, `autobahn_charging_stations()`, `autobahn_closure_details()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_parking_lorry_details()`, `autobahn_roadwork_details()`, `autobahn_roadworks()`, `autobahn_warning_details()`, `autobahn_warnings()`, `autobahn_webcam_details()`, `autobahn_webcams()`

Examples

```
## Not run:
autobahn_roadworks()

## End(Not run)
```

autobahn_roadworks *List Autobahn roadworks*

Description

List Autobahn roadworks

Usage

```
autobahn_roadworks(road_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>road_id</code>	Road identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns current roadworks for a specific Autobahn road id. API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per roadwork entry:

road_id Road identifier (character).
identifier Entry identifier (character).
title Title (character).
subtitle Subtitle (character).
display_type Display type (character).
icon Icon identifier (character).
is_blocked Whether the road is blocked (logical).

future Whether the entry is future-dated (logical).
start_timestamp Start timestamp string (character).
start_time Parsed start time (POSIXct).
point Point coordinate string (character).
extent Extent coordinate string (character).
coordinate_lat Latitude (character).
coordinate_long Longitude (character).
description Description text (character).
footer Footer text (character).
route_recommendation Route recommendations (list-column).
lorry_parking_feature_icons Lorry parking feature icons (list-column).

With `flatten = TRUE`, the two list-columns are transformed according to `flatten_mode`.

See Also

[autobahn_roads\(\)](#) to list available roads, and [autobahn_roadwork_details\(\)](#) for detail records.

Other Autobahn: [autobahn_charging_station_details\(\)](#), [autobahn_charging_stations\(\)](#), [autobahn_closure_details\(\)](#), [autobahn_closures\(\)](#), [autobahn_parking_lorries\(\)](#), [autobahn_parking_lorry_details\(\)](#), [autobahn_roads\(\)](#), [autobahn_roadwork_details\(\)](#), [autobahn_warning_details\(\)](#), [autobahn_warnings\(\)](#), [autobahn_webcam_details\(\)](#), [autobahn_webcams\(\)](#)

Examples

```
## Not run:
roads <- autobahn_roads()
autobahn_roadworks(roads$road_id[[1]], flatten = TRUE)

## End(Not run)
```

autobahn_roadwork_details

Get Autobahn roadwork details

Description

Get Autobahn roadwork details

Usage

```
autobahn_roadwork_details(roadwork_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

roadwork_id	Roadwork identifier.
flatten	Logical; if TRUE, simplify nested list columns according to flatten_mode. Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when flatten = TRUE: "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns full details for a single roadwork entry. API documentation: <https://autobahn.api.bund.dev>.

Value

A one-row tibble for the selected roadwork with the same columns as `autobahn_roadworks()`. `road_id` is typically NA for detail calls.

See Also

`autobahn_roadworks()` to list roadworks.

Other Autobahn: `autobahn_charging_station_details()`, `autobahn_charging_stations()`, `autobahn_closure_details()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_parking_lorry_details()`, `autobahn_roads()`, `autobahn_roadworks()`, `autobahn_warning_details()`, `autobahn_warnings()`, `autobahn_webcam_details()`, `autobahn_webcams()`

Examples

```
## Not run:
roads <- autobahn_roads()
roadworks <- autobahn_roadworks(roads$road_id[[1]])
autobahn_roadwork_details(roadworks$identifier[[1]])

## End(Not run)
```

autobahn_warnings *List Autobahn warnings*

Description

List Autobahn warnings

Usage

```
autobahn_warnings(road_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>road_id</code>	Road identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns current warnings for a specific Autobahn road id. API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per warning entry:

road_id Road identifier (character).
identifier Entry identifier (character).
title Title (character).
subtitle Subtitle (character).
display_type Display type (character).
icon Icon identifier (character).
is_blocked Whether the road is blocked (logical).
future Whether the entry is future-dated (logical).
start_timestamp Start timestamp string (character).
start_time Parsed start time (POSIXct).
point Point coordinate string (character).
extent Extent coordinate string (character).
coordinate_lat Latitude (character).
coordinate_long Longitude (character).
description Description text (character).
footer Footer text (character).
route_recommendation Route recommendations (list-column).
lorry_parking_feature_icons Lorry parking feature icons (list-column).

With `flatten = TRUE`, the two list-columns are transformed according to `flatten_mode`.

See Also

[autobahn_roads\(\)](#) to list roads and [autobahn_warning_details\(\)](#) for details.

Other Autobahn: [autobahn_charging_station_details\(\)](#), [autobahn_charging_stations\(\)](#), [autobahn_closure_details\(\)](#), [autobahn_closures\(\)](#), [autobahn_parking_lorries\(\)](#), [autobahn_parking_lorry_details\(\)](#), [autobahn_roads\(\)](#), [autobahn_roadwork_details\(\)](#), [autobahn_roadworks\(\)](#), [autobahn_warning_details\(\)](#), [autobahn_webcam_details\(\)](#), [autobahn_webcams\(\)](#)

Examples

```
## Not run:
roads <- autobahn_roads()
autobahn_warnings(roads$road_id[[1]], flatten = TRUE)

## End(Not run)
```

autobahn_warning_details

Get Autobahn warning details

Description

Get Autobahn warning details

Usage

```
autobahn_warning_details(warning_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

warning_id	Warning identifier.
flatten	Logical; if TRUE, simplify nested list columns according to flatten_mode. Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when flatten = TRUE: <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via tidyr::unnest_longer(). Warning: this can significantly increase the number of rows.

Details

Returns full details for a single warning entry. API documentation: <https://autobahn.api.bund.dev>.

Value

A one-row tibble for the selected warning with the same columns as `autobahn_warnings()`. `road_id` is typically NA for detail calls.

See Also

`autobahn_warnings()` to list warnings.

Other Autobahn: `autobahn_charging_station_details()`, `autobahn_charging_stations()`, `autobahn_closure_details()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_parking_lorry_details()`, `autobahn_roads()`, `autobahn_roadwork_details()`, `autobahn_roadworks()`, `autobahn_warnings()`, `autobahn_webcam_details()`, `autobahn_webcams()`

Examples

```
## Not run:
roads <- autobahn_roads()
warnings <- autobahn_warnings(roads$road_id[[1]])
autobahn_warning_details(warnings$identifier[[1]])

## End(Not run)
```

autobahn_webcams	<i>List Autobahn webcams</i>
------------------	------------------------------

Description

List Autobahn webcams

Usage

```
autobahn_webcams(road_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>road_id</code>	Road identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

Returns webcam entries for a specific Autobahn road id. API documentation: <https://autobahn.api.bund.dev>.

Value

A tibble with one row per webcam entry:

road_id Road identifier (character).

identifier Entry identifier (character).

title Title (character).

subtitle Subtitle (character).

display_type Display type (character).

icon Icon identifier (character).

is_blocked Whether the road is blocked (logical).

future Whether the entry is future-dated (logical).

start_timestamp Start timestamp string (character).

start_time Parsed start time (POSIXct).

point Point coordinate string (character).

extent Extent coordinate string (character).

coordinate_lat Latitude (character).

coordinate_long Longitude (character).

description Description text (character).

footer Footer text (character).

route_recommendation Route recommendations (list-column).

lorry_parking_feature_icons Lorry parking feature icons (list-column).

With `flatten = TRUE`, the two list-columns are transformed according to `flatten_mode`.

See Also

[autobahn_roads\(\)](#) and [autobahn_webcam_details\(\)](#).

Other Autobahn: [autobahn_charging_station_details\(\)](#), [autobahn_charging_stations\(\)](#), [autobahn_closure_details\(\)](#), [autobahn_closures\(\)](#), [autobahn_parking_lorries\(\)](#), [autobahn_parking_lorry_details\(\)](#), [autobahn_roads\(\)](#), [autobahn_roadwork_details\(\)](#), [autobahn_roadworks\(\)](#), [autobahn_warning_details\(\)](#), [autobahn_warnings\(\)](#), [autobahn_webcam_details\(\)](#)

Examples

```
## Not run:  
roads <- autobahn_roads()  
autobahn_webcams(roads$road_id[[1]], flatten = TRUE)  
  
## End(Not run)
```

`autobahn_webcam_details`*Get Autobahn webcam details*

Description

Get Autobahn webcam details

Usage

```
autobahn_webcam_details(webcam_id, flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>webcam_id</code>	Webcam identifier.
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns full details for a single webcam entry. API documentation: <https://autobahn.api.bund.dev>.

Value

A one-row tibble for the selected webcam with the same columns as `autobahn_webcams()`. `road_id` is typically NA for detail calls.

See Also

`autobahn_webcams()` to list webcams.

Other Autobahn: `autobahn_charging_station_details()`, `autobahn_charging_stations()`, `autobahn_closure_details()`, `autobahn_closures()`, `autobahn_parking_lorries()`, `autobahn_parking_lorry_details()`, `autobahn_roads()`, `autobahn_roadwork_details()`, `autobahn_roadworks()`, `autobahn_warning_details()`, `autobahn_warnings()`, `autobahn_webcams()`

Examples

```
## Not run:
roads <- autobahn_roads()
webcams <- autobahn_webcams(roads$road_id[[1]])
autobahn_webcam_details(webcams$identifier[[1]])

## End(Not run)
```

bewerberboerse_details

Retrieve Bewerberboerse candidate details

Description

Retrieve Bewerberboerse candidate details

Usage

```
bewerberboerse_details(referenznummer, flatten = FALSE, flatten_mode = "json")
```

Arguments

referenznummer Bewerber referenznummer.

flatten Logical; if TRUE, simplify nested list columns according to `flatten_mode`. Default FALSE keeps list columns as-is.

flatten_mode How to handle list columns when `flatten = TRUE`:

- "drop" Remove list columns entirely. Use when nested data is not needed.
- "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.
- "unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.
Warning: this can significantly increase the number of rows.

Details

Fetches details for a single candidate. The `referenznummer` typically comes from `bewerberboerse_search()`. See <https://bundesapi.github.io/bewerberboerse-api/>.

Value

A tibble with the same columns as `bewerberboerse_search()`, filtered to the requested `referenznummer`.

refnr Candidate reference number (character).

verfuegbarkeit_von Availability date text (character).

aktualisierungsdatum Last update date text (character).

veroeffentlichungsdatum Publication date text (character).

stellenart Position type (character).

arbeitszeit_modelle Work-time models, collapsed string (character).

berufe Occupations, collapsed string (character).

letzte_taetigkeit_jahr Year of most recent activity (character).

letzte_taetigkeit_bezeichnung Label of most recent activity (character).

letzte_taetigkeit_aktuell Whether latest activity is current (logical).

hat_email Email contact available (logical).

hat_telefon Phone contact available (logical).

hat_adresse Postal address available (logical).

ort City/locality (character).

plz Postal code (character).

umkreis Radius text returned by API (character).

region Region/state (character).

land Country (character).

mehrere_arbeitsorte Multiple work locations flag (logical).

ausbildungen Education entries (list-column).

erfahrung Experience entries (list-column).

operation_id Operation identifier (character).

page Returned page index (integer).

size Returned page size (integer).

max_ergebnisse Total matching result count (integer).

See Also

[bewerberboerse_search\(\)](#) to find candidates and [bunddev_auth_set\(\)](#) for auth.

Other Bewerberboerse: [bewerberboerse_search\(\)](#)

Examples

```
## Not run:
Sys.setenv(BEWERBERBOERSE_API_KEY = "jobboerse-bewerbersuche-ui")
bunddev_auth_set("bewerberboerse", type = "api_key", env_var = "BEWERBERBOERSE_API_KEY")
bewerberboerse_details("12345", flatten = TRUE)

## End(Not run)
```

 bewerberboerse_search *Search the Bewerberboerse API*

Description

Search the Bewerberboerse API

Usage

```
bewerberboerse_search(params = list(), flatten = FALSE, flatten_mode = "json")
```

Arguments

params	<p>Named list of query parameters:</p> <ul style="list-style-type: none"> was Search terms (skills, professions, keywords) (character). ausbildungsart Training type code, e.g. au, ba, fo (character). wo Location query (place/PLZ/state/country) (character). umkreis Radius in km around wo (integer). angebotsart Offer type code, e.g. ar, az (character). arbeitszeit Work time code, e.g. vz, tz (character). berufserfahrung Experience level code, e.g. be, bg (character). vertragsart Contract type code, e.g. be, ub (character). behinderung Accessibility filter code (character). page Page index (integer). size Number of results per page (integer).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	<p>How to handle list columns when <code>flatten = TRUE</code>:</p> <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

The Bewerberboerse API provides access to candidate listings. Authentication uses an API key passed as X-API-Key (clientId jobboerse-bewerbersuche-ui). See <https://bundesapi.github.io/bewerberboerse-api/> for official docs.

Use `bunddev_auth_set()` to configure the key and `bunddev_parameters()` to inspect available query parameters.

Value

A tibble with one row per candidate profile:

refnr Candidate reference number (character).
verfuegbarkeit_von Availability date text (character).
aktualisierungsdatum Last update date text (character).
veroeffentlichungsdatum Publication date text (character).
stellenart Position type (character).
arbeitszeit_modelle Work-time models, collapsed string (character).
berufe Occupations, collapsed string (character).
letzte_taetigkeit_jahr Year of most recent activity (character).
letzte_taetigkeit_bezeichnung Label of most recent activity (character).
letzte_taetigkeit_aktuell Whether latest activity is current (TRUE/FALSE).
hat_email Email contact available (TRUE/FALSE).
hat_telefon Phone contact available (TRUE/FALSE).
hat_adresse Postal address available (TRUE/FALSE).
ort City/locality (character).
plz Postal code (character).
umkreis Radius text returned by API (character).
region Region/state (character).
land Country (character).
mehrere_arbeitsorte Multiple work locations flag (TRUE/FALSE).
ausbildungen Education entries (list-column).
erfahrung Experience entries (list-column).
operation_id Operation identifier used by the tidier (character).
page Returned page index (integer).
size Returned page size (integer).
max_ergebnisse Total matching result count (integer).

See Also

[bewerberboerse_details\(\)](#) for detailed entries and [bunddev_auth_set\(\)](#) for authentication setup.
 Other Bewerberboerse: [bewerberboerse_details\(\)](#)

Examples

```
## Not run:
Sys.setenv(BEWERBERBOERSE_API_KEY = "jobboerse-bewerbersuche-ui")
bunddev_auth_set("bewerberboerse", type = "api_key", env_var = "BEWERBERBOERSE_API_KEY")
bewerberboerse_search(params = list(was = "data", size = 10), flatten = TRUE)

## End(Not run)
```

bunddev_auth_get	<i>Get authentication configuration for an API</i>
------------------	--

Description

Get authentication configuration for an API

Usage

```
bunddev_auth_get(api)
```

Arguments

api	Registry id.
-----	--------------

Details

Returns the stored auth configuration for the API or a default none entry if no auth has been configured.

Value

A named list with auth settings for api. If no configuration was set, returns defaults with type = "none".

See Also

[bunddev_auth_set\(\)](#) to configure credentials.

Other Authentication: [bunddev_auth_set\(\)](#)

Examples

```
bunddev_auth_get("jobsuche")
```

bunddev_auth_set	<i>Set authentication configuration for an API</i>
------------------	--

Description

Set authentication configuration for an API

Usage

```

bunddev_auth_set(
  api,
  type = "api_key",
  env_var,
  scheme = NULL,
  oauth_url = NULL,
  oauth_secret_env = NULL,
  oauth_default_id = NULL,
  oauth_token_header = "OAuthAccessToken",
  oauth_fallback_header = "X-API-Key"
)

```

Arguments

<code>api</code>	Registry id.
<code>type</code>	Authentication type ("none", "api_key", or "oauth2").
<code>env_var</code>	Environment variable containing credentials (for <code>api_key</code>) or client ID (for <code>oauth2</code>).
<code>scheme</code>	Authentication scheme used in the Authorization header (e.g., "Bearer", "ApiKey", "OAuth").
<code>oauth_url</code>	OAuth2 token endpoint URL (for <code>oauth2</code> type).
<code>oauth_secret_env</code>	OAuth2 client secret environment variable (for <code>oauth2</code> type).
<code>oauth_default_id</code>	Default OAuth2 client ID if <code>env_var</code> is not set (for <code>oauth2</code> type).
<code>oauth_token_header</code>	Header name for OAuth token (default "OAuthAccessToken").
<code>oauth_fallback_header</code>	Header name when no secret available (default "X-API-Key").

Details

Store auth configuration used by `bunddev_call()` and adapter helpers.

For API key auth: set `env_var` to the name of an environment variable containing the key. The key is sent as `Authorization: {scheme} <key>`.

For OAuth2 client credentials: set `oauth_url` to the token endpoint, `env_var` to the client ID env var, and `oauth_secret_env` to the client secret env var. If the secret is available, fetches an OAuth token; otherwise falls back to sending the client ID as an API key.

Value

A named list with stored authentication settings:

type Authentication type ("none", "api_key", "oauth2").

env_var Environment variable name for credential lookup.

scheme Header scheme/template used for auth header construction.

oauth_url OAuth token endpoint URL (oauth2 only).
oauth_secret_env Environment variable name for OAuth client secret.
oauth_default_id Fallback OAuth client ID when env_var is unset.
oauth_token_header Header name used for OAuth access tokens.
oauth_fallback_header Header name used for fallback API key mode.

See Also

[bunddev_auth_get\(\)](#) to inspect the stored configuration, and [bunddev_call\(\)](#) to make authenticated requests.

Other Authentication: [bunddev_auth_get\(\)](#)

Examples

```
# API key authentication
Sys.setenv(JOBBOERSE_API_KEY = "jobboerse-jobsuche")
bunddev_auth_set("jobsuche", type = "api_key", env_var = "JOBBOERSE_API_KEY", scheme = "X-API-Key")

# OAuth2 client credentials
bunddev_auth_set("berufssprachkurssuche",
  type = "oauth2",
  oauth_url = "https://rest.arbeitsagentur.de/oauth/gettoken_cc",
  env_var = "BERUFSSPRACHKURSSUCHE_API_KEY",
  oauth_secret_env = "BERUFSSPRACHKURSSUCHE_CLIENT_SECRET",
  oauth_default_id = "bd24f42e-ad0b-4005-b834-23bb6800dc6c")
```

bunddev_cache_dir	<i>Locate the bunddev cache directory</i>
-------------------	---

Description

Locate the bunddev cache directory

Usage

```
bunddev_cache_dir()
```

Details

The cache directory is used to store downloaded OpenAPI specs and cached API responses. During R CMD check, bunddev uses a temporary cache directory to avoid leaving persistent files behind. Use this to inspect or clean cached files.

Value

Path to the bunddev cache directory (character scalar).

See Also

[bunddev_spec\(\)](#) to download specs, and [bunddev_spec_path\(\)](#) to locate a specific spec file.

Other OpenAPI: [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:
bunddev_cache_dir()

## End(Not run)
```

bunddev_call	<i>Call an API operation</i>
--------------	------------------------------

Description

Call an API operation

Usage

```
bunddev_call(
  api,
  operation_id = NULL,
  params = list(),
  path = NULL,
  method = NULL,
  parse = c("json", "text", "raw", "xml"),
  base_url = NULL,
  body = NULL,
  body_type = c("json", "form"),
  headers = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

api	Registry id.
operation_id	OpenAPI operationId (use this OR path+method).
params	Named list of query and path parameters passed to the selected operation.
path	API path (use with method instead of operation_id).
method	HTTP method (use with path instead of operation_id).

parse	Response parsing mode.
base_url	Optional base URL override.
body	Optional request body (for POST/PUT requests).
body_type	Body encoding type ("json" or "form").
headers	Optional named list of custom HTTP headers.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

This is the low-level OpenAPI caller. It supports two modes:

- Use `operation_id` to lookup endpoints by their OpenAPI `operationId`
- Use `path + method` for APIs without `operationIds`

The function fills path parameters from `params`, applies auth (if configured), and optionally caches GET responses when `safe = TRUE`.

Use [bunddev_parameters\(\)](#) and [bunddev_parameter_values\(\)](#) to discover valid parameters before calling.

Value

Parsed response according to `parse`:

`parse = "json"` Parsed JSON as an R list.

`parse = "text"` Character scalar with response body text.

`parse = "raw"` Raw vector with response bytes.

`parse = "xml"` xml2 document object.

See Also

[bunddev_call_tidy\(\)](#) for adapter-specific tidy outputs, and [bunddev_auth_set\(\)](#) to configure API keys.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:
# Retrieve Autobahn road ids (raw response)
bunddev_call("autobahn", "list-autobahnen")

## End(Not run)
```

`bunddev_call_tidy` *Call an API operation and return tidy data*

Description

Call an API operation and return tidy data

Usage

```
bunddev_call_tidy(api, operation_id, params = list(), ...)
```

Arguments

<code>api</code>	Registry id.
<code>operation_id</code>	OpenAPI operationId.
<code>params</code>	Named list of parameters forwarded to bunddev_call() .
<code>...</code>	Additional arguments passed to the tidier.

Details

This helper calls [bunddev_call\(\)](#) and then applies the adapter-specific tidier for the API. If no tidier is registered, a tibble containing the raw response is returned.

Value

A tibble with tidied endpoint data when a tidier is registered for `api`. Otherwise a one-row tibble with a single list-column row containing the parsed response object.

See Also

[bunddev_call\(\)](#) for raw responses, and [bunddev_parameters\(\)](#) to discover available parameters.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:  
# Tidy Autobahn roadworks  
bunddev_call_tidy("autobahn", "list-roadworks", params = list(roadId = "A1"))  
  
## End(Not run)
```

bunddev_endpoints *Retrieve endpoints for a given API*

Description

Retrieve endpoints for a given API

List OpenAPI endpoints for an API

Usage

```
bunddev_endpoints(id)
```

```
bunddev_endpoints(id)
```

Arguments

`id` Registry id.

Details

Parses the cached OpenAPI spec and returns every available path + method with its operationId and summary, if provided.

Value

A tibble with columns `method`, `path`, `operation_id`, `summary`

A tibble with one row per OpenAPI operation:

method HTTP method in lower-case, e.g. "get" (character).

path OpenAPI path template, e.g. "/v1/items/{id}" (character).

operation_id OpenAPI operationId (character or NA).

summary Operation summary from the spec (character or NA).

See Also

[bunddev_parameters\(\)](#) to inspect parameters and [bunddev_spec\(\)](#) to access the full spec.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:  
bunddev_endpoints("autobahn")  
  
## End(Not run)
```

bunddev_info	<i>Get a registry entry by id</i>
--------------	-----------------------------------

Description

Get a registry entry by id

Usage

```
bunddev_info(id)
```

Arguments

id Registry id.

Details

Use this to access the spec URL, documentation URL, and authentication requirements for a single API.

Value

A one-row tibble with the same columns as [bunddev_registry\(\)](#).

See Also

[bunddev_list\(\)](#) for discovery and [bunddev_registry\(\)](#) for the full table.

Other Registry: [bunddev_list\(\)](#), [bunddev_registry\(\)](#), [bunddev_search\(\)](#)

Examples

```
bunddev_info("smard")
```

bunddev_list	<i>List registry entries</i>
--------------	------------------------------

Description

List registry entries

Usage

```
bunddev_list(tag = NULL, auth = NULL)
```

Arguments

tag	Optional tag to filter on.
auth	Optional auth type to filter on.

Details

Use this to quickly narrow down APIs by topic or authentication type. Tags correspond to the taxonomy in the bundled registry.

Value

A tibble with the same columns as [bunddev_registry\(\)](#), filtered by tag and/or auth when provided.

See Also

[bunddev_registry\(\)](#) for the full table and [bunddev_info\(\)](#) for one entry.

Other Registry: [bunddev_info\(\)](#), [bunddev_registry\(\)](#), [bunddev_search\(\)](#)

Examples

```
bunddev_list(tag = "jobs")  
bunddev_list(auth = "api_key")
```

`bunddev_ms_to_posix` *Convert milliseconds to POSIXct*

Description

Convert milliseconds to POSIXct

Usage

```
bunddev_ms_to_posix(value, tz = "Europe/Berlin")
```

Arguments

<code>value</code>	Timestamp in milliseconds.
<code>tz</code>	Timezone for conversion.

Details

Converts epoch milliseconds to POSIXct in the requested timezone.

Value

POSIXct timestamp.

See Also

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
bunddev_ms_to_posix(1704067200000, tz = "Europe/Berlin")
```

`bunddev_parameters` *List OpenAPI parameters for an API*

Description

List OpenAPI parameters for an API

Usage

```
bunddev_parameters(id, name = NULL, path = NULL, method = NULL)
```

Arguments

id	Registry id.
name	Optional parameter name to filter.
path	Optional endpoint path to filter.
method	Optional HTTP method to filter.

Details

Returns one row per parameter defined in the OpenAPI spec. Enumerations are stored in a list-column (enum). Use the filter arguments to narrow down results to a specific endpoint or parameter name.

Value

A tibble with one row per parameter definition:

- method** HTTP method in lower-case (character).
- path** OpenAPI path template where the parameter applies (character).
- name** Parameter name (character).
- location** Parameter location such as "query" or "path" (character).
- required** Whether the parameter is required (TRUE/FALSE).
- description** Parameter description from the spec (character or NA).
- schema_type** OpenAPI schema type, e.g. "string" (character or NA).
- enum** List-column of allowed values (list of character vectors).

See Also

[bunddev_parameters_for\(\)](#) for adapter-specific parameters and [bunddev_parameter_values\(\)](#) for enum values.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:
bunddev_parameters("smard", name = "resolution")

## End(Not run)
```

`bunddev_parameters_for`*List OpenAPI parameters for a specific adapter*

Description

List OpenAPI parameters for a specific adapter

Usage

```
bunddev_parameters_for(endpoint)
```

Arguments

`endpoint` Adapter function or its name.

Details

Resolves the adapter to an OpenAPI path/method mapping and filters parameters accordingly.

Value

A tibble with the same columns as `bunddev_parameters()`, filtered to the OpenAPI path/method associated with `endpoint`.

See Also

`bunddev_parameters()` for the full API parameter table.

Other OpenAPI: `bunddev_cache_dir()`, `bunddev_call()`, `bunddev_call_tidy()`, `bunddev_endpoints()`, `bunddev_ms_to_posix()`, `bunddev_parameter_values()`, `bunddev_parameters()`, `bunddev_spec()`, `bunddev_spec_path()`, `bunddev_timestamp_to_ms()`

Examples

```
## Not run:  
bunddev_parameters_for(smard_timeseries)  
  
## End(Not run)
```

bunddev_parameter_values

Extract parameter enum values

Description

Extract parameter enum values

Usage

```
bunddev_parameter_values(endpoint, name)
```

Arguments

endpoint	Adapter function or its name.
name	Parameter name.

Details

Returns unique enum values for a parameter defined on the adapter endpoint.

Value

A character vector of unique enum values. Returns character(0) if no enum is declared for the selected parameter.

See Also

[bunddev_parameters_for\(\)](#) to inspect all parameters for an adapter.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:  
bunddev_parameter_values(smard_timeseries, "resolution")  
  
## End(Not run)
```

`bunddev_rate_limit_get`*Get API rate limit configuration*

Description

Get API rate limit configuration

Usage

```
bunddev_rate_limit_get(api)
```

Arguments

`api` Registry id.

Details

If no explicit limit was set, the function tries to infer one from the registry entry. The result is used by adapter helpers when `safe = TRUE`.

Value

A named list with:

max_per_hour Configured/inferred calls-per-hour limit (integer or NA).

timestamps Numeric Unix timestamps of tracked requests.

See Also

[bunddev_rate_limit_set\(\)](#) to override the default.

Other Throttling: [bunddev_rate_limit_set\(\)](#)

Examples

```
bunddev_rate_limit_get("smard")
```

`bunddev_rate_limit_set`*Set API rate limit configuration*

Description

Set API rate limit configuration

Usage

```
bunddev_rate_limit_set(api, max_per_hour)
```

Arguments

<code>api</code>	Registry id.
<code>max_per_hour</code>	Maximum number of calls per hour.

Details

Use this to override or enforce a per-hour rate limit for a given API. The default is inferred from the registry entry when available.

Value

A named list with:

max_per_hour Maximum allowed calls per hour (integer).

timestamps Numeric Unix timestamps of tracked requests.

See Also

[bunddev_rate_limit_get\(\)](#) to inspect the current setting.

Other Throttling: [bunddev_rate_limit_get\(\)](#)

Examples

```
bunddev_rate_limit_set("smard", max_per_hour = 60)
```

bunddev_registry	<i>Read the bundled API registry</i>
------------------	--------------------------------------

Description

Read the bundled API registry

Usage

```
bunddev_registry()
```

Details

The registry is bundled with the package and contains metadata such as the API title, provider, documentation URL, OpenAPI spec URL, authentication type, and any declared rate limits. The data originates from bund.dev and the bundesAPI registry.

Value

A tibble with one row per API registry entry:

id Stable API identifier used throughout bunddev (character).

title Human-readable API title (character).

provider API provider or owning institution (character).

spec_url OpenAPI specification URL (character).

docs_url Upstream API documentation URL (character).

auth Authentication type ("none", "api_key", "oauth2").

rate_limit Declared upstream rate limit text (character or NA).

tags List-column of taxonomy tags (list of character vectors).

See Also

[bunddev_list\(\)](#) for filtered listings, [bunddev_search\(\)](#) for keyword searches, and [bunddev_info\(\)](#) for a single entry.

Other Registry: [bunddev_info\(\)](#), [bunddev_list\(\)](#), [bunddev_search\(\)](#)

Examples

```
registry <- bunddev_registry()
head(registry, 3)
```

bunddev_search	<i>Search registry entries</i>
----------------	--------------------------------

Description

Search registry entries

Usage

```
bunddev_search(q)
```

Arguments

q Search query.

Details

Searches across registry ids, titles, providers, and tags using a simple substring match.

Value

A tibble with the same columns as [bunddev_registry\(\)](#), containing entries that match q.

See Also

[bunddev_list\(\)](#) to filter by tag or auth, and [bunddev_info\(\)](#) for details on a single API.

Other Registry: [bunddev_info\(\)](#), [bunddev_list\(\)](#), [bunddev_registry\(\)](#)

Examples

```
bunddev_search("weather")
```

bunddev_spec	<i>Retrieve a cached API spec</i>
--------------	-----------------------------------

Description

Retrieve a cached API spec

Usage

```
bunddev_spec(id, refresh = FALSE)
```

Arguments

id	Registry id.
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Downloads the OpenAPI spec from the registry if it is missing or when refresh = TRUE. Parsed specs are returned as lists.

Value

Parsed OpenAPI specification as a nested list.

See Also

[bunddev_endpoints\(\)](#) to list operations and [bunddev_parameters\(\)](#) to inspect parameters.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec_path\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:
bunddev_spec("smard")

## End(Not run)
```

`bunddev_spec_path` *Build a cache path for an API spec*

Description

Build a cache path for an API spec

Usage

```
bunddev_spec_path(id)
```

Arguments

id	Registry id.
----	--------------

Details

Determines the cache file name based on the spec URL extension (.yaml/json).

Value

File path where the API specification is cached (character scalar).

See Also

[bunddev_spec\(\)](#) to download and parse specs.

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_timestamp_to_ms\(\)](#)

Examples

```
## Not run:
bunddev_spec_path("smard")

## End(Not run)
```

`bunddev_timestamp_to_ms`

Convert timestamps to milliseconds

Description

Convert timestamps to milliseconds

Usage

```
bunddev_timestamp_to_ms(value, tz = "Europe/Berlin")
```

Arguments

<code>value</code>	Timestamp as numeric (ms), Date, or POSIXct.
<code>tz</code>	Timezone for Date/POSIXct conversion.

Details

This helper standardizes timestamps for APIs that expect epoch milliseconds. Dates are interpreted in the supplied timezone.

Value

Numeric timestamp in milliseconds.

See Also

Other OpenAPI: [bunddev_cache_dir\(\)](#), [bunddev_call\(\)](#), [bunddev_call_tidy\(\)](#), [bunddev_endpoints\(\)](#), [bunddev_ms_to_posix\(\)](#), [bunddev_parameter_values\(\)](#), [bunddev_parameters\(\)](#), [bunddev_parameters_for\(\)](#), [bunddev_spec\(\)](#), [bunddev_spec_path\(\)](#)

Examples

```
bunddev_timestamp_to_ms(as.POSIXct("2024-01-01 00:00:00", tz = "Europe/Berlin"))
```

```
bundeshaushalt_budget_data
```

Query Bundeshaushalt budget data

Description

Query Bundeshaushalt budget data

Usage

```
bundeshaushalt_budget_data(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters: year Budget year (integer, required). account Account type ("income" or "expenses", required). quota Target value mode, e.g. planned/actual values (character). unit Budget unit/granularity level (character). id Optional budget node id to drill into a specific subtree (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundeshaushalt_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

The Bundshaushalt API provides budget data for federal income and expenses. Required query parameters are year and account. API documentation: <https://github.com/bundesAPI/bundshaushalt-api>.

Value

A one-row tibble with budget metadata and nested payloads:

account Requested account type (character).

entity Entity label from the API response (character).

level_cur Current hierarchy level (integer).

level_max Maximum hierarchy level (integer).

modify_date Upstream modification date (character).

quota Quota/target mode (character).

timestamp Response timestamp in milliseconds (numeric).

timestamp_time Response timestamp as POSIXct in Europe/Berlin.

unit Selected unit/granularity (character).

year Budget year (integer).

details Detailed budget rows (list-column).

children Child nodes in the budget hierarchy (list-column).

parents Parent nodes in the hierarchy (list-column).

related Related nodes from the API response (list-column).

See Also

[bunddev_parameters\(\)](#) to inspect parameter metadata from the OpenAPI spec.

Examples

```
## Not run:  
bundshaushalt_budget_data(params = list(year = 2021, account = "expenses"))  
  
## End(Not run)
```

bundesrat_aktuelles *List current Bundesrat news*

Description

List current Bundesrat news

Usage

```
bundesrat_aktuelles(view = "renderXml", safe = TRUE, refresh = FALSE)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundrat_aktuelles", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns current news items from the Bundesrat app feed. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` in the current news feed. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronolog](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_aktuelles()

## End(Not run)
```

bundesrat_mitglieder *List Bundesrat members*

Description

List Bundesrat members

Usage

```
bundesrat_mitglieder(view = "renderXml", safe = TRUE, refresh = FALSE)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde" "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns Bundesrat member entries from the mobile feed. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` in the member feed. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologi](#)
[bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#),
[bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_mitglieder()

## End(Not run)
```

 bundesrat_plenum_aktuelle_sitzung

List Bundesrat current plenum session entries

Description

List Bundesrat current plenum session entries

Usage

```
bundesrat_plenum_aktuelle_sitzung(
  view = "renderXml",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesrat", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns entries for the current plenum session. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` for the current session feed. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_plenum_aktuelle_sitzung()

## End(Not run)
```

```
bundesrat_plenum_chronologisch
  List Bundesrat plenum entries in chronological order
```

Description

List Bundesrat plenum entries in chronological order

Usage

```
bundesrat_plenum_chronologisch(
  view = "renderXml",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns plenum entries ordered chronologically. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` in chronological order. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_plenum_chronologisch()

## End(Not run)
```

bundesrat_plenum_kompakt

List Bundesrat plenum compact entries

Description

List Bundesrat plenum compact entries

Usage

```
bundesrat_plenum_kompakt(view = "renderXml", safe = TRUE, refresh = FALSE)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns plenum compact entries for Bundesrat sessions. API documentation: [https://bundesrat.
api.bund.dev](https://bundesrat.api.bund.dev).

Value

A tibble with one row per XML `<item>` in the plenum compact feed. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, **dateOfIssue_time**, **imageDate_time**, **layoutDate_time**, **startdate_time**, **stopdate_time**
Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_plenum_kompakt()

## End(Not run)
```

bundesrat_plenum_naechste_sitzungen
Get Bundesrat upcoming plenum sessions

Description

Get Bundesrat upcoming plenum sessions

Usage

```
bundesrat_plenum_naechste_sitzungen(  
  view = "render[iOSDetailsWithoutInnerDate]",  
  safe = TRUE,  
  refresh = FALSE  
)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesrat", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns upcoming Bundesrat sessions. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML <item> in the upcoming sessions feed. Column names correspond to the XML child element names.

<tag> One column per unique scalar child element (character).

<repeated_tag> List-column when a child element name appears more than once in a single <item>.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_plenum_naechste_sitzungen()

## End(Not run)
```

bundesrat_praesidium *List Bundesrat presidium entries*

Description

List Bundesrat presidium entries

Usage

```
bundesrat_praesidium(view = "renderXml", safe = TRUE, refresh = FALSE)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesrat_praesidium", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns Bundesrat presidium entries. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML <item> in the presidium feed. Column names correspond to the XML child element names.

<tag> One column per unique scalar child element (character).

<repeated_tag> List-column when a child element name appears more than once in a single <item>.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_praesidium()

## End(Not run)
```

bundesrat_startlist *List Bundesrat API endpoints*

Description

List Bundesrat API endpoints

Usage

```
bundesrat_startlist(view = "renderXml", safe = TRUE, refresh = FALSE)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesrat", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the start list for the Bundesrat mobile API, including the URLs for other available resources. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` in the start list feed. Column names correspond to the XML child element names within each `<item>`.

`<tag>` One column per unique scalar child element (character). Exact names depend on the XML feed structure.

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time

Parsed POSIXct (Europe/Berlin) added when the corresponding source field (date, dateOfIssue, imageDate, layoutDate, startdate, stopdate) is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_stimmverteilung\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_startlist()

## End(Not run)
```

```
bundesrat_stimmverteilung
  Get Bundesrat voting distribution
```

Description

Get Bundesrat voting distribution

Usage

```
bundesrat_stimmverteilung(
  view = "render[iOSDetailsWithoutInnerDate]",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesrat_stimmverteilung", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the Bundesrat voting distribution. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` in the voting distribution feed. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_termine\(\)](#)

Examples

```
## Not run:
bundesrat_stimmverteilung()

## End(Not run)
```

bundesrat_termine *List Bundesrat dates and events*

Description

List Bundesrat dates and events

Usage

```
bundesrat_termine(view = "renderXml", safe = TRUE, refresh = FALSE)
```

Arguments

view	Rendering mode for the XML output.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesrat_termine", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns scheduled Bundesrat dates and events. API documentation: <https://bundesrat.api.bund.dev>.

Value

A tibble with one row per XML `<item>` in the dates/events feed. Column names correspond to the XML child element names.

`<tag>` One column per unique scalar child element (character).

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

date_time, dateOfIssue_time, imageDate_time, layoutDate_time, startdate_time, stopdate_time
 Parsed POSIXct (Europe/Berlin) added when the corresponding source field is present.

See Also

Other Bundesrat: [bundesrat_aktuelles\(\)](#), [bundesrat_mitglieder\(\)](#), [bundesrat_plenum_aktuelle_sitzung\(\)](#), [bundesrat_plenum_chronologisch\(\)](#), [bundesrat_plenum_kompakt\(\)](#), [bundesrat_plenum_naechste_sitzungen\(\)](#), [bundesrat_praesidium\(\)](#), [bundesrat_startlist\(\)](#), [bundesrat_stimmverteilung\(\)](#)

Examples

```
## Not run:
bundesrat_termin()

## End(Not run)
```

bundestag_article	<i>Get a Bundestag news article</i>
-------------------	-------------------------------------

Description

Get a Bundestag news article

Usage

```
bundestag_article(article_id, safe = TRUE, refresh = FALSE)
```

Arguments

article_id	Article id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a single Bundestag news article in XML format. API documentation: <https://bundestag.api.bund.dev/>.

Value

A one-row tibble with fields extracted from the XML document. Column names correspond to XML element names.

<tag> One column per unique scalar child element of the XML root (character). Exact names depend on the XML document structure.

<repeated_tag> List-column when a child element name appears more than once.

See Also

Other Bundestag: [bundestag_ausschuesse\(\)](#), [bundestag_ausschuss\(\)](#), [bundestag_conferences\(\)](#), [bundestag_mdb_bio\(\)](#), [bundestag_mdb_index\(\)](#), [bundestag_speaker\(\)](#), [bundestag_video_feed\(\)](#)

Examples

```
## Not run:
bundestag_article(849630)

## End(Not run)
```

bundestag_ausschuesse *List Bundestag committees*

Description

List Bundestag committees

Usage

```
bundestag_ausschuesse(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_ausschuesse", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the committee index from Bundestag XML feeds. API documentation: <https://bundestag.api.bund.dev/>.

Value

A tibble with one row per committee `<item>` from the index feed. Column names correspond to XML child element names within each `<item>`.

`<tag>` One column per unique scalar child element in each `<item>` (character). Exact names depend on the feed structure.

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

See Also

Other Bundestag: [bundestag_article\(\)](#), [bundestag_ausschuss\(\)](#), [bundestag_conferences\(\)](#), [bundestag_mdb_bio\(\)](#), [bundestag_mdb_index\(\)](#), [bundestag_speaker\(\)](#), [bundestag_video_feed\(\)](#)

Examples

```
## Not run:
bundestag_ausschuss()

## End(Not run)
```

bundestag_ausschuss *Get Bundestag committee details*

Description

Get Bundestag committee details

Usage

```
bundestag_ausschuss(ausschuss_id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>ausschuss_id</code>	Committee id (e.g. "a11").
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns details for a single committee. API documentation: <https://bundestag.api.bund.dev/>.

Value

A one-row tibble with committee detail fields extracted from XML. Column names correspond to XML element names.

<tag> One column per unique scalar child element of the XML root (character). Exact names depend on the XML document structure.

<repeated_tag> List-column when a child element name appears more than once.

See Also

Other Bundestag: [bundestag_article\(\)](#), [bundestag_ausschuesse\(\)](#), [bundestag_conferences\(\)](#), [bundestag_mdb_bio\(\)](#), [bundestag_mdb_index\(\)](#), [bundestag_speaker\(\)](#), [bundestag_video_feed\(\)](#)

Examples

```
## Not run:  
bundestag_ausschuss("a11")  
  
## End(Not run)
```

`bundestag_conferences` *Get Bundestag conferences overview*

Description

Get Bundestag conferences overview

Usage

```
bundestag_conferences(safe = TRUE, refresh = FALSE)
```

Arguments

<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns conference overview data from the plenum feed. API documentation: <https://bundestag.api.bund.dev/>.

Value

A tibble with one row per <item> entry from the conferences feed. Column names correspond to XML child element names within each <item>.

<tag> One column per unique scalar child element in each <item> (character). Exact names depend on the feed structure.

<repeated_tag> List-column when a child element name appears more than once in a single <item>.

See Also

Other Bundestag: [bundestag_article\(\)](#), [bundestag_ausschuesse\(\)](#), [bundestag_ausschuss\(\)](#), [bundestag_mdb_bio\(\)](#), [bundestag_mdb_index\(\)](#), [bundestag_speaker\(\)](#), [bundestag_video_feed\(\)](#)

Examples

```
## Not run:
bundestag_conferences()

## End(Not run)
```

bundestag_lobbyregister_search

Search the Bundestag lobbyregister

Description

Search the Bundestag lobbyregister

Usage

```
bundestag_lobbyregister_search(
  q = NULL,
  sort = NULL,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

q	Optional search string.
sort	Optional sorting order.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_lobbyregister_search", "cache")</code> .

refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns search results from the Bundestag lobbyregister. API documentation: <https://bundesapi.github.io/bundestag-lobbyregister-api/>.

Value

A one-row tibble with search metadata:

source Source label from the API payload (character).

source_url Source URL (character).

search_url URL of the executed search (character).

search_date Search timestamp/date string (character).

search_parameters Effective search parameter object (list-column).

result_count Total result count (integer).

results Search result entries (list-column).

Examples

```
## Not run:
bundestag_lobbyregister_search(q = "energie")

## End(Not run)
```

bundestag_mdb_bio *Get a Bundestag biography*

Description

Get a Bundestag biography

Usage

```
bundestag_mdb_bio(mdb_id, safe = TRUE, refresh = FALSE)
```

Arguments

mdb_id	Member id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_mdb_index", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns details for a single member of parliament. API documentation: <https://bundestag.api.bund.dev/>.

Value

A one-row tibble with biography fields extracted from XML. Column names correspond to XML element names.

<tag> One column per unique scalar child element of the XML root (character). Exact names depend on the XML document structure.

<repeated_tag> List-column when a child element name appears more than once.

See Also

Other Bundestag: [bundestag_article\(\)](#), [bundestag_ausschuesse\(\)](#), [bundestag_ausschuss\(\)](#), [bundestag_conferences\(\)](#), [bundestag_mdb_index\(\)](#), [bundestag_speaker\(\)](#), [bundestag_video_feed\(\)](#)

Examples

```
## Not run:  
bundestag_mdb_bio(1769)  
  
## End(Not run)
```

`bundestag_mdb_index` *List Bundestag members of parliament*

Description

List Bundestag members of parliament

Usage

```
bundestag_mdb_index(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the index of members of parliament. API documentation: <https://bundestag.api.bund.dev/>.

Value

A tibble with one row per member `<item>` from the index feed. Column names correspond to XML child element names within each `<item>`.

`<tag>` One column per unique scalar child element in each `<item>` (character). Exact names depend on the feed structure.

`<repeated_tag>` List-column when a child element name appears more than once in a single `<item>`.

See Also

Other Bundestag: [bundestag_article\(\)](#), [bundestag_ausschuesse\(\)](#), [bundestag_ausschuss\(\)](#), [bundestag_conferences\(\)](#), [bundestag_mdb_bio\(\)](#), [bundestag_speaker\(\)](#), [bundestag_video_feed\(\)](#)

Examples

```
## Not run:
bundestag_mdb_index()

## End(Not run)
```

bundestag_speaker	<i>Get the current Bundestag speaker</i>
-------------------	--

Description

Get the current Bundestag speaker

Usage

```
bundestag_speaker(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the current speaker from the plenum feed. API documentation: <https://bundestag.api.bund.dev/>.

Value

A one-row tibble with fields extracted from the XML document (same structure rules as `bundestag_article()`).

<tag> One column per unique scalar child element of the XML root (character). Exact names depend on the XML document structure.

<repeated_tag> List-column when a child element name appears more than once.

See Also

Other Bundestag: `bundestag_article()`, `bundestag_ausschuesse()`, `bundestag_ausschuss()`, `bundestag_conferences()`, `bundestag_mdb_bio()`, `bundestag_mdb_index()`, `bundestag_video_feed()`

Examples

```
## Not run:  
bundestag_speaker()  
  
## End(Not run)
```

`bundestag_video_feed` *Get a Bundestag video feed entry*

Description

Get a Bundestag video feed entry

Usage

```
bundestag_video_feed(content_id, safe = TRUE, refresh = FALSE)
```

Arguments

content_id	Video content id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a video feed entry from the Bundestag webtv service. API documentation: <https://bundestag.api.bund.dev/>.

Value

A one-row tibble with video feed fields extracted from XML. Column names correspond to XML element names.

<tag> One column per unique scalar child element of the XML root (character). Exact names depend on the XML document structure.

<repeated_tag> List-column when a child element name appears more than once.

See Also

Other Bundestag: [bundestag_article\(\)](#), [bundestag_ausschuesse\(\)](#), [bundestag_ausschuss\(\)](#), [bundestag_conferences\(\)](#), [bundestag_mdb_bio\(\)](#), [bundestag_mdb_index\(\)](#), [bundestag_speaker\(\)](#)

Examples

```
## Not run:
bundestag_video_feed(7529016)

## End(Not run)
```

dashboard_deutschland_geo

Get Dashboard Deutschland GeoJSON

Description

Get Dashboard Deutschland GeoJSON

Usage

```
dashboard_deutschland_geo(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns GeoJSON data for Germany and the federal states. API documentation: <https://dashboard.deutschland.api.bund.dev/>.

Value

A one-row tibble with GeoJSON container metadata:

type GeoJSON object type (character).
name Dataset name (character).
title Dataset title (character).
version Version identifier (character).
copyright_short Short copyright label (character).
copyright_url Copyright URL (character).
crs Coordinate reference system metadata (list-column).
features GeoJSON feature collection entries (list-column).

See Also

Other Dashboard Deutschland: [dashboard_deutschland_get\(\)](#), [dashboard_deutschland_indicators\(\)](#)

Examples

```
## Not run:
dashboard_deutschland_get(flatten = TRUE)

## End(Not run)
```

```
dashboard_deutschland_get
  List Dashboard Deutschland entries
```

Description

List Dashboard Deutschland entries

Usage

```
dashboard_deutschland_get(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundedeutschland", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

Returns the full list of dashboard entries and metadata for each item. API documentation: <https://dashboard.deutschland.api.bund.dev/>.

Value

A tibble with one row per dashboard tile:

id Tile identifier (character).
name Title in German (character).
name_en Title in English (character).
description Description in German (character).
description_en Description in English (character).
teaser Teaser text in German (character).
teaser_en Teaser text in English (character).
conclusion Conclusion text in German (character).
conclusion_en Conclusion text in English (character).
category Category metadata (list-column).
tags Tag metadata (list-column).
image Image path/url (character).
clicks Click counter (integer).
order_id Ordering index (integer).
trending Whether tile is marked as trending (TRUE/FALSE).
top Whether tile is marked as top (TRUE/FALSE).
layout_tiles Layout tile metadata (list-column).
layout_mode Layout mode identifier (character).

See Also

Other Dashboard Deutschland: [dashboard_deutschland_geo\(\)](#), [dashboard_deutschland_indicators\(\)](#)

Examples

```
## Not run:  
dashboard_deutschland_get()  
  
## End(Not run)
```

dashboard_deutschland_indicators

Query Dashboard Deutschland indicators

Description

Query Dashboard Deutschland indicators

Usage

```
dashboard_deutschland_indicators(ids = NULL, safe = TRUE, refresh = FALSE)
```

Arguments

<code>ids</code>	Indicator ids, semicolon-separated or as a character vector.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns indicator data for the specified ids. API documentation: <https://dashboard.deutschland.api.bund.dev/>.

Value

A tibble with one row per indicator entry:

id Indicator id (character).
date Timestamp in milliseconds since epoch (numeric).
date_time Timestamp as POSIXct in Europe/Berlin.
json Indicator payload as JSON string (character).
title Indicator title (character).

See Also

Other Dashboard Deutschland: [dashboard_deutschland_geo\(\)](#), [dashboard_deutschland_get\(\)](#)

Examples

```
## Not run:  
dashboard_deutschland_indicators("tile_1667811574092")  
  
## End(Not run)
```

 ddb_institutions *List DDB institutions*

Description

List DDB institutions

Usage

```
ddb_institutions(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of institution query parameters: hasItems Only institutions with items (TRUE/FALSE). sector Sector filter (character). offset Start index for pagination (integer). length Maximum number of returned items (integer). zoomlevel Optional map zoom level parameter (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns institutions registered in the DDB. Requires the DDB API key.

Value

A [tibble](#) with one row per institution and columns:

id Character. Institution identifier.

label Character. Display name.

latitude Numeric. Geographic latitude.

longitude Numeric. Geographic longitude.

sector Character. Cultural sector code.

children List. Nested child institutions.

Falls back to a one-row tibble with list-column response when the API returns a non-homogeneous payload.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DDB: [ddb_institution_sectors\(\)](#), [ddb_search\(\)](#)

Examples

```
## Not run:
ddb_institutions(params = list(hasItems = TRUE))

## End(Not run)
```

ddb_institution_sectors

List DDB institution sectors

Description

List DDB institution sectors

Usage

```
ddb_institution_sectors(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns institution sector metadata. Requires the DDB API key.

Value

A [tibble](#) with one row per sector and columns:

value Character. Sector name.

count Integer. Number of institutions in sector.

Falls back to a one-row tibble with list-column response when the API returns a non-homogeneous payload.

See Also

Other DDB: [ddb_institutions\(\)](#), [ddb_search\(\)](#)

Examples

```
## Not run:  
ddb_institution_sectors()  
  
## End(Not run)
```

ddb_search

Search Deutsche Digitale Bibliothek

Description

Search Deutsche Digitale Bibliothek

Usage

```
ddb_search(  
  query,  
  params = list(),  
  safe = TRUE,  
  refresh = FALSE,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

query	Search query string.
params	Named list of query parameters. Common keys: query Search string (set automatically from query). offset Start index for pagination (integer). length Maximum number of returned items (integer). facet Facet selection(s) supported by the DDB search API (character). sort Sort key/order (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns search results from the Deutsche Digitale Bibliothek API. You need an API key from <https://www.deutsche-digitale-bibliothek.de/user/apikey>. The key is sent in the Authorization header as `OAuth oauth_consumer_key="<key>"`.

Configure authentication via `bunddev_auth_set()` using a template-style scheme, or set the `DDB_API_KEY` environment variable directly.

Value

A one-row tibble with one list-column:

response Full parsed DDB search payload (list-column).

See Also

`bunddev_parameters()` to inspect available query parameters. `bunddev_auth_set()` to configure authentication.

Other DDB: `ddb_institution_sectors()`, `ddb_institutions()`

Examples

```
## Not run:
# Recommended: use bunddev_auth_set with template scheme
Sys.setenv(DDB_API_KEY = "<api-key>")
bunddev_auth_set("ddb", type = "api_key", env_var = "DDB_API_KEY",
                 scheme = "OAuth oauth_consumer_key=%s")
ddb_search(query = "berlin", params = list(rows = 5))

## End(Not run)
```

destatis_catalogue_cubes

List Destatis cubes

Description

List Destatis cubes

Usage

```
destatis_catalogue_cubes(
  params = list(),
  username = "GAST",
  password = "GAST",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

params	<p>Named list of Genesis API query parameters. Common keys:</p> <p>username Genesis username (added automatically from username if missing).</p> <p>password Genesis password (added automatically from password if missing).</p> <p>name Table/cube name for data endpoints (also set by name argument).</p> <p>selection Selection filter expression (character).</p> <p>area Regional scope code (character).</p> <p>pagelength Maximum number of returned entries (integer).</p> <p>language Response language code (character).</p> <p>searchcriterion Catalogue search field (character).</p> <p>sortcriterion Catalogue sort field (character).</p> <p>type Resource type filter (character).</p> <p>startyear Start year filter (integer).</p> <p>endyear End year filter (integer).</p>
username	<p>Genesis username (default "GAST").</p>

password	Genesis password (default "GAST").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundde "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the cube catalogue from the Destatis Genesis API. API documentation: <https://github.com/bundesAPI/destatis-api>.

Value

A one-row tibble with one list-column response containing the parsed catalogue payload.

response Catalogue endpoints: full API payload as list-column.

name Data endpoints: requested table/cube name (character).

data Data endpoints: full payload as list-column.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Destatis: [destatis_catalogue_tables\(\)](#), [destatis_data_cube\(\)](#), [destatis_data_table\(\)](#)

Examples

```
## Not run:
destatis_catalogue_cubes()

## End(Not run)
```

destatis_catalogue_tables

List Destatis tables

Description

List Destatis tables

Usage

```
destatis_catalogue_tables(
  params = list(),
  username = "GAST",
  password = "GAST",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

params	Named list of Genesis API query parameters. Common keys: username Genesis username (added automatically from username if missing). password Genesis password (added automatically from password if missing). name Table/cube name for data endpoints (also set by name argument). selection Selection filter expression (character). area Regional scope code (character). pagelength Maximum number of returned entries (integer). language Response language code (character). searchcriterion Catalogue search field (character). sortcriterion Catalogue sort field (character). type Resource type filter (character). startyear Start year filter (integer). endyear End year filter (integer).
username	Genesis username (default "GAST").
password	Genesis password (default "GAST").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bunddev", "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the table catalogue from the Destatis Genesis API. API documentation: <https://github.com/bundesAPI/destatis-api>.

Value

A one-row tibble with one list-column response containing the parsed catalogue payload.

response Catalogue endpoints: full API payload as list-column.

name Data endpoints: requested table/cube name (character).

data Data endpoints: full payload as list-column.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Destatis: [destatis_catalogue_cubes\(\)](#), [destatis_data_cube\(\)](#), [destatis_data_table\(\)](#)

Examples

```
## Not run:
destatis_catalogue_tables()

## End(Not run)
```

destatis_data_cube *Retrieve Destatis cube data*

Description

Retrieve Destatis cube data

Usage

```
destatis_data_cube(
  name,
  params = list(),
  username = "GAST",
  password = "GAST",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

name	Cube id.
params	Named list of Genesis API query parameters. Common keys: username Genesis username (added automatically from username if missing). password Genesis password (added automatically from password if missing). name Table/cube name for data endpoints (also set by name argument). selection Selection filter expression (character). area Regional scope code (character). pagelength Maximum number of returned entries (integer). language Response language code (character). searchcriterion Catalogue search field (character). sortcriterion Catalogue sort field (character). type Resource type filter (character). startyear Start year filter (integer). endyear End year filter (integer).
username	Genesis username (default "GAST").
password	Genesis password (default "GAST").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundles/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns cube data as text (csv by default). API documentation: <https://github.com/bundesAPI/destatis-api>.

Value

A one-row tibble with:

name Requested cube id (character).

data Raw response body (typically CSV/text) (character).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Destatis: [destatis_catalogue_cubes\(\)](#), [destatis_catalogue_tables\(\)](#), [destatis_data_table\(\)](#)

Examples

```
## Not run:
destatis_data_cube("21231BJ001")

## End(Not run)
```

destatis_data_table *Retrieve Destatis table data*

Description

Retrieve Destatis table data

Usage

```
destatis_data_table(
  name,
  params = list(),
  username = "GAST",
  password = "GAST",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

name Table id.

params Named list of Genesis API query parameters. Common keys:

- username** Genesis username (added automatically from username if missing).
- password** Genesis password (added automatically from password if missing).
- name** Table/cube name for data endpoints (also set by name argument).
- selection** Selection filter expression (character).
- area** Regional scope code (character).

	pagelength Maximum number of returned entries (integer).
	language Response language code (character).
	searchcriterion Catalogue search field (character).
	sortcriterion Catalogue sort field (character).
	type Resource type filter (character).
	startyear Start year filter (integer).
	endyear End year filter (integer).
username	Genesis username (default "GAST").
password	Genesis password (default "GAST").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesAPI/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns table data as text (csv by default). API documentation: <https://github.com/bundesAPI/destatis-api>.

Value

A one-row tibble with:

name Requested table id (character).

data Raw response body (typically CSV/text) (character).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Destatis: [destatis_catalogue_cubes\(\)](#), [destatis_catalogue_tables\(\)](#), [destatis_data_cube\(\)](#)

Examples

```
## Not run:
destatis_data_table("12411-0001")
```

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

deutschlandatlas_query

Query Deutschlandatlas indicators

Description

Query Deutschlandatlas indicators

Usage

```
deutschlandatlas_query(
  table = "p_apo_f_ZA2022",
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

table	Table id (default "p_apo_f_ZA2022").
params	Named list of ArcGIS query parameters: where SQL-like filter expression (required by this adapter). f Output format ("json" default). outFields Fields to return ("*" default). returnGeometry Whether to include geometry ("true"/"false"). spatialRel Spatial relation, e.g. "esriSpatialRelIntersects". geometry Optional geometry filter (JSON string or R list).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

The Deutschlandatlas API is backed by an ArcGIS feature service. You must supply a where filter and output format f (usually "json"). API documentation: <https://github.com/AndreasFischer1985/deutschlandatlas-api>.

Value

A tibble with one row per ArcGIS feature. Attribute names are normalized to lower snake_case. Includes a geometry list-column.

Attribute columns One column per ArcGIS attributes field (sanitized to snake_case).

geometry Geometry object as list-column when requested/available.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Examples

```
## Not run:
deutschlandatlas_query(
  table = "p_apo_f_ZA2022",
  params = list(
    where = "1=1",
    outFields = "*",
    f = "json",
    returnGeometry = "false",
    resultRecordCount = 5
  )
)

## End(Not run)
```

diga_catalog_entries *List DiGA catalog entries*

Description

List DiGA catalog entries

Usage

```
diga_catalog_entries(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

<code>params</code>	Named list of FHIR search parameters: <code>_count</code> Maximum bundle size per request (integer, default 1000). <code>_profile</code> FHIR profile URL used to restrict resource type (character).
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev_cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <code>"drop"</code> Remove list columns entirely. Use when nested data is not needed. <code>"json"</code> Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . <code>"unnest"</code> Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per catalog entry and columns:

resourceType Character. FHIR resource type, always "CatalogEntry".

id Character. FHIR resource identifier.

status Character. Entry status.

orderable Logical. Whether the entry is orderable.

meta List. Resource metadata (profile, timestamps).

identifier List. Business identifiers.

type List. Catalog entry type coding.

referencedItem List. Referenced device definition.

extension List. FHIR profile extensions.

additionalCharacteristic List. Additional characteristics.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DiGA: [diga_charge_item_definitions\(\)](#), [diga_device_definitions\(\)](#), [diga_organizations\(\)](#), [diga_questionnaire_responses\(\)](#), [diga_questionnaires\(\)](#)

diga_charge_item_definitions

List DiGA prescription units

Description

List DiGA prescription units

Usage

```
diga_charge_item_definitions(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of FHIR search parameters: _count Maximum bundle size per request (integer, default 1000). _profile FHIR profile URL used to restrict resource type (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundles/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per charge item definition and columns:

resourceType Character. FHIR resource type, always "ChargeItemDefinition".

id Character. FHIR resource identifier.

url Character. Canonical URL of the resource.

status Character. Publication status.

- description** Character. Description text.
- meta** List. Resource metadata (profile, timestamps).
- identifier** List. Business identifiers.
- code** List. Charge item code.
- instance** List. Referenced device definitions.
- applicability** List. Applicability conditions.
- propertyGroup** List. Pricing properties.
- extension** List. FHIR profile extensions.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DiGA: [diga_catalog_entries\(\)](#), [diga_device_definitions\(\)](#), [diga_organizations\(\)](#), [diga_questionnaire_responses\(\)](#), [diga_questionnaires\(\)](#)

diga_device_definitions

List DiGA device definitions

Description

List DiGA device definitions

Usage

```
diga_device_definitions(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

- | | |
|---------|---|
| params | Named list of FHIR search parameters:
_count Maximum bundle size per request (integer, default 1000).
_profile FHIR profile URL used to restrict resource type (character). |
| safe | Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundlecache")</code> . |
| refresh | Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE). |
| flatten | Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is. |

flatten_mode How to handle list columns when `flatten = TRUE`:

- "drop" Remove list columns entirely. Use when nested data is not needed.
- "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.
- "unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.
Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per device definition and columns:

resourceType Character. FHIR resource type, always "DeviceDefinition".

id Character. FHIR resource identifier.

url Character. Canonical URL of the resource.

status Character. Publication status.

description Character. Device description text.

meta List. Resource metadata (profile, timestamps).

identifier List. Business identifiers.

type List. Device type coding.

deviceName List. Device name entries.

property List. Device properties.

extension List. FHIR profile extensions.

See Also

Other DiGA: [diga_catalog_entries\(\)](#), [diga_charge_item_definitions\(\)](#), [diga_organizations\(\)](#), [diga_questionnaire_responses\(\)](#), [diga_questionnaires\(\)](#)

Examples

```
## Not run: dig_device_definitions()
```

diga_organizations *List DiGA manufacturers*

Description

List DiGA manufacturers

Usage

```
diga_organizations(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

<code>params</code>	Named list of FHIR search parameters: <code>_count</code> Maximum bundle size per request (integer, default 1000). <code>_profile</code> FHIR profile URL used to restrict resource type (character).
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundles/cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <code>"drop"</code> Remove list columns entirely. Use when nested data is not needed. <code>"json"</code> Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . <code>"unnest"</code> Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per organization and columns:

resourceType Character. FHIR resource type, always "Organization".

id Character. FHIR resource identifier.

name Character. Organization display name.

active Logical. Whether the organization is active.

meta List. Resource metadata (profile, timestamps).

identifier List. Business identifiers.

type List. Organization type coding.

telecom List. Contact details.

address List. Postal addresses.

extension List. FHIR profile extensions.

See Also

`bunddev_parameters()` to inspect available query parameters.

Other DiGA: `diga_catalog_entries()`, `diga_charge_item_definitions()`, `diga_device_definitions()`, `diga_questionnaire_responses()`, `diga_questionnaires()`

diga_questionnaires *List DiGA questionnaires*

Description

List DiGA questionnaires

Usage

```
diga_questionnaires(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

<code>params</code>	Named list of FHIR search parameters: <code>_count</code> Maximum bundle size per request (integer, default 1000). <code>_profile</code> FHIR profile URL used to restrict resource type (character).
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundlecache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <code>"drop"</code> Remove list columns entirely. Use when nested data is not needed. <code>"json"</code> Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . <code>"unnest"</code> Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per questionnaire and columns:

resourceType Character. FHIR resource type, always "Questionnaire".

id Character. FHIR resource identifier.

url Character. Canonical URL of the resource.

status Character. Publication status.

name Character. Computer-friendly name.

title Character. Human-readable title.

meta List. Resource metadata (profile, timestamps).

identifier List. Business identifiers.

item List. Questionnaire items and groups.

extension List. FHIR profile extensions.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DiGA: [diga_catalog_entries\(\)](#), [diga_charge_item_definitions\(\)](#), [diga_device_definitions\(\)](#), [diga_organizations\(\)](#), [diga_questionnaire_responses\(\)](#)

diga_questionnaire_responses

List DiGA questionnaire responses

Description

List DiGA questionnaire responses

Usage

```
diga_questionnaire_responses(  
  params = list(),  
  safe = TRUE,  
  refresh = FALSE,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

<code>params</code>	Named list of FHIR search parameters: <code>_count</code> Maximum bundle size per request (integer, default 1000). <code>_profile</code> FHIR profile URL used to restrict resource type (character).
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundlecache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <code>"drop"</code> Remove list columns entirely. Use when nested data is not needed. <code>"json"</code> Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . <code>"unnest"</code> Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per questionnaire response and columns:

resourceType Character. FHIR resource type, always "QuestionnaireResponse".

id Character. FHIR resource identifier.

status Character. Response completion status.

authored Character. When the response was authored.

meta List. Resource metadata (profile, timestamps).

identifier List. Business identifiers.

questionnaire Character. Reference to source questionnaire.

item List. Response items and answers.

extension List. FHIR profile extensions.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DiGA: [diga_catalog_entries\(\)](#), [diga_charge_item_definitions\(\)](#), [diga_device_definitions\(\)](#), [diga_organizations\(\)](#), [diga_questionnaires\(\)](#)

dip_bundestag_aktivitaet

Get a DIP Aktivität

Description

Get a DIP Aktivität

Usage

```
dip_bundestag_aktivitaet(
  aktivitaet_id,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

aktivitaet_id	Aktivität id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundestag", "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected Aktivitaet:

id Aktivitaet identifier (character).
aktivitaetsart Activity type, e.g. "Rede" (character).
typ Entity type, always "Aktivität" (character).
dokumentart Document type: "Drucksache" or "Plenarprotokoll" (character).
wahlperiode Legislative period (integer).
datum Date of associated document (character, ISO date).
aktualisiert Last update timestamp (character, ISO datetime).

titel Title (character).

fundstelle Document reference details (list-column).

vorgangsbezug Up to 4 related Vorgänge (list-column).

vorgangsbezug_anzahl Total number of related Vorgänge (integer).

deskriptor Thesaurus descriptors (list-column).

abstract Summary text (character or NA).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgang_position\(\)](#), [dip_bundestag_vorgang_position_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_aktivitaet(1493545)

## End(Not run)
```

```
dip_bundestag_aktivitaet_list
      List DIP Aktivitäten
```

Description

List DIP Aktivitäten

Usage

```
dip_bundestag_aktivitaet_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per Aktivitaet and columns:

id Character. Unique Aktivitaet identifier.

aktivitaetsart Character. Activity type.

typ Character. Entity type, always "Aktivitaet".

dokumentart Character. Document type.

wahlperiode Integer. Legislative period.

datum Character. Date of associated document.

aktualisiert Character. Last update timestamp.

titel Character. Title text.

vorgangsbezug_anzahl Integer. Total related Vorgaenge.

abstract Character. Summary text.

fundstelle List. Document reference details.

vorgangsbezug List. Related Vorgaenge.

deskriptor List. Thesaurus descriptor entries.

Wrapped in a one-row tibble with `num_found` (integer), `cursor` (character), and `documents` (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_drucksache\(\)](#), [dip_bundestag_drucksache_li](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_aktivitaet_list()

## End(Not run)
```

dip_bundestag_drucksache

Get a DIP Drucksache

Description

Get a DIP Drucksache

Usage

```
dip_bundestag_drucksache(
  drucksache_id,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

drucksache_id	Drucksache id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected Drucksache:

id Drucksache identifier (character).
typ Entity type, always "Dokument" (character).
dokumentart Document type, always "Drucksache" (character).
drucksachetyp Drucksache subtype, e.g. "Antrag" (character).
dokumentnummer Document number, e.g. "19/1" (character).
wahlperiode Legislative period (integer).
herausgeber Publisher: "BT" or "BR" (character).
datum Publication date (character, ISO date).
aktualisiert Last update timestamp (character, ISO datetime).
titel Title (character).
autoren_anzeige Up to 4 authors for display (list-column).
autoren_anzahl Total author count (integer).
fundstelle Document reference details (list-column).
pdf_hash MD5 checksum of the PDF file (character or NA).
urheber Authoring bodies (list-column).
vorgangsbezug Up to 4 related Vorgänge (list-column).
vorgangsbezug_anzahl Total number of related Vorgänge (integer).
ressort Government departments involved (list-column).
anlagen Appendix description (character or NA).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_drucksache(68852)

## End(Not run)
```

dip_bundestag_drucksache_list
List DIP Drucksachen

Description

List DIP Drucksachen

Usage

```
dip_bundestag_drucksache_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_drucksachen_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per Drucksache and columns:

- id** Character. Unique Drucksache identifier.
- typ** Character. Entity type, always "Dokument".
- dokumentart** Character. Document type, always "Drucksache".
- drucksachetyp** Character. Drucksache subtype.
- dokumentnummer** Character. Document number.
- wahlperiode** Integer. Legislative period.
- herausgeber** Character. Publisher: "BT" or "BR".
- datum** Character. Publication date.
- aktualisiert** Character. Last update timestamp.
- titel** Character. Title text.
- autoren_anzahl** Integer. Total author count.
- pdf_hash** Character. MD5 checksum of PDF file.
- vorgangsbezug_anzahl** Integer. Total related Vorgaenge.
- anlagen** Character. Appendix description.
- autoren_anzeige** List. Up to 4 display authors.
- fundstelle** List. Document reference details.
- urheber** List. Authoring bodies.
- vorgangsbezug** List. Related Vorgaenge.
- ressort** List. Government departments involved.

Wrapped in a one-row tibble with num_found (integer), cursor (character), and documents (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_drucksache_list()

## End(Not run)
```

dip_bundestag_drucksache_text
Get a DIP Drucksache text

Description

Get a DIP Drucksache text

Usage

```
dip_bundestag_drucksache_text(  
  drucksache_id,  
  params = list(),  
  safe = TRUE,  
  refresh = FALSE  
)
```

Arguments

drucksache_id	Drucksache id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected Drucksache text record. Contains all columns from [dip_bundestag_drucksache\(\)](#) plus:

text Full document text (character).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_drucksache_text(68852)

## End(Not run)
```

```
dip_bundestag_drucksache_text_list
  List DIP Drucksache texts
```

Description

List DIP Drucksache texts

Usage

```
dip_bundestag_drucksache_text_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per Drucksache text and columns:

- id** Character. Unique Drucksache identifier.
- typ** Character. Entity type, always "Dokument".
- dokumentart** Character. Document type, always "Drucksache".
- drucksachetyp** Character. Drucksache subtype.
- dokumentnummer** Character. Document number.
- wahlperiode** Integer. Legislative period.
- herausgeber** Character. Publisher: "BT" or "BR".
- datum** Character. Publication date.
- aktualisiert** Character. Last update timestamp.
- titel** Character. Title text.
- autoren_anzahl** Integer. Total author count.
- pdf_hash** Character. MD5 checksum of PDF file.
- vorgangsbezug_anzahl** Integer. Total related Vorgaenge.
- anlagen** Character. Appendix description.
- text** Character. Full document text.
- autoren_anzeige** List. Up to 4 display authors.
- fundstelle** List. Document reference details.
- urheber** List. Authoring bodies.
- vorgangsbezug** List. Related Vorgaenge.
- ressort** List. Government departments involved.

Wrapped in a one-row tibble with num_found (integer), cursor (character), and documents (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_drucksache_text_list()

## End(Not run)
```

dip_bundestag_person *Get a DIP Person*

Description

Get a DIP Person

Usage

```
dip_bundestag_person(person_id, params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

person_id	Person id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundestag", "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected person:

id Person identifier (character).
nachname Last name (character).
vorname First name (character).
namenszusatz Name affix, e.g. "von der" (character or NA).
typ Entity type, always "Person" (character).
wahlperiode Legislative period of first associated document (integer or NA).
basisdatum Date of first associated document (character or NA).
datum Date of most recent associated document (character or NA).
aktualisiert Last update timestamp (character, ISO datetime).
titel Display title with role (character).
person_roles Roles and alternative names (list-column).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_person(1728)

## End(Not run)
```

```
dip_bundestag_person_list
      List DIP Personen
```

Description

List DIP Personen

Usage

```
dip_bundestag_person_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .

refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per Person and columns:

id Character. Unique person identifier.

nachname Character. Last name.

vorname Character. First name.

namenzusatz Character. Name affix.

typ Character. Entity type, always "Person".

wahlperiode Integer. Legislative period of first document.

basisdatum Character. Date of first associated document.

datum Character. Date of most recent document.

aktualisiert Character. Last update timestamp.

titel Character. Display title with role.

person_roles List. Roles and alternative names.

Wrapped in a one-row tibble with `num_found` (integer), `cursor` (character), and `documents` (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_person_list()

## End(Not run)
```

dip_bundestag_plenarprotokoll
Get a DIP Plenarprotokoll

Description

Get a DIP Plenarprotokoll

Usage

```
dip_bundestag_plenarprotokoll(  
  plenarprotokoll_id,  
  params = list(),  
  safe = TRUE,  
  refresh = FALSE  
)
```

Arguments

plenarprotokoll_id	Plenarprotokoll id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected Plenarprotokoll:

id Plenarprotokoll identifier (character).
dokumentart Document type, always "Plenarprotokoll" (character).
typ Entity type, always "Dokument" (character).
dokumentnummer Document number, e.g. "19/1" (character).
wahlperiode Legislative period (integer).
herausgeber Publisher assignment (character).
datum Session date (character, ISO date).

- aktualisiert** Last update timestamp (character, ISO datetime).
- titel** Title (character).
- fundstelle** Document reference details (list-column).
- pdf_hash** MD5 checksum of the PDF file (character or NA).
- vorgangsbezug** Up to 4 related Vorgänge (list-column).
- vorgangsbezug_anzahl** Total number of related Vorgänge (integer).
- sitzungsbemerkung** Session remark, e.g. "Sondersitzung" (character or NA).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_plenarprotokoll(908)

## End(Not run)
```

```
dip_bundestag_plenarprotokoll_list
      List DIP Plenarprotokolle
```

Description

List DIP Plenarprotokolle

Usage

```
dip_bundestag_plenarprotokoll_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A [tibble](#) with one row per Plenarprotokoll and columns:

id Character. Unique Plenarprotokoll identifier.

dokumentart Character. Document type, always "Plenarprotokoll".

typ Character. Entity type, always "Dokument".

dokumentnummer Character. Document number.

wahlperiode Integer. Legislative period.

datum Character. Session date.

aktualisiert Character. Last update timestamp.

titel Character. Title text.

pdf_hash Character. MD5 checksum of PDF file.

vorgangsbezug_anzahl Integer. Total related Vorgaenge.

sitzungsbemerkung Character. Session remark.

herausgeber List. Publisher assignment.

fundstelle List. Document reference details.

vorgangsbezug List. Related Vorgaenge.

Wrapped in a one-row tibble with `num_found` (integer), `cursor` (character), and `documents` (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_plenarprotokoll_list()

## End(Not run)
```

```
dip_bundestag_plenarprotokoll_text
    Get a DIP Plenarprotokoll text
```

Description

Get a DIP Plenarprotokoll text

Usage

```
dip_bundestag_plenarprotokoll_text(
  plenarprotokoll_id,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

plenarprotokoll_id	Plenarprotokoll id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .

refresh Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected Plenarprotokoll text record. Contains all columns from [dip_bundestag_plenarprotokoll_list\(\)](#) plus:

text Full document text (character).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgang_position\(\)](#), [dip_bundestag_vorgang_position_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_plenarprotokoll_text(908)

## End(Not run)
```

```
dip_bundestag_plenarprotokoll_text_list
  List DIP Plenarprotokoll texts
```

Description

List DIP Plenarprotokoll texts

Usage

```
dip_bundestag_plenarprotokoll_text_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Value

A **tibble** with one row per Plenarprotokoll text and columns:

id Character. Unique Plenarprotokoll identifier.

dokumentart Character. Document type, always "Plenarprotokoll".

typ Character. Entity type, always "Dokument".

dokumentnummer Character. Document number.

wahlperiode Integer. Legislative period.

datum Character. Session date.

aktualisiert Character. Last update timestamp.

titel Character. Title text.

pdf_hash Character. MD5 checksum of PDF file.

vorgangsbezug_anzahl Integer. Total related Vorgaenge.

sitzungsbemerkung Character. Session remark.

text Character. Full document text.

herausgeber List. Publisher assignment.

fundstelle List. Document reference details.

vorgangsbezug List. Related Vorgaenge.

Wrapped in a one-row tibble with `num_found` (integer), `cursor` (character), and `documents` (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_vorgang\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_plenarprotokoll_text_list()

## End(Not run)
```

dip_bundestag_vorgang *Get a DIP Vorgang*

Description

Get a DIP Vorgang

Usage

```
dip_bundestag_vorgang(
  vorgang_id,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

vorgang_id	Vorgang id.
params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bunddev/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns metadata for a single Vorgang.

Value

A one-row tibble for the selected Vorgang:

- id** Vorgang identifier (character).
- typ** Entity type, always "Vorgang" (character).
- beratungsstand** Consultation status, e.g. "Abgeschlossen" (character).
- vorgangstyp** Vorgang type, e.g. "Geschäftsordnung" (character).
- wahlperiode** Legislative period (integer).
- initiative** Initiating factions/bodies (list-column).
- datum** Date of most recent associated document (character, ISO date).
- aktualisiert** Last update timestamp (character, ISO datetime).
- titel** Title (character).
- abstract** Summary text (character or NA).
- sachgebiet** Subject areas (list-column).
- deskriptor** Thesaurus descriptors (list-column of named lists).
- gesta** GESTA reference number (character or NA).
- zustimmungsbeduerftigkeit** Consent requirements (list-column).
- kom** EU COM reference number (character or NA).
- ratsdok** EU Council document number (character or NA).
- verkuendung** Promulgation details (list-column).
- inkrafttreten** Entry-into-force details (list-column).
- archiv** Archive reference (character or NA).
- mitteilung** Supplementary note (character or NA).
- vorgang_verlinkung** Linked Vorgänge (list-column).
- sek** EU SEC reference number (character or NA).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_vorgang(84343)

## End(Not run)
```

```
dip_bundestag_vorgangsposition
    Get a DIP Vorgangsposition
```

Description

Get a DIP Vorgangsposition

Usage

```
dip_bundestag_vorgangsposition(
  vorgangsposition_id,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

vorgangsposition_id	Vorgangsposition id.
params	Named list of query parameters. Common keys: <ul style="list-style-type: none"> format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_vorgangsposition_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble for the selected Vorgangsposition:

id Vorgangsposition identifier (character).
vorgangsposition Step description (character).
zuordnung Assignment: "BT", "BR", "BV", or "EK" (character).
gang Key step in the legislative process (logical).
fortsetzung Continuation of a previous debate (logical).
nachtrag Addendum entry (logical).
vorgangstyp Parent Vorgang type (character).
typ Entity type, always "Vorgangsposition" (character).
titel Title of the parent Vorgang (character).
dokumentart Document type: "Drucksache" or "Plenarprotokoll" (character).
vorgang_id Parent Vorgang id (character).
datum Date of associated document (character, ISO date).
aktualisiert Last update timestamp (character, ISO datetime).
fundstelle Document reference details (list-column).
urheber Authoring bodies (list-column).
ueberweisung Committee referrals (list-column).
aktivitaet_anzeige Up to 4 associated activities for display (list-column).
aktivitaet_anzahl Total number of associated activities (integer).
ressort Government departments involved (list-column).
beschlussfassung Voting decisions (list-column).
ratsdok EU Council document number (character or NA).
kom EU COM reference number (character or NA).
sek EU SEC reference number (character or NA).
mitberaten Co-deliberated Vorgangspositionen (list-column).
abstract Summary text (character or NA).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
dip_bundestag_vorgangsposition(173376)

## End(Not run)
```

dip_bundestag_vorgangsposition_list
List DIP Vorgangsposition entries

Description

List DIP Vorgangsposition entries

Usage

```
dip_bundestag_vorgangsposition_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundestag_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns metadata for Vorgangsposition entries.

Value

A [tibble](#) with one row per Vorgangsposition and columns:

- id** Character. Unique Vorgangsposition identifier.
- vorgangsposition** Character. Step description.
- gang** Logical. Key step in legislative process.
- fortsetzung** Logical. Continuation of prior debate.
- nachtrag** Logical. Addendum entry.
- vorgangstyp** Character. Parent Vorgang type.
- typ** Character. Entity type, always "Vorgangsposition".
- titel** Character. Title of the parent Vorgang.
- dokumentart** Character. Document type.
- vorgang_id** Character. Parent Vorgang identifier.
- datum** Character. Date of associated document.
- aktualisiert** Character. Last update timestamp.
- aktivitaet_anzahl** Integer. Total associated activities.
- ratsdok** Character. EU Council document number.
- kom** Character. EU COM reference number.
- sek** Character. EU SEC reference number.
- abstract** Character. Summary text.
- zuordnung** List. Assignment to BT/BR/BV/EK.
- fundstelle** List. Document reference details.
- urheber** List. Authoring bodies.
- ueberweisung** List. Committee referrals.
- aktivitaet_anzeige** List. Up to 4 display activities.
- ressort** List. Government departments involved.
- beschlussfassung** List. Voting decisions.
- mitberaten** List. Co-deliberated Vorgangspositionen.

Wrapped in a one-row tibble with `num_found` (integer), `cursor` (character), and `documents` (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang\(\)](#), [dip_bundestag_vorgang_list\(\)](#), [dip_bundestag_vorgangsposition\(\)](#)

Examples

```
## Not run:
dip_bundestag_vorgangsposition_list()

## End(Not run)
```

```
dip_bundestag_vorgang_list
  List DIP Vorgang entries
```

Description

List DIP Vorgang entries

Usage

```
dip_bundestag_vorgang_list(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters. Common keys: format Response format ("json" recommended; default set by adapter). apikey Optional API key as query parameter (prefer auth header). fuzzy Free-text search string (character). sort Sort field/order definition (character). cursor Cursor for pagination (character). limit Page size/maximum number of documents (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundestag_vorgang_list_cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to flatten_mode. Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when flatten = TRUE: "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns metadata for DIP Vorgang entries. Requires an API key. Obtain a key from <https://dip.bundestag.de/ueber-dip/hilfe/api>.

Configure authentication via [bunddev_auth_set\(\)](#) or set the DIP_BUNDESTAG_API_KEY environment variable directly.

Value

A [tibble](#) with one row per Vorgang and columns:

id Character. Unique Vorgang identifier.
typ Character. Entity type, always "Vorgang".
beratungsstand Character. Consultation status.
vorgangstyp Character. Vorgang type.
wahlperiode Integer. Legislative period.
datum Character. Date of last associated document.
aktualisiert Character. Last update timestamp.
titel Character. Title text.
abstract Character. Summary text.
gesta Character. GESTA reference number.
kom Character. EU COM reference number.
ratsdok Character. EU Council document number.
archiv Character. Archive reference.
mitteilung Character. Supplementary note.
sek Character. EU SEC reference number.
initiative List. Initiating factions or bodies.
sachgebiet List. Subject areas.
deskriptor List. Thesaurus descriptor entries.
zustimmungsbeduerftigkeit List. Consent requirements.
verkuendung List. Promulgation details.
inkrafttreten List. Entry-into-force details.
vorgang_verlinkung List. Linked Vorgaenge.

Wrapped in a one-row tibble with num_found (integer), cursor (character), and documents (list-column) when not yet unpacked.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters. [bunddev_auth_set\(\)](#) to configure authentication.

Other DIP Bundestag: [dip_bundestag_aktivitaet\(\)](#), [dip_bundestag_aktivitaet_list\(\)](#), [dip_bundestag_drucksache_list\(\)](#), [dip_bundestag_drucksache_text\(\)](#), [dip_bundestag_drucksache_text_list\(\)](#), [dip_bundestag_person\(\)](#), [dip_bundestag_person_list\(\)](#), [dip_bundestag_plenarprotokoll\(\)](#), [dip_bundestag_plenarprotokoll_list\(\)](#), [dip_bundestag_plenarprotokoll_text\(\)](#), [dip_bundestag_plenarprotokoll_text_list\(\)](#), [dip_bundestag_vorgang\(\)](#), [dip_bundestag_vorgangsposition\(\)](#), [dip_bundestag_vorgangsposition_list\(\)](#)

Examples

```
## Not run:
# Recommended: use bunddev_auth_set
Sys.setenv(DIP_BUNDESTAG_API_KEY = "<api-key>")
bunddev_auth_set(
  "dip_bundestag",
  type = "api_key",
  env_var = "DIP_BUNDESTAG_API_KEY",
  scheme = "ApiKey"
)
dip_bundestag_vorgang_list()

## End(Not run)
```

dwd_alpine_forecast_text

Fetch DWD alpine forecast text

Description

Fetch DWD alpine forecast text

Usage

```
dwd_alpine_forecast_text(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Retrieves alpine forecast text from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A one-row tibble:

text Alpine forecast text (character).

See Also

[dwd_sea_warning_text\(\)](#) and [dwd_avalanche_warnings\(\)](#).

Other DWD: [dwd_avalanche_warnings\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_station_overview\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:
dwd_alpine_forecast_text()

## End(Not run)
```

```
dwd_avalanche_warnings
      Fetch DWD avalanche warnings
```

Description

Fetch DWD avalanche warnings

Usage

```
dwd_avalanche_warnings(safe = TRUE, refresh = FALSE)
```

Arguments

<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Retrieves avalanche warnings from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A one-row tibble:

raw Full parsed avalanche warning payload (list-column).

See Also

[dwd_alpine_forecast_text\(\)](#) and [dwd_sea_warning_text\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_station_overview\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:
dwd_avalanche_warnings()

## End(Not run)
```

dwd_coast_warnings *Fetch DWD coastal warnings*

Description

Fetch DWD coastal warnings

Usage

```
dwd_coast_warnings(
  language = c("de", "en"),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

language	Language code ("de" or "en").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Retrieves coastal warnings from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A tibble with one row per coastal warning:

region_id Warning region identifier (character).

type Warning type code (numeric).

level Warning severity level (numeric).

description Warning description HTML (character).

description_text Warning description plain text (character).

event Event name (character).

headline Warning headline (character).

See Also

[dwd_warnings_nowcast\(\)](#) and [dwd_municipality_warnings\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_avalanche_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_station_overview\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:
dwd_coast_warnings(language = "de", flatten = TRUE)

## End(Not run)
```

dwd_crowd_reports	<i>Fetch DWD crowd reports</i>
-------------------	--------------------------------

Description

Fetch DWD crowd reports

Usage

```
dwd_crowd_reports(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Retrieves crowd-sourced weather reports from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A tibble with one row per crowd report:

meldung_id Report identifier (numeric).
timestamp Report timestamp in milliseconds (numeric).
timestamp_time Parsed report time (POSIXct).
lat Latitude (character).
lon Longitude (character).
place Place name (character).
category Weather category (character).
auspraegung Severity/manifestation (character).
zusatz_attribute Additional attributes (list-column).

See Also

[dwd_station_overview\(\)](#) and [dwd_warnings_nowcast\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_avalanche_warnings\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_station_overview\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:
dwd_crowd_reports(flatten = TRUE)

## End(Not run)
```

`dwd_municipality_warnings`*Fetch DWD municipality warnings*

Description

Fetch DWD municipality warnings

Usage

```
dwd_municipality_warnings(  
  language = c("de", "en"),  
  safe = TRUE,  
  refresh = FALSE,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

<code>language</code>	Language code ("de" or "en").
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Retrieves municipality warnings from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A tibble with the same columns as `dwd_warnings_nowcast()`.

See Also

[dwd_warnings_nowcast\(\)](#) and [dwd_coast_warnings\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_avalanche_warnings\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_station_overview\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:  
dwd_municipality_warnings(language = "de", flatten = TRUE)  
  
## End(Not run)
```

```
dwd_sea_warning_text  Fetch DWD sea warning text
```

Description

Fetch DWD sea warning text

Usage

```
dwd_sea_warning_text(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Retrieves sea warning text from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A one-row tibble:

text Sea warning text (character).

See Also

[dwd_alpine_forecast_text\(\)](#) and [dwd_avalanche_warnings\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_avalanche_warnings\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_station_overview\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:
dwd_sea_warning_text()

## End(Not run)
```

dwd_station_overview *Fetch DWD station overview data*

Description

Fetch DWD station overview data

Usage

```
dwd_station_overview(
  station_ids,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

station_ids	Station identifiers.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Retrieves station overview data from the DWD App API. API documentation: <https://dwd.api.bund.dev>. resources or the API documentation.

Value

A tibble with one row per station id:

station_id Station id (character).

forecast1 Forecast block 1 (list-column).

forecast2 Forecast block 2 (list-column).

forecast_start Forecast start timestamp (character).

forecast_start_time Forecast start as POSIXct in Europe/Berlin.

days Daily forecast summaries (list-column).

warnings Warning entries (list-column).

three_hour_summaries Three-hour summaries (list-column).

forecast1_start_time forecast1 start as POSIXct.

forecast2_start_time forecast2 start as POSIXct.

See Also

[dwd_crowd_reports\(\)](#) and [dwd_warnings_nowcast\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_avalanche_warnings\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_warnings_nowcast\(\)](#)

Examples

```
## Not run:  
dwd_station_overview(c("10865", "G005"), flatten = TRUE)  
  
## End(Not run)
```

dwd_warnings_nowcast *Fetch DWD nowcast warnings*

Description

Fetch DWD nowcast warnings

Usage

```
dwd_warnings_nowcast(  
  language = c("de", "en"),  
  safe = TRUE,  
  refresh = FALSE,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

language	Language code ("de" or "en").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Retrieves nowcast weather warnings from the DWD App API. API documentation: <https://dwd.api.bund.dev>.

Value

A tibble with one row per warning:

type Warning type code (numeric).

level Warning severity level (numeric).

start Start timestamp in milliseconds (numeric).

start_time Parsed start time (POSIXct).

end End timestamp in milliseconds (numeric).

end_time Parsed end time (POSIXct).

description Warning description HTML (character).

description_text Warning description plain text (character).

event Event name (character).

headline Warning headline (character).

regions Affected regions (list-column).

urls Related URLs (list-column).

is_vorabinfo Whether this is a preliminary warning (logical).

With `flatten = TRUE`, the list-columns are transformed according to `flatten_mode`.

See Also

[dwd_municipality_warnings\(\)](#) and [dwd_coast_warnings\(\)](#).

Other DWD: [dwd_alpine_forecast_text\(\)](#), [dwd_avalanche_warnings\(\)](#), [dwd_coast_warnings\(\)](#), [dwd_crowd_reports\(\)](#), [dwd_municipality_warnings\(\)](#), [dwd_sea_warning_text\(\)](#), [dwd_station_overview\(\)](#)

Examples

```
## Not run:
dwd_warnings_nowcast(language = "de", flatten = TRUE)

## End(Not run)
```

eco_visio_counters *List Eco-Visio counters for an organization*

Description

List Eco-Visio counters for an organization

Usage

```
eco_visio_counters(id_organisme, safe = TRUE, refresh = FALSE)
```

Arguments

id_organisme	Organization ID (e.g., 4586 for Bike Count Display).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundles/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The Eco-Visio API provides access to bicycle and pedestrian counter data. This function returns all counters registered for a specific organization. API documentation: <https://github.com/bundesAPI/eco-visio-api>.

Value

A tibble with one row per counter. Common columns:

- id** Counter identifier (character).
- name** Counter name/label (character).
- lat** Latitude (numeric).
- lon** Longitude (numeric).
- flow_ids** Flow/practice identifiers (list-column).
- raw** Original counter object (list-column).

See Also

[eco_visio_data\(\)](#) to retrieve measurement data for a counter.
Other Eco Visio: [eco_visio_data\(\)](#)

Examples

```
## Not run:
eco_visio_counters(4586)

## End(Not run)
```

eco_visio_data	<i>Get Eco-Visio counter measurement data</i>
----------------	---

Description

Get Eco-Visio counter measurement data

Usage

```
eco_visio_data(
  id_organisme,
  id_pdc,
  interval,
  flow_ids,
  begin = NULL,
  end = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

id_organisme	Organization ID.
id_pdc	Counter ID.
interval	Aggregation interval (1-6: 1=15min, 2=hours, 3=days, 4=weeks, 5=months, 6=years).
flow_ids	Practice IDs (semicolon-separated string or character vector).
begin	Optional start date (Date or "YYYY-MM-DD" string).
end	Optional end date (Date or "YYYY-MM-DD" string).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns measurement data for a specific counter. The `flow_ids` parameter specifies which traffic types to include (e.g., pedestrians, cyclists). Use `eco_visio_counters()` to discover available counters and their flow IDs.

Value

A tibble with one row per measurement interval:

date Interval timestamp/date label from the API (character).

count Count value for the selected flow(s) (integer).

date_time date converted to POSIXct in Europe/Berlin.

See Also

[eco_visio_counters\(\)](#) to list available counters.

Other Eco Visio: [eco_visio_counters\(\)](#)

Examples

```
## Not run:
# Get daily data for a counter
eco_visio_data(
  id_organisme = 4586,
  id_pdc = 100125331,
  interval = 4,
  flow_ids = "101125331"
)

## End(Not run)
```

feiertage_list

List German public holidays

Description

List German public holidays

Usage

```
feiertage_list(
  jahr = NULL,
  nur_land = NULL,
  nur_daten = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

jahr	Year to query (defaults to current year on the API).
nur_land	Optional Bundesland code to filter.
nur_datan	Logical; return only date values (1) or include names (0).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The Feiertage API returns German public holidays from a Wikipedia-based dataset. The endpoint supports filtering by year and Bundesland. Official docs: <https://github.com/bundesAPI/feiertage-api>.

Value

A tibble with one row per holiday:

region Bundesland code (character, NA for nationwide results).

holiday Holiday name (character).

date Date in YYYY-MM-DD format (character).

note Optional holiday note from the API (character or NA).

date_time Date as POSIXct in Europe/Berlin.

See Also

[bunddev_parameters\(\)](#) for available query parameters.

Examples

```
## Not run:  
feiertage_list(jahr = 2024)  
feiertage_list(jahr = 2024, nur_land = "BY")  
  
## End(Not run)
```

handelsregister_search

Search the Handelsregister portal

Description

This adapter scrapes the public Handelsregister search form because no official OpenAPI specification is available for this service.

Usage

```
handelsregister_search(
  schlagwoerter,
  schlagwort_optionen = c("all", "min", "exact"),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

schlagwoerter	Search terms.
schlagwort_optionen	Keyword options.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundlesAPI/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

The Handelsregister search is provided via a public web form. This helper automates the form flow and parses the result table into a tidy tibble. API documentation: <https://github.com/bundesAPI/handelsregister>.

The registry notes that more than 60 requests per hour may violate the terms of use. Use `safe = TRUE` to respect the built-in rate limiting.

Value

A tibble with one row per register entry:

name Company/entity name (character).

court Register court and register label (character).

register_num Extracted register number (character).

state Federal state (character).

status Status text from the listing (character).

status_current Uppercase status code derived from status (character).

documents_text Document summary text (character).

documents_count Number of linked documents (integer).

documents_links Document URLs (list-column of character vectors).

history Additional parsed history tokens (list-column).

See Also

[bunddev_rate_limit_get\(\)](#) to inspect the configured limit.

Examples

```
## Not run:
handelsregister_search("Deutsche Bahn", schlagwort_optionen = "all")

## End(Not run)
```

hilfsmittel_nachweisschema

Get Hilfsmittel verification schema details

Description

Get Hilfsmittel verification schema details

Usage

```
hilfsmittel_nachweisschema(
  id,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

<code>id</code>	Nachweisschema id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel/cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns detail information for a Nachweisschema. API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A `tibble` with one row for the requested verification schema and columns:

id Character. Schema identifier.

name Character. Schema name.

produktmusterErforderlich Logical. Whether a product sample is required.

nachweisschemaKategorieZuweisungen List. Category assignments (nested objects).

nachweisAbschnittAnforderungen List. Section requirements (nested objects).

See Also

Other Hilfsmittel: [hilfsmittel_produk_t\(\)](#), [hilfsmittel_produk_tart\(\)](#), [hilfsmittel_produk_te\(\)](#), [hilfsmittel_produk_tgruppe\(\)](#), [hilfsmittel_tree\(\)](#), [hilfsmittel_untergruppe\(\)](#)

Examples

```
## Not run:
hilfsmittel_nachweisschema("a3d37017-2c91-4d6d-bbbe-4002d2868044")

## End(Not run)
```

hilfsmittel_produk *Get Hilfsmittel product details*

Description

Get Hilfsmittel product details

Usage

```
hilfsmittel_produk(
  id,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

id	Produkt id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesAPI/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns detail information for a single product. API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A [tibble](#) with one row for the requested product and columns:

organisationId Character. Organisation identifier.

produktartId Character. Product-type identifier.

nummer Integer. Sequential product number.

name Character. Product name.
artikelnummern List. Article numbers (character vector).
typenAusfuehrungen List. Type/model variants (character vector).
aufnahmeDatum Character. Admission date (ISO-8601).
aenderungsDatum Character. Last-modified date (ISO-8601).
zehnSteller Character. Ten-digit product code.
herstellerName Character. Manufacturer name.
istHerausgenommen Logical. Whether product is withdrawn.
istAbrechnungsposition Logical. Whether product is a billing item.
id Character. Product identifier.
displayName Character. Combined display label.
produktgruppeNummer Integer. Product-group number.
anwendungsortNummer Integer. Application-site number.
untergruppeNummer Integer. Subgroup number.
produktartNummer Integer. Product-type number.
produktartBezeichnung Character. Product-type name.
kontruktionsmerkmale List. Construction features (nested objects).
aufnahmeDatum_time POSIXct. Parsed admission date (Europe/Berlin).
aenderungsDatum_time POSIXct. Parsed last-modified date (Europe/Berlin).

See Also

Other Hilfsmittel: [hilfsmittel_nachweisschema\(\)](#), [hilfsmittel_produkart\(\)](#), [hilfsmittel_produkte\(\)](#), [hilfsmittel_produkgruppe\(\)](#), [hilfsmittel_tree\(\)](#), [hilfsmittel_untergruppe\(\)](#)

Examples

```
## Not run:
hilfsmittel_produkart("f41f52a6-5d2d-4dd3-9d0e-39675ceca7f3")

## End(Not run)
```

hilfsmittel_produkart

Get Hilfsmittel product type details

Description

Get Hilfsmittel product type details

Usage

```

hilfsmittel_produkart(
  id,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)

```

Arguments

<code>id</code>	Produktart id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns details for a product type (Produktart). API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A `tibble` with one row for the requested product type and columns:

id Character. Product-type identifier.

versionId Character. Internal version identifier.

nummer Integer. Product-type number.

beschreibung Character. Description text.

bezeichnung Character. Product-type name.

untergruppeId Character. Parent subgroup identifier.

indikation Character. Indication text.

siebenSteller Character. Seven-digit code.

isNnElement Logical. Internal flag (usage unclear).

konstruktionsmerkmalschemaId Character. Construction-feature schema id.

untergruppe Integer. Always NA.

aufnahmeDatum Character. Admission date (ISO-8601).

aenderungDatum Character. Last-modified date (ISO-8601).

aufnahmeDatum_time POSIXct. Parsed admission date (Europe/Berlin).

aenderungDatum_time POSIXct. Parsed last-modified date (Europe/Berlin).

See Also

Other Hilfsmittel: [hilfsmittel_nachweisschema\(\)](#), [hilfsmittel_produkte\(\)](#), [hilfsmittel_produkte\(\)](#), [hilfsmittel_produkgruppe\(\)](#), [hilfsmittel_tree\(\)](#), [hilfsmittel_untergruppe\(\)](#)

Examples

```
## Not run:
hilfsmittel_produkart("e6b913ef-cf21-4c5f-826d-f866516c3c65")

## End(Not run)
```

hilfsmittel_produkte *List Hilfsmittel products*

Description

List Hilfsmittel products

Usage

```
hilfsmittel_produkte(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed.

"json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.

"unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.

Warning: this can significantly increase the number of rows.

Details

Returns the full product list (large payload). API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A **tibble** with one row per product and columns:

organisationId Character. Organisation identifier.

produktartId Character. Product-type identifier.

nummer Integer. Sequential product number.

name Character. Product name.

artikelnummern List. Article numbers (character vector).

typenAusfuehrungen List. Type/model variants (character vector).

aufnahmeDatum Character. Admission date (ISO-8601).

aenderungsDatum Character. Last-modified date (ISO-8601).

zehnSteller Character. Ten-digit product code.

herstellerName Character. Manufacturer name.

istHerausgenommen Logical. Whether product is withdrawn.

istAbrechnungsposition Logical. Whether product is a billing item.

id Character. Product identifier.

displayName Character. Combined display label.

aufnahmeDatum_time POSIXct. Parsed admission date (Europe/Berlin).

aenderungsDatum_time POSIXct. Parsed last-modified date (Europe/Berlin).

See Also

Other Hilfsmittel: [hilfsmittel_nachweisschema\(\)](#), [hilfsmittel_produkte\(\)](#), [hilfsmittel_produkart\(\)](#), [hilfsmittel_produkgruppe\(\)](#), [hilfsmittel_tree\(\)](#), [hilfsmittel_untergruppe\(\)](#)

Examples

```
## Not run:  
hilfsmittel_produkte()  
  
## End(Not run)
```

 hilfsmittel_produkgruppe

Get Hilfsmittel product group details

Description

Get Hilfsmittel product group details

Usage

```

hilfsmittel_produkgruppe(
  id,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)

```

Arguments

id	Produktgruppe id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddesAPI/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns details for a product group (Produktgruppe). API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A [tibble](#) with one row for the requested product group and columns:

id Character. Product group identifier.

versionId Character. Internal version identifier.

bezeichnung Character. Product group name.

nummer Integer. Product group number.

definition Character. Description text.

indikation Character. Indication text.

querverweis Character. Cross-reference to other groups.

isNnElement Logical. Internal flag (usage unclear).

sachbearbeiterGkvId Character. GKV case-worker identifier.

sachbearbeiterGkv Character. Always NA.

sachbearbeiterMdsId Character. MDS case-worker identifier.

sachbearbeiterMds Character. Always NA.

aufnahmeDatum Character. Admission date (ISO-8601).

aenderungDatum Character. Last-modified date (ISO-8601).

veroeffentlichungsDokumentFileStoreId Character. Publication document id.

merkblattFileStoreId Character. Always NA.

isCollectionReviewable Logical. Internal flag (usage unclear).

aufnahmeDatum_time POSIXct. Parsed admission date (Europe/Berlin).

aenderungDatum_time POSIXct. Parsed last-modified date (Europe/Berlin).

See Also

Other Hilfsmittel: [hilfsmittel_nachweisschema\(\)](#), [hilfsmittel_produkkt\(\)](#), [hilfsmittel_produkktart\(\)](#), [hilfsmittel_produkkte\(\)](#), [hilfsmittel_tree\(\)](#), [hilfsmittel_untergruppe\(\)](#)

Examples

```
## Not run:
tree <- hilfsmittel_tree(level = 1)
hilfsmittel_produkktgruppe(tree$id[[1]])

## End(Not run)
```

hilfsmittel_tree	<i>List Hilfsmittel tree nodes</i>
------------------	------------------------------------

Description

List Hilfsmittel tree nodes

Usage

```
hilfsmittel_tree(level, safe = TRUE, refresh = FALSE)
```

Arguments

level	Tree level to retrieve (1-4).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns nodes from the Hilfsmittel product tree up to the selected level. API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A **tibble** with one row per tree node and columns:

- id** Character. Node identifier.
- parentId** Character. Parent node identifier (NA for roots).
- displayValue** Character. Node display label.
- xStellerDisplayValue** Character. xSteller with label.
- xSteller** Character. Position code of the node.
- level** Integer. Tree depth level.

See Also

Other Hilfsmittel: [hilfsmittel_nachweisschema\(\)](#), [hilfsmittel_produktdatum\(\)](#), [hilfsmittel_produktdatumart\(\)](#), [hilfsmittel_produktdatumtyp\(\)](#), [hilfsmittel_produktdatumgruppe\(\)](#), [hilfsmittel_untergruppe\(\)](#)

Examples

```
## Not run:
hilfsmittel_tree(level = 1)

## End(Not run)
```

hilfsmittel_untergruppe

Get Hilfsmittel subgroup details

Description

Get Hilfsmittel subgroup details

Usage

```

hilfsmittel_untergruppe(
  id,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)

```

Arguments

<code>id</code>	Untergruppe id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns details for a subgroup (Untergruppe). API documentation: <https://github.com/bundesAPI/hilfsmittel-api>.

Value

A **tibble** with one row for the requested subgroup and columns:

- id** Character. Subgroup identifier.
- versionId** Character. Internal version identifier.
- nummer** Integer. Subgroup number.
- anwendungsortId** Character. Application-site identifier.
- bezeichnung** Character. Subgroup name.
- produktgruppeId** Character. Parent product-group identifier.
- nachweisschemaId** Character. Verification-schema identifier.
- sechsSteller** Character. Six-digit code.
- isNnElement** Logical. Internal flag (usage unclear).
- produktgruppe** Numeric. Always NA.

anwendungsort Numeric. Always NA.

aufnahmeDatum Character. Admission date (ISO-8601).

aenderungsDatum Character. Last-modified date (ISO-8601).

aufnahmeDatum_time POSIXct. Parsed admission date (Europe/Berlin).

aenderungsDatum_time POSIXct. Parsed last-modified date (Europe/Berlin).

See Also

Other Hilfsmittel: [hilfsmittel_nachweisschema\(\)](#), [hilfsmittel_produkkt\(\)](#), [hilfsmittel_produkktart\(\)](#), [hilfsmittel_produkte\(\)](#), [hilfsmittel_produkgruppe\(\)](#), [hilfsmittel_tree\(\)](#)

Examples

```
## Not run:
hilfsmittel_untergruppe("c92d1976-d3cb-4b9f-bcdf-805272a9ea86")

## End(Not run)
```

hochwasserzentralen_bundeslaender

List flood portal states and connected regions

Description

List flood portal states and connected regions

Usage

```
hochwasserzentralen_bundeslaender(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundeslaender", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns metadata for all Bundeslaender and connected regions in the hochwasserzentralen.de portal.
API documentation: <https://hochwasserzentralen.api.bund.dev/>.

Value

A tibble with one row per Bundesland/region entry. Columns are created dynamically from the upstream service response fields. If the response contains GeoJSON features, each feature's properties are used as columns; otherwise each top-level field becomes a column. Scalar values become atomic columns and nested structures become list-columns.

See Also

Other Hochwasserzentralen: [hochwasserzentralen_bundesland_geojson\(\)](#), [hochwasserzentralen_bundesland_info_hochwasserzentralen_lagepegel\(\)](#), [hochwasserzentralen_pegel_info\(\)](#)

Examples

```
## Not run:
hochwasserzentralen_bundeslaender()

## End(Not run)
```

```
hochwasserzentralen_bundesland_geojson
  Get Bundesland GeoJSON boundaries
```

Description

Get Bundesland GeoJSON boundaries

Usage

```
hochwasserzentralen_bundesland_geojson(
  version,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

version	GeoJSON version identifier (e.g., "20211130").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.

`flatten_mode` How to handle list columns when `flatten = TRUE`:

- "drop" Remove list columns entirely. Use when nested data is not needed.
- "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.
- "unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.
Warning: this can significantly increase the number of rows.

Details

Returns GeoJSON boundaries for Bundeslaender and connected regions from the hochwasserzentralen.de portal. API documentation: <https://hochwasserzentralen.api.bund.dev/>.

Value

A one-row tibble with:

- type** GeoJSON object type (character).
- name** GeoJSON dataset name (character).
- features** GeoJSON feature list (list-column).

See Also

Other Hochwasserzentralen: [hochwasserzentralen_bundeslaender\(\)](#), [hochwasserzentralen_bundesland_info\(\)](#), [hochwasserzentralen_lagepegel\(\)](#), [hochwasserzentralen_pegel_info\(\)](#)

Examples

```
## Not run:
hochwasserzentralen_bundesland_geojson("20211130", flatten = TRUE)

## End(Not run)
```

hochwasserzentralen_bundesland_info

Get flood portal metadata for a Bundesland

Description

Get flood portal metadata for a Bundesland

Usage

```
hochwasserzentralen_bundesland_info(
  bundesland_id,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

bundesland_id	Bundesland id (e.g., "HE").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns metadata for a single Bundesland or region in the hochwasserzentralen.de portal. API documentation: <https://hochwasserzentralen.api.bund.dev/>.

Value

A one-row tibble of metadata for the selected Bundesland id. Columns are created dynamically from the upstream service response fields. If the response contains GeoJSON features, each feature's properties are used as columns; otherwise each top-level field becomes a column. Scalar values become atomic columns and nested structures become list-columns.

See Also

Other Hochwasserzentralen: [hochwasserzentralen_bundeslaender\(\)](#), [hochwasserzentralen_bundesland_geojson\(\)](#), [hochwasserzentralen_lagepegel\(\)](#), [hochwasserzentralen_pegel_info\(\)](#)

Examples

```
## Not run:  
hochwasserzentralen_bundesland_info("HE")  
  
## End(Not run)
```

```
hochwasserzentralen_lagepegel  
  List flood gauge locations
```

Description

List flood gauge locations

Usage

```
hochwasserzentralen_lagepegel(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns latitude/longitude coordinates for all pegel identifiers available in the hochwasserzentralen.de portal. API documentation: <https://hochwasserzentralen.api.bund.dev/>.

Value

A tibble with one row per gauge location. Columns are created dynamically from the upstream service response fields. If the response contains GeoJSON features, each feature's properties are used as columns; otherwise each top-level field becomes a column. Scalar values become atomic columns and nested structures become list-columns.

See Also

Other Hochwasserzentralen: [hochwasserzentralen_bundeslaender\(\)](#), [hochwasserzentralen_bundesland_geojson\(\)](#), [hochwasserzentralen_bundesland_info\(\)](#), [hochwasserzentralen_pegel_info\(\)](#)

Examples

```
## Not run:
hochwasserzentralen_lagepegel()

## End(Not run)
```

hochwasserzentralen_pegel_info
Get flood gauge information

Description

Get flood gauge information

Usage

```
hochwasserzentralen_pegel_info(pegelnummer, safe = TRUE, refresh = FALSE)
```

Arguments

pegelnummer	Pegelnummer identifier (e.g., "HE_24820206").
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde" "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns metadata for a single flood gauge (pegel) from hochwasserzentralen.de. API documentation: <https://hochwasserzentralen.api.bund.dev/>.

Value

A one-row tibble of gauge metadata for the selected `pegelnummer`. Columns are created dynamically from the upstream service response fields. If the response contains GeoJSON features, each feature's properties are used as columns; otherwise each top-level field becomes a column. Scalar values become atomic columns and nested structures become list-columns.

See Also

Other Hochwasserzentralen: [hochwasserzentralen_bundeslaender\(\)](#), [hochwasserzentralen_bundesland_geojson\(\)](#), [hochwasserzentralen_bundesland_info\(\)](#), [hochwasserzentralen_lagepegel\(\)](#)

Examples

```
## Not run:
hochwasserzentralen_pegel_info("HE_24820206")

## End(Not run)
```

jobsuche_logo	<i>Fetch Jobsuche employer logo</i>
---------------	-------------------------------------

Description

Fetch Jobsuche employer logo

Usage

```
jobsuche_logo(hash_id, safe = TRUE, refresh = FALSE)
```

Arguments

hash_id	Logo hash id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev_auth_set", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the raw logo bytes for a given employer hash id. Use this together with listings returned by [jobsuche_search\(\)](#) or [jobsuche_search_app\(\)](#).

See <https://jobsuche.api.bund.dev> for API documentation.

Value

A one-row tibble with:

hash_id Requested logo hash id (character).

logo Raw logo bytes (list-column with raw vector).

See Also

[jobsuche_search\(\)](#) for listings and [bunddev_auth_set\(\)](#) for auth setup.

Other Jobsuche: [jobsuche_search\(\)](#), [jobsuche_search_app\(\)](#)

Examples

```
## Not run:
Sys.setenv(JOBBOERSE_API_KEY = "jobboerse-jobsuche")
bunddev_auth_set("jobsuche", type = "api_key", env_var = "JOBBOERSE_API_KEY")
logo <- jobsuche_logo("abc123")

## End(Not run)
```

jobsuche_search

Search Jobsuche listings

Description

Search Jobsuche listings

Usage

```
jobsuche_search(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters: was Free-text job/keyword query (character). wo Location query (character). berufsfeld Occupational field filter (character). page Page index (integer). size Number of results per page (integer). arbeitgeber Employer filter (character). veroeffentlichseit Only offers published within N days (integer). zeitarbeit Filter temporary-agency postings (TRUE/FALSE). angebotsart Offer type code(s), semicolon-separated (integer/character). befristung Contract duration code(s), semicolon-separated (integer/character). arbeitszeit Work time model code(s), semicolon-separated (character). behinderung Accessibility/disability filter (TRUE/FALSE). corona Only postings with corona flag (TRUE/FALSE). umkreis Radius in kilometers around wo (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

The Jobsuche API provides access to job listings from the Bundesagentur fuer Arbeit. Authentication is required via an API key passed as X-API-Key. See <https://jobsuche.api.bund.dev> for the official API documentation.

Use `bunddev_auth_set()` to configure the key and `bunddev_parameters()` to discover supported query parameters.

Value

A tibble with one row per job listing:

hash_id Posting hash id (character).
beruf Job title/occupation (character).
refnr Reference number (character).
arbeitgeber Employer name (character).
aktuelle_veroeffentlichungsdatum Published date/time text (character).
aktuelle_veroeffentlichungsdatum_time Published timestamp as POSIXct in Europe/Berlin.
eintrittsdatum Start date text (character).
eintrittsdatum_time Start timestamp as POSIXct in Europe/Berlin.
arbeitsort_plz Postal code (character).
arbeitsort_ort City/locality (character).
arbeitsort_strasse Street (character).
arbeitsort_region Region/state (character).
arbeitsort_land Country code/name (character).
arbeitsort_lat Latitude (numeric).
arbeitsort_lon Longitude (numeric).
modifikations_timestamp Modification timestamp text (character).
modifikations_timestamp_time Modification timestamp as POSIXct in Europe/Berlin.
page Returned page number (character).
size Returned page size (character).
max_ergebnisse Total result count (character).
facetten Facet metadata (list-column).

See Also

[jobsuche_search_app\(\)](#) for the app endpoint, [jobsuche_logo\(\)](#) to fetch employer logos, and [bunddev_auth_set\(\)](#) for authentication.

Other Jobsuche: [jobsuche_logo\(\)](#), [jobsuche_search_app\(\)](#)

Examples

```
## Not run:
Sys.setenv(JOBBOERSE_API_KEY = "jobboerse-jobsuche")
bunddev_auth_set("jobsuche", type = "api_key", env_var = "JOBBOERSE_API_KEY")
jobsuche_search(params = list(was = "data", size = 5), flatten = TRUE)

## End(Not run)
```

jobsuche_search_app *Search Jobsuche listings (app endpoint)*

Description

Search Jobsuche listings (app endpoint)

Usage

```
jobsuche_search_app(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	<p>Named list of query parameters:</p> <ul style="list-style-type: none"> was Free-text job/keyword query (character). wo Location query (character). berufsfeld Occupational field filter (character). page Page index (integer). size Number of results per page (integer). arbeitgeber Employer filter (character). veroeffentlichseit Only offers published within N days (integer). zeitarbeit Filter temporary-agency postings (TRUE/FALSE). angebotsart Offer type code(s), semicolon-separated (integer/character). befristung Contract duration code(s), semicolon-separated (integer/character). arbeitszeit Work time model code(s), semicolon-separated (character). behinderung Accessibility/disability filter (TRUE/FALSE). corona Only postings with corona flag (TRUE/FALSE). umkreis Radius in kilometers around wo (integer).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed.

"json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.

"unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.

Warning: this can significantly increase the number of rows.

Details

This uses the mobile app endpoint of the Jobsuche API. It shares the same authentication mechanism and parameters as `jobsuche_search()`.

See <https://jobsuche.api.bund.dev> for API documentation.

Value

A tibble with the same columns as `jobsuche_search()`.

hash_id Posting hash id (character).

beruf Job title/occupation (character).

refnr Reference number (character).

arbeitgeber Employer name (character).

aktuelle_veroeffentlichungsdatum Published date/time text (character).

aktuelle_veroeffentlichungsdatum_time Published timestamp as POSIXct (Europe/Berlin).

eintrittsdatum Start date text (character).

eintrittsdatum_time Start timestamp as POSIXct (Europe/Berlin).

arbeitsort_plz Postal code (character).

arbeitsort_ort City/locality (character).

arbeitsort_strasse Street (character).

arbeitsort_region Region/state (character).

arbeitsort_land Country code/name (character).

arbeitsort_lat Latitude (numeric).

arbeitsort_lon Longitude (numeric).

modifikations_timestamp Modification timestamp text (character).

modifikations_timestamp_time Modification timestamp as POSIXct (Europe/Berlin).

page Returned page number (character).

size Returned page size (character).

max_ergebnisse Total result count (character).

facetten Facet metadata (list-column).

See Also

`jobsuche_search()` for the standard endpoint and `bunddev_parameters()` for parameter discovery.

Other Jobsuche: `jobsuche_logo()`, `jobsuche_search()`

Examples

```
## Not run:
Sys.setenv(JOBBOERSE_API_KEY = "jobboerse-jobsuche")
bunddev_auth_set("jobsuche", type = "api_key", env_var = "JOBBOERSE_API_KEY")
jobsuche_search_app(params = list(was = "data", size = 5), flatten = TRUE)

## End(Not run)
```

klinikatlas_german_places

List German places from Bundes-Klinik-Atlas

Description

List German places from Bundes-Klinik-Atlas

Usage

```
klinikatlas_german_places(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns place metadata used by the Bundes-Klinik-Atlas search UI. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A [tibble](#) with one row per place entry.

See Also

Other Klinik Atlas: [klinikatlas_german_states\(\)](#), [klinikatlas_hospital_detail\(\)](#), [klinikatlas_icd_codes\(\)](#), [klinikatlas_locations\(\)](#), [klinikatlas_ops_codes\(\)](#), [klinikatlas_search\(\)](#), [klinikatlas_states\(\)](#)

Examples

```
## Not run:
klinikatlas_german_places()

## End(Not run)
```

`klinikatlas_german_states`*List German states with coordinates from Bundes-Klinik-Atlas*

Description

List German states with coordinates from Bundes-Klinik-Atlas

Usage

```
klinikatlas_german_states(safe = TRUE, refresh = FALSE)
```

Arguments

<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundeklinikatlas", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns federal states with map coordinates used by the Bundes-Klinik-Atlas. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A `tibble` with one row per federal state.

See Also

Other Klinik Atlas: `klinikatlas_german_places()`, `klinikatlas_hospital_detail()`, `klinikatlas_icd_codes()`, `klinikatlas_locations()`, `klinikatlas_ops_codes()`, `klinikatlas_search()`, `klinikatlas_states()`

Examples

```
## Not run:  
klinikatlas_german_states()  
  
## End(Not run)
```

`klinikatlas_hospital_detail`*Fetch raw hospital detail HTML from Bundes-Klinik-Atlas*

Description

Fetch raw hospital detail HTML from Bundes-Klinik-Atlas

Usage

```
klinikatlas_hospital_detail(id, params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

<code>id</code>	Integer hospital identifier from <code>klinikatlas_search()</code> .
<code>params</code>	Optional named list of upstream query parameters such as <code>tx_tverzhospitaldata_show[department]</code> or <code>cHash</code> .
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundelcache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the upstream HTML detail page as text. This wrapper intentionally avoids HTML scraping so the package remains aligned with the documented API surface. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

Character scalar containing the raw HTML response.

See Also

Other Klinik Atlas: `klinikatlas_german_places()`, `klinikatlas_german_states()`, `klinikatlas_icd_codes()`, `klinikatlas_locations()`, `klinikatlas_ops_codes()`, `klinikatlas_search()`, `klinikatlas_states()`

Examples

```
## Not run:  
html <- klinikatlas_hospital_detail(2610)  
substr(html, 1, 200)  
  
## End(Not run)
```

klinikatlas_icd_codes *List ICD codes from Bundes-Klinik-Atlas*

Description

List ICD codes from Bundes-Klinik-Atlas

Usage

```
klinikatlas_icd_codes(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesAPI", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns ICD code metadata used by the Bundes-Klinik-Atlas search UI. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A `tibble` with one row per ICD code.

See Also

Other Klinik Atlas: [klinikatlas_german_places\(\)](#), [klinikatlas_german_states\(\)](#), [klinikatlas_hospital_details\(\)](#), [klinikatlas_locations\(\)](#), [klinikatlas_ops_codes\(\)](#), [klinikatlas_search\(\)](#), [klinikatlas_states\(\)](#)

Examples

```
## Not run:  
klinikatlas_icd_codes()  
  
## End(Not run)
```

klinikatlas_locations *List Bundes-Klinik-Atlas hospital locations*

Description

List Bundes-Klinik-Atlas hospital locations

Usage

```
klinikatlas_locations(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesAPI", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the location directory for hospitals listed in the Bundes-Klinik-Atlas. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A [tibble](#) with one row per hospital location.

See Also

Other Klinik Atlas: [klinikatlas_german_places\(\)](#), [klinikatlas_german_states\(\)](#), [klinikatlas_hospital_details\(\)](#), [klinikatlas_icd_codes\(\)](#), [klinikatlas_ops_codes\(\)](#), [klinikatlas_search\(\)](#), [klinikatlas_states\(\)](#)

Examples

```
## Not run:  
klinikatlas_locations()  
  
## End(Not run)
```

klinikatlas_ops_codes *List OPS codes from Bundes-Klinik-Atlas*

Description

List OPS codes from Bundes-Klinik-Atlas

Usage

```
klinikatlas_ops_codes(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundesAPI", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns OPS code metadata used by the Bundes-Klinik-Atlas search UI. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A `tibble` with one row per OPS code.

See Also

Other Klinik Atlas: [klinikatlas_german_places\(\)](#), [klinikatlas_german_states\(\)](#), [klinikatlas_hospital_details\(\)](#), [klinikatlas_icd_codes\(\)](#), [klinikatlas_locations\(\)](#), [klinikatlas_search\(\)](#), [klinikatlas_states\(\)](#)

Examples

```
## Not run:  
klinikatlas_ops_codes()  
  
## End(Not run)
```

klinikatlas_search *Search Bundes-Klinik-Atlas hospitals*

Description

Search Bundes-Klinik-Atlas hospitals

Usage

```
klinikatlas_search(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of upstream query parameters. Parameter names are forwarded unchanged and should match the OpenAPI spec, for example <code>tx_solr[searchlabel]</code> , <code>tx_solr[rows]</code> , or <code>tx_solr[icd]</code> .
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Searches hospitals using the documented Bundes-Klinik-Atlas search endpoint. Parameter names are intentionally left in the upstream bracketed form so they can be passed through unchanged. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A [tibble](#) with one row per search result and the nested API payload retained in list-columns.

See Also

Other Klinik Atlas: [klinikatlas_german_places\(\)](#), [klinikatlas_german_states\(\)](#), [klinikatlas_hospital_detail\(\)](#), [klinikatlas_icd_codes\(\)](#), [klinikatlas_locations\(\)](#), [klinikatlas_ops_codes\(\)](#), [klinikatlas_states\(\)](#)

Examples

```
## Not run:
klinikatlas_search(params = list(
  "tx_solr[rows]" = 5,
  "tx_solr[start]" = 0,
  "tx_solr[searchlabel]" = "Berlin"
))

## End(Not run)
```

klinikatlas_states *List Bundes-Klinik-Atlas state summaries*

Description

List Bundes-Klinik-Atlas state summaries

Usage

```
klinikatlas_states(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns aggregate Klinik-Atlas metrics by federal state. API documentation: <https://github.com/bundesAPI/klinikatlas-api>.

Value

A [tibble](#) with one row per federal state.

See Also

Other Klinik Atlas: [klinikatlas_german_places\(\)](#), [klinikatlas_german_states\(\)](#), [klinikatlas_hospital_detail\(\)](#), [klinikatlas_icd_codes\(\)](#), [klinikatlas_locations\(\)](#), [klinikatlas_ops_codes\(\)](#), [klinikatlas_search\(\)](#)

Examples

```
## Not run:
klinikatlas_states()

## End(Not run)
```

ladestationen_query *Query charging stations*

Description

Query charging stations

Usage

```
ladestationen_query(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of ArcGIS query parameters. Common keys: geometry Geometry filter (required by this adapter). Can be a JSON string or an R list that is converted to JSON. geometryType Geometry type, e.g. "esriGeometryEnvelope". where SQL-like filter expression (default often "1=1"). outFields Fields to return (default "*"). returnGeometry Whether to include feature geometry ("true"/"false"). outSR Output spatial reference id, e.g. 4326. f Output format (default "json"). resultRecordCount Maximum number of returned rows (integer). token Optional ArcGIS access token if required by upstream service.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundles", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> :

- "drop" Remove list columns entirely. Use when nested data is not needed.
- "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.
- "unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.
Warning: this can significantly increase the number of rows.

Details

The Ladesaehlenregister API is backed by an ArcGIS feature service. You must supply a geometry filter and outFields. The ArcGIS service may require a token query parameter even though the API docs describe the service as public. API documentation: <https://ladestationen.api.bund.dev>.

Value

A tibble with one row per ArcGIS feature. Attribute names are normalized to lower snake_case. Includes a geometry list-column with the feature geometry object.

Attribute columns One column per ArcGIS attributes field (normalized to snake_case).

geometry Feature geometry object as list-column.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Examples

```
## Not run:
geometry <- jsonlite::toJSON(
  list(
    xmin = 13.3, ymin = 52.4, xmax = 13.5, ymax = 52.6,
    spatialReference = list(wkid = 4326)
  ),
  auto_unbox = TRUE
)
ladestationen_query(params = list(
  geometry = geometry,
  geometryType = "esriGeometryEnvelope",
  where = "1=1",
  outFields = "*",
  outSR = 4326,
  f = "json",
  returnGeometry = "false",
  resultRecordCount = 5
))

## End(Not run)
```

lebensmittelwarnung_warnings
List food and product warnings

Description

List food and product warnings

Usage

```
lebensmittelwarnung_warnings(  
  food = list(),  
  products = list(),  
  safe = TRUE,  
  refresh = FALSE,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

food	Optional request options for food warnings (rows, start, sort, fq).
products	Optional request options for product warnings (rows, start, sort, fq).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

The Lebensmittelwarnungen API returns German food and product safety warnings. Requests are sent as JSON to the `warnings/merged` endpoint. API documentation: <https://lebensmittelwarnung.api.bund.dev>. <https://github.com/bundesAPI/lebensmittelwarnung-api>.

The API expects an Authorization header. By default the adapter uses the public key documented in the OpenAPI spec. Override it with `bunddev_auth_set()` if needed.

Value

A tibble with one row per warning:

type Warning type from the source index (character).

archived Whether the warning is archived (TRUE/FALSE).

id Warning identifier (integer).

link Link to the warning details page (character).

title Warning title (character).

warning Warning text/summary (character).

affected_states Affected German states (list-column).

product Structured product details (list-column).

rapex_information RAPEX metadata when present (list-column).

safety_information Additional safety metadata (list-column).

published_date Published timestamp in milliseconds since epoch (numeric).

published_date_time Published timestamp as POSIXct in Europe/Berlin.

num_found Total number of matching warnings in the response (integer).

See Also

[bunddev_auth_set\(\)](#) to configure the authorization header.

Examples

```
## Not run:
lebensmittelwarnung_warnings(food = list(rows = 10), products = list(rows = 0))

## End(Not run)
```

```
luftqualitaet_airquality
```

List air quality measurements

Description

List air quality measurements

Usage

```
luftqualitaet_airquality(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The Luftqualitaet API provides air quality data and metadata from the Umweltbundesamt. Use query parameters to filter by date/time and station. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A *tibble* with one row per station/time observation. Column names are derived from the indices metadata in the API response and typically include a station id key column followed by date end (CET), an overall air-quality index, an incompleteness flag, and per-component sub-arrays (component id, value, index, percent). Exact column names and count vary by query parameters.

See Also

`bunddev_parameters()` to inspect available query parameters. `luftqualitaet_measures()` for measurement metadata and `luftqualitaet_components()` for pollutant components.

Other Luftqualitaet: `luftqualitaet_airquality_limits()`, `luftqualitaet_annualbalances()`, `luftqualitaet_components()`, `luftqualitaet_measures()`, `luftqualitaet_measures_limits()`, `luftqualitaet_meta()`, `luftqualitaet_networks()`, `luftqualitaet_scopes()`, `luftqualitaet_stationsettings()`, `luftqualitaet_stationtypes()`, `luftqualitaet_thresholds()`, `luftqualitaet_transgressions()`, `luftqualitaet_transgressiontypes()`

Examples

```
## Not run:
luftqualitaet_airquality(params = list(
  date_from = "2024-01-01",
  date_to = "2024-01-02"
))

## End(Not run)
```

luftqualitaet_airquality_limits
Get air quality date limits

Description

Get air quality date limits

Usage

```
luftqualitaet_airquality_limits(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns date limits for air quality measurements. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per station. Column names are derived from the indices metadata in the API response and typically include a `station_id` key column followed by `limit date start min (CET)` and `limit date start max (CET)`.

See Also

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_annualbalances\(\)](#), [luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_annualbalances
List annual balances

Description

List annual balances

Usage

```
luftqualitaet_annualbalances(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns annual balance data for a component and year. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per annual-balance record. Column names are derived from the indices metadata in the API response and typically include station id, component id, year, value, and transgression type.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_component\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_components

List components

Description

List components

Usage

```
luftqualitaet_components(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns available pollutant components. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per component. Column names are derived from the indices metadata in the API response and typically include a component id/code key column followed by component id, component code, component symbol, component unit, and component name.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_measures

List measurements metadata

Description

List measurements metadata

Usage

```
luftqualitaet_measures(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns measurement metadata for stations, components, and scopes. Official docs: <https://luftqualitaet.api.bund.dev>.

Value

A *tibble* with one row per station/time measurement. Column names are derived from the indices metadata in the API response and typically include a station id key column and per-observation arrays containing component id, scope id, value, date end, and index. Exact column names and count vary by query parameters.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#), [luftqualitaet_components\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

```
luftqualitaet_measures_limits
  Get measurement date limits
```

Description

Get measurement date limits

Usage

```
luftqualitaet_measures_limits(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns date limits for measurement metadata. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A `tibble` with one row per scope/component/station combination. Column names are derived from the indices metadata in the API response and typically include `scope id`, `component id`, `station id`, `limit date start min (CET)`, and `limit date start max (CET)`.

See Also

Other Luftqualitaet: `luftqualitaet_airquality()`, `luftqualitaet_airquality_limits()`, `luftqualitaet_annualba`, `luftqualitaet_components()`, `luftqualitaet_measures()`, `luftqualitaet_meta()`, `luftqualitaet_networks()`, `luftqualitaet_scopes()`, `luftqualitaet_stationsettings()`, `luftqualitaet_stationtypes()`, `luftqualitaet_thresholds()`, `luftqualitaet_transgressions()`, `luftqualitaet_transgressiontypes()`

luftqualitaet_meta *List combined metadata*

Description

List combined metadata

Usage

```
luftqualitaet_meta(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns combined metadata for a given use case. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with combined metadata. The structure depends on the use parameter and may include sections for components, scopes, networks, stations, limits, and cross-references. Column names are derived from the indices metadata in the API response and vary by use value.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#)
[luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#),
[luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#),
[luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#),
[luftqualitaet_transgressiontypes\(\)](#)

 luftqualitaet_networks

List networks

Description

List networks

Usage

```
luftqualitaet_networks(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns networks used by the Luftqualitaet API. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per monitoring network. Column names are derived from the indices metadata in the API response and typically include a network id/code key column followed by network id, network code, and network name.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#), [luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtyp](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_scopes *List scopes*

Description

List scopes

Usage

```
luftqualitaet_scopes(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bunddev", "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns scopes used by the Luftqualitaet API. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per scope. Column names are derived from the indices metadata in the API response and typically include a scope id/code key column followed by scope id, scope code, scope time base, scope time scope, scope time is max, and scope name.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualbaeuerliche_emissionslimits\(\)](#), [luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_stationsettings
List station settings

Description

List station settings

Usage

```
luftqualitaet_stationsettings(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns station settings metadata. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A `tibble` with one row per station setting. Column names are derived from the indices metadata in the API response and typically include station setting id, station setting name, and station setting short name.

See Also

`bunddev_parameters()` to inspect available query parameters.

Other Luftqualitaet: `luftqualitaet_airquality()`, `luftqualitaet_airquality_limits()`, `luftqualitaet_annualba`
`luftqualitaet_components()`, `luftqualitaet_measures()`, `luftqualitaet_measures_limits()`,
`luftqualitaet_meta()`, `luftqualitaet_networks()`, `luftqualitaet_scopes()`, `luftqualitaet_stationtypes()`,
`luftqualitaet_thresholds()`, `luftqualitaet_transgressions()`, `luftqualitaet_transgressiontypes()`

luftqualitaet_stationtypes
List station types

Description

List station types

Usage

```
luftqualitaet_stationtypes(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bunddev", "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns station types metadata. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per station type. Column names are derived from the indices metadata in the API response and typically include station type id and station type name.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#)
[luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#),
[luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#),
[luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

 luftqualitaet_thresholds

List thresholds

Description

List thresholds

Usage

```
luftqualitaet_thresholds(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bunddev/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns threshold metadata for components and scopes. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A [tibble](#) with one row per threshold entry. Column names are derived from the indices metadata in the API response and typically include threshold id, component id, scope id, threshold min, threshold max, and threshold index.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#), [luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_transgressions\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_transgressions
List transgressions

Description

List transgressions

Usage

```
luftqualitaet_transgressions(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns exceedances (transgressions) data. API documentation: <https://luftqualitaet.api.bund.dev>.

Value

A **tibble** with one row per station transgression record. Column names are derived from the indices metadata in the API response and typically include `station id`, `day_first`, `day_recent`, `value of year`, and monthly values for January through December.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Luftqualitaet: [luftqualitaet_airquality\(\)](#), [luftqualitaet_airquality_limits\(\)](#), [luftqualitaet_annualba](#), [luftqualitaet_components\(\)](#), [luftqualitaet_measures\(\)](#), [luftqualitaet_measures_limits\(\)](#), [luftqualitaet_meta\(\)](#), [luftqualitaet_networks\(\)](#), [luftqualitaet_scopes\(\)](#), [luftqualitaet_stationsettings\(\)](#), [luftqualitaet_stationtypes\(\)](#), [luftqualitaet_thresholds\(\)](#), [luftqualitaet_transgressiontypes\(\)](#)

luftqualitaet_transgressiontypes
List transgression types

Description

List transgression types

Usage

```
luftqualitaet_transgressiontypes(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Common keys: date_from Start date/time filter (character, ISO-8601/date). date_to End date/time filter (character, ISO-8601/date). component Pollutant component id/code (character). scope Scope id/code (character). station Station id/code (character). network Network id/code (character). lang Language code for labels/descriptions (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bunddev/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns available transgression types. API documentation: <https://luftqualitaet.api.bunddev>.

Value

A `tibble` with one row per transgression type. Column names are derived from the indices metadata in the API response and typically include exceedance type id and exceedance type name.

See Also

`bunddev_parameters()` to inspect available query parameters.

Other Luftqualitaet: `luftqualitaet_airquality()`, `luftqualitaet_airquality_limits()`, `luftqualitaet_annualba`, `luftqualitaet_components()`, `luftqualitaet_measures()`, `luftqualitaet_measures_limits()`, `luftqualitaet_meta()`, `luftqualitaet_networks()`, `luftqualitaet_scopes()`, `luftqualitaet_stationsettings()`, `luftqualitaet_stationtypes()`, `luftqualitaet_thresholds()`, `luftqualitaet_transgressions()`

 marktstammdaten_filters_gaserzeugung

List MaStR filter options for gas generation

Description

List MaStR filter options for gas generation

Usage

```
marktstammdaten_filters_gaserzeugung(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns filter definitions for public gas generation data. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A **tibble** with one row per available filter and columns:

FilterName Character. Machine-readable filter field name.

ListObject List. Selectable options (named with Name/Value).

Type Character. Filter data type ("text", "number", "multidropdown", "date", or "boolean").

See Also

Other Marktstammdaten: [marktstammdaten_filters_gasverbrauch\(\)](#), [marktstammdaten_filters_stromerzeugung\(\)](#), [marktstammdaten_filters_stromverbrauch\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_gasverbrauch\(\)](#), [marktstammdaten_stromerzeugung\(\)](#), [marktstammdaten_stromverbrauch\(\)](#)

Examples

```
## Not run:
marktstammdaten_filters_gaserzeugung()

## End(Not run)
```

marktstammdaten_filters_gasverbrauch

List MaStR filter options for gas consumption

Description

List MaStR filter options for gas consumption

Usage

```
marktstammdaten_filters_gasverbrauch(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

Returns filter definitions for public gas consumption data. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A [tibble](#) with one row per available filter and columns:

FilterName Character. Machine-readable filter field name.

ListObject List. Selectable options (named with Name/Value).

Type Character. Filter data type ("text", "number", "multidropdown", "date", or "boolean").

See Also

Other Marktstammdaten: [marktstammdaten_filters_gaserzeugung\(\)](#), [marktstammdaten_filters_stromerzeugung\(\)](#), [marktstammdaten_filters_stromverbrauch\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_gasverbrauch\(\)](#), [marktstammdaten_stromerzeugung\(\)](#), [marktstammdaten_stromverbrauch\(\)](#)

Examples

```
## Not run:  
marktstammdaten_filters_gasverbrauch()  
  
## End(Not run)
```

marktstammdaten_filters_stromerzeugung

List MaStR filter options for electricity generation

Description

List MaStR filter options for electricity generation

Usage

```
marktstammdaten_filters_stromerzeugung(  
  safe = TRUE,  
  refresh = FALSE,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns filter definitions for public electricity generation data. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A [tibble](#) with one row per available filter and columns:

FilterName Character. Machine-readable filter field name.

ListObject List. Selectable options (named with Name/Value).

Type Character. Filter data type ("text", "number", "multidropdown", "date", or "boolean").

See Also

Other Marktstammdaten: [marktstammdaten_filters_gaserzeugung\(\)](#), [marktstammdaten_filters_gasverbrauch\(\)](#), [marktstammdaten_filters_stromverbrauch\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_gasverbrauch\(\)](#), [marktstammdaten_stromerzeugung\(\)](#), [marktstammdaten_stromverbrauch\(\)](#)

Examples

```
## Not run:
marktstammdaten_filters_stromerzeugung()

## End(Not run)
```

 marktstammdaten_filters_stromverbrauch

List MaStR filter options for electricity consumption

Description

List MaStR filter options for electricity consumption

Usage

```
marktstammdaten_filters_stromverbrauch(
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns filter definitions for public electricity consumption data. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A [tibble](#) with one row per available filter and columns:

FilterName Character. Machine-readable filter field name.

ListObject List. Selectable options (named with Name/Value).

Type Character. Filter data type ("text", "number", "multidropdown", "date", or "boolean").

See Also

Other Marktstammdaten: [marktstammdaten_filters_gaserzeugung\(\)](#), [marktstammdaten_filters_gasverbrauch\(\)](#), [marktstammdaten_filters_stromerzeugung\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_gasverbrauch\(\)](#), [marktstammdaten_stromerzeugung\(\)](#), [marktstammdaten_stromverbrauch\(\)](#)

Examples

```
## Not run:
marktstammdaten_filters_stromverbrauch()

## End(Not run)
```

```
marktstammdaten_gaserzeugung
      List MaStR gas generation data
```

Description

List MaStR gas generation data

Usage

```
marktstammdaten_gaserzeugung(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters: sort Sort definition string (character). page Page index (integer). pageSize Page size/number of returned entries (integer). filter Serialized filter expression from filter endpoints (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> :

- "drop" Remove list columns entirely. Use when nested data is not needed.
- "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.
- "unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.
Warning: this can significantly increase the number of rows.

Details

Returns public gas generation data from the MaStR. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A **tibble** with one row per gas generation unit and columns determined by the MaStR API response. Common columns include MaStRNummer, EinheitName, Bundesland, Ort, Plz, BetriebsStatusName, InbetriebnahmeDatum, and EinheitRegistrierungsdatum (see `marktstammdaten_filters_gaserzeugung()` for available filter fields). Additionally:

total Numeric. Total number of matching records upstream.

***Datum_time** POSIXct. Parsed date column (Europe/Berlin) added for every source column whose name contains Datum.

See Also

`bunddev_parameters()` to inspect available query parameters.

Other Marktstammdaten: `marktstammdaten_filters_gaserzeugung()`, `marktstammdaten_filters_gasverbrauch()`, `marktstammdaten_filters_stromerzeugung()`, `marktstammdaten_filters_stromverbrauch()`, `marktstammdaten_gasverbrauch()`, `marktstammdaten_stromerzeugung()`, `marktstammdaten_stromverbrauch()`

Examples

```
## Not run:
marktstammdaten_gaserzeugung(params = list(page = 1, pageSize = 5))

## End(Not run)
```

marktstammdaten_gasverbrauch
List MaStR gas consumption data

Description

List MaStR gas consumption data

Usage

```
marktstammdaten_gasverbrauch(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters: sort Sort definition string (character). page Page index (integer). pageSize Page size/number of returned entries (integer). filter Serialized filter expression from filter endpoints (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns public gas consumption data from the MaStR. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A [tibble](#) with one row per gas consumption unit and columns determined by the MaStR API response. Common columns include `MaStRNummer`, `EinheitName`, `Bundesland`, `Ort`, `Plz`, `BetriebsStatusName`, and `EinheitRegistrierungsdatum` (see [marktstammdaten_filters_gasverbrauch\(\)](#) for available filter fields). Additionally:

total Numeric. Total number of matching records upstream.

***Datum_time** POSIXct. Parsed date column (Europe/Berlin) added for every source column whose name contains `Datum`.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Marktstammdaten: [marktstammdaten_filters_gaserzeugung\(\)](#), [marktstammdaten_filters_gasverbrauch\(\)](#), [marktstammdaten_filters_stromerzeugung\(\)](#), [marktstammdaten_filters_stromverbrauch\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_stromerzeugung\(\)](#), [marktstammdaten_stromverbrauch\(\)](#)

Examples

```
## Not run:
marktstammdaten_gasverbrauch(params = list(page = 1, pageSize = 5))

## End(Not run)
```

```
marktstammdaten_stromerzeugung
      List MaStR electricity generation data
```

Description

List MaStR electricity generation data

Usage

```
marktstammdaten_stromerzeugung(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters: sort Sort definition string (character). page Page index (integer). pageSize Page size/number of returned entries (integer). filter Serialized filter expression from filter endpoints (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev_cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.

`flatten_mode` How to handle list columns when `flatten = TRUE`:

- "drop" Remove list columns entirely. Use when nested data is not needed.
- "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the **default**.
- "unnest" Expand list columns into multiple rows via `tidyr::unnest_longer()`.
Warning: this can significantly increase the number of rows.

Details

Returns public electricity generation data from the MaStR. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A **tibble** with one row per electricity generation unit and columns determined by the MaStR API response. Common columns include `MaStRNummer`, `EinheitName`, `Bundesland`, `Ort`, `Plz`, `Bruttoleistung`, `Nettonennleistung`, `EnergietraegerName`, `BetriebsStatusName`, `InbetriebnahmeDatum`, and `EinheitRegistrierungsdatum` (see [marktstammdaten_filters_stromerzeugung\(\)](#) for available filter fields). Additionally:

total Numeric. Total number of matching records upstream.

***Datum_time** POSIXct. Parsed date column (Europe/Berlin) added for every source column whose name contains `Datum`.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Marktstammdaten: [marktstammdaten_filters_gaserzeugung\(\)](#), [marktstammdaten_filters_gasverbrauch\(\)](#), [marktstammdaten_filters_stromerzeugung\(\)](#), [marktstammdaten_filters_stromverbrauch\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_gasverbrauch\(\)](#), [marktstammdaten_stromverbrauch\(\)](#)

Examples

```
## Not run:
marktstammdaten_stromerzeugung(params = list(page = 1, pageSize = 5))

## End(Not run)
```

marktstammdaten_stromverbrauch

List MaStR electricity consumption data

Description

List MaStR electricity consumption data

Usage

```
marktstammdaten_stromverbrauch(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	Named list of query parameters: sort Sort definition string (character). page Page index (integer). pageSize Page size/number of returned entries (integer). filter Serialized filter expression from filter endpoints (character).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundles/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns public electricity consumption data from the MaStR. API documentation: <https://github.com/bundesAPI/marktstammdaten-api>.

Value

A [tibble](#) with one row per electricity consumption unit and columns determined by the MaStR API response. Common columns include `MaStRNummer`, `EinheitName`, `Bundesland`, `Ort`, `Plz`, `BetriebsStatusName`, and `EinheitRegistrierungsdatum` (see [marktstammdaten_filters_stromverbrauch\(\)](#) for available filter fields). Additionally:

total Numeric. Total number of matching records upstream.

***Datum_time** POSIXct. Parsed date column (Europe/Berlin) added for every source column whose name contains `Datum`.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Marktstammdaten: [marktstammdaten_filters_gaserzeugung\(\)](#), [marktstammdaten_filters_gasverbrauch\(\)](#), [marktstammdaten_filters_stromerzeugung\(\)](#), [marktstammdaten_filters_stromverbrauch\(\)](#), [marktstammdaten_gaserzeugung\(\)](#), [marktstammdaten_gasverbrauch\(\)](#), [marktstammdaten_stromerzeugung\(\)](#)

Examples

```
## Not run:
marktstammdaten_stromverbrauch(params = list(page = 1, pageSize = 5))

## End(Not run)
```

mudab_parameters	<i>List MUDAB parameters</i>
------------------	------------------------------

Description

List MUDAB parameters

Usage

```
mudab_parameters(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns measurement parameters from the MUDAB database. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per parameter and columns:

metadataid Integer. Metadata identifier.

COMPT_DS Character. Compartment code (BL, CW, CS, or CF).

PARAMETER Character. Parameter abbreviation.

PARAMETERGRUPPE Character. Parameter group code.

PARAM_NAME Character. Parameter name.

PARGROUP Character. Parameter group code (alternate).

PARAMGROUP_NAME Character. Parameter group name.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_parameters(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_parameters_biologie

List MUDAB parameters (Biologie)

Description

List MUDAB parameters (Biologie)

Usage

```
mudab_parameters_biologie(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundde "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns parameter entries for the Biologie compartment. API documentation: [https://mudab.
api.bund.dev](https://mudab.api.bund.dev).

Value

A [tibble](#) with one row per parameter and columns:

metadataid Integer. Metadata identifier.

COMPT_DS Character. Compartment code (BL, CW, CS, or CF).

PARAMETER Character. Parameter abbreviation.

PARAMETERGRUPPE Character. Parameter group code.

PARAM_NAME Character. Parameter name.

PARGROUP Character. Parameter group code (alternate).

PARAMGROUP_NAME Character. Parameter group name.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_parameters_biologie(range = list(from = 0, count = 5))

## End(Not run)
```

 mudab_parameters_biota

List MUDAB parameters (Biota)

Description

List MUDAB parameters (Biota)

Usage

```
mudab_parameters_biota(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundled/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns parameter entries for the Biota compartment. API documentation: <https://mudab.api.bund.dev>.

Value

A **tibble** with one row per parameter and columns:

metadataid Integer. Metadata identifier.

COMPT_DS Character. Compartment code (BL, CW, CS, or CF).

PARAMETER Character. Parameter abbreviation.

PARAMETERGRUPPE Character. Parameter group code.

PARAM_NAME Character. Parameter name.

PARGROUP Character. Parameter group code (alternate).

PARAMGROUP_NAME Character. Parameter group name.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biological\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_parameters_biota(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_parameters_sediment

List MUDAB parameters (Sediment)

Description

List MUDAB parameters (Sediment)

Usage

```
mudab_parameters_sediment(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundled/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns parameter entries for the Sediment compartment. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per parameter and columns:

metadataid Integer. Metadata identifier.

COMPT_DS Character. Compartment code (BL, CW, CS, or CF).

PARAMETER Character. Parameter abbreviation.

PARAMETERGRUPPE Character. Parameter group code.

PARAM_NAME Character. Parameter name.

PARGROUP Character. Parameter group code (alternate).

PARAMGROUP_NAME Character. Parameter group name.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_parameters_sediment(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_parameters_wasser

List MUDAB parameters (Wasser)

Description

List MUDAB parameters (Wasser)

Usage

```
mudab_parameters_wasser(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundled/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns parameter entries for the Wasser compartment. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per parameter and columns:

metadataid Integer. Metadata identifier.

COMPT_DS Character. Compartment code (BL, CW, CS, or CF).

PARAMETER Character. Parameter abbreviation.

PARAMETERGRUPPE Character. Parameter group code.

PARAM_NAME Character. Parameter name.

PARGROUP Character. Parameter group code (alternate).

PARAMGROUP_NAME Character. Parameter group name.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_parameters_wasser(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_parameter_values

List MUDAB parameter values

Description

List MUDAB parameter values

Usage

```
mudab_parameter_values(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns measurement values for parameters from the MUDAB database. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per measurement value and columns:

metadataid Integer. Metadata identifier.

STATNAME_ST Character. Station name.

STATNAME_DATE_TIME Character. Composite station-date-time string.

PARAMETERID_PM Character. Parameter identifier.

PARAMCODE_PM Character. Parameter code name.

DATE_STM Character. Measurement date (YYYYMMDD).

TIME_STM Character. Measurement time (HHMM).

VALUE_MS Character. Measured value.

datetime_time POSIXct. Parsed date-time (Europe/Berlin).

See Also

Other MUDAB: [mudab_parameters\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_parameter_values(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_plc_measurements

List MUDAB PLC measurement values

Description

List MUDAB PLC measurement values

Usage

```
mudab_plc_measurements(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns PLC station measurement values. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per PLC measurement and columns:

metadataid Integer. Metadata identifier.

NUMMER Integer. Running row number.

NAME Integer. Measurement name code.

STATION_CODE Character. Station code.

STATION_NAME Character. Station name.

VALUE Numeric. Measured value.

PARAM_TYPE Character. Aggregation type ("TOT" or "AVE").

PARAM_ID Character. Numeric code for NAME.

PERIOD_NAME Character. Measurement period (year).

PERIOD_TYPE Character. Aggregation duration ("A").

LAND_CD Character. Federal state code.

ST_LAT Numeric. Latitude.

ST_LON Numeric. Longitude.

SUBCM_CODE Character. Sub-compartment code.

SUBCM_NAME Character. Sub-compartment name.

MON_TYPE Character. Monitoring type.

VAL_UNIT Character. Measurement unit.

AREA Numeric. Area value.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biological\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:  
mudab_plc_measurements(range = list(from = 0, count = 5))  
  
## End(Not run)
```

mudab_plc_parameters *List MUDAB PLC parameters*

Description

List MUDAB PLC parameters

Usage

```
mudab_plc_parameters(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns parameters measured at PLC stations. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per PLC parameter and columns:

metadataid Integer. Metadata identifier.

PARAMETER Character. Parameter abbreviation.

PARAMETERGRUPPE Character. Parameter group code.

STATION_CODE Character. Station code.

LETZTE_MESSUNG Character. Year of last measurement.

PRKEY Character. Unique key (station + parameter).

STATION_NAME Character. Station name.

ANZ_MESSWERTE Integer. Number of measured values.

SUBCM_CODE Character. Sub-compartment code.

SUBCM_NAME Character. Sub-compartment name.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_plc_parameters(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_plc_stations	<i>List MUDAB PLC stations</i>
--------------------	--------------------------------

Description

List MUDAB PLC stations

Usage

```
mudab_plc_stations(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns HELCOM PLC stations. API documentation: <https://mudab.api.bund.dev>.

Value

A [tibble](#) with one row per PLC station and columns:

STATION_NAME Character. Station name.
STATION_CODE Character. Station code.
LAND_CD Character. Federal state code.
ST_LAT Numeric. Latitude.
ST_LON Numeric. Longitude.
SUBCM_CODE Character. Sub-compartment code.
SUBCM_NAME Character. Sub-compartment name.
MON_TYPE Character. Monitoring type.

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_project_stations\(\)](#), [mudab_stations\(\)](#)

Examples

```
## Not run:
mudab_plc_stations(range = list(from = 0, count = 5))

## End(Not run)
```

```
mudab_project_stations
```

List MUDAB project stations

Description

List MUDAB project stations

Usage

```
mudab_project_stations(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundelcache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns project stations from the MUDAB database. API documentation: <https://mudab.api.bund.dev>.

Value

A **tibble** with one row per project station and columns:

metadataid Integer. Metadata identifier.

PROJECTSTATIONID Integer. Project station identifier.

NAME_PS Character. Project station name.

REGION Character. Region of the station.

INSTITUT Character. Responsible institute.

See Also

Other MUDAB: `mudab_parameter_values()`, `mudab_parameters()`, `mudab_parameters_biologie()`, `mudab_parameters_biota()`, `mudab_parameters_sediment()`, `mudab_parameters_wasser()`, `mudab_plc_measurements()`, `mudab_plc_parameters()`, `mudab_plc_stations()`, `mudab_stations()`

Examples

```
## Not run:
mudab_project_stations(range = list(from = 0, count = 5))

## End(Not run)
```

mudab_stations	<i>List MUDAB stations</i>
----------------	----------------------------

Description

List MUDAB stations

Usage

```
mudab_stations(
  filter = NULL,
  range = NULL,
  orderby = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Optional filter definition.
range	Optional range specification.
orderby	Optional ordering specification.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns measurement stations from the MUDAB database. API documentation: <https://mudab.api.bund.dev>.

Value

A **tibble** with one row per measurement station and columns:

metadataaid Integer. Metadata identifier.
STATNAME_ST Character. Station name.
NAME_PS Character. Project station name.
STATIONTYPE_ST Character. Station type.
COMPT_DS Character. Compartment code (BL, CW, CS, or CF).

See Also

Other MUDAB: [mudab_parameter_values\(\)](#), [mudab_parameters\(\)](#), [mudab_parameters_biologie\(\)](#), [mudab_parameters_biota\(\)](#), [mudab_parameters_sediment\(\)](#), [mudab_parameters_wasser\(\)](#), [mudab_plc_measurements\(\)](#), [mudab_plc_parameters\(\)](#), [mudab_plc_stations\(\)](#), [mudab_project_stations\(\)](#)

Examples

```
## Not run:
mudab_stations(range = list(from = 0, count = 5))

## End(Not run)
```

nina_archive_mowas *Get MOWAS archive entry*

Description

Get MOWAS archive entry

Usage

```
nina_archive_mowas(identifier, safe = TRUE, refresh = FALSE)
```

Arguments

identifier	Warning identifier.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

identifier Warning identifier (character).

data Archive entry response payload (list).

See Also

Other NINA: [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltips\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_archive_mowas_mapping
Get MOWAS archive mapping

Description

Get MOWAS archive mapping

Usage

```
nina_archive_mowas_mapping(identifier, safe = TRUE, refresh = FALSE)
```

Arguments

identifier	Warning identifier.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

identifier Warning identifier (character).

data Archive mapping response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_covid_infos *Get COVID info data*

Description

Get COVID info data

Usage

```
nina_covid_infos(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

data COVID info response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_covid_map	<i>Get COVID map data</i>
----------------	---------------------------

Description

Get COVID map data

Usage

```
nina_covid_map(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

data COVID map response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_covid_rules	<i>Get COVID rules for an ARS</i>
------------------	-----------------------------------

Description

Get COVID rules for an ARS

Usage

```
nina_covid_rules(ars, safe = TRUE, refresh = FALSE)
```

Arguments

ars	ARS code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

ars The ARS code used in the request (character).

data COVID rules response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_covid_ticker *Get COVID ticker*

Description

Get COVID ticker

Usage

```
nina_covid_ticker(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

data COVID ticker response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_covid_ticker_message *Get COVID ticker message*

Description

Get COVID ticker message

Usage

```
nina_covid_ticker_message(id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>id</code>	Ticker message id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

id Ticker message identifier (character).

data Ticker message response payload (list).

See Also

Other NINA: `nina_archive_mowas()`, `nina_archive_mowas_mapping()`, `nina_covid_infos()`, `nina_covid_map()`, `nina_covid_rules()`, `nina_covid_ticker()`, `nina_dashboard()`, `nina_event_code()`, `nina_event_codes()`, `nina_faqs()`, `nina_logo()`, `nina_logos()`, `nina_mapdata()`, `nina_mowas_rss()`, `nina_notfalltipps()`, `nina_version()`, `nina_warning()`, `nina_warning_geojson()`, `nina_warning_json()`, `nina_warnings()`

<code>nina_dashboard</code>	<i>Get NINA dashboard data</i>
-----------------------------	--------------------------------

Description

Get NINA dashboard data

Usage

```
nina_dashboard(ars, safe = TRUE, refresh = FALSE)
```

Arguments

<code>ars</code>	ARS code.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns dashboard data for the given ARS code. API documentation: <https://nina.api.bund.de/>.

Value

A one-row tibble.

ars The ARS code used in the request (character).

data Dashboard response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_event_code	<i>Get an event code file</i>
-----------------	-------------------------------

Description

Get an event code file

Usage

```
nina_event_code(filename, safe = TRUE, refresh = FALSE)
```

Arguments

filename Event code filename.

safe Logical; if TRUE (default), apply rate-limiting and cache GET responses to `tools::R_user_dir("bundle", "cache")`.

refresh Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

filename Event code file name (character).

bytes Raw file content (list of raw vectors).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_event_codes *List NINA event codes*

Description

List NINA event codes

Usage

```
nina_event_codes(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A tibble with one row per event code entry. Columns are bound from the upstream eventCodes array, plus:

last_modification_date Feed-level modification timestamp in milliseconds (numeric).

last_modification_time Parsed modification timestamp (POSIXct, Europe/Berlin).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_faqs *List FAQs*

Description

List FAQs

Usage

```
nina_faqs(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A tibble with one row per FAQ entry.

question FAQ question text (character).

answer FAQ answer text (character).

last_modification_date Modification timestamp in milliseconds (numeric).

last_modification_time Parsed modification timestamp (POSIXct, Europe/Berlin).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltips\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_logo	<i>Get a logo file</i>
-----------	------------------------

Description

Get a logo file

Usage

```
nina_logo(filename, safe = TRUE, refresh = FALSE)
```

Arguments

filename	Logo file name.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

filename Logo file name (character).

bytes Raw file content (list of raw vectors).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_logos

List NINA logos

Description

List NINA logos

Usage

```
nina_logos(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A tibble with one row per logo metadata entry. Columns are bound from the upstream logos array, plus:

last_modification_date Feed-level modification timestamp in milliseconds (numeric).

last_modification_time Parsed modification timestamp (POSIXct, Europe/Berlin).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_mapdata	<i>List map data</i>
--------------	----------------------

Description

List map data

Usage

```
nina_mapdata(
  source = c("katwarn", "biwapp", "mowas", "dwd", "lhp", "police"),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

source	Map data source.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Valid sources are: katwarn, biwapp, mowas, dwd, lhp, police.

Value

A tibble with one row per map-data warning entry.

id Warning identifier (character).

version Warning version (integer).

start_date Start date string from the API (character).

severity Warning severity level (character).

urgency Warning urgency level (character).

type Warning type (character).

i18n_title Internationalised title entries (list).

trans_keys Translation key entries (list).

start_date_time Parsed start date (POSIXct, Europe/Berlin).

source Map data source used in the request (character).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_mowas_rss	<i>Get MOWAS RSS feed</i>
----------------	---------------------------

Description

Get MOWAS RSS feed

Usage

```
nina_mowas_rss(ars, safe = TRUE, refresh = FALSE)
```

Arguments

ars	ARS code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

ars The ARS code used in the request (character).

rss RSS XML text returned by the API (character).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_notfalltips	<i>List emergency tips</i>
------------------	----------------------------

Description

List emergency tips

Usage

```
nina_notfalltips(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundelcache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A tibble with one row per emergency-tip category.

title Category title (character).

tips Tips for this category (list).

last_modification_date Modification timestamp in milliseconds (numeric).

last_modification_time Parsed modification timestamp (POSIXct, Europe/Berlin).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_version	<i>Get data version info</i>
--------------	------------------------------

Description

Get data version info

Usage

```
nina_version(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Value

A one-row tibble.

data Version/data-version response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_warning

Get a NINA travel warning by content id

Description

Get a NINA travel warning by content id

Usage

```
nina_warning(content_id, safe = TRUE, refresh = FALSE)
```

Arguments

content_id	Travel warning content id.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundled/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns full travel warning content, including HTML blocks.

Value

A tibble with detailed warning content rows. Same column structure as [nina_warnings\(\)](#): entry scalar fields (`snake_case`), plus `content_id`, `response_country`, `response_last_modified`, `last_modified_time`, and `effective_time` (POSIXct, Europe/Berlin).

See Also

`nina_warnings()` for ids.

Other NINA: `nina_archive_mowas()`, `nina_archive_mowas_mapping()`, `nina_covid_infos()`, `nina_covid_map()`, `nina_covid_rules()`, `nina_covid_ticker()`, `nina_covid_ticker_message()`, `nina_dashboard()`, `nina_event_code()`, `nina_event_codes()`, `nina_faqs()`, `nina_logo()`, `nina_logos()`, `nina_mapdata()`, `nina_mowas_rss()`, `nina_notfalltipps()`, `nina_version()`, `nina_warning_geojson()`, `nina_warning_json()`, `nina_warnings()`

Examples

```
## Not run:
warnings <- nina_warnings()
nina_warning(warnings$content_id[[1]])

## End(Not run)
```

nina_warnings	<i>List NINA travel warnings</i>
---------------	----------------------------------

Description

List NINA travel warnings

Usage

```
nina_warnings(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The NINA API provides warnings and app data from the Bundesamt fuer Bevoelkerungsschutz. This endpoint returns a list of travel warnings with metadata. API documentation: <https://nina.api.bund.dev>.

Value

A tibble with one row per warning entry. Columns include scalar fields from each entry (names converted to snake_case) plus the following added metadata:

content_id Content identifier from the NINA feed (character).

response_country Feed-level country code (character).

response_last_modified Feed-level modification timestamp in milliseconds (numeric).

last_modified_time Entry-level last-modified timestamp (POSIXct, Europe/Berlin).

effective_time Entry-level effective timestamp (POSIXct, Europe/Berlin).

See Also

[nina_warning\(\)](#) for a single warning and [nina_mapdata\(\)](#) for map-based alerts.

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warning_json\(\)](#)

Examples

```
## Not run:
nina_warnings()

## End(Not run)
```

nina_warning_geojson *Get a NINA warning (GeoJSON)*

Description

Get a NINA warning (GeoJSON)

Usage

```
nina_warning_geojson(identifier, safe = TRUE, refresh = FALSE)
```

Arguments

identifier	Warning identifier.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundled/cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a warning in GeoJSON format.

Value

A one-row tibble.

identifier Warning identifier (character).

geojson Full GeoJSON response payload (list).

See Also

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_json\(\)](#), [nina_warnings\(\)](#)

nina_warning_json	<i>Get a NINA warning (JSON)</i>
-------------------	----------------------------------

Description

Get a NINA warning (JSON)

Usage

```
nina_warning_json(identifier, safe = TRUE, refresh = FALSE)
```

Arguments

identifier	Warning identifier.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a warning in JSON format.

Value

A one-row tibble with CAP warning fields.

identifier Warning identifier (character).

sender Sender identifier (character).

sent Sent timestamp string (character).

status Alert status (character).

msg_type Message type (character).

scope Alert scope (character).

code Alert code entries (list).

info Alert info blocks (list).

sent_time Parsed sent timestamp (POSIXct, Europe/Berlin).

See Also

[nina_mapdata\(\)](#) to discover identifiers.

Other NINA: [nina_archive_mowas\(\)](#), [nina_archive_mowas_mapping\(\)](#), [nina_covid_infos\(\)](#), [nina_covid_map\(\)](#), [nina_covid_rules\(\)](#), [nina_covid_ticker\(\)](#), [nina_covid_ticker_message\(\)](#), [nina_dashboard\(\)](#), [nina_event_code\(\)](#), [nina_event_codes\(\)](#), [nina_faqs\(\)](#), [nina_logo\(\)](#), [nina_logos\(\)](#), [nina_mapdata\(\)](#), [nina_mowas_rss\(\)](#), [nina_notfalltipps\(\)](#), [nina_version\(\)](#), [nina_warning\(\)](#), [nina_warning_geojson\(\)](#), [nina_warnings\(\)](#)

Examples

```
## Not run:  
map <- nina_mapdata("mowas")  
nina_warning_json(map$id[[1]])  
  
## End(Not run)
```

pegel_online_measurements

Get Pegel-Online measurements

Description

Get Pegel-Online measurements

Usage

```
pegel_online_measurements(
  station,
  timeseries,
  start = NULL,
  end = NULL,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

station	Station UUID, name, or number.
timeseries	Timeseries shortname.
start	Start timestamp in ISO 8601.
end	End timestamp in ISO 8601.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns measurement values for a station timeseries. Timestamps must be ISO 8601 strings. API documentation: <https://pegel-online.api.bund.dev>.

Value

A tibble with one row per measurement:

timestamp Measurement timestamp string (character).

value Measured value (numeric).

timestamp_time Measurement timestamp as POSIXct in Europe/Berlin.

See Also

[pegel_online_timeseries\(\)](#) for metadata and [pegel_online_measurements_plot\(\)](#) for PNG plots.

Other Pegel Online: [pegel_online_measurements_plot\(\)](#), [pegel_online_station\(\)](#), [pegel_online_stations\(\)](#), [pegel_online_timeseries\(\)](#), [pegel_online_waters\(\)](#)

Examples

```
## Not run:
stations <- pegel_online_stations(params = list(limit = 1))
pegel_online_measurements(stations$uuid[[1]], "W",
  start = "2024-01-01T00:00:00Z",
```

```
    end = "2024-01-02T00:00:00Z"  
  )  
  
  ## End(Not run)
```

```
pegel_online_measurements_plot  
  Get Pegel-Online measurements plot
```

Description

Get Pegel-Online measurements plot

Usage

```
pegel_online_measurements_plot(  
  station,  
  timeseries,  
  start = NULL,  
  end = NULL,  
  safe = TRUE,  
  refresh = FALSE  
)
```

Arguments

station	Station UUID, name, or number.
timeseries	Timeseries shortname.
start	Start timestamp in ISO 8601.
end	End timestamp in ISO 8601.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundel/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a PNG plot for the measurements endpoint. API documentation: <https://pegel-online.api.bund.dev>.

Value

A one-row tibble with:

station Requested station id (character).
timeseries Requested timeseries shortname (character).
png Raw PNG bytes (list-column with raw vector).

See Also

[pegel_online_measurements\(\)](#) for numeric values.

Other Pegel Online: [pegel_online_measurements\(\)](#), [pegel_online_station\(\)](#), [pegel_online_stations\(\)](#), [pegel_online_timeseries\(\)](#), [pegel_online_waters\(\)](#)

Examples

```
## Not run:
stations <- pegel_online_stations(params = list(limit = 1))
pegel_online_measurements_plot(stations$uuid[[1]], "W",
  start = "2024-01-01T00:00:00Z",
  end = "2024-01-02T00:00:00Z"
)

## End(Not run)
```

pegel_online_station *Get a Pegel-Online station*

Description

Get a Pegel-Online station

Usage

```
pegel_online_station(
  station,
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

station	Station UUID, name, or number.
params	Named list of query parameters: <ul style="list-style-type: none"> includeTimeseries Include timeseries metadata (TRUE/FALSE). includeCurrentMeasurement Include current measurement metadata (TRUE/FALSE). includeCharacteristicValues Include characteristic values (TRUE/FALSE). waters Filter by water shortnames (character vector). ids Filter by station ids (character vector). timeseries Timeseries shortname filter (character). fuzzyId Fuzzy station id/name search (character).

	latitude Latitude center for geo search (numeric).
	longitude Longitude center for geo search (numeric).
	radius Radius for geo search (numeric).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev/pegel_online_station/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

Fetches a single station record. API documentation: <https://pegel-online.api.bund.dev>.

Value

A one-row tibble with the same columns as `pegel_online_stations()`.

See Also

`bunddev_parameters()` to inspect available query parameters. `pegel_online_stations()` and `pegel_online_timeseries()`.

Other Pegel Online: `pegel_online_measurements()`, `pegel_online_measurements_plot()`, `pegel_online_stations()`, `pegel_online_timeseries()`, `pegel_online_waters()`

Examples

```
## Not run:
stations <- pegel_online_stations(params = list(limit = 1))
pegel_online_station(stations$suuid[[1]])

## End(Not run)
```

pegel_online_stations *List Pegel-Online stations*

Description

List Pegel-Online stations

Usage

```
pegel_online_stations(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params	<p>Named list of query parameters:</p> <ul style="list-style-type: none"> includeTimeseries Include timeseries metadata (TRUE/FALSE). includeCurrentMeasurement Include current measurement metadata (TRUE/FALSE). includeCharacteristicValues Include characteristic values (TRUE/FALSE). waters Filter by water shortnames (character vector). ids Filter by station ids (character vector). timeseries Timeseries shortname filter (character). fuzzyId Fuzzy station id/name search (character). latitude Latitude center for geo search (numeric). longitude Longitude center for geo search (numeric). radius Radius for geo search (numeric).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	<p>How to handle list columns when <code>flatten = TRUE</code>:</p> <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

The Pegel-Online API provides water level station metadata. Use query parameters to filter stations by water, ids, or location. API documentation: <https://pegel-online.api.bund.dev>.

Value

A tibble with one row per station:

uuid Station UUID (character).

number Station number/code (character).

shortname Short station name (character).

longname Long station name (character).

km River kilometer (numeric).

agency Responsible agency (character).

longitude Longitude (numeric).

latitude Latitude (numeric).

water Water metadata (list-column).

timeseries Timeseries metadata (list-column).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters. [pegel_online_station\(\)](#) for a single station and [pegel_online_measurements\(\)](#) for time series values.

Other Pegel Online: [pegel_online_measurements\(\)](#), [pegel_online_measurements_plot\(\)](#), [pegel_online_station\(\)](#), [pegel_online_timeseries\(\)](#), [pegel_online_waters\(\)](#)

Examples

```
## Not run:  
pegel_online_stations(params = list(limit = 5))  
  
## End(Not run)
```

pegel_online_timeseries

Get Pegel-Online timeseries metadata

Description

Get Pegel-Online timeseries metadata

Usage

```
pegel_online_timeseries(
  station,
  timeseries,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

station	Station UUID, name, or number.
timeseries	Timeseries shortname.
params	Named list of query parameters: includeTimeseries Include timeseries metadata (TRUE/FALSE). includeCurrentMeasurement Include current measurement metadata (TRUE/FALSE). includeCharacteristicValues Include characteristic values (TRUE/FALSE). waters Filter by water shortnames (character vector). ids Filter by station ids (character vector). timeseries Timeseries shortname filter (character). fuzzyId Fuzzy station id/name search (character). latitude Latitude center for geo search (numeric). longitude Longitude center for geo search (numeric). radius Radius for geo search (numeric).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns metadata for a timeseries, including unit and gauge zero. Official docs: <https://pegel-online.api.bund.dev>.

Value

A one-row tibble with timeseries metadata:

shortname Timeseries shortname (character).
longname Timeseries long name (character).
unit Measurement unit (character).
equidistance Sampling interval/equidistance (numeric).
gauge_zero Gauge zero metadata (list-column).
characteristic_values Characteristic values (list-column).

current_value Current measured value (numeric).

current_timestamp Current timestamp as string (character).

current_timestamp_time Current timestamp as POSIXct in Europe/Berlin.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters. [pegel_online_measurements\(\)](#) for measurement values.

Other Pegel Online: [pegel_online_measurements\(\)](#), [pegel_online_measurements_plot\(\)](#), [pegel_online_station\(\)](#), [pegel_online_stations\(\)](#), [pegel_online_waters\(\)](#)

Examples

```
## Not run:
stations <- pegel_online_stations(params = list(limit = 1))
pegel_online_timeseries(stations$uuid[[1]], "W")

## End(Not run)
```

pegel_online_waters *List Pegel-Online waters*

Description

List Pegel-Online waters

Usage

```
pegel_online_waters(
  params = list(),
  safe = TRUE,
  refresh = FALSE,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

params Named list of query parameters:

- includeTimeseries** Include timeseries metadata (TRUE/FALSE).
- includeCurrentMeasurement** Include current measurement metadata (TRUE/FALSE).
- includeCharacteristicValues** Include characteristic values (TRUE/FALSE).
- waters** Filter by water shortnames (character vector).
- ids** Filter by station ids (character vector).
- timeseries** Timeseries shortname filter (character).

	fuzzyId Fuzzy station id/name search (character).
	latitude Latitude center for geo search (numeric).
	longitude Longitude center for geo search (numeric).
	radius Radius for geo search (numeric).
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Lists waters available in Pegel-Online. API documentation: <https://pegel-online.api.bunddev>.

Value

A tibble with one row per water:

shortname Water shortname (character).

longname Water long name (character).

stations Associated stations (list-column).

timeseries Associated timeseries metadata (list-column).

See Also

`bunddev_parameters()` to inspect available query parameters. `pegel_online_stations()` for station metadata.

Other Pegel Online: `pegel_online_measurements()`, `pegel_online_measurements_plot()`, `pegel_online_station()`, `pegel_online_stations()`, `pegel_online_timeseries()`

Examples

```
## Not run:
pegel_online_waters(params = list(limit = 5))

## End(Not run)
```

psm_anwendungen *List approved applications*

Description

List approved applications

Usage

```
psm_anwendungen(
  kennr = NULL,
  awg_id = NULL,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

kennr	Optional product identification number.
awg_id	Optional application identifier (16 characters).
params	Named list of query parameters. Frequently used keys: kennr Product id (e.g. "024780-00"). wirkstoffId Active ingredient id. awg_id Application id. kultur Crop filter code. kultur_gruppe Crop group filter code. kode Code filter for catalog lookups. kodeliste Code list name. sprache Language code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde/cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns approved applications (Anwendungsgebiete) which define the combinations of products, crops, and pests for which use is permitted.

Value

A [tibble](#) with one row per approved application and columns:

ANTRAGNR Character. Application request number.

ANWENDUNGEN_ANZ_JE_BEFALL Numeric. Applications per infestation.

ANWENDUNGEN_MAX_JE_KULTUR Numeric. Max applications per crop.

ANWENDUNGEN_MAX_JE_VEGETATION Numeric. Max applications per vegetation period.

ANWENDUNGSBEREICH Character. Area of use code.

ANWENDUNGSTECHNIK Character. Application technique.

AWGNR Character. Application group number.

AWG_ID Character. Application identifier (16 chars).

AW_ABSTAND_BIS Numeric. Application interval upper bound.

AW_ABSTAND_EINHEIT Character. Interval unit.

AW_ABSTAND_VON Numeric. Application interval lower bound.

EINSATZGEBIET Character. Field of use code.

GENEHMIGUNG Character. Approval flag.

HUK Character. Home-and-garden flag.

KENNR Character. Product identification number.

KULTUR_ERL Character. Crop clarification text.

SCHADORG_ERL Character. Pest clarification text.

STADIUM_KULTUR_BEM Character. Crop growth-stage note.

STADIUM_KULTUR_BIS Character. Crop growth-stage upper bound.

STADIUM_KULTUR_KODELISTE Numeric. Crop growth-stage code list.

STADIUM_KULTUR_VON Character. Crop growth-stage lower bound.

STADIUM_SCHADORG_BEM Character. Pest growth-stage note.

STADIUM_SCHADORG_BIS Character. Pest growth-stage upper bound.

STADIUM_SCHADORG_KODELISTE Numeric. Pest growth-stage code list.

STADIUM_SCHADORG_VON Character. Pest growth-stage lower bound.

WIRKUNGSBEREICH Character. Efficacy area code.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters. [psm_mittel\(\)](#) to list products.

Other Pflanzenschutzmittelzulassung: [psm_kultur_gruppen\(\)](#), [psm_mittel\(\)](#), [psm_schadorg_gruppen\(\)](#), [psm_stand\(\)](#), [psm_wirkstoffe\(\)](#)

Examples

```
## Not run:
psm_anwendungen(kennr = "024780-00")

## End(Not run)
```

psm_kultur_gruppen *List crop groups*

Description

List crop groups

Usage

```
psm_kultur_gruppen(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params	Named list of query parameters. Frequently used keys: kennr Product id (e.g. "024780-00"). wirkstoffId Active ingredient id. avg_id Application id. kultur Crop filter code. kultur_gruppe Crop group filter code. kode Code filter for catalog lookups. kodeliste Code list name. sprache Language code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns hierarchical crop group classifications.

Value

A [tibble](#) with one row per crop-to-group mapping and columns:

KULTUR Character. Crop code (child).

KULTUR_GRUPPE Character. Crop group code (parent).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Pflanzenschutzmittelzulassung: [psm_anwendungen\(\)](#), [psm_mittel\(\)](#), [psm_schadorg_gruppen\(\)](#), [psm_stand\(\)](#), [psm_wirkstoffe\(\)](#)

Examples

```
## Not run:
psm_kultur_gruppen()

## End(Not run)
```

psm_mittel *List approved plant protection products*

Description

List approved plant protection products

Usage

```
psm_mittel(kennr = NULL, params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

kennr	Optional product identification number (9 characters).
params	Named list of query parameters. Frequently used keys: kennr Product id (e.g. "024780-00"). wirkstoffld Active ingredient id. awg_id Application id. kultur Crop filter code. kultur_gruppe Crop group filter code. kode Code filter for catalog lookups. kodeliste Code list name. sprache Language code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundde "cache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The Pflanzenschutzmittelzulassung API provides access to Germany's plant protection product database from the Bundesamt fuer Verbraucherschutz und Lebensmittelsicherheit (BVL). This function returns approved pesticides. API documentation: <https://github.com/bundesAPI/pflanzenschutzmittelzulassung>

Value

A [tibble](#) with one row per plant protection product and columns:

FORMULIERUNG_ART Character. Formulation type code.

KENNR Character. Product identification number.

MITTELNAME Character. Product trade name.

ZUL_ENDE Character. Approval end date.

ZUL_ERSTMALIG_AM Character. Initial approval date.

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters. [psm_wirkstoffe\(\)](#) to list active ingredients, [psm_stand\(\)](#) for data version.

Other Pflanzenschutzmittelzulassung: [psm_anwendungen\(\)](#), [psm_kultur_gruppen\(\)](#), [psm_schadorg_gruppen\(\)](#), [psm_stand\(\)](#), [psm_wirkstoffe\(\)](#)

Examples

```
## Not run:
psm_mittel()
psm_mittel(kennr = "024780-00")

## End(Not run)
```

psm_schadorg_gruppen *List pest groups*

Description

List pest groups

Usage

```
psm_schadorg_gruppen(params = list(), safe = TRUE, refresh = FALSE)
```

Arguments

params Named list of query parameters. Frequently used keys:

- kennr** Product id (e.g. "024780-00").
- wirkstoffld** Active ingredient id.
- awg_id** Application id.
- kultur** Crop filter code.
- kultur_gruppe** Crop group filter code.
- kode** Code filter for catalog lookups.

	kodeliste Code list name.
	sprache Language code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns hierarchical pest/pathogen group classifications.

Value

A [tibble](#) with one row per pest-to-group mapping and columns:

SCHADORG Character. Pest organism code (child).

SCHADORG_GRUPPE Character. Pest group code (parent).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters.

Other Pflanzenschutzmittelzulassung: [psm_anwendungen\(\)](#), [psm_kultur_gruppen\(\)](#), [psm_mittel\(\)](#), [psm_stand\(\)](#), [psm_wirkstoffe\(\)](#)

Examples

```
## Not run:
psm_schadorg_gruppen()

## End(Not run)
```

psm_stand

Get data version

Description

Get data version

Usage

```
psm_stand(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the release date/version of the plant protection product database.

Value

A [tibble](#) with one row and columns:

DATUM Character. Release date of the current data set.

See Also

Other Pflanzenschutzmittelzulassung: [psm_anwendungen\(\)](#), [psm_kultur_gruppen\(\)](#), [psm_mittel\(\)](#), [psm_schadorg_gruppen\(\)](#), [psm_wirkstoffe\(\)](#)

Examples

```
## Not run:
psm_stand()

## End(Not run)
```

psm_wirkstoffe	<i>List active ingredients</i>
----------------	--------------------------------

Description

List active ingredients

Usage

```
psm_wirkstoffe(
  wirkstoffId = NULL,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

wirkstoffId	Optional active ingredient ID.
params	Named list of query parameters. Frequently used keys: kennr Product id (e.g. "024780-00"). wirkstoffId Active ingredient id. avg_id Application id. kultur Crop filter code. kultur_gruppe Crop group filter code.

	kode Code filter for catalog lookups.
	kodeliste Code list name.
	sprache Language code.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns active ingredients (Wirkstoffe) from the plant protection product database.

Value

A [tibble](#) with one row per active ingredient and columns:

GENEHMIGT Character. Approval flag.

KATEGORIE Character. Ingredient category.

WIRKNR Character. Active ingredient number.

WIRKSTOFFNAME Character. Active ingredient name (German).

WIRKSTOFFNAME_EN Character. Active ingredient name (English).

See Also

[bunddev_parameters\(\)](#) to inspect available query parameters. [psm_mittel\(\)](#) to list products.

Other Pflanzenschutzmittelzulassung: [psm_anwendungen\(\)](#), [psm_kultur_gruppen\(\)](#), [psm_mittel\(\)](#), [psm_schadorg_gruppen\(\)](#), [psm_stand\(\)](#)

Examples

```
## Not run:
psm_wirkstoffe()

## End(Not run)
```

regionalatlas_query *Query Regionalatlas data*

Description

Query Regionalatlas data

Usage

```
regionalatlas_query(
  table,
  where = "1=1",
  out_fields = "*",
  return_geometry = FALSE,
  params = list(),
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

table	Table name containing indicators (e.g., "ai002_1_5" for population density). See Details for common tables.
where	Optional SQL WHERE clause for filtering.
out_fields	Fields to return (default "*" for all).
return_geometry	Logical; include geometry in response.
params	Additional ArcGIS query parameters merged with adapter defaults. Common keys: layer JSON-encoded layer definition (set automatically from table). where SQL-like filter expression (set from where). outFields Fields to return (set from out_fields). returnGeometry Whether to include geometry ("true"/"false"). f Output format ("json" by default). spatialRel Spatial relation, e.g. "esriSpatialRelIntersects".
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to tools::R_user_dir("bundelcache").
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The Regionalatlas API provides access to over 160 regional indicators from the German statistical offices. Data is available at various administrative levels (Bundeslaender, Regierungsbezirke, Kreise, Gemeinden). API documentation: <https://github.com/bundesAPI/regionalatlas-api>.

Common tables and their indicators:

- ai002_1_5: Population (ai0201=density, ai0202=change, ai0208=foreigners %)
- ai002_4_5: Age (ai0218=average age, ai0219=mother age at 1st child)
- ai008_1_5: Employment (ai0801=unemployment rate)
- ai_s_01: Disposable income per capita
- ai_s_04: SGB-II quota

- ai017_1: GDP per employee
- ai005: Federal election results

Regional levels in data: typ 1=Bundeslaender, 2=Regierungsbezirke, 3=Kreise, 5=Gemeinden. Filter using WHERE clause, e.g., where = "typ = 1".

Value

A [tibble](#) with one row per geographic feature and columns:

ags Character. Official municipality key (Amtlicher Gemeindegemeinschaftsschlüssel).

ags2 Character. Join key for indicator tables.

gen Character. Region name.

typ Integer. Administrative level (1=Bundeslaender, 2=Regierungsbezirke, 3=Kreise, 5=Gemeinden).

jahr Integer. Reference year.

jahr2 Integer. Indicator reference year.

ai... Numeric. Indicator values whose names depend on the selected table (e.g. ai0201 for population density).

See Also

[bunddev_parameters\(\)](#) for available query parameters.

Examples

```
## Not run:
# Population density indicators
regionalatlas_query("ai002_1_5")

# Filter for Bundeslaender only
regionalatlas_query("ai002_1_5", where = "typ = 1")

# Age data for Kreise
regionalatlas_query("ai002_4_5", where = "typ = 3")

## End(Not run)
```

smard_indices

List available SMARD timestamps

Description

List available SMARD timestamps

Usage

```
smard_indices(
  filter,
  region = "DE",
  resolution = "hour",
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

filter	Filter id.
region	Region code.
resolution	Data resolution.
safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The SMARD API provides power market data published by the Bundesnetzagentur. Use this helper to retrieve available timestamps for a given filter/region and resolution. API documentation: <https://smard.api.bund.dev>.

Note: The registry rate limit states that more than 60 requests per hour are not permitted. Use `safe = TRUE` to respect the limit.

Value

A tibble with one column:

timestamp Available SMARD timestamp in milliseconds since epoch (numeric).

See Also

[smard_timeseries\(\)](#) and [smard_table\(\)](#) for data retrieval, and [bunddev_parameters\(\)](#) for parameter discovery.

Other SMARD: [smard_table\(\)](#), [smard_timeseries\(\)](#)

Examples

```
## Not run:
smard_indices(410, region = "DE", resolution = "hour")

## End(Not run)
```

smard_table	<i>Fetch SMARD table data</i>
-------------	-------------------------------

Description

Fetch SMARD table data

Usage

```
smard_table(filter, region = "DE", timestamp, safe = TRUE, refresh = FALSE)
```

Arguments

<code>filter</code>	Filter id.
<code>region</code>	Region code.
<code>timestamp</code>	Timestamp from indices (ms), POSIXct, or Date. Timestamps are interpreted in the Europe/Berlin timezone.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns table-style SMARD data for a single timestamp. Use `smard_indices()` to obtain a valid timestamp. API documentation: <https://smard.api.bund.dev>.

Value

A tibble with one row per returned value/version:

timestamp Timestamp in milliseconds since epoch (numeric).

time Timestamp as POSIXct in Europe/Berlin.

value Reported value (numeric).

name Version/series label when provided by the API (character).

See Also

`smard_indices()` for timestamps and `smard_timeseries()` for time series.

Other SMARD: `smard_indices()`, `smard_timeseries()`

Examples

```
## Not run:
indices <- smard_indices(410, region = "DE", resolution = "hour")
smard_table(410, region = "DE", timestamp = indices$timestamp[[1]])

## End(Not run)
```

<code>smard_timeseries</code>	<i>Fetch SMARD timeseries data</i>
-------------------------------	------------------------------------

Description

Fetch SMARD timeseries data

Usage

```
smard_timeseries(
  filter,
  region = "DE",
  resolution = "hour",
  timestamp,
  safe = TRUE,
  refresh = FALSE
)
```

Arguments

<code>filter</code>	Filter id.
<code>region</code>	Region code.
<code>resolution</code>	Data resolution.
<code>timestamp</code>	Timestamp from indices (ms), POSIXct, or Date. Timestamps are interpreted in the Europe/Berlin timezone.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundle", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a tidy time series for a single filter/region. Use `smard_indices()` to obtain a valid timestamp. API documentation: <https://smard.api.bund.dev>.

Value

A tibble with one row per time point:

timestamp Timestamp in milliseconds since epoch (numeric).

time Timestamp as POSIXct in Europe/Berlin.

value Series value for the selected filter/region (numeric).

See Also

[smard_indices\(\)](#) for timestamps and [smard_table\(\)](#) for table output.

Other SMARD: [smard_indices\(\)](#), [smard_table\(\)](#)

Examples

```
## Not run:
indices <- smard_indices(410, region = "DE", resolution = "hour")
smard_timeseries(410, region = "DE", resolution = "hour", timestamp = indices$timestamp[[1]])

## End(Not run)
```

tagesschau_channels *Fetch Tagesschau channels*

Description

Fetch Tagesschau channels

Usage

```
tagesschau_channels(flatten = FALSE, flatten_mode = "json")
```

Arguments

flatten	Logical; if TRUE, simplify nested list columns according to flatten_mode. Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when flatten = TRUE: "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via tidyr::unnest_longer() . Warning: this can significantly increase the number of rows.

Details

Lists the Tagesschau channels endpoint. API documentation: <https://tagesschau.api.bund.dev/>.

Value

A tibble with the same columns as `tagesschau_homepage()`, restricted to channel entries.

See Also

`tagesschau_news()` and `tagesschau_homepage()`.

Other Tagesschau: `tagesschau_homepage()`, `tagesschau_news()`, `tagesschau_search()`

Examples

```
## Not run:
tagesschau_channels(flatten = TRUE)

## End(Not run)
```

tagesschau_homepage	<i>Fetch Tagesschau homepage items</i>
---------------------	--

Description

Fetch Tagesschau homepage items

Usage

```
tagesschau_homepage(flatten = FALSE, flatten_mode = "json")
```

Arguments

<code>flatten</code>	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
<code>flatten_mode</code>	How to handle list columns when <code>flatten = TRUE</code> : <ul style="list-style-type: none"> "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default. "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code>. Warning: this can significantly increase the number of rows.

Details

Fetches the Tagesschau homepage feed as provided by the ARD Tagesschau API. API documentation: <https://tagesschau.api.bund.dev/>.

Note: The registry rate limit allows up to 60 requests per hour. Usage of content is restricted to private, non-commercial use unless otherwise stated by the source (see Tagesschau CC license notes).

Value

A tibble with one row per item:

section Feed section, e.g. "news" or "regional" (character).

sophora_id Sophora CMS identifier (character).

external_id External identifier (character).

title Article title (character).

date Publication date string (character).

date_time Parsed publication date (POSIXct).

headline Topline/kicker text (character).

first_sentence First sentence of the article (character).

details API details URL (character).

detailsweb Web details URL (character).

share_url Share URL (character).

update_check_url Update-check URL (character).

region_id Region identifier (character).

ressort Editorial section/ressort (character).

type Content type (character).

breaking_news Whether the item is breaking news (logical).

copyright Copyright notice (character).

alttext Image alt text (character).

teaser_image Teaser image metadata (list-column).

tracking Tracking metadata (list-column).

tags Associated tags (list-column).

images Image collection (list-column).

streams Media streams (list-column).

geotags Geographic tags (list-column).

branding_image Branding image metadata (list-column).

first_frame First video frame metadata (list-column).

With `flatten = TRUE`, the list-columns are transformed according to `flatten_mode`.

See Also

[tagesschau_news\(\)](#), [tagesschau_search\(\)](#), and [tagesschau_channels\(\)](#).

Other Tagesschau: [tagesschau_channels\(\)](#), [tagesschau_news\(\)](#), [tagesschau_search\(\)](#)

Examples

```
## Not run:
tagesschau_homepage(flatten = TRUE)

## End(Not run)
```

tagesschau_news	<i>Fetch Tagesschau news items</i>
-----------------	------------------------------------

Description

Fetch Tagesschau news items

Usage

```
tagesschau_news(  
  regions = NULL,  
  ressort = NULL,  
  flatten = FALSE,  
  flatten_mode = "json"  
)
```

Arguments

regions	Optional region ids.
ressort	Optional ressort filter.
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Returns current news items with optional filters for region or ressort. API documentation: <https://tagesschau.api.bund.dev/>.

Value

A tibble with the same columns as `tagesschau_homepage()`, filtered by optional region/ressort parameters.

See Also

[tagesschau_homepage\(\)](#) and [tagesschau_search\(\)](#).

Other Tagesschau: [tagesschau_channels\(\)](#), [tagesschau_homepage\(\)](#), [tagesschau_search\(\)](#)

Examples

```
## Not run:
tagesschau_news(ressort = "inland", flatten = TRUE)

## End(Not run)
```

tagesschau_search	<i>Search Tagesschau content</i>
-------------------	----------------------------------

Description

Search Tagesschau content

Usage

```
tagesschau_search(
  search_text = NULL,
  page_size = NULL,
  result_page = NULL,
  flatten = FALSE,
  flatten_mode = "json"
)
```

Arguments

search_text	Query string.
page_size	Results per page.
result_page	Result page index.
flatten	Logical; if TRUE, simplify nested list columns according to <code>flatten_mode</code> . Default FALSE keeps list columns as-is.
flatten_mode	How to handle list columns when <code>flatten = TRUE</code> : "drop" Remove list columns entirely. Use when nested data is not needed. "json" Convert each list element to a JSON string. Preserves all data in a text-queryable format. This is the default . "unnest" Expand list columns into multiple rows via <code>tidyr::unnest_longer()</code> . Warning: this can significantly increase the number of rows.

Details

Searches Tagesschau content by free-text query. API documentation: <https://tagesschau.api.bund.dev/>.

Value

A tibble with the same columns as `tagesschau_homepage()`, restricted to search results.

See Also

[tagesschau_news\(\)](#) and [tagesschau_homepage\(\)](#).

Other Tagesschau: [tagesschau_channels\(\)](#), [tagesschau_homepage\(\)](#), [tagesschau_news\(\)](#)

Examples

```
## Not run:
tagesschau_search(search_text = "energie", page_size = 10, flatten = TRUE)

## End(Not run)
```

travelwarning_healthcare

List healthcare documents

Description

List healthcare documents

Usage

```
travelwarning_healthcare(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns healthcare documents provided by the Auswaertiges Amt. API documentation: <https://travelwarning.api.bund.dev>.

Value

A tibble with one row per healthcare document:

content_id Character. Entry/content identifier.

parent_content_id Character. Parent identifier for nested records.

response_country Character. Country code from response metadata.

response_last_modified Numeric. Feed-level modification timestamp.

response_last_modified_time POSIXct. Parsed feed-level timestamp.

last_modified_time POSIXct. Entry-level modification timestamp.

effective_time POSIXct. Entry effective timestamp.
last_modified Numeric. Raw entry modification timestamp in ms.
name Character. Document name.
url Character. Download URL.

See Also

Other Travelwarning: [travelwarning_representatives_country\(\)](#), [travelwarning_representatives_germany\(\)](#), [travelwarning_state_names\(\)](#), [travelwarning_warning\(\)](#), [travelwarning_warnings\(\)](#)

Examples

```
## Not run:  
travelwarning_healthcare()  
  
## End(Not run)
```

travelwarning_representatives_country
List German representatives in foreign countries

Description

List German representatives in foreign countries

Usage

```
travelwarning_representatives_country(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a list of German representatives in foreign countries. API documentation: <https://travelwarning.api.bund.dev>.

Value

A tibble with one row per German representative abroad:

content_id Character. Entry/content identifier.
parent_content_id Character. Parent identifier for nested records.
response_country Character. Country code from response metadata.
response_last_modified Numeric. Feed-level modification timestamp.
response_last_modified_time POSIXct. Parsed feed-level timestamp.
last_modified_time POSIXct. Entry-level modification timestamp.
effective_time POSIXct. Entry effective timestamp.
last_modified Numeric. Raw entry modification timestamp in ms.
title Character. Representation title.
leader Character. Head of mission.
locales Character. Languages spoken.
country Character. Country.
zip Character. Postal code.
city Character. City.
region Character. Region.
street Character. Street.
number Character. House number.
departments Character. Department details.
fax Character. Fax number.
telephone Character. Phone number.
mail Character. Email address.
misc Character. Miscellaneous notes.
url Character. External link.
postal Character. Postal address.
type Character. Address type (e.g. Botschaft).
remark Character. Remark.
open Character. Opening hours.

See Also

Other Travelwarning: [travelwarning_healthcare\(\)](#), [travelwarning_representatives_germany\(\)](#), [travelwarning_state_names\(\)](#), [travelwarning_warning\(\)](#), [travelwarning_warnings\(\)](#)

Examples

```
## Not run:
travelwarning_representatives_country()

## End(Not run)
```

travelwarning_representatives_germany
List foreign representatives in Germany

Description

List foreign representatives in Germany

Usage

```
travelwarning_representatives_germany(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns a list of foreign representatives in Germany. API documentation: <https://travelwarning.api.bund.dev>.

Value

A tibble with one row per foreign representative in Germany:

content_id Character. Entry/content identifier.
parent_content_id Character. Parent identifier for nested records.
response_country Character. Country code from response metadata.
response_last_modified Numeric. Feed-level modification timestamp.
response_last_modified_time POSIXct. Parsed feed-level timestamp.
last_modified_time POSIXct. Entry-level modification timestamp.
effective_time POSIXct. Entry effective timestamp.
last_modified Numeric. Raw entry modification timestamp in ms.
title Character. Representation title.
leader Character. Head of mission.
locales Character. Languages spoken.
country Character. Country.
zip Character. Postal code.
city Character. City.
region Character. Region.

street Character. Street.
number Character. House number.
departments Character. Department details.
fax Character. Fax number.
telephone Character. Phone number.
mail Character. Email address.
misc Character. Miscellaneous notes.
url Character. External link.
postal Character. Postal address.
type Character. Address type (e.g. Botschaft).
remark Character. Remark.
open Character. Opening hours.

See Also

Other Travelwarning: [travelwarning_healthcare\(\)](#), [travelwarning_representatives_country\(\)](#), [travelwarning_state_names\(\)](#), [travelwarning_warning\(\)](#), [travelwarning_warnings\(\)](#)

Examples

```
## Not run:
travelwarning_representatives_germany()

## End(Not run)
```

```
travelwarning_state_names
      List state names documents
```

Description

List state names documents

Usage

```
travelwarning_state_names(safe = TRUE, refresh = FALSE)
```

Arguments

safe	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
refresh	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns documents with state name information. API documentation: <https://travelwarning.api.bund.dev>.

Value

A tibble with one row per state-name document:

content_id Character. Entry/content identifier.
parent_content_id Character. Parent identifier for nested records.
response_country Character. Country code from response metadata.
response_last_modified Numeric. Feed-level modification timestamp.
response_last_modified_time POSIXct. Parsed feed-level timestamp.
last_modified_time POSIXct. Entry-level modification timestamp.
effective_time POSIXct. Entry effective timestamp.
last_modified Numeric. Raw entry modification timestamp in ms.
name Character. Document name.
url Character. Download URL.

See Also

Other Travelwarning: [travelwarning_healthcare\(\)](#), [travelwarning_representatives_country\(\)](#), [travelwarning_representatives_germany\(\)](#), [travelwarning_warning\(\)](#), [travelwarning_warnings\(\)](#)

Examples

```
## Not run:  
travelwarning_state_names()  
  
## End(Not run)
```

`travelwarning_warning` *Get a travel warning by content id*

Description

Get a travel warning by content id

Usage

```
travelwarning_warning(content_id, safe = TRUE, refresh = FALSE)
```

Arguments

<code>content_id</code>	Travel warning content id.
<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bundde", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

Returns the full travel warning content, including HTML blocks. API documentation: <https://travelwarning.api.bund.dev>.

Value

A tibble with one row per warning content block. Same metadata columns as `travelwarning_warnings()`, plus the full warning detail fields:

content_id	Character. Entry/content identifier.
parent_content_id	Character. Parent identifier for nested records.
response_country	Character. Country code from response metadata.
response_last_modified	Numeric. Feed-level modification timestamp in milliseconds.
response_last_modified_time	POSIXct. Parsed feed-level timestamp (Europe/Berlin).
last_modified_time	POSIXct. Entry-level modification timestamp (Europe/Berlin).
effective_time	POSIXct. Entry effective timestamp (Europe/Berlin).
last_modified	Numeric. Raw entry modification timestamp in ms.
effective	Numeric. Raw entry effective timestamp in ms.
title	Character. Country travel warning title.
country_code	Character. Two-letter country code.
iso3_country_code	Character. Three-letter ISO country code.
country_name	Character. Country name (German).
warning	Logical. Full travel warning issued.
partial_warning	Logical. Partial travel warning issued.
situation_warning	Logical. Situation-based travel warning.
situation_part_warning	Logical. Situation-based partial warning.
content	Character. HTML content of the warning.

See Also

`travelwarning_warnings()` to list available ids.

Other Travelwarning: `travelwarning_healthcare()`, `travelwarning_representatives_country()`, `travelwarning_representatives_germany()`, `travelwarning_state_names()`, `travelwarning_warnings()`

Examples

```
## Not run:
warnings <- travelwarning_warnings()
travelwarning_warning(warnings$content_id[[1]])

## End(Not run)
```

```
travelwarning_warnings
      List travel warnings
```

Description

List travel warnings

Usage

```
travelwarning_warnings(safe = TRUE, refresh = FALSE)
```

Arguments

<code>safe</code>	Logical; if TRUE (default), apply rate-limiting and cache GET responses to <code>tools::R_user_dir("bunddev", "cache")</code> .
<code>refresh</code>	Logical; if TRUE, ignore cached responses and re-fetch from the API (default FALSE).

Details

The travelwarning API provides travel and safety information from the Auswaertiges Amt. This endpoint returns a list of all travel warnings with metadata. API documentation: <https://travelwarning.api.bund.dev>.

Value

A tibble with one row per warning entry.

content_id Character. Entry/content identifier.

parent_content_id Character. Parent identifier for nested records.

response_country Character. Country code from response metadata.

response_last_modified Numeric. Feed-level modification timestamp in milliseconds.

response_last_modified_time POSIXct. Parsed feed-level timestamp (Europe/Berlin).

last_modified_time POSIXct. Entry-level modification timestamp (Europe/Berlin).

effective_time POSIXct. Entry effective timestamp (Europe/Berlin).

last_modified Numeric. Raw entry modification timestamp in ms.

effective Numeric. Raw entry effective timestamp in ms.
title Character. Country travel warning title.
country_code Character. Two-letter country code.
country_name Character. Country name (German).
warning Logical. Full travel warning issued.
partial_warning Logical. Partial travel warning issued.
situation_warning Logical. Situation-based travel warning.
situation_part_warning Logical. Situation-based partial warning.

See Also

[travelwarning_warning\(\)](#) for full details of a single warning.

Other Travelwarning: [travelwarning_healthcare\(\)](#), [travelwarning_representatives_country\(\)](#), [travelwarning_representatives_germany\(\)](#), [travelwarning_state_names\(\)](#), [travelwarning_warning\(\)](#)

Examples

```
## Not run:  
travelwarning_warnings()  
  
## End(Not run)
```

Index

- * **Abfallnavi**
 - [abfallnavi_fraktionen](#), [6](#)
 - [abfallnavi_fraktionen_hausnummern](#), [7](#)
 - [abfallnavi_fraktionen_strassen](#), [8](#)
 - [abfallnavi_hausnummern](#), [9](#)
 - [abfallnavi_kalender_download](#), [10](#)
 - [abfallnavi_ort](#), [11](#)
 - [abfallnavi_orte](#), [12](#)
 - [abfallnavi_strassen](#), [13](#)
 - [abfallnavi_termine_hausnummern](#), [14](#)
 - [abfallnavi_termine_strassen](#), [15](#)
- * **Ausbildungssuche**
 - [ausbildungssuche_details](#), [16](#)
 - [ausbildungssuche_search](#), [17](#)
- * **Authentication**
 - [bunddev_auth_get](#), [40](#)
 - [bunddev_auth_set](#), [40](#)
- * **Autobahn**
 - [autobahn_charging_station_details](#), [21](#)
 - [autobahn_charging_stations](#), [19](#)
 - [autobahn_closure_details](#), [23](#)
 - [autobahn_closures](#), [22](#)
 - [autobahn_parking_lorries](#), [24](#)
 - [autobahn_parking_lorry_details](#), [26](#)
 - [autobahn_roads](#), [27](#)
 - [autobahn_roadwork_details](#), [29](#)
 - [autobahn_roadworks](#), [28](#)
 - [autobahn_warning_details](#), [32](#)
 - [autobahn_warnings](#), [30](#)
 - [autobahn_webcam_details](#), [35](#)
 - [autobahn_webcams](#), [33](#)
- * **Bewerberboerse**
 - [bewerberboerse_details](#), [36](#)
 - [bewerberboerse_search](#), [38](#)
- * **Bundeshaushalt**
 - [bundeshaushalt_budget_data](#), [59](#)
- * **Bundesrat**
 - [bundesrat_aktuelles](#), [61](#)
 - [bundesrat_mitglieder](#), [62](#)
 - [bundesrat_plenum_aktuelle_sitzung](#), [63](#)
 - [bundesrat_plenum_chronologisch](#), [64](#)
 - [bundesrat_plenum_kompakt](#), [65](#)
 - [bundesrat_plenum_naechste_sitzungen](#), [66](#)
 - [bundesrat_praesidium](#), [67](#)
 - [bundesrat_startlist](#), [68](#)
 - [bundesrat_stimmverteilung](#), [69](#)
 - [bundesrat_termine](#), [70](#)
- * **Bundestag Lobbyregister**
 - [bundestag_lobbyregister_search](#), [75](#)
- * **Bundestag**
 - [bundestag_article](#), [71](#)
 - [bundestag_ausschuesse](#), [72](#)
 - [bundestag_ausschuss](#), [73](#)
 - [bundestag_conferences](#), [74](#)
 - [bundestag_mdb_bio](#), [76](#)
 - [bundestag_mdb_index](#), [77](#)
 - [bundestag_speaker](#), [78](#)
 - [bundestag_video_feed](#), [79](#)
- * **DDB**
 - [ddb_institution_sectors](#), [86](#)
 - [ddb_institutions](#), [85](#)
 - [ddb_search](#), [87](#)
- * **DIP Bundestag**
 - [dip_bundestag_aktivitaet](#), [105](#)
 - [dip_bundestag_aktivitaet_list](#), [106](#)
 - [dip_bundestag_drucksache](#), [108](#)
 - [dip_bundestag_drucksache_list](#), [110](#)
 - [dip_bundestag_drucksache_text](#), [112](#)
 - [dip_bundestag_drucksache_text_list](#), [113](#)
 - [dip_bundestag_person](#), [115](#)
 - [dip_bundestag_person_list](#), [116](#)
 - [dip_bundestag_plenarprotokoll](#), [118](#)
 - [dip_bundestag_plenarprotokoll_list](#), [118](#)

- 119
- dip_bundestag_plenarprotokoll_text, 121
- dip_bundestag_plenarprotokoll_text_list, 122
- dip_bundestag_vorgang, 124
- dip_bundestag_vorgang_list, 130
- dip_bundestag_vorgangsposition, 126
- dip_bundestag_vorgangsposition_list, 128
- * **DWD**
 - dwd_alpine_forecast_text, 132
 - dwd_avalanche_warnings, 133
 - dwd_coast_warnings, 134
 - dwd_crowd_reports, 135
 - dwd_municipality_warnings, 137
 - dwd_sea_warning_text, 138
 - dwd_station_overview, 139
 - dwd_warnings_nowcast, 140
- * **Dashboard Deutschland**
 - dashboard_deutschland_geo, 80
 - dashboard_deutschland_get, 82
 - dashboard_deutschland_indicators, 84
- * **Destatis**
 - destatis_catalogue_cubes, 89
 - destatis_catalogue_tables, 90
 - destatis_data_cube, 92
 - destatis_data_table, 93
- * **Deutschlandatlas**
 - deutschlandatlas_query, 95
- * **DiGA**
 - diga_catalog_entries, 96
 - diga_charge_item_definitions, 98
 - diga_device_definitions, 99
 - diga_organizations, 100
 - diga_questionnaire_responses, 103
 - diga_questionnaires, 102
- * **Eco Visio**
 - eco_visio_counters, 142
 - eco_visio_data, 143
- * **Feiertage**
 - feiertage_list, 144
- * **Handelsregister**
 - handelsregister_search, 146
- * **Hilfsmittel**
 - hilfsmittel_nachweisschema, 147
 - hilfsmittel_produk, 149
 - hilfsmittel_produkart, 150
 - hilfsmittel_produkte, 152
 - hilfsmittel_produkgruppe, 154
 - hilfsmittel_tree, 155
 - hilfsmittel_untergruppe, 156
- * **Hochwasserzentralen**
 - hochwasserzentralen_bundeslaender, 158
 - hochwasserzentralen_bundesland_geojson, 159
 - hochwasserzentralen_bundesland_info, 160
 - hochwasserzentralen_lagepegel, 161
 - hochwasserzentralen_pegel_info, 162
- * **Jobsuche**
 - jobsuche_logo, 163
 - jobsuche_search, 164
 - jobsuche_search_app, 167
- * **Klinik Atlas**
 - klinatlas_german_places, 169
 - klinatlas_german_states, 170
 - klinatlas_hospital_detail, 171
 - klinatlas_icd_codes, 172
 - klinatlas_locations, 173
 - klinatlas_ops_codes, 174
 - klinatlas_search, 175
 - klinatlas_states, 176
- * **Ladestationen**
 - ladestationen_query, 177
- * **Lebensmittelwarnung**
 - lebensmittelwarnung_warnings, 179
- * **Luftqualitaet**
 - luftqualitaet_airquality, 180
 - luftqualitaet_airquality_limits, 182
 - luftqualitaet_annualbalances, 183
 - luftqualitaet_components, 184
 - luftqualitaet_measures, 185
 - luftqualitaet_measures_limits, 186
 - luftqualitaet_meta, 187
 - luftqualitaet_networks, 188
 - luftqualitaet_scopes, 189
 - luftqualitaet_stationsettings, 190
 - luftqualitaet_stationtypes, 191
 - luftqualitaet_thresholds, 192
 - luftqualitaet_transgressions, 193

- luftqualitaet_transgressionstypes, [194](#)
- * **MUDAB**
 - mudab_parameter_values, [213](#)
 - mudab_parameters, [206](#)
 - mudab_parameters_biolegie, [207](#)
 - mudab_parameters_biota, [209](#)
 - mudab_parameters_sediment, [210](#)
 - mudab_parameters_wasser, [211](#)
 - mudab_plc_measurements, [214](#)
 - mudab_plc_parameters, [216](#)
 - mudab_plc_stations, [217](#)
 - mudab_project_stations, [218](#)
 - mudab_stations, [220](#)
- * **Marktstammdaten**
 - marktstammdaten_filters_gaserzeugung, [195](#)
 - marktstammdaten_filters_gasverbrauch, [196](#)
 - marktstammdaten_filters_stromerzeugung, [197](#)
 - marktstammdaten_filters_stromverbrauch, [199](#)
 - marktstammdaten_gaserzeugung, [200](#)
 - marktstammdaten_gasverbrauch, [201](#)
 - marktstammdaten_stromerzeugung, [203](#)
 - marktstammdaten_stromverbrauch, [204](#)
- * **NINA**
 - nina_archive_mowas, [221](#)
 - nina_archive_mowas_mapping, [222](#)
 - nina_covid_infos, [222](#)
 - nina_covid_map, [223](#)
 - nina_covid_rules, [224](#)
 - nina_covid_ticker, [225](#)
 - nina_covid_ticker_message, [225](#)
 - nina_dashboard, [226](#)
 - nina_event_code, [227](#)
 - nina_event_codes, [228](#)
 - nina_faqs, [228](#)
 - nina_logo, [229](#)
 - nina_logos, [230](#)
 - nina_mapdata, [231](#)
 - nina_mowas_rss, [232](#)
 - nina_notfalltipps, [233](#)
 - nina_version, [233](#)
 - nina_warning, [234](#)
 - nina_warning_geojson, [236](#)
 - nina_warning_json, [237](#)
 - nina_warnings, [235](#)
- * **OpenAPI**
 - bunddev_cache_dir, [42](#)
 - bunddev_call, [43](#)
 - bunddev_call_tidy, [45](#)
 - bunddev_endpoints, [46](#)
 - bunddev_ms_to_posix, [49](#)
 - bunddev_parameter_values, [52](#)
 - bunddev_parameters, [49](#)
 - bunddev_parameters_for, [51](#)
 - bunddev_spec, [56](#)
 - bunddev_spec_path, [57](#)
 - bunddev_timestamp_to_ms, [58](#)
- * **Pegel Online**
 - pegel_online_measurements, [238](#)
 - pegel_online_measurements_plot, [240](#)
 - pegel_online_station, [241](#)
 - pegel_online_stations, [243](#)
 - pegel_online_timeseries, [244](#)
 - pegel_online_waters, [246](#)
- * **Pflanzenschutzmittelzulassung**
 - psm_anwendungen, [248](#)
 - psm_kultur_gruppen, [250](#)
 - psm_mittel, [251](#)
 - psm_schadorg_gruppen, [252](#)
 - psm_stand, [253](#)
 - psm_wirkstoffe, [254](#)
- * **Regionalatlas**
 - regionalatlas_query, [255](#)
- * **Registry**
 - bunddev_info, [47](#)
 - bunddev_list, [48](#)
 - bunddev_registry, [55](#)
 - bunddev_search, [56](#)
- * **SMARD**
 - smard_indices, [257](#)
 - smard_table, [259](#)
 - smard_timeseries, [260](#)
- * **Tagesschau**
 - tagesschau_channels, [261](#)
 - tagesschau_homepage, [262](#)
 - tagesschau_news, [264](#)
 - tagesschau_search, [265](#)
- * **Throttling**
 - bunddev_rate_limit_get, [53](#)

- bunddev_rate_limit_set, 54
- * **Travelwarning**
 - travelwarning_healthcare, 266
 - travelwarning_representatives_country, 267
 - travelwarning_representatives_germany, 269
 - travelwarning_state_names, 270
 - travelwarning_warning, 271
 - travelwarning_warnings, 273
- abfallnavi_fraktionen, 6, 7–9, 11–15
- abfallnavi_fraktionen_hausnummern, 7, 7, 8, 9, 11–15
- abfallnavi_fraktionen_strassen, 7, 8, 9, 11–15
- abfallnavi_hausnummern, 7, 8, 9, 11–15
- abfallnavi_kalender_download, 7–9, 10, 11–15
- abfallnavi_ort, 7–9, 11, 11, 12–15
- abfallnavi_orte, 7–9, 11, 12, 13–15
- abfallnavi_strassen, 7–9, 11, 12, 13, 14, 15
- abfallnavi_strassen(), 12
- abfallnavi_termine_hausnummern, 7–9, 11–13, 14, 15
- abfallnavi_termine_strassen, 7–9, 11–14, 15
- ausbildungssuche_details, 16, 19
- ausbildungssuche_details(), 19
- ausbildungssuche_search, 16, 17
- ausbildungssuche_search(), 16
- autobahn_charging_station_details, 20, 21, 23, 24, 26, 27, 29, 30, 32–35
- autobahn_charging_station_details(), 20
- autobahn_charging_stations, 19, 21, 23, 24, 26, 27, 29, 30, 32–35
- autobahn_charging_stations(), 21
- autobahn_closure_details, 20, 21, 23, 23, 26, 27, 29, 30, 32–35
- autobahn_closure_details(), 23
- autobahn_closures, 20, 21, 22, 24, 26, 27, 29, 30, 32–35
- autobahn_closures(), 24
- autobahn_parking_lorries, 20, 21, 23, 24, 24, 27, 29, 30, 32–35
- autobahn_parking_lorries(), 27
- autobahn_parking_lorry_details, 20, 21, 23, 24, 26, 26, 27, 29, 30, 32–35
- autobahn_parking_lorry_details(), 26
- autobahn_roads, 20, 21, 23, 24, 26, 27, 27, 29, 30, 32–35
- autobahn_roads(), 20, 23, 26, 29, 32, 34
- autobahn_roadwork_details, 20, 21, 23, 24, 26, 27, 29, 29, 32–35
- autobahn_roadwork_details(), 29
- autobahn_roadworks, 20, 21, 23, 24, 26, 27, 28, 30, 32–35
- autobahn_roadworks(), 27, 30
- autobahn_warning_details, 20, 21, 23, 24, 26, 27, 29, 30, 32, 32, 34, 35
- autobahn_warning_details(), 32
- autobahn_warnings, 20, 21, 23, 24, 26, 27, 29, 30, 30, 33–35
- autobahn_warnings(), 27, 33
- autobahn_webcam_details, 20, 21, 23, 24, 26, 27, 29, 30, 32–34, 35
- autobahn_webcam_details(), 34
- autobahn_webcams, 20, 21, 23, 24, 26, 27, 29, 30, 32, 33, 33, 35
- autobahn_webcams(), 35
- bewerberboerse_details, 36, 39
- bewerberboerse_details(), 39
- bewerberboerse_search, 37, 38
- bewerberboerse_search(), 36, 37
- bunddev_auth_get, 40, 42
- bunddev_auth_get(), 42
- bunddev_auth_set, 40, 40
- bunddev_auth_set(), 18, 19, 37–40, 44, 88, 131, 164–166, 179, 180
- bunddev_cache_dir, 42, 44–46, 49–52, 57, 58
- bunddev_call, 43, 43, 45, 46, 49–52, 57, 58
- bunddev_call(), 41, 42, 45
- bunddev_call_tidy, 43, 44, 45, 46, 49–52, 57, 58
- bunddev_call_tidy(), 44
- bunddev_endpoints, 43–45, 46, 49–52, 57, 58
- bunddev_endpoints(), 57
- bunddev_info, 47, 48, 55, 56
- bunddev_info(), 48, 55, 56
- bunddev_list, 47, 48, 55, 56
- bunddev_list(), 47, 55, 56
- bunddev_ms_to_posix, 43–46, 49, 50–52, 57, 58

- bunddev_parameter_values, [43–46](#), [49–51](#), [52](#), [57](#), [58](#)
- bunddev_parameter_values(), [44](#), [50](#)
- bunddev_parameters, [43–46](#), [49](#), [49](#), [51](#), [52](#), [57](#), [58](#)
- bunddev_parameters(), [19](#), [38](#), [44–46](#), [51](#), [57](#), [60](#), [86](#), [88](#), [90](#), [91](#), [93](#), [94](#), [96](#), [97](#), [99](#), [102–104](#), [106](#), [108](#), [109](#), [111](#), [112](#), [114](#), [116](#), [117](#), [119](#), [121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#), [145](#), [165](#), [168](#), [178](#), [181](#), [183–185](#), [187–194](#), [201](#), [203](#), [204](#), [206](#), [242](#), [244](#), [246](#), [247](#), [249](#), [250](#), [252](#), [253](#), [255](#), [257](#), [258](#)
- bunddev_parameters_for, [43–46](#), [49](#), [50](#), [51](#), [52](#), [57](#), [58](#)
- bunddev_parameters_for(), [50](#), [52](#)
- bunddev_rate_limit_get, [53](#), [54](#)
- bunddev_rate_limit_get(), [54](#), [147](#)
- bunddev_rate_limit_set, [53](#), [54](#)
- bunddev_rate_limit_set(), [53](#)
- bunddev_registry, [47](#), [48](#), [55](#), [56](#)
- bunddev_registry(), [47](#), [48](#), [56](#)
- bunddev_search, [47](#), [48](#), [55](#), [56](#)
- bunddev_search(), [55](#)
- bunddev_spec, [43–46](#), [49–52](#), [56](#), [58](#)
- bunddev_spec(), [43](#), [46](#), [58](#)
- bunddev_spec_path, [43–46](#), [49–52](#), [57](#), [57](#), [58](#)
- bunddev_spec_path(), [43](#)
- bunddev_timestamp_to_ms, [43–46](#), [49–52](#), [57](#), [58](#), [58](#)
- bundeshaushalt_budget_data, [59](#)
- bundesrat_aktuelles, [61](#), [62](#), [63](#), [65–71](#)
- bundesrat_mitglieder, [61](#), [62](#), [63](#), [65–71](#)
- bundesrat_plenum_aktuelle_sitzung, [61](#), [62](#), [63](#), [65–71](#)
- bundesrat_plenum_chronologisch, [61–63](#), [64](#), [66–71](#)
- bundesrat_plenum_kompakt, [61–63](#), [65](#), [65](#), [67–71](#)
- bundesrat_plenum_naechste_sitzungen, [61–63](#), [65](#), [66](#), [66](#), [68–71](#)
- bundesrat_praesidium, [61–63](#), [65–67](#), [67](#), [69–71](#)
- bundesrat_startlist, [61–63](#), [65–68](#), [68](#), [70](#), [71](#)
- bundesrat_stimmverteilung, [61–63](#), [65–69](#), [69](#), [71](#)
- bundesrat_termine, [61–63](#), [65–70](#), [70](#)
- bundestag_article, [71](#), [73–75](#), [77–80](#)
- bundestag_article(), [79](#)
- bundestag_ausschuesse, [72](#), [72](#), [74](#), [75](#), [77–80](#)
- bundestag_ausschuss, [72](#), [73](#), [73](#), [75](#), [77–80](#)
- bundestag_conferences, [72–74](#), [74](#), [77–80](#)
- bundestag_lobbyregister_search, [75](#)
- bundestag_mdb_bio, [72–75](#), [76](#), [78–80](#)
- bundestag_mdb_index, [72–75](#), [77](#), [77](#), [79](#), [80](#)
- bundestag_speaker, [72–75](#), [77](#), [78](#), [78](#), [80](#)
- bundestag_video_feed, [72–75](#), [77–79](#), [79](#)
- dashboard_deutschland_geo, [80](#), [83](#), [84](#)
- dashboard_deutschland_get, [81](#), [82](#), [84](#)
- dashboard_deutschland_indicators, [81](#), [83](#), [84](#)
- ddb_institution_sectors, [86](#), [86](#), [88](#)
- ddb_institutions, [85](#), [87](#), [88](#)
- ddb_search, [86](#), [87](#), [87](#)
- destatis_catalogue_cubes, [89](#), [91](#), [93](#), [94](#)
- destatis_catalogue_tables, [90](#), [90](#), [93](#), [94](#)
- destatis_data_cube, [90](#), [91](#), [92](#), [94](#)
- destatis_data_table, [90](#), [91](#), [93](#), [93](#)
- deutschlandatlas_query, [95](#)
- diga_catalog_entries, [96](#), [99](#), [100](#), [102–104](#)
- diga_charge_item_definitions, [97](#), [98](#), [100](#), [102–104](#)
- diga_device_definitions, [97](#), [99](#), [99](#), [102–104](#)
- diga_organizations, [97](#), [99](#), [100](#), [100](#), [103](#), [104](#)
- diga_questionnaire_responses, [97](#), [99](#), [100](#), [102](#), [103](#), [103](#)
- diga_questionnaires, [97](#), [99](#), [100](#), [102](#), [102](#), [104](#)
- dip_bundestag_aktivitaet, [105](#), [108](#), [109](#), [111](#), [112](#), [114](#), [116](#), [117](#), [119](#), [121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_aktivitaet_list, [106](#), [106](#), [109](#), [111](#), [112](#), [114](#), [116](#), [117](#), [119](#), [121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_drucksache, [106](#), [108](#), [108](#), [111](#), [112](#), [114](#), [116](#), [117](#), [119](#), [121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_drucksache(), [112](#)

- dip_bundestag_drucksache_list, [106](#), [108](#),
[109](#), [110](#), [112](#), [114](#), [116](#), [117](#), [119](#),
[121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_drucksache_text, [106](#), [108](#),
[109](#), [111](#), [112](#), [114](#), [116](#), [117](#), [119](#),
[121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_drucksache_text_list,
[106](#), [108](#), [109](#), [111](#), [112](#), [113](#), [116](#),
[117](#), [119](#), [121](#), [122](#), [124](#), [125](#), [127](#),
[129](#), [131](#)
- dip_bundestag_person, [106](#), [108](#), [109](#), [111](#),
[112](#), [114](#), [115](#), [117](#), [119](#), [121](#), [122](#),
[124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_person_list, [106](#), [108](#), [109](#),
[111](#), [112](#), [114](#), [116](#), [116](#), [119](#), [121](#),
[122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_plenarprotokoll, [106](#), [108](#),
[109](#), [111](#), [112](#), [114](#), [116](#), [117](#), [118](#),
[121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [131](#)
- dip_bundestag_plenarprotokoll(), [122](#)
- dip_bundestag_plenarprotokoll_list,
[106](#), [108](#), [109](#), [111](#), [112](#), [114](#), [116](#),
[117](#), [119](#), [119](#), [122](#), [124](#), [125](#), [127](#),
[129](#), [131](#)
- dip_bundestag_plenarprotokoll_text,
[106](#), [108](#), [109](#), [111](#), [112](#), [114](#), [116](#),
[117](#), [119](#), [121](#), [121](#), [124](#), [125](#), [127](#),
[129](#), [131](#)
- dip_bundestag_plenarprotokoll_text_list,
[106](#), [108](#), [109](#), [111](#), [112](#), [114](#), [116](#),
[117](#), [119](#), [121](#), [122](#), [122](#), [125](#), [127](#),
[129](#), [131](#)
- dip_bundestag_vorgang, [106](#), [108](#), [109](#), [111](#),
[112](#), [114](#), [116](#), [117](#), [119](#), [121](#), [122](#),
[124](#), [124](#), [127](#), [129](#), [131](#)
- dip_bundestag_vorgang_list, [106](#), [108](#),
[109](#), [111](#), [112](#), [114](#), [116](#), [117](#), [119](#),
[121](#), [122](#), [124](#), [125](#), [127](#), [129](#), [130](#)
- dip_bundestag_vorgangsposition, [106](#),
[108](#), [109](#), [111](#), [112](#), [114](#), [116](#), [117](#),
[119](#), [121](#), [122](#), [124](#), [125](#), [126](#), [129](#),
[131](#)
- dip_bundestag_vorgangsposition_list,
[106](#), [108](#), [109](#), [111](#), [112](#), [114](#), [116](#),
[117](#), [119](#), [121](#), [122](#), [124](#), [125](#), [127](#),
[128](#), [131](#)
- dwd_alpine_forecast_text, [132](#), [133](#), [135](#),
[136](#), [138](#), [140](#), [141](#)
- dwd_alpine_forecast_text(), [133](#), [138](#)
- dwd_avalanche_warnings, [133](#), [133](#), [135](#),
[136](#), [138](#), [140](#), [141](#)
- dwd_avalanche_warnings(), [133](#), [138](#)
- dwd_coast_warnings, [133](#), [134](#), [136](#), [138](#),
[140](#), [141](#)
- dwd_coast_warnings(), [138](#), [141](#)
- dwd_crowd_reports, [133](#), [135](#), [135](#), [138](#), [140](#),
[141](#)
- dwd_crowd_reports(), [140](#)
- dwd_municipality_warnings, [133](#), [135](#), [136](#),
[137](#), [138](#), [140](#), [141](#)
- dwd_municipality_warnings(), [135](#), [141](#)
- dwd_sea_warning_text, [133](#), [135](#), [136](#), [138](#),
[138](#), [140](#), [141](#)
- dwd_sea_warning_text(), [133](#)
- dwd_station_overview, [133](#), [135](#), [136](#), [138](#),
[139](#), [141](#)
- dwd_station_overview(), [136](#)
- dwd_warnings_nowcast, [133](#), [135](#), [136](#), [138](#),
[140](#), [140](#)
- dwd_warnings_nowcast(), [135–138](#), [140](#)
- eco_visio_counters, [142](#), [144](#)
- eco_visio_counters(), [143](#), [144](#)
- eco_visio_data, [142](#), [143](#)
- eco_visio_data(), [142](#)
- feiertage_list, [144](#)
- handelsregister_search, [146](#)
- hilfsmittel_nachweisschema, [147](#), [150](#),
[152](#), [153](#), [155](#), [156](#), [158](#)
- hilfsmittel_produk, [148](#), [149](#), [152](#), [153](#),
[155](#), [156](#), [158](#)
- hilfsmittel_produkart, [148](#), [150](#), [150](#),
[153](#), [155](#), [156](#), [158](#)
- hilfsmittel_produkte, [148](#), [150](#), [152](#), [152](#),
[155](#), [156](#), [158](#)
- hilfsmittel_produkgruppe, [148](#), [150](#), [152](#),
[153](#), [154](#), [156](#), [158](#)
- hilfsmittel_tree, [148](#), [150](#), [152](#), [153](#), [155](#),
[155](#), [158](#)
- hilfsmittel_untergruppe, [148](#), [150](#), [152](#),
[153](#), [155](#), [156](#), [156](#)
- hochwasserzentralen_bundeslaender, [158](#),
[160–163](#)
- hochwasserzentralen_bundesland_geojson,
[159](#), [159](#), [161–163](#)

- hochwasserzentralen_bundesland_info, [159](#), [160](#), [160](#), [162](#), [163](#)
- hochwasserzentralen_lagepegel, [159–161](#), [161](#), [163](#)
- hochwasserzentralen_pegel_info, [159–162](#), [162](#)
- jobsuche_logo, [163](#), [166](#), [168](#)
- jobsuche_logo(), [166](#)
- jobsuche_search, [164](#), [164](#), [168](#)
- jobsuche_search(), [164](#), [168](#)
- jobsuche_search_app, [164](#), [166](#), [167](#)
- jobsuche_search_app(), [164](#), [166](#)
- klinikatlas_german_places, [169](#), [170–174](#), [176](#)
- klinikatlas_german_states, [169](#), [170](#), [171–174](#), [176](#)
- klinikatlas_hospital_detail, [169](#), [170](#), [171](#), [172–174](#), [176](#)
- klinikatlas_icd_codes, [169–171](#), [172](#), [173](#), [174](#), [176](#)
- klinikatlas_locations, [169–172](#), [173](#), [174](#), [176](#)
- klinikatlas_ops_codes, [169–173](#), [174](#), [176](#)
- klinikatlas_search, [169–174](#), [175](#), [176](#)
- klinikatlas_search(), [171](#)
- klinikatlas_states, [169–174](#), [176](#), [176](#)
- ladestationen_query, [177](#)
- lebensmittelwarnung_warnings, [179](#)
- luftqualitaet_airquality, [180](#), [182–194](#)
- luftqualitaet_airquality_limits, [181](#), [182](#), [183–194](#)
- luftqualitaet_annualbalances, [181](#), [182](#), [183](#), [184–194](#)
- luftqualitaet_components, [181–183](#), [184](#), [185–194](#)
- luftqualitaet_components(), [181](#)
- luftqualitaet_measures, [181–184](#), [185](#), [186–194](#)
- luftqualitaet_measures(), [181](#)
- luftqualitaet_measures_limits, [181–185](#), [186](#), [187–194](#)
- luftqualitaet_meta, [181–186](#), [187](#), [188–194](#)
- luftqualitaet_networks, [181–187](#), [188](#), [189–194](#)
- luftqualitaet_scopes, [181–188](#), [189](#), [190–194](#)
- luftqualitaet_stationsettings, [181–189](#), [190](#), [191–194](#)
- luftqualitaet_stationtypes, [181–190](#), [191](#), [192–194](#)
- luftqualitaet_thresholds, [181–191](#), [192](#), [193](#), [194](#)
- luftqualitaet_transgressions, [181–192](#), [193](#), [194](#)
- luftqualitaet_transgressiontypes, [181–193](#), [194](#)
- marktstammdaten_filters_gaserzeugung, [195](#), [197](#), [198](#), [200](#), [201](#), [203](#), [204](#), [206](#)
- marktstammdaten_filters_gaserzeugung(), [201](#)
- marktstammdaten_filters_gasverbrauch, [196](#), [196](#), [198](#), [200](#), [201](#), [203](#), [204](#), [206](#)
- marktstammdaten_filters_gasverbrauch(), [202](#)
- marktstammdaten_filters_stromerzeugung, [196](#), [197](#), [197](#), [200](#), [201](#), [203](#), [204](#), [206](#)
- marktstammdaten_filters_stromerzeugung(), [204](#)
- marktstammdaten_filters_stromverbrauch, [196–198](#), [199](#), [201](#), [203](#), [204](#), [206](#)
- marktstammdaten_filters_stromverbrauch(), [205](#)
- marktstammdaten_gaserzeugung, [196–198](#), [200](#), [200](#), [203](#), [204](#), [206](#)
- marktstammdaten_gasverbrauch, [196–198](#), [200](#), [201](#), [201](#), [204](#), [206](#)
- marktstammdaten_stromerzeugung, [196–198](#), [200](#), [201](#), [203](#), [203](#), [206](#)
- marktstammdaten_stromverbrauch, [196–198](#), [200](#), [201](#), [203](#), [204](#), [204](#)
- mudab_parameter_values, [207](#), [208](#), [210–212](#), [213](#), [215](#), [217–220](#)
- mudab_parameters, [206](#), [208](#), [210–212](#), [214](#), [215](#), [217–220](#)
- mudab_parameters_biologie, [207](#), [207](#), [210–212](#), [214](#), [215](#), [217–220](#)
- mudab_parameters_biota, [207](#), [208](#), [209](#), [211](#), [212](#), [214](#), [215](#), [217–220](#)

- mudab_parameters_sediment, 207, 208, 210, 210, 212, 214, 215, 217–220
 mudab_parameters_wasser, 207, 208, 210, 211, 211, 214, 215, 217–220
 mudab_plc_measurements, 207, 208, 210–212, 214, 214, 217–220
 mudab_plc_parameters, 207, 208, 210–212, 214, 215, 216, 218–220
 mudab_plc_stations, 207, 208, 210–212, 214, 215, 217, 217, 219, 220
 mudab_project_stations, 207, 208, 210–212, 214, 215, 217, 218, 218, 220
 mudab_stations, 207, 208, 210–212, 214, 215, 217–219, 220

 nina_archive_mowas, 221, 222–230, 232–238
 nina_archive_mowas_mapping, 221, 222, 223–230, 232–238
 nina_covid_infos, 221, 222, 222, 224–230, 232–238
 nina_covid_map, 221–223, 223, 224–230, 232–238
 nina_covid_rules, 221–224, 224, 225–230, 232–238
 nina_covid_ticker, 221–224, 225, 226–230, 232–238
 nina_covid_ticker_message, 221–225, 225, 227–230, 232–238
 nina_dashboard, 221–226, 226, 227–230, 232–238
 nina_event_code, 221–227, 227, 228–230, 232–238
 nina_event_codes, 221–227, 228, 229, 230, 232–238
 nina_faqs, 221–228, 228, 230, 232–238
 nina_logo, 221–229, 229, 230, 232–238
 nina_logos, 221–230, 230, 232–238
 nina_mapdata, 221–230, 231, 232–238
 nina_mapdata(), 236, 238
 nina_mowas_rss, 221–230, 232, 232, 233–238
 nina_notfalltipps, 221–230, 232, 233, 234–238
 nina_version, 221–230, 232, 233, 233, 235–238
 nina_warning, 221–230, 232–234, 234, 236–238
 nina_warning(), 236
 nina_warning_geojson, 221–230, 232–236, 236, 238
 nina_warning_json, 221–230, 232–237, 237
 nina_warnings, 221–230, 232–235, 235, 237, 238
 nina_warnings(), 234, 235

 pegel_online_measurements, 238, 241, 242, 244, 246, 247
 pegel_online_measurements(), 241, 244, 246
 pegel_online_measurements_plot, 239, 240, 242, 244, 246, 247
 pegel_online_measurements_plot(), 239
 pegel_online_station, 239, 241, 241, 244, 246, 247
 pegel_online_station(), 244
 pegel_online_stations, 239, 241, 242, 243, 246, 247
 pegel_online_stations(), 242, 247
 pegel_online_timeseries, 239, 241, 242, 244, 244, 247
 pegel_online_timeseries(), 239, 242
 pegel_online_waters, 239, 241, 242, 244, 246, 246

 psm_anwendungen, 248, 250, 252–255
 psm_kultur_gruppen, 249, 250, 252–255
 psm_mittel, 249, 250, 251, 253–255
 psm_mittel(), 249, 255
 psm_schadorg_gruppen, 249, 250, 252, 252, 254, 255
 psm_stand, 249, 250, 252, 253, 253, 255
 psm_stand(), 252
 psm_wirkstoffe, 249, 250, 252–254, 254
 psm_wirkstoffe(), 252

 regionalatlas_query, 255

 smard_indices, 257, 259, 261
 smard_indices(), 259–261
 smard_table, 258, 259, 261
 smard_table(), 258, 261
 smard_timeseries, 258, 259, 260
 smard_timeseries(), 258, 259

 tagesschau_channels, 261, 263, 264, 266
 tagesschau_channels(), 263
 tagesschau_homepage, 262, 262, 264, 266

tagesschau_homepage(), 262, 264–266
tagesschau_news, 262, 263, 264, 266
tagesschau_news(), 262, 263, 266
tagesschau_search, 262–264, 265
tagesschau_search(), 263, 264
tibble, 7–9, 11–15, 86, 87, 97, 98, 100, 101,
103, 104, 107, 111, 114, 117, 120,
123, 129, 131, 148, 149, 151, 153,
154, 156, 157, 169, 170, 172–176,
181–195, 197–199, 201, 202, 204,
205, 207–209, 211–213, 215, 216,
218–220, 249, 250, 252–255, 257
tidyr::unnest_longer(), 17, 19, 21, 22,
24–26, 28, 30–33, 35, 36, 38, 59, 76,
81, 82, 85, 88, 95, 97, 98, 100–102,
104, 107, 110, 113, 117, 120, 123,
128, 130, 134, 136, 137, 139, 141,
146, 148, 149, 151, 153, 154, 157,
160, 165, 168, 175, 178, 179, 195,
196, 198, 199, 201, 202, 204, 205,
219, 242, 243, 247, 261, 262, 264,
265
travelwarning_healthcare, 266, 268,
270–272, 274
travelwarning_representatives_country,
267, 267, 270–272, 274
travelwarning_representatives_germany,
267, 268, 269, 271, 272, 274
travelwarning_state_names, 267, 268, 270,
270, 272, 274
travelwarning_warning, 267, 268, 270, 271,
271, 274
travelwarning_warning(), 274
travelwarning_warnings, 267, 268,
270–272, 273
travelwarning_warnings(), 272