Package: kwb.tenders (via r-universe)

June 16, 2026

Title R Package for Automated Monitoring of German Public Procurement Portals (Vergabeportale) for KWB-Relevant Tenders

Version 0.0.0.9000

Description Logs into public procurement portals (starting with Vergabemarktplatz Brandenburg), scrapes published tenders, scores them for relevance to KWB research topics (e.g. groundwater) and renders an overview report.

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.tenders

BugReports https://github.com/KWB-R/kwb.tenders/issues

Depends R (>= 4.1.0)

Imports chromote, dplyr, httr, jsonlite, openxlsx, pdftools, rvest, stringr, tibble, utils, xml2, yaml

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Config/pak/sysreqs chromium make libicu-dev libjpeg-dev libxml2-dev libssl-dev libpoppler-cpp-dev poppler-data

Repository https://kwb-r.r-universe.dev

Date/Publication 2026-06-16 13:28:03 UTC

RemoteUrl https://github.com/KWB-R/kwb.tenders

RemoteRef HEAD

RemoteSha cf72ef2d3dc05c6a09875adb5531732202fb3b47

Contents

`apply_title_excludes` *Veto out-of-scope tenders (construction / building / maintenance)*

Description

Drops tenders that are not a fit for a research institute, two ways:

1. **title** contains a building/maintenance term (see `tender_excludes()`) and no strong water keyword rescues it (so a "Grundwasser..." title is kept);
2. **CPV** shows a works / maintenance / cleaning code (`45...` Bau, `50...` Reparatur/Wartung, `9046/9047/9061/9064/9091...` Reinigung) without an engineering-services code (`71...`); hard veto, so even "Neubau Klaeranlage" or "Reinigung Faulbehaelter" is dropped while "Ingenieurleistungen ..." stays.

Sets `is_relevant = FALSE` and records the reason in an `excluded` column. Matching folds umlauts / is case-insensitive.

Usage

```
apply_title_excludes(
  df,
  title_cols = c("Kurzbezeichnung", "Bezeichnung", "Titel"),
  keywords = tender_keywords(),
  excludes = tender_excludes()
)
```

Arguments

`df`	A scored tibble (must contain `is_relevant`).
`title_cols`	Candidate title columns (those present are used).
`keywords`	Keyword groups, for the strong-keyword rescue (default `tender_keywords()`).
`excludes`	Exclusion list (default `tender_excludes()`).

Value

`df` with vetoed rows' `is_relevant` set `FALSE` and an `excluded` column.

`berlin_tenders` *Vergabeplattform Berlin connector (HTTP, login-free)*

Description

Reads the Berlin notices (berlin.de, iTWO tender backend) over HTTP and scores them (`score_layered()`). The paginated HTML list (`?start=N`) is the primary source: it covers the full look-back window and carries the iTWO detail link per notice in a `data-href` attribute, with a date-based early stop. If the HTML cannot be parsed it falls back to the RSS feed (latest ~50), which is also used to backfill any missing links. No browser and no login required.

Usage

```
berlin_tenders(
  keywords = tender_keywords(),
  cpv_map = tender_cpv_map(),
  since_days = 30,
  max_pages = 60,
  relevant_only = TRUE,
  verbose = TRUE
)
```

Arguments

keywords	Keyword groups (default tender_keywords()).
cpv_map	CPV-to-group map (default tender_cpv_map()).
since_days	Stop paging once a page is entirely older than this many days (the list is newest-first; default 30). NULL pages up to max_pages.
max_pages	Safety cap on pages fetched (default 60; 10 notices/page).
relevant_only	Return only relevant tenders (default TRUE).
verbose	Print progress (default TRUE).

Value

A scored tibble with Plattform = "Vergabeplattform Berlin".

Examples

```
## Not run:
berlin_tenders(since_days = 30)

## End(Not run)
```

---

check_tenders *Check Vergabemarktplatz Brandenburg for relevant tenders (single-portal report)*

Description

Convenience wrapper around vmp_bb_tenders() that also writes the overview report. For the combined multi-portal run see screen_all_portals().

Usage

```
check_tenders(
  dir = "reports",
  headless = TRUE,
  login = FALSE,
  max_pages = Inf,
  publication_types = c("ExAnte", "Tender"),
  contracting_rules = "VOL",
  screen_details = TRUE,
  max_detail = Inf,
  screen_notice = FALSE,
  max_notice = Inf,
  username = Sys.getenv("VMP_BB_USERNAME"),
  password = Sys.getenv("VMP_BB_PASSWORD"),
  keywords = tender_keywords()
)
```

Arguments

`dir`	Output directory for the report and caches (default `"reports"`).
`headless`	Run chromote headless (default `TRUE`).
`login`	Log in before scraping (default `FALSE`; the search is public).
`max_pages`	Maximum number of result pages to scrape (default `Inf`).
`publication_types`, `contracting_rules`	Search filter passed to `vmp_bb_scrape_tenders()`.
`screen_details`	Detail-page layer (default `TRUE`; see `enrich_with_details()`).
`max_detail`	Maximum number of detail pages to screen (default `Inf`).
`screen_notice`	Notice-PDF layer (default `FALSE`; forces `login = TRUE`; see `enrich_with_notice()`).
`max_notice`	Maximum number of new notice PDFs to read (default `Inf`).
`username`, `password`	Credentials when `login = TRUE` (default env vars `VMP_BB_USERNAME` / `VMP_BB_PASSWORD`).
`keywords`	Keyword list for relevance scoring (default `tender_keywords()`).

Value

Invisibly, the scored tibble of all tenders.

Examples

```
## Not run:
check_tenders() # public search, all pages
check_tenders(max_pages = 2) # quick test

## End(Not run)
```

`combine_tenders` *Combine scored tender tibbles from several portal connectors*

Description

Row-binds the per-portal results, filling columns absent in some sources with `NA`, and guarantees a `Plattform` column. Each input should be a scored tibble (see `score_relevance()`) as returned by a portal connector.

Usage

```
combine_tenders(tenders_list)
```

Arguments

`tenders_list`	A list of data frames (one per portal). `NULL` entries and zero-row frames are dropped.

Value

One combined data frame (an empty data frame if all inputs are empty).

Examples

```
a <- data.frame(Plattform = "A", Kurzbezeichnung = "x", stringsAsFactors = FALSE)
b <- data.frame(Plattform = "B", cpv = "71351500-8", stringsAsFactors = FALSE)
combine_tenders(list(a, b))
```

`cosinex_tenders` *Scrape + score a cosinex Vergabemarktplatz instance (generic connector)*

Description

Shared engine behind `vmp_bb_tenders()`, `vmp_nrw_tenders()` and `dtvp_tenders()`: opens a chromote session, optionally logs in, scrapes the extended-search results, scores them (`score_relevance()`), enriches via the detail (and optional notice) layers, applies the title/CPV exclusions (`apply_title_excludes()`) and tags `Plattform = plattform`. The detail and notice caches are namespaced by `slug`, so several portals can share one `cache_dir` without clobbering each other.

Usage

```
cosinex_tenders(
  base_url,
  plattform,
  slug,
  mount = "VMPCenter",
  keywords = tender_keywords(),
  login = FALSE,
  max_pages = Inf,
  since_days = NULL,
  publication_types = c("ExAnte", "Tender"),
  contracting_rules = "VOL",
  screen_details = TRUE,
  max_detail = Inf,
  screen_notice = FALSE,
  max_notice = Inf,
  username = "",
  password = "",
  cache_dir = "reports",
  relevant_only = FALSE,
  headless = TRUE
)
```

Arguments

base_url	Portal host, e.g. "https://www.evergabe.nrw.de".
plattform	Display name written to the Plattform column.
slug	Short id used for the per-portal cache files (e.g. "vmp_nrw").
mount	cosinex mount segment: "VMPCenter" (Land marketplaces) or "Center" (DTVP).
keywords	Keyword list for relevance scoring (default tender_keywords()).
login	Log in before scraping (default FALSE; the search is public).
max_pages	Maximum number of result pages to scrape (default Inf).
since_days	If set, stop paging once a result page is entirely older than this many days (the search is sorted newest-first). Bounds the scrape for large portals/award histories; NULL scrapes up to max_pages. The precise date trim happens later in screen_portals().
publication_types, contracting_rules	Search filter passed to vmp_bb_scrape_tenders().
screen_details	Detail-page layer (default TRUE; see enrich_with_details()).
max_detail	Maximum number of detail pages to screen (default Inf).
screen_notice	Notice-PDF layer (default FALSE; forces login = TRUE; see enrich_with_notice()).
max_notice	Maximum number of new notice PDFs to read (default Inf).
username, password	Credentials when login = TRUE (default env vars VMP_BB_USERNAME / VMP_BB_PASSWORD).
cache_dir	Directory for the detail/notice caches (default "reports").
relevant_only	Return only relevant tenders (default FALSE; the combined multi-portal run in screen_all_portals() sets this TRUE).
headless	Run chromote headless (default TRUE).

Value

A scored tibble with a Plattform column.

Examples

```
## Not run:
cosinex_tenders("https://www.evergabe.nrw.de", "Vergabemarktplatz NRW",
                slug = "vmp_nrw", max_pages = 2)

## End(Not run)
```

cpv_labels *CPV code -> German label lookup*

Description

Reads the bundled CPV label table (`inst/extdata/cpv_labels.csv`, columns `code`, `name`). Edit/extend that file (or drop in the full official CPV list) to cover more codes.

Usage

```
cpv_labels(
  path = system.file("extdata", "cpv_labels.csv", package = "kwb.tenders")
)
```

Arguments

`path`	CSV file with columns `code`, `name`.

Value

A named character vector (names = CPV codes, values = German labels).

Examples

```
head(cpv_labels())
```

cpv_summary *Summarise all CPV codes found across the tenders*

Description

Aggregates the CPV codes collected by `enrich_with_details()` into a table: one row per code (`cpv_id`) with its German label (`cpv_name`, via `cpv_labels()`), the number of tenders it appears in (`n_tenders`) and the KWB research group(s) it maps to (`groups`). Used as the "CPV" sheet of the report.

Usage

```
cpv_summary(
  tenders,
  cpv_map = tender_cpv_map(),
  keywords = tender_keywords(),
  labels = cpv_labels()
)
```

Arguments

tenders	A tibble with a cpv column (comma-separated CPV codes).
cpv_map	CPV-to-group mapping (default tender_cpv_map()).
keywords	Keyword groups, for group display names (default tender_keywords()).
labels	CPV code -> name lookup (default cpv_labels()).

Value

A data.frame with columns cpv_id, cpv_name, n_tenders, groups, sorted by descending frequency.

Examples

```
cpv_summary(data.frame(cpv = c("90700000-4, 90733000-4", "90700000-4")))
```

---

dedupe_tenders *Merge duplicate tenders that appear on several portals*

Description

The same tender is often syndicated across sources (a federal tender in the Datenservice *and* in TED, a Land tender on its cosinex marketplace *and* the Datenservice, ...). Rows whose normalised title matches are collapsed to one, keeping the highest-priority platform's record (Datenservice > TED > cosinex > Berlin) and listing every source in Plattform; the relevance groups are unioned. Only titles with >= 20 normalised characters are matched, so short generic titles are never merged.

Usage

```
dedupe_tenders(tenders, verbose = TRUE)
```

Arguments

tenders	A combined scored tibble (see combine_tenders()).
verbose	Print how many rows were merged (default TRUE).

Value

tenders with cross-portal duplicates merged (fewer or equal rows).

Examples

```
a <- data.frame(Kurzbezeichnung = "Erneuerung Schaltanlage Wasserwerk Lodmannshagen",
                Plattform = "TED (EU)", groups = "Grundwasser", stringsAsFactors = FALSE)
b <- data.frame(Kurzbezeichnung = "Erneuerung Schaltanlage Wasserwerk Lodmannshagen",
                 Plattform = "Oeffentliche Vergabe (Bund)", groups = "Grundwasser",
                 stringsAsFactors = FALSE)
dedupe_tenders(combine_tenders(list(a, b)))
```

dtvp_tenders *Deutsches Vergabeportal (DTVP) connector (cosinex)*

Description

Thin wrapper around `cosinex_tenders()` for the Deutsches Vergabeportal (`dtvp.de`). DTVP uses the `"Center"` mount; its published search is login-free (registration is only needed to submit bids).

Usage

```
dtvp_tenders(keywords = tender_keywords(), ...)
```

Arguments

`keywords`	Keyword groups (default `tender_keywords()`).
`...`	Further arguments passed to `cosinex_tenders()` (e.g. `login`, `publication_types`, `contracting_rules`, `since_days`, `max_pages`, `cache_dir`, `relevant_only`).

Value

A scored tibble with `Plattform = "Deutsches Vergabeportal (DTVP)"`.

Examples

```
## Not run:
dtvp_tenders(max_pages = 2)

## End(Not run)
```

enrich_with_details *Enrich tenders with a detail-page relevance layer (rendered text + CPV codes)*

Description

For ongoing tenders that are not yet in `cache`, renders the public detail page via `session`, matches the keyword groups against its full text and maps its CPV codes to groups. Cached tenders are reused without re-fetching. The matching group(s) are merged into `groups`/`is_relevant`; adds columns `detail_groups`, `cpv`, `cpv_groups`, `match_source`. The updated cache is returned as `attr(result, "detail_cache")`.

Usage

```
enrich_with_details(
  session,
  tenders,
  keywords = tender_keywords(),
  cpv_map = tender_cpv_map(),
  ongoing_only = TRUE,
  max_detail = Inf,
  delay = 0.2,
  cache = NULL
)
```

Arguments

session	A session from vmp_bb_session().
tenders	A scored tibble (see score_relevance()).
keywords	Keyword groups (default tender_keywords()).
cpv_map	CPV-to-group mapping (default tender_cpv_map()).
ongoing_only	Only screen tenders whose deadline has not passed (default TRUE).
max_detail	Maximum number of *new* detail pages to render per call (default Inf).
delay	Seconds between detail pages (politeness; default 0.2).
cache	Detail cache from a previous run (see read_detail_cache()).

Value

tenders with the detail layer merged in; the updated cache is in attr(result, "detail_cache").

---

enrich_with_notice *Enrich tenders with a notice-PDF (Bekanntmachung) relevance layer*

Description

For ongoing tenders not yet cached, reads the published announcement PDF(s) via the logged-in session and matches the keyword groups against the text. Adds a notice_groups column, merges it into groups/is_relevant and adds the notice source to match_source. Requires a logged-in session. The updated cache is returned as attr(result, "notice_cache").

Usage

```
enrich_with_notice(
  session,
  tenders,
  keywords = tender_keywords(),
  ongoing_only = TRUE,
  max_notice = Inf,
```

```
  delay = 0.3,
  cache = NULL
)
```

Arguments

session	A logged-in session from vmp_bb_session().
tenders	A tibble (typically already passed through enrich_with_details()).
keywords	Keyword groups (default tender_keywords()).
ongoing_only	Only screen ongoing tenders (default TRUE).
max_notice	Maximum number of *new* notice PDFs to read (default Inf).
delay	Seconds between tenders (default 0.3).
cache	Notice cache from a previous run (see read_notice_cache()).

Value

tenders with the notice layer merged in.

---

oeffentlichevergabe_tenders

Screen the Datenservice Oeffentlicher Einkauf (oeffentlichevergabe.de)

Description

Login-free connector: downloads the OCDS notice export for the last days days, parses each notice and scores it with score_layered() (title full rule, description strong-only, CPV mapped). Returns relevant tenders with a Plattform column, ready for combine_tenders() / write_tender_report().

Usage

```
oeffentlichevergabe_tenders(
  keywords = tender_keywords(),
  cpv_map = tender_cpv_map(),
  days = 7,
  end = Sys.Date(),
  relevant_only = TRUE,
  verbose = TRUE
)
```

Arguments

keywords	Keyword groups (default tender_keywords()).
cpv_map	CPV-to-group map (default tender_cpv_map()).
days	Number of past days to fetch (default 7; the API serves data up to the previous day).
end	Most recent date to consider (default Sys.Date()).
relevant_only	Keep only relevant tenders (default TRUE).
verbose	Print per-day progress (default TRUE).

Value

A scored tibble of (relevant) tenders; empty data frame if none.

Examples

```
## Not run:
oeffentlichevergabe_tenders(days = 3)

## End(Not run)
```

---

read_detail_cache *Read / write the detail-screening cache*

Description

The cache (one row per already-screened tender) lets the scheduled job screen only *new* tenders and reuse earlier results; persisted with the report so it survives across runs.

Usage

```
read_detail_cache(path)

write_detail_cache(cache, path)
```

Arguments

path	Cache file path (.rds).
cache	A cache data.frame (columns tender_id, detail_groups, cpv, cpv_groups).

Value

read_detail_cache() returns the cache data.frame (empty if absent); write_detail_cache() returns path invisibly.

read_notice_cache *Read / write the notice-screening cache*

Description

Read / write the notice-screening cache

Usage

```
read_notice_cache(path)

write_notice_cache(cache, path)
```

Arguments

path	Cache file path (.rds).
cache	A cache data.frame (tender_id, notice_groups).

Value

read_notice_cache() a data.frame (empty if absent); write_notice_cache() returns path invisibly.

score_layered *Layered relevance scoring for portal connectors (title + long text + CPV)*

Description

Scores a tender tibble the way the VMP-BB pipeline does, but in one call for connectors that already ship a description and CPV codes (e.g. the API portals): title_cols use the full rule (>=1 strong OR >=2 supporting), text_cols (long free text) are matched STRONG-only (incidental supporting hits in long text are noise), and cpv_col codes are mapped to groups. The three group sets are merged into groups, with match_source (title/detail/cpv), cpv_groups, score and is_relevant.

Usage

```
score_layered(
  df,
  title_cols,
  text_cols = character(),
  cpv_col = NULL,
  keywords = tender_keywords(),
  cpv_map = tender_cpv_map(),
  exclude = TRUE
)
```

Arguments

df	A data frame of tenders.
title_cols	Columns scored with the full rule (e.g. the title).
text_cols	Columns scored strong-only (e.g. description); default none.
cpv_col	Name of a comma/space-separated CPV column, or NULL.
keywords	Keyword groups (default tender_keywords()).
cpv_map	CPV-to-group map (default tender_cpv_map()).
exclude	Apply apply_title_excludes() afterwards to drop construction / building / maintenance tenders (default TRUE).

Value

df with groups, cpv_groups, match_source, score, is_relevant added, sorted by descending score.

---

score_relevance *Score tenders for relevance to KWB research groups*

Description

Case-insensitive substring matching of each group's keywords against all character columns. A tender matches a group if it contains at least one strong keyword or at least two supporting keywords; it is relevant if it matches at least one group.

Usage

```
score_relevance(tenders, keywords = tender_keywords())
```

Arguments

tenders	A data frame / tibble of tenders (e.g. from vmp_bb_scrape_tenders()).
keywords	Keyword groups (default tender_keywords()). May also be a single group as list(strong = ..., supporting = ...).

Value

tenders with added columns groups (matching group names, comma separated), matched_keywords, score and is_relevant, sorted by descending score.

Examples

```
df <- data.frame(
  Bezeichnung = c("Grundwassermonitoring Brunnen", "Kanalsanierung Sensorik"),
  stringsAsFactors = FALSE
)
res <- score_relevance(df)
res[, c("Bezeichnung", "groups", "score")]
```

`screen_all_portals` *Screen all configured portals into one combined report*

Description

Convenience entry point (used by the scheduled GitHub Action): wires the built-in connectors – the cosinex marketplaces Vergabemarktplatz Brandenburg (`vmp_bb_tenders()`), Vergabemarktplatz NRW (`vmp_nrw_tenders()`) and DTVP (`dtvp_tenders()`), Vergabeplattform Berlin (`berlin_tenders()`), the federal Datenservice (`oeffentlichevergabe_tenders()`) and TED (`ted_tenders()`) – and runs them through `screen_portals()`. The searches are login-free (only VMP-BB optionally logs in for the notice layer), and a portal that fails is skipped (the others still produce the report).

Usage

```
screen_all_portals(
  dir = "reports",
  vmp_bb = TRUE,
  nrw = TRUE,
  dtvp = TRUE,
  berlin = TRUE,
  oeffentlichevergabe = TRUE,
  ted = TRUE,
  vmp_bb_login = FALSE,
  vmp_bb_notice = FALSE,
  nrw_login = FALSE,
  nrw_notice = FALSE,
  since_days = 30,
  cosinex_contracting_rules = "VOL",
  keywords = tender_keywords(),
  verbose = TRUE
)
```

Arguments

`dir`	Output directory (default `"reports"`).
`vmp_bb`, `nrw`, `dtvp`, `berlin`, `oeffentlichevergabe`, `ted`	Enable each source (all `TRUE`).
`vmp_bb_login`, `vmp_bb_notice`	Log in / read notice PDFs for VMP-BB (default `FALSE`; need `VMP_BB_*` secrets).
`nrw_login`, `nrw_notice`	Log in / read notice PDFs for Vergabemarktplatz NRW (default `FALSE`; need an NRW account + `VMP_NRW_*` secrets).
`since_days`	Unified look-back window in days, applied to every portal by publication date (default `30`): the API connectors fetch this many days and a final filter trims all sources (incl. VMP-BB) to the same window.

cosinex_contracting_rules
: Procurement regulations (Vergabeart) for the cosinex portals (Brandenburg/NRW/DTVP), default "VOL" (VgV / VOL/A / UVgO; excludes VOB/Bau). See vmp_bb_scrape_tenders() for other values. The API portals have no such filter (construction is excluded via the CPV-45 veto).

keywords
: Keyword groups (default tender_keywords()).

verbose
: Print progress (default TRUE).

Value

Invisibly, the combined scored tibble.

Examples

```
## Not run:
screen_all_portals(vmp_bb_login = TRUE, vmp_bb_notice = TRUE)

## End(Not run)
```

---

screen_portals *Run several portal connectors, combine and write one report*

Description

Calls each source connector (a function returning a scored tender tibble), tagging it with a Plattform, combines the results with combine_tenders() and writes one report via write_tender_report(). A source that errors is logged and skipped, so one portal failing does not abort the run.

Usage

```
screen_portals(
  sources,
  dir = "reports",
  portal = "tenders",
  keywords = tender_keywords(),
 keep_types = c("Ausschreibung", "Geplante Ausschreibung", "Vergebener Auftrag"),
  since_days = NULL,
  dedupe = TRUE,
  verbose = TRUE
)
```

Arguments

sources
: A named list of functions, each returning a scored tibble (e.g. list("TED" = function() ted_tenders())). The name is used as the Plattform if the connector does not set one.

dir
: Output directory (default "reports").

portal	File-name id for the combined report (default "tenders").
keywords	Passed to connectors that take it (currently informational).
keep_types	Keep only these Veroeffentlichungstyp values (default: Ausschreibung, Geplante Ausschreibung and Vergebener Auftrag -> own section each). NULL keeps all types.
since_days	If set, keep only notices whose Veroeffentlicht (publication date) is within the last since_days days; NULL (default) applies no date filter. Used to unify the look-back window across portals.
dedupe	Merge cross-portal duplicates with dedupe_tenders() before writing (default TRUE).
verbose	Print progress (default TRUE).

Value

Invisibly, the combined scored tibble.

Examples

```
## Not run:
screen_portals(list(
  "Oeffentliche Vergabe" = function() oeffentlichevergabe_tenders(days = 7),
  "TED" = function() ted_tenders()
))

## End(Not run)
```

---

ted_tenders *Screen TED (Tenders Electronic Daily) for relevant tenders*

Description

Login-free EU connector. Full-text queries terms (German water terms) restricted to countries, fetches matching notices and scores them with score_layered(). Returns relevant tenders with a Plattform column.

Usage

```
ted_tenders(
  keywords = tender_keywords(),
  cpv_map = tender_cpv_map(),
  terms = ted_default_terms(),
  countries = "DEU",
  since_days = 90,
  scope = "ACTIVE",
  max_pages = 5,
  page_size = 100,
  relevant_only = TRUE,
  verbose = TRUE
)
```

Arguments

keywords	Keyword groups (default tender_keywords()).
cpv_map	CPV-to-group map (default tender_cpv_map()).
terms	Full-text query terms (default: a built-in water-term set).
countries	Place-of-performance country codes (default "DEU"; NULL/character() for EU-wide).
since_days	Only notices published within the last N days (default 90, via TED today(-N)); NULL to disable. Past-deadline notices are dropped too.
scope	Notice scope ("ACTIVE", "ALL", "LATEST"; default "ACTIVE").
max_pages, page_size	Pagination caps (default 5 x 100).
relevant_only	Keep only relevant tenders (default TRUE).
verbose	Print progress (default TRUE).

Value

A scored tibble of (relevant) tenders; empty data frame if none.

Examples

```
## Not run:
ted_tenders(max_pages = 1)

## End(Not run)
```

---

tender_cpv_map *CPV-code to research-group mapping*

Description

CPV-code to research-group mapping

Usage

```
tender_cpv_map(
  path = system.file("extdata", "cpv_groups.yml", package = "kwb.tenders")
)
```

Arguments

path	YAML file mapping CPV prefixes to group slugs (inst/extdata/cpv_groups.yml).

Value

A list of entries, each a list with prefix and groups.

Examples

```
str(tender_cpv_map())
```

`tender_detail_text` *Fetch a tender detail page (rendered) and extract its text + CPV codes*

Description

Navigates the (JavaScript-rendered) public detail page via the chromote session and reads the rendered text. No login required.

Usage

```
tender_detail_text(session, url, wait = 10)
```

Arguments

`session`	A session from `vmp_bb_session()`.
`url`	Project detail URL (the `Aktion` column).
`wait`	Maximum seconds to wait for the page to render (default `10`).

Value

A list with `text` (rendered page text) and `cpv` (character vector).

Examples

```
## Not run:
session <- vmp_bb_session()
tender_detail_text(session, tenders$Aktion[1])

## End(Not run)
```

`tender_excludes` *Title-level exclusion (veto) terms*

Description

Reads `inst/extdata/keywords_exclude.yml` – terms that mark a tender as not relevant when they appear in its title (and no strong water keyword does). Used by `apply_title_excludes()`. This file is deliberately ignored by `tender_keywords()`; it is not a research group.

Usage

```
tender_excludes(
  path = system.file("extdata", "keywords_exclude.yml", package = "kwb.tenders")
)
```

Arguments

path	YAML file with a terms: list (and optional name).

Value

A list with name and terms (character vector).

---

tender_keywords *KWB research-group keywords*

Description

Reads the keyword lists for all KWB research groups shipped with the package (inst/extdata/keywords.yml). Each group has a display name and strong / supporting keyword vectors.

Usage

```
tender_keywords(dir = system.file("extdata", package = "kwb.tenders"))
```

Arguments

dir	Directory holding the per-group keyword files (keywords_<slug>.yml, one file per research group).

Value

A named list of groups (named by slug), each a list with name, strong, supporting.

Examples

```
names(tender_keywords())
```

---

tender_notice_text *Fetch and extract the text of a tender's announcement (notice) PDF(s)*

Description

Opens the (logged-in) detail page, finds the published Bekanntmachung PDF link(s) and returns their combined extracted text. Requires a logged-in session (see vmp_bb_login()); no bidder registration needed.

Usage

```
tender_notice_text(session, detail_url, max_pdfs = 3)
```

Arguments

session	A logged-in session from vmp_bb_session().
detail_url	The tender's detail URL (the Aktion column).
max_pdfs	Maximum number of PDFs to read per tender (default 3).

Value

The combined PDF text (empty string if none/!accessible).

Examples

```
## Not run:
s <- vmp_bb_session(); vmp_bb_login(s)
tender_notice_text(s, tenders$Aktion[1])

## End(Not run)
```

---

vmp_bb_login *Log in to Vergabemarktplatz Brandenburg (optional)*

Description

Logs in via the Keycloak SSO form. Note: the public tender search works *without* login (see vmp_bb_scrape_tenders()), so logging in is optional.

Usage

```
vmp_bb_login(
  session,
  username = Sys.getenv("VMP_BB_USERNAME"),
  password = Sys.getenv("VMP_BB_PASSWORD"),
  auth_url = VMP_BB_AUTH_URL
)
```

Arguments

session	A session from vmp_bb_session().
username, password	Credentials (default env vars VMP_BB_USERNAME / VMP_BB_PASSWORD).
auth_url	Login (Keycloak SSO) URL (default the Brandenburg one; other cosinex portals pass their own via cosinex_urls()).

Value

The session, invisibly. Errors if the login is rejected.

Examples

```
## Not run:
session <- vmp_bb_session()
vmp_bb_login(session)

## End(Not run)
```

`vmp_bb_scrape_tenders` *Search for and scrape tender results*

Description

Applies a filter via the portal's deep-link (the search state is a base64 JSON in the URL hash) and scrapes the result table across pages. Works without login (the search is public).

Usage

```
vmp_bb_scrape_tenders(
  session,
  publication_types = c("ExAnte", "Tender"),
  contracting_rules = "VOL",
  max_pages = Inf,
  search_url = VMP_BB_SEARCH_URL,
  stop_before = NULL
)
```

Arguments

`session`	A session from `vmp_bb_session()`.
`publication_types`	Publication types to include. Default `c("ExAnte", "Tender")` (Beabsichtigte Ausschreibung + Ausschreibung). Further option: `"ExPost"` (Vergebener Auftrag).
`contracting_rules`	Procurement regulations to include. Default `"VOL"` (VgV / VOL/A / UVgO). Others: `"VOB"`, `"VSVGV"`, `"SEKTVO"`, `"OTHER"`.
`max_pages`	Maximum number of result pages to scrape (default `Inf`).
`search_url`	Extended-search URL (default the Brandenburg one; other cosinex portals pass their own via `cosinex_urls()`).
`stop_before`	Optional `Date`: stop paging once a result page is entirely older than this (results are sorted newest-first). Bounds the scrape for large portals/award histories; `NULL` (default) scrapes up to `max_pages`.

Value

A tibble with one row per tender (all pages combined). The `Aktion` column holds the project detail URL; the `Veroeffentlichungstyp` column labels each row ("Ausschreibung" / "Geplante Ausschreibung").

Examples

```
## Not run:
session <- vmp_bb_session()
tenders <- vmp_bb_scrape_tenders(session, max_pages = 2)

## End(Not run)
```

`vmp_bb_session` *Start a chromote browser session*

Description

Creates a headless Chrome session via `chromote`. The portal performs cross-origin SSO redirects, which a direct chromote session handles reliably.

Usage

```
vmp_bb_session(headless = TRUE)
```

Arguments

`headless`	Kept for API compatibility. The chromote backend always runs headless; `FALSE` only emits a note. Use `session$view()` to watch a live session in your browser.

Value

A chromote::ChromoteSession object.

Examples

```
## Not run:
session <- vmp_bb_session()

## End(Not run)
```

`vmp_bb_tenders` *Scrape + score Vergabemarktplatz Brandenburg (portal connector)*

Description

The VMP-BB connector for `screen_portals()` / `screen_all_portals()`: a thin wrapper around `cosinex_tenders()` pinned to Vergabemarktplatz Brandenburg. It opens a chromote session, optionally logs in, scrapes tenders, scores them (`score_relevance()`), enriches via the detail and (optional) notice layers, applies the title exclusions (`apply_title_excludes()`) and tags `Plattform = "Vergabemarktplatz Brandenburg"`. Returns the scored tibble (it writes no report); the detail/notice screening caches are read/written under `cache_dir`.

Usage

```
vmp_bb_tenders(
  keywords = tender_keywords(),
  login = FALSE,
  max_pages = Inf,
  since_days = NULL,
  publication_types = c("ExAnte", "Tender"),
  contracting_rules = "VOL",
  screen_details = TRUE,
  max_detail = Inf,
  screen_notice = FALSE,
  max_notice = Inf,
  username = Sys.getenv("VMP_BB_USERNAME"),
  password = Sys.getenv("VMP_BB_PASSWORD"),
  cache_dir = "reports",
  relevant_only = FALSE,
  headless = TRUE
)
```

Arguments

keywords	Keyword list for relevance scoring (default tender_keywords()).
login	Log in before scraping (default FALSE; the search is public).
max_pages	Maximum number of result pages to scrape (default Inf).
since_days	If set, stop scraping pages older than this many days (results are newest-first); NULL (default) scrapes up to max_pages.
publication_types, contracting_rules	Search filter passed to vmp_bb_scrape_tenders().
screen_details	Detail-page layer (default TRUE; see enrich_with_details()).
max_detail	Maximum number of detail pages to screen (default Inf).
screen_notice	Notice-PDF layer (default FALSE; forces login = TRUE; see enrich_with_notice()).
max_notice	Maximum number of new notice PDFs to read (default Inf).
username, password	Credentials when login = TRUE (default env vars VMP_BB_USERNAME / VMP_BB_PASSWORD).
cache_dir	Directory for the detail/notice caches (default "reports").
relevant_only	Return only relevant tenders (default FALSE; the combined multi-portal run in screen_all_portals() sets this TRUE).
headless	Run chromote headless (default TRUE).

Value

A scored tibble with a Plattform column.

Examples

```
## Not run:
vmp_bb_tenders(max_pages = 2)

## End(Not run)
```

`vmp_nrw_tenders` *Vergabemarktplatz NRW connector (cosinex)*

Description

Thin wrapper around `cosinex_tenders()` for Vergabemarktplatz NRW (`evergabe.nrw.de`). The published search is login-free; an optional login (`login = TRUE`, or `screen_notice = TRUE` for the Bekanntmachung-PDF layer) uses the same cosinex Keycloak flow as Brandenburg and needs an NRW account.

Usage

```
vmp_nrw_tenders(
  keywords = tender_keywords(),
  username = Sys.getenv("VMP_NRW_USERNAME"),
  password = Sys.getenv("VMP_NRW_PASSWORD"),
  ...
)
```

Arguments

`keywords`	Keyword groups (default `tender_keywords()`).
`username, password`	NRW credentials for the optional login (default env vars `VMP_NRW_USERNAME` / `VMP_NRW_PASSWORD`).
`...`	Further arguments passed to `cosinex_tenders()` (e.g. `login`, `screen_notice`, `publication_types`, `contracting_rules`, `since_days`, `max_pages`, `cache_dir`, `relevant_only`).

Value

A scored tibble with `Plattform = "Vergabemarktplatz NRW"`.

Examples

```
## Not run:
vmp_nrw_tenders(max_pages = 2)

## End(Not run)
```

`write_tender_report` *Write a tender overview report (Excel + Markdown + HTML)*

Description

Writes a dated Excel workbook (sheets "Relevant", "Alle", "Neu"), a `latest.md` summary, a browsable `latest.html` (for GitHub Pages) and a small state file used to flag tenders that are new since the previous run.

Usage

```
write_tender_report(
  tenders,
  dir = "reports",
  portal = "vmp-bb",
  date = Sys.time()
)
```

Arguments

`tenders`	A scored tibble (see `score_relevance()`).
`dir`	Output directory (created if needed). Default `"reports"`.
`portal`	Short portal id used in file names. Default `"vmp-bb"`.
`date`	Report timestamp (default `Sys.time()`); its date part names the files, the full timestamp (Europe/Berlin) shows in the "Stand" line.

Value

Invisibly, a list with the written file paths and counts.

Examples

```
## Not run:
tenders <- score_relevance(vmp_bb_scrape_tenders(session))
write_tender_report(tenders)

## End(Not run)
```

Index