Package: duckspending (via r-universe)

May 31, 2026

Title Access to USAspending Data in a DuckLake

Version 0.1.0

Description Provides a connection to USAspending data in a DuckLake, allowing users to perform bulk analyses difficult to achieve with the USAspending API.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Imports cli, DBI, duckdb, lifecycle, methods, rlang, tibble, utils, vctrs

Suggests connections, dbplyr, dplyr, jsonlite, knitr, labelled, rmarkdown, rscontract, testthat (>= 3.0.0), withr, pkgdown, rprojroot

VignetteBuilder knitr

Config/testthat/edition 3

URL https://usaspending.grant-witness.us/r-package/

Contact info@grant-witness.us

Config/pak/sysreqs xz-utils

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

Date/Publication 2026-05-20 00:13:56 UTC

RemoteUrl https://usaspending.grant-witness.us/r-package/duckspending

RemoteRef HEAD

RemoteSha fd2dfb428242dad1aa9cd63c77217afcfa4a9269

Contents

`connection_view.duckspending_connection`
IDE Connections pane integration for a duckspending connection

---

Description

S3 method for `connections::connection_view()`. Builds an `rscontract::rscontract_spec()` that wires the pane's drill-down to `pane_object_list()` and `pane_object_columns()` so the IDE sees a filtered, user-facing view of the DuckLake.

Usage

```
connection_view.duckspending_connection(
  con,
  connection_code = "",
  host = "",
  name = "",
  connection_id = ""
)
```

Arguments

`con`	A `duckspending_connection`.
`connection_code`	Optional connection code shown by the IDE.
`host`, `name`, `connection_id`	See `connections::connection_view()`.

Value

The result of `rscontract::rscontract_open()`, invisibly.

duckspending_attach *Attach an additional snapshot to an open connection*

Description

Caches the snapshot's catalog locally (if needed), `ATTACH`es it under a hidden alias, and creates a top-level schema named after the date with one view per user table. After this call the snapshot is queryable as `<conn>.<YYYYMMDD>.<table>`.

Usage

```
duckspending_attach(snapshot, conn = duckspending_connection())
```

Arguments

`snapshot`	Snapshot identifier (YYYYMMDD character, `Date`, or numeric).
`conn`	A `duckspending_connection`. Defaults to the cached default `duckspending_connection()`.

Details

If `conn` is the package's cached default connection (the one `duckspending_connection()` returns with no args), the cache entry is updated in place so subsequent default-arg callers see the new state.

Value

`conn`, invisibly, with the new snapshot in its metadata.

duckspending_attached *Currently-attached snapshots on a connection*

Description

Currently-attached snapshots on a connection

Usage

```
duckspending_attached(conn = duckspending_connection())
```

Arguments

`conn`	A `duckspending_connection`. Defaults to the cached default `duckspending_connection()`.

Value

Character vector of YYYYMMDD dates, newest first.

`duckspending_clear_cache` *Clear the on-disk HTTP cache for a duckspending connection*

Description

Calls `cache_httpfs_clear_cache()` on the underlying DuckDB session to delete all cached parquet blocks from disk. Useful when you want to force fresh fetches or reclaim disk space.

Usage

```
duckspending_clear_cache(conn = duckspending_connection())
```

Arguments

`conn`	A `duckspending_connection`. Defaults to the cached default `duckspending_connection()`.

Details

Has no effect (with a message) if the connection was opened with `cache = FALSE` or if the `cache_httpfs` extension failed to load.

Value

`conn`, invisibly.

`duckspending_connection` *Connect to the USAspending DuckLake archive*

Description

Opens a DuckDB connection with one or more monthly DuckLake snapshots attached. Each attached snapshot is exposed as a top-level schema named after its date (`"20260206"`, `"20260306"`, ...). A `latest` schema mirrors the newest attached snapshot for convenience.

Usage

```
duckspending_connection(
  snapshots = "latest",
  base_url = default_base_url(),
  catalog_dir = tools::R_user_dir("duckspending", which = "cache"),
  download_catalog = TRUE,
  open_pane = TRUE,
  cache = .duckspending_cache_default(),
  new_conn = FALSE
)
```

Arguments

`snapshots`	Snapshot selector. One of: • `"latest"` (default): attach only the newest snapshot per `duckspending_snapshots()`. • `"all"`: attach every snapshot in the manifest. • A character vector of YYYYMMDD strings, or coercible (`Date` or numeric), naming the snapshots to attach.
`base_url`	Base URL of the public archive. Override for testing via `options(duckspending.base_url=)` or the `DUCKSPENDING_BASE_URL` env var; defaults to https://usaspending.grant-witness.us.
`catalog_dir`	Local cache directory for snapshot catalog files.
`download_catalog`	Logical; when `TRUE` (default) catalog files are cached locally before attaching, when `FALSE` the catalog is attached directly over HTTP.
`open_pane`	Logical; when `TRUE` (default) and the **connections** package is available, register the connection with the IDE Connections pane.
`cache`	Logical; when `TRUE` (default) the `cache_httpfs` community extension is loaded to persist fetched parquet blocks on disk, speeding up repeated and parallel queries. Override via `options(duckspending.cache = FALSE)` or `DUCKSPENDING_CACHE=false`. Set to `FALSE` to skip caching.
`new_conn`	Logical; when `TRUE` force a new connection even if a cached default exists.

Details

By default a single snapshot — the newest one announced by the public manifest — is attached. To attach a specific set, pass `snapshots = ...`.

The first call with default arguments caches its connection in a package-level environment; subsequent default calls reuse it, so all the other `duckspending_*` helpers that default `conn = duckspending_connection()` share a single underlying DuckDB session. Calls with non-default arguments are not cached.

Value

A `duckspending_connection` (`duckdb_connection` subclass).

Examples

```
## Not run:
con <- duckspending_connection()
duckspending_tbl("subaward_search")  # uses the cached default
DBI::dbDisconnect(con, shutdown = TRUE)

## End(Not run)
```

`duckspending_describe` *Describe snapshots, tables, and columns visible on a connection*

Description

Three modes:

- With no `table`, returns one row per user-facing table in each attached snapshot (plus the `latest` alias).
- With a `table`, returns a column-level summary for that table in the requested `snapshot` (defaults to `latest`).

Usage

```
duckspending_describe(
  table = NULL,
  snapshot = NULL,
  descriptions = duckspending_descriptions(),
  include_source = FALSE,
  conn = duckspending_connection()
)
```

Arguments

`table`	Optional table name. When `NULL`, lists snapshot/table pairs.
`snapshot`	Snapshot selector for the column-level mode. `NULL` uses the `latest` schema.
`descriptions`	Optional list returned by `duckspending_descriptions()`. When supplied, populates the `description` column.
`include_source`	If `TRUE`, the column-level mode also returns a `source` column attributing each description (e.g. `"Data Dictionary Crosswalk"`, `"Inferred"`). Defaults to `FALSE` to keep the common case tidy.
`conn`	A `duckspending_connection`. Defaults to the cached default `duckspending_connection()`.

Details

Internal bookkeeping (DuckLake metadata, hidden `_dl_*` databases, `_snapshots` marker, DuckDB metadata schemas) is filtered out.

Value

A tibble.

`duckspending_descriptions` *Fetch and parse a descriptions JSON file for a USAspending DuckLake*

Description

Descriptions live in a JSON file at the bucket root next to the catalog (default URL is derived from the production catalog URL). When the file is unreachable (e.g. not yet published, transient network error), returns `NULL` so callers can fall back to schema-only output.

Usage

```
duckspending_descriptions(
  url = NULL,
  dir = tools::R_user_dir("duckspending", which = "cache"),
  refresh = FALSE
)
```

Arguments

`url`	URL or local path to a descriptions JSON. Defaults to `<base_url>/descriptions.json`, where `base_url` is the active archive URL (`default_base_url()`).
`dir`	Directory to cache the file in.
`refresh`	Force a re-check this session (ignores the per-session memoisation). The conditional ETag/Last-Modified check still applies, so a still-fresh remote file isn't re-downloaded.

Details

The file is cached under `tools::R_user_dir()` and refreshed conditionally via the `ETag` / `Last-Modified` headers: the first call per R session sends a `HEAD` request, and a `GET` only follows when the remote validators differ from the cached ones. Subsequent calls in the same session reuse the cached parse without any HTTP traffic.

If the per-session check fails (e.g. no network), the failure is silent: a cached copy is returned when available, `NULL` otherwise, and the next call within the same session re-attempts the check — so a transient outage doesn't permanently disable the help-text refresh.

Expected JSON shape:

```
{
  "tables": {
    "<schema>.<table>": {
      "description": "...",
      "columns": {"<column>": "..."}
    }
  }
}
```

Value

A parsed list, or `NULL` if the file could not be fetched.

---

duckspending_detach *Detach a snapshot from an open connection*

Description

Drops the user-facing schema (`"YYYYMMDD"`) and DETACHes the hidden underlying catalog. Removes the snapshot from the connection metadata. Refuses to detach the snapshot currently bound to `latest`.

Usage

```
duckspending_detach(snapshot, conn = duckspending_connection())
```

Arguments

`snapshot`	Snapshot identifier.
`conn`	A `duckspending_connection`. Defaults to the cached default `duckspending_connection()`.

Value

`conn`, invisibly.

---

duckspending_label *Apply column descriptions as labels to a data frame*

Description

Stamps `attr(col, "label")` on every column in `data` whose name has a matching description in the USAspending descriptions dictionary. Both Positron's Data Explorer and RStudio's `View()` honour `attr(., "label")` and surface it under the column header — the same convention used by the `labelled` and `haven` packages.

Usage

```
duckspending_label(
  data,
  table = NULL,
  descriptions = duckspending_descriptions()
)
```

Arguments

`data`	A data frame or tibble.
`table`	Optional table name. When supplied, only descriptions for that table are used (unambiguous; recommended when you know the source). When `NULL`, descriptions from all tables are merged into a flat lookup; on collisions (same column name in multiple tables), the first match wins.
`descriptions`	Optional list returned by `duckspending_descriptions()`. `NULL` makes this a no-op (useful when the dictionary is unreachable).

Details

Columns whose names aren't in the dictionary are left untouched.

Value

`data`, with `attr(., "label")` set on matching columns.

`duckspending_label_default` *Resolve whether labels should be applied by default*

Description

Checks (in order): `options(duckspending.label = ...)`, the `DUCKSPENDING_LABEL` env var, and falls back to `TRUE`. The env var is treated as falsy if set to `"false"`, `"0"`, `"no"`, `"off"` (case-insensitive); anything else (including unset-but-checked) is truthy.

Usage

```
duckspending_label_default()
```

Value

A single logical.

`duckspending_latest` *The newest available snapshot date.*

Description

Convenience equivalent of `duckspending_snapshots()[1]`. Returns `NA_character_` if no manifest is reachable.

Usage

```
duckspending_latest(
  refresh = FALSE,
  base_url = default_base_url(),
  cache_dir = tools::R_user_dir("duckspending", which = "cache")
)
```

Arguments

`refresh`	Logical; bypass the local cache and re-fetch.
`base_url`	Base URL (default: production). Override with `options(duckspending.base_url = ...)` for testing.
`cache_dir`	Local cache directory.

Value

YYYYMMDD character scalar or `NA_character_`.

`duckspending_snapshots`
List available USAspending snapshots

Description

Fetches the small text manifest at `{base_url}/snapshots.txt` and returns the date strings, newest first. The manifest is cached in `tools::R_user_dir("duckspending", "cache")` for one hour to keep repeat calls cheap.

Usage

```
duckspending_snapshots(
  refresh = FALSE,
  base_url = default_base_url(),
  cache_dir = tools::R_user_dir("duckspending", which = "cache"),
  ttl_seconds = 3600L
)
```

Arguments

`refresh`	Logical; bypass the local cache and re-fetch.
`base_url`	Base URL (default: production). Override with `options(duckspending.base_url = ...)` for testing.
`cache_dir`	Local cache directory.
`ttl_seconds`	Cache TTL in seconds (default 1 hour).

Value

A character vector of YYYYMMDD dates, newest first. Empty character if the manifest is not found.

`duckspending_table_helpers` *Convenience accessors for the most-used USAspending tables*

Description

Each function below returns a non-collected `dplyr` table reference (i.e. a `tbl_lazy`) for one of the headline tables in the archive. They are thin wrappers around `duckspending_tbl()` that exist so the table names users care about most are exported, autocompletable, and discoverable via `?duckspending` — you don't have to remember the exact column-name spelling of `subaward_search` etc.

Usage

```
duckspending_subawards(
  snapshot = NULL,
  label = duckspending_label_default(),
  conn = duckspending_connection()
)

duckspending_assistance_transactions(
  snapshot = NULL,
  label = duckspending_label_default(),
  conn = duckspending_connection()
)

duckspending_procurement_transactions(
  snapshot = NULL,
  label = duckspending_label_default(),
  conn = duckspending_connection()
)

duckspending_agencies(
  snapshot = NULL,
```

```
  label = duckspending_label_default(),
  conn = duckspending_connection()
)

duckspending_recipients(
  snapshot = NULL,
  label = duckspending_label_default(),
  conn = duckspending_connection()
)
```

Arguments

snapshot	Optional snapshot selector (YYYYMMDD, Date, or numeric); defaults to the connection's latest schema.
label	Whether to attach column descriptions as attr(., "label") when the result is collect()ed. Defaults to duckspending_label_default().
conn	A duckspending_connection. Defaults to the cached default duckspending_connection().

Details

Everything is lazy: build a dplyr pipeline and call dplyr::collect() when you want the result in R, or run dplyr::show_query() to inspect the SQL. To use the legacy raw tables not listed here, call duckspending_tbl() directly.

Value

A tbl_lazy reference.

Functions

- duckspending_subawards(): Sub-award table (subaward_search) — one row per disclosed sub-award.
- duckspending_assistance_transactions(): Federal Assistance (FABS) transactions — source_assistance_transaction.
- duckspending_procurement_transactions(): Federal Procurement (FPDS) transactions — source_procurement_transaction.
- duckspending_agencies(): Toptier agencies (toptier_agency) — one row per department/agency at the top of the federal hierarchy.
- duckspending_recipients(): Recipient profile table (recipient_profile) — one row per known funding recipient with rolled-up totals.

`duckspending_tables` *Tables published in the USAspending DuckLake archive*

Description

This help topic is rendered dynamically from `<base>/descriptions.json` on the public archive, so it always reflects the descriptions currently shipping alongside the catalog – no package release is required when descriptions change.

Details

The table below lists every user-facing table in the archive together with a short description. Column-level descriptions are available via `duckspending_describe()`.

If the descriptions JSON is unreachable (network outage, file not yet published) the table will say so and you can still inspect the schema interactively with `duckspending_describe()`.

Table	Description
agency	Joining table that pairs each subtier agency with its top
appropriation_account_balances	DATA Act File A: TAS-level totals for each agency repo
award_category	Reference table mapping internal award type/category c
budget_authority	Aggregated agency-level budget authority figures by fisc
bureau_title_lookup	Crosswalk of agency/bureau codes to human-readable bu
c_to_d_linkage_updates	Manual corrections to the linkage between File C financ
cgac	Common Government-wide Accounting Classification (C
dabs_submission_window_schedule	Schedule of DATA Act reporting periods: open/close da
disaster_emergency_fund_code	Disaster Emergency Fund Codes (DEFC) — tag obligat
duns	Crosswalk between legacy DUNS numbers and current U
external_data_load_date	Tracks when each external feed (SAM.gov, FPDS, FABS
external_data_type	Reference table naming the external data sources tracke
federal_account	Federal Account groupings — the parent level above TA
financial_accounts_by_awards	DATA Act File C: the bridge between budget appropria
financial_accounts_by_program_activity_object_class	DATA Act File B: how each TAS spent money, broken d
frec	Federal Reporting Entity Codes — used by certain Trea
frec_map	FREC-to-CGAC reconciliation table used to map FREC
gtas_sf133_balances	GTAS/SF-133 government-wide budget execution balan
historic_parent_duns	Legacy parent-DUNS crosswalk for historical recipient m
historical_appropriation_account_balances	Pre-DATA Act TAS-level account balances, sourced fron
naics	North American Industry Classification System codes —
object_class	Object class codes (OMB Circular A-11) — classify fede
office	Contracting offices and sub-organizations within subtier
overall_totals	Top-line fiscal-year aggregates (total budget authority) u
parent_award	IDV (Indefinite Delivery Vehicle) and ordering-agreemer
psc	Product and Service Codes — more granular than NAIC
recipient_profile	Recipient-level summary records used for recipient searc
ref_city_county_state_code	City × county × state FIPS code reference used for plac
ref_country_code	ISO and USAspending country codes used in place-of-pe

ref_population_cong_district	ACS-derived population estimates by state × congressio
ref_population_county	ACS-derived population estimates by state × county. U
ref_program_activity	Program activity codes — identify the specific activity c
references_cfda	Assistance Listings (formerly CFDA — Catalog of Feder
references_definition	Glossary table with plain-language and official definition
source_assistance_transaction	Assistance transaction actions (grants, loans, direct payr
source_procurement_transaction	Contract transaction actions (FPDS) — one row per cor
state_data	State-level metadata: FIPS codes, abbreviations, and A
subaward_search	Subaward records reported by prime recipients to FSRS
submission_attributes	One row per DATA Act submission. The authoritative s
subtier_agency	Subtier agencies — bureaus, administrations, and offices
toptier_agency	Cabinet-level agencies (Department of Defense, Departm
treasury_appropriation_account	Treasury Account Symbol (TAS) definitions — the atom
zips_grouped	ZIP-to-county/congressional-district crosswalk used by g

See Also

duckspending_describe() for live, column-level details on a connected DuckLake; duckspending_descriptions(for the raw parsed JSON.

`duckspending_tbl` *Lazy table accessor for a USAspending DuckLake snapshot*

Description

Returns a non-collected `dplyr` table reference to `<snapshot>.<name>`, where `<snapshot>` defaults to the connection's `latest` schema. Verifies the table exists first so users get a clear error instead of a deferred SQL failure.

Usage

```
duckspending_tbl(
  name,
  snapshot = NULL,
  label = duckspending_label_default(),
  conn = duckspending_connection()
)
```

Arguments

`name`	Table name (no quoting needed).
`snapshot`	Optional snapshot selector. One of: • `NULL` (default): use the connection's `latest` schema. • `"latest"`: explicit equivalent of `NULL`. • YYYYMMDD character / `Date` / numeric naming an attached snapshot.

label	Whether to attach column descriptions as attr(., "label") when the lazy table is collect()ed. Defaults to duckspending_label_default() (controlled by options(duckspending.label) and DUCKSPENDING_LABEL).
conn	A duckspending_connection. Defaults to the cached default duckspending_connection().

Value

A tbl_lazy (dbplyr) reference to the table.

---

duckspending_view *Open the IDE Connections pane for a duckspending connection*

Description

Thin convenience wrapper around connections::connection_view() that dispatches to the duckspending_connection S3 method. Useful after creating a connection with open_pane = FALSE, or to re-open the pane after closing it.

Usage

```
duckspending_view(conn = duckspending_connection())
```

Arguments

conn	A duckspending_connection. Defaults to the cached default duckspending_connection().

Value

conn, invisibly.

Index