Package 'awdb'

July 22, 2025

Title Query the USDA NWCC Air and Water Database REST API

Version 0.1.2

Description Query the four endpoints of the 'Air and Water Database (AWDB) REST
API' maintained by the National Water and Climate Center (NWCC) at the
United States Department of Agriculture (USDA). Endpoints include data,
forecast, reference-data, and metadata. The package is extremely light
weight, with 'Rust' via 'extendr' doing most of the heavy lifting to
deserialize and flatten deeply nested 'JSON' responses. The AWDB can be
found at <https://wcc.sc.egov.usda.gov/awdbRestApi/swagger-ui/index.html>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2.9000

Config/rextendr/version 0.3.1.9001

SystemRequirements Cargo (Rust's package manager), rustc

Imports cli, httr2, rlang (>= 1.1.0), sf

Depends R (>= 4.2)

LazyData true

URL <https://github.com/kbvernon/awdb>, <https://kbvernon.github.io/awdb/>

BugReports <https://github.com/kbvernon/awdb/issues>

NeedsCompilation yes

Author Kenneth Blake Vernon [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0003-0098-5092>)

Maintainer Kenneth Blake Vernon <kenneth.b.vernon@gmail.com>

Repository CRAN

Date/Publication 2025-04-14 21:00:05 UTC

Contents

---

awdb_options   *Build List of Additional Query Parameters*

Description

This is a helper function to make it easier to handle additional query parameters. Provides defaults
for each and does type checking.

Usage

```
set_options(
  networks = "*",
  duration = "daily",
  begin_date = NULL,
  end_date = NULL,
  period_reference = "end",
  central_tendency = NULL,
  return_flags = FALSE,
  return_original_values = FALSE,
  return_suspect_values = FALSE,
  begin_publication_date = NULL,
  end_publication_date = NULL,
  exceedence_probabilities = NULL,
  forecast_periods = NULL,
  station_names = NULL,
  dco_codes = NULL,
  county_names = NULL,
  hucs = NULL,
  return_forecast_metadata = FALSE,
  return_reservoir_metadata = FALSE,
  return_element_metadata = FALSE,
  active_only = TRUE,
  request_size = 10L
)

## S3 method for class 'awdb_options'
print(x, ...)
```

Arguments

networks	character vector, abbreviations or codes for station networks of interest (e.g., "USGS" refers to all USGS soil monitoring stations). Default is *, for "all networks". See Details for available networks and codes.
duration	character scalar, the temporal resolution of the element measurements. Available values include `daily` (default), `hourly`, `semimonthly`, `monthly`, `calendar_year`, and `water_year`.
begin_date	character scalar, start date for time period of interest. Date must be in format ″YYYY-MM-DD″.
end_date	character scalar, end date for time period of interest. Date must be in format ″YYYY-MM-DD″.
period_reference	
	character scalar, reporting convention to use when returning instantaneous data. Default is ″end″.
central_tendency	
	character scalar, the central tendency to return for each element value. Available options include `NULL` (default, no central tendency returned), `median` and `average`.
return_flags	boolean scalar, whether to return flags with each element value. Default is `FALSE`.
return_original_values	
	boolean scalar, whether to return original element values. Default is `FALSE`.
return_suspect_values	
	boolean scalar, whether to return suspect element values. Default is `FALSE`.
begin_publication_date	
	character scalar, the beginning of the publication period for which to retrieve data. Date must be in format `YYYY-MM-DD`. If `NULL`, assumes start of the current water year.
end_publication_date	
	character scalar, the end of the publication period for which to retrieve data. Date must be in format `YYYY-MM-DD`. If `NULL`, assumes current day.
exceedence_probabilities	
	integer vector, the probability that streamflow will exceed a specified level.
forecast_periods	
	character vector, the time period over which to make streamflow forecasts.
station_names	character vector, used to subset stations by their names. Default is `NULL`.
dco_codes	character vector, used to subset stations to those that fall in specified DCOs. Default is `NULL`.
county_names	character vector, used to subset stations to those that fall in specified counties. Default is `NULL`.
hucs	integer vector, used to subset stations to those that fall in specified hydrologic units. Default is `NULL`.
return_forecast_metadata	
	boolean scalar, whether to return forecast metadata with station locations. Will be included as a list column. Default is `FALSE`.

```
return_reservoir_metadata
```
boolean scalar, whether to return reservoir metadata with station locations. Will be included as a list column. Default is FALSE.

```
return_element_metadata
```
boolean scalar, whether to return element metadata with station locations. Will be included as a list column. Default is FALSE.

`active_only`   boolean scalar, whether to include only active stations. Default is TRUE.

`request_size`   integer scalar, number of individual stations to include in each query. This helps to meet rate limits imposed by the API. If you are getting a request error, you might try lowering this number. Default is 10L.

`x`   an awdb_options list

`...`   ignored

Value

an awdb_options list

Examples

```
set_options()
```

---

datasets	*Datasets*

Description

Arbitrary bounding boxes drawn around potential areas of interest.

Usage

```
bear_lake

cascades
```

Format

An object of class sfc_POLYGON (inherits from sfc) of length 1.

Details

Areas of interest include:

- Bear Lake along the border between Utah and Idaho. See location on Google Maps.
- Cascades in west central Oregon. See location on Google Maps.

Source

Coordinates digitized manually.

`get_elements`   *Get Station Elements*

Description

Get station elements from the USDA National Water and Climate Center Air and Water Database REST API. Elements are soil, snow, stream, and weather variables measured at AWDB stations.

Usage

```
get_elements(aoi = NULL, elements, awdb_options = set_options(), as_sf = FALSE)
```

Arguments

`aoi`	`sfc` POLYGON scalar, the area of interest used for performing a spatial filter on available stations in `network`. If `NULL` (the default), no spatial filter is performed.
`elements`	character vector, abbreviations or codes for variables of interest (e.g., "SMS" for "Soil Moisture Percent"). See Details for available elements and codes.
`awdb_options`	an `awdb_options` list with additional query parameters.
`as_sf`	boolean scalar, whether to return the data as an `sf` table. Default is `FALSE`. Repeating the spatial data across each station element and its time series can be costly.

Details

This endpoint will accept the following query parameters via `set_options()`:

- `duration`
- `begin_date`
- `end_date`
- `period_reference`
- `central_tendency`
- `return_flags`
- `return_original_values`
- `return_suspect_values`

The following can also be passed to filter stations:

- `station_names`
- `dco_codes`
- `county_names`
- `hucs`
- `active_only`

You may also specify `networks` and `request_size`. The `networks` parameter is used internally to build unique station triplet identifiers of the form `station:state:network` which are then passed to the endpoint, so it serves to filter stations to just those networks. The `request_size` parameter is for handling rate limits, which are based on the number of elements - a hard value to measure directly, so this parameter is more a rule of thumb than a strict standard. If processing is slow for you, you may find experimenting with this parameter useful.

See `set_options()` for more details.

Element Format:

Elements are specified as triplets of the form `elementCode:heightDepth:ordinal`. Any part of the element triplet can contain the `*` wildcard character. Both `heightDepth` and `ordinal` are optional. The unit of `heightDepth` is inches. If `ordinal` is not specified, it is assumed to be 1. Here are some examples:

- `"WTEQ"` - return all snow water equivalent values.
- `"SMS:-8"` - return soil moisture values observed 8 inches below the surface.
- `"SMS:*"` - return soil moisture values for all measured depths.

Value

if `as_sf`, an sf table, otherwise a simple data.frame. The number of rows depends on the number of stations and element parameters. Time series data are included as a list column named `"element_values"`.

Examples

```
# get snow water equivalent values around Bear Lake
get_elements(bear_lake, elements = "WTEQ")

# return as sf table
get_elements(bear_lake, elements = "WTEQ", as_sf = TRUE)
```

---

get_forecasts   *Get Station Forecasts*

Description

Get station forecasts from the USDA National Water and Climate Center Air and Water Database REST API. These will almost always be streamflow forecasts, set with `elements = "SRVO"`, but some others are also available, albeit with extremely limited spatial representation (see Details).

Usage

```
get_forecasts(
  aoi = NULL,
  elements,
  awdb_options = set_options(),
  as_sf = FALSE
)
```

Arguments

`aoi`	`sfc` POLYGON scalar, the area of interest used for performing a spatial filter on available stations in `network`. If `NULL` (the default), no spatial filter is performed.
`elements`	character vector, abbreviations or codes for variables of interest (e.g., "SMS" for "Soil Moisture Percent"). See Details for available elements and codes.
`awdb_options`	an `awdb_options` list with additional query parameters.
`as_sf`	boolean scalar, whether to return the data as an sf table. Default is `FALSE`. Repeating the spatial data across each station element and its time series can be costly.

Details

This endpoint will accept the following query parameters via `set_options()`:

- `begin_publication_date`
- `end_publication_date`
- `exceedence_probabilities`
- `forecast_periods`

The following can also be passed to filter stations:

- `station_names`
- `dco_codes`
- `county_names`
- `hucs`
- `active_only`

You may also specify `networks` and `request_size`. The `networks` parameter is used internally to build unique station triplet identifiers of the form `station:state:network` which are then passed to the endpoint, so it serves to filter stations to just those networks. The `request_size` parameter is for handling rate limits, which are based on the number of elements - a hard value to measure directly, so this parameter is more a rule of thumb than a strict standard. If processing is slow for you, you may find experimenting with this parameter useful.

Note that the `duration` parameter is ignored - or, more precisely, it is set to `NULL`.

See `set_options()` for more details.

Element Format:

Elements are specified as triplets of the form `elementCode:heightDepth:ordinal`. Any part of the element triplet can contain the `*` wildcard character. Both `heightDepth` and `ordinal` are optional. The unit of `heightDepth` is inches. If `ordinal` is not specified, it is assumed to be 1. Here are some examples:

- `"WTEQ"` - return all snow water equivalent values.
- `"SMS:-8"` - return soil moisture values observed 8 inches below the surface.
- `"SMS:*"` - return soil moisture values for all measured depths.

Forecast Elements:

Almost all forecasts are reported in SRVO, the adjusted streamflow set which accounts for upstream operations such as reservoir operations and diversions. JDAY, RESC, and REST are mostly there to maintain historical forecasts made at Lake Tahoe (the birthplace of the snow survey). In general, it's recommended to use SRVO.

Value

if as_sf, an sf table, otherwise a simple data.frame. The number of rows depends on the number of stations and element parameters. Time series data are included as a list column named "forecast_values".

Examples

```
# get streamflow forecasts
get_forecasts(cascades, elements = "SRVO")

# return as sf table
get_forecasts(cascades, elements = "SRVO", as_sf = TRUE)
```

---

get_references *Get Data Dictionary*

Description

Get references from the USDA National Water and Climate Center Air and Water Database REST API. References provide descriptions of all codes used in the AWDB.

Usage

```
get_references(reference_type = "elements")
```

Arguments

reference_type character scalar, the name of the reference. Potential values include dcos, durations, elements (default), forecastPeriods, functions, instruments, networks, physicalElements, states, and units.

Value

a data.frame with reference data

Examples

```
get_references("elements")
```

---

get_stations   *Get Station Metadata*

Description

Get station metadata from the USDA National Water and Climate Center Air and Water Database REST API. This includes their spatial coordinates.

Usage

```
get_stations(aoi = NULL, elements, awdb_options = set_options())
```

Arguments

aoi	sfc POLYGON scalar, the area of interest used for performing a spatial filter on available stations in network. If NULL (the default), no spatial filter is performed.
elements	character vector, abbreviations or codes for variables of interest (e.g., "SMS" for "Soil Moisture Percent"). See Details for available elements and codes.
awdb_options	an awdb_options list with additional query parameters.

Details

This endpoint will accept the following query parameters via set_options():

- station_names
- dco_codes
- county_names
- hucs
- return_forecast_metadata
- return_reservoir_metadata
- return_element_metadata
- active_only

You may also specify networks. The networks parameter is used internally to build unique station triplet identifiers of the form station:state:network, so it serves to filter stations to just those networks.

See set_options() for more details.

Element Format:

Elements are specified as triplets of the form elementCode:heightDepth:ordinal. Any part of the element triplet can contain the * wildcard character. Both heightDepth and ordinal are optional. The unit of heightDepth is inches. If ordinal is not specified, it is assumed to be 1. Here are some examples:

- ″WTEQ″ - return all snow water equivalent values.
- ″SMS:-8″ - return soil moisture values observed 8 inches below the surface.
- ″SMS:*″ - return soil moisture values for all measured depths.

Value

an sf table with station metadata.

Examples

```
# get all stations in aoi
get_stations(
  bear_lake,
  elements = "*"
)

# get all stations in aoi that measure WTEQ
get_stations(
  bear_lake,
  elements = "WTEQ"
)

# get all stations in aoi that are part of SNTL network
get_stations(
  bear_lake,
  elements = "*",
  awdb_options = set_options(networks = "SNTL")
)
```

Index