Package: flowcam (via r-universe)

November 10, 2026

Type Package

Title Download and Animate USGS Stream-Gage Camera Images

Version 0.1.0.9000

Description A tidy interface to the U.S. Geological Survey (USGS) National Imagery Management System (NIMS) API (<https://api.waterdata.usgs.gov/docs/nims/overview/>), which stores and serves images collected by stream-gage cameras at monitoring locations across the United States. Provides functions to discover cameras by NWIS site number or camera ID, list and filter available images by date and time, download images at multiple resolutions, and assemble downloaded frames into animated GIFs or MP4 videos. Integrates with the 'dataRetrieval' package (<https://doi-usgs.github.io/dataRetrieval/>) to enrich camera records with watershed metadata (drainage area, hydrologic unit code, state, county) and to retrieve co-located streamflow, stage, or water-quality time series — enabling direct comparison of visual stream conditions with measured observations. API authentication uses the 'API_USGS_PAT' environment variable shared with 'dataRetrieval', so a single key covers both packages.

License AGPL (>= 3)

URL https://connorb.github.io/flowcam/,

https://github.com/ConnorB/flowcam,

https://connorb.r-universe.dev/flowcam

X-schema.org-keywords usgs, stream gage, streamgage, hydrology, water data, river cameras, NIMS, NWIS, timelapse, flood monitoring, camera images, rstats

BugReports https://github.com/ConnorB/flowcam/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports cli (>= 3.0.0), httr2 (>= 1.0.0), lifecycle, rlang (>= 1.0.0), tibble (>= 3.0.0)

Suggests av, dataRetrieval (>= 2.7.17), gifski, sf, httptest2, jpeg, knitr, png, quarto, testthat (>= 3.0.0), withr

VignetteBuilder quarto

Config/testthat/edition 3

Config/roxygen2/version 8.0.0.9000

RoxygenNote 8.0.0.9000

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-08-11 15:48:16 UTC

RemoteUrl https://github.com/ConnorB/flowcam

RemoteRef HEAD

RemoteSha fcf70e185a0ead41e35246ffc53ba7613d9c655f

Contents

build_image_url 2
download_images 3
find_cameras 5
find_gage_cameras 6
get_flow_statistics 7
get_network_cameras 9
get_site_data_availability 10
get_site_field_measurements 11
get_site_streamflow 13
get_timelapse_url 15
list_images 16
make_gif 17
make_video 19
set_usgs_api_key 21

Index **22**

`build_image_url` *Build full image URLs from camera metadata and filenames*

Description

Combines a camera's base directory with one or more filenames returned by `list_images()` to produce download-ready image URLs.

Usage

```
build_image_url(camera_row, filename, size = c("small", "overlay", "thumb"))
```

Arguments

`camera_row`	A single-row tibble (or named list) from `find_cameras()`. Must contain the directory column corresponding to `size`.
`filename`	Character vector of one or more filenames from `list_images()`.
`size`	Image size. One of: • `"small"` (default): ~720 px wide (`smallDir`) • `"overlay"`: full-size overlay image (`overlayDir`) • `"thumb"`: thumbnail ~200 px tall (`thumbDir`)

Value

A character vector of full image URLs, one per element of `filename`.

Examples

```
## Not run:
cam   <- find_cameras(cam_id = "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire")
imgs  <- list_images(cam$camId[[1]], limit = 5)
urls  <- build_image_url(cam, imgs$filename, size = "small")

## End(Not run)
```

`download_images` *Download camera images to disk*

Description

Lists available images for a camera and downloads them to a local directory. Images are fetched directly from USGS S3 storage; no API key is required for the downloads themselves.

Usage

```
download_images(
  cam_id = NULL,
  dest_dir,
  size = "small",
  limit = 1000L,
  time = NULL,
  overwrite = FALSE,
  site_id = NULL
)
```

Arguments

cam_id	Character. Camera identifier. Use find_cameras() to look up valid IDs. Cannot be used together with site_id.
dest_dir	Character. Path to an existing local directory where images will be saved.
size	Image size passed to build_image_url(). One of "small" (default), "overlay", or "thumb".
limit	Integer between 1 and 50000. Page size for the internal list_images() call. Default is 1000. All images in the requested time range are downloaded via automatic pagination.
time	POSIXct, Date, or character vector of length 1 or 2 used to filter images by capture time. A length-1 value is treated as a start (on or after). A length-2 vector sets the start and end; use NA for an open bound. See list_images() for full details.
overwrite	Logical. If FALSE (default), skip files that already exist in dest_dir (resume behaviour).
site_id	Character. NWIS site number (e.g. "05366800" or "USGS-05366800"). Cannot be used together with cam_id. If the site has more than one camera, supply cam_id directly.

Details

Identify the camera with either cam_id or site_id — provide exactly one. site_id accepts both bare NWIS numbers ("05366800") and the USGS- prefixed form ("USGS-05366800"). When a site has multiple cameras, specify the desired camera via cam_id instead.

Value

A character vector of local file paths, invisibly. Failed downloads are represented as NA.

Examples

```
## Not run:
# By cam_id
paths <- download_images(
  "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
  dest_dir = tempdir(),
  limit    = 5
)

# By USGS site ID
paths <- download_images(
  site_id  = "USGS-05366800",
  dest_dir = tempdir(),
  limit    = 5
)

## End(Not run)
```

`find_cameras` *Query NIMS cameras*

Description

Returns a tibble of camera metadata from the USGS National Imagery Management System (NIMS). With no arguments, all cameras are returned. Filter by a NWIS site ID or a specific camera ID.

Usage

```
find_cameras(site_id = NULL, cam_id = NULL, return_fields = NULL)
```

Arguments

`site_id`	Character. NWIS site number (e.g. `"05366800"`). Cannot be used together with `cam_id`.
`cam_id`	Character. Camera identifier (e.g. `"WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire"`). Cannot be used together with `site_id`.
`return_fields`	Character vector of camera fields to include in the response (e.g. `c("camName", "newestImageDT")`). `camId` is always returned regardless of this setting. When `NULL` (the default), all fields are returned.

Value

A tibble with one row per camera. Columns depend on `return_fields`. `lat` and `lng` are returned as numeric. Datetime columns are POSIXct (UTC). The `ingest` column is a list-column containing each camera's ingestion configuration.

Examples

```
## Not run:
# All cameras
find_cameras()

# Cameras at a specific gage
find_cameras(site_id = "05366800")

# A specific camera
find_cameras(cam_id = "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire")

# Only selected fields
find_cameras(return_fields = c("camName", "newestImageDT"))

## End(Not run)
```

`find_gage_cameras` *Query NIMS cameras and enrich with NWIS site metadata*

Description

Calls `find_cameras()` for the given NWIS site ID, then joins the result with site metadata from `dataRetrieval::read_waterdata_monitoring_location()`. Optionally appends a `data_types` list-column with the available time series at the site via `get_site_data_availability()`. Requires the `dataRetrieval` package.

Usage

```
find_gage_cameras(site_id, include_availability = TRUE)
```

Arguments

`site_id`	Character. A single 8-to-15-digit NWIS site number (e.g. `"05366800"`).
`include_availability`	Logical. When `TRUE` (default), a `data_types` list-column is appended containing a tibble of available time series for the site (from `get_site_data_availability()`). Set to `FALSE` to skip this extra API call and return only the camera and site metadata.

Value

A tibble with all camera columns from `find_cameras()` plus these site metadata columns when available: `monitoring_location_name`, `state_name`, `county_name`, `hydrologic_unit_code`, `drainage_area` (total drainage area in square miles), and `altitude` (elevation in feet above the stated vertical datum). When `include_availability = TRUE`, a `data_types` list-column is also included; each element is a tibble with columns from `get_site_data_availability()`.

Examples

```
## Not run:
# With available time series (default)
find_gage_cameras("05366800")

# Site metadata only, no data availability call
find_gage_cameras("05366800", include_availability = FALSE)

## End(Not run)
```

`get_flow_statistics` *Retrieve historical flow statistics for a USGS gage*

Description

Provides historical context for interpreting camera imagery — "is this flow high or low?" — by returning day-of-year percentile curves or period-of-record summaries.

Usage

```
get_flow_statistics(
  site_id = NULL,
  cam_id = NULL,
  parameter_code = "00060",
  type = c("daily_normals", "period_summary"),
  computation = "percentile",
  time = NULL
)
```

Arguments

`site_id`	Character. A single NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Exactly one of `site_id` or `cam_id` must be provided.
`cam_id`	Character. A camera identifier. If supplied, the function resolves the associated NWIS site ID via `find_cameras()`. Exactly one of `site_id` or `cam_id` must be provided.
`parameter_code`	Character. One or more five-digit USGS parameter codes. Default `"00060"` (discharge, ft^3/s).
`type`	Character. `"daily_normals"` (default) returns day-of-year and month-of-year percentile/statistic curves, suitable for overlaying on a time series plot. `"period_summary"` returns calendar-month, calendar-year, and water-year summaries.
`computation`	Character vector. One or more of `"percentile"` (default), `"arithmetic_mean"`, `"minimum"`, `"maximum"`, `"median"`. When `NULL` or `NA`, all computation types are returned.
`time`	For `type = "period_summary"` only. A POSIXct, Date, or character vector of length 1 or 2 (start, end) used to restrict the date range of the summary. Follows the same semantics as the `time` argument in `get_site_streamflow()`. `NULL` (default) returns the full period of record. Ignored when `type = "daily_normals"`.

Details

Wraps `dataRetrieval::read_waterdata_stats_por()` (`type = "daily_normals"`) or `dataRetrieval::read_wa` (`type = "period_summary"`).

Value

A tibble. All types include columns:

`site_id` Bare NWIS site number.

`parameter_code` Five-digit parameter code.

`unit_of_measure` Units string.

`computation` Statistical method (e.g. `"percentile"`).

`percentile` Integer percentile (e.g. `50L`); `NA` for non-percentile computations.

`value` Numeric statistic value.

`sample_count` Number of observations used to compute the statistic.

`type = "daily_normals"` additionally includes:

`month_day` Character MM-DD representing the day or month of year.

`type = "period_summary"` additionally includes:

`interval_type` One of `"calendar_month"`, `"calendar_year"`, or `"water_year"`.

`start_date` Date. Start of the summary interval.

`end_date` Date. End of the summary interval.

Returns a zero-row tibble (with correct column types) when no data are found. Requires the dataRetrieval package.

Examples

```
## Not run:
# Day-of-year discharge percentile curves (the 10/25/50/75/90th percentiles)
get_flow_statistics("05366800")

# Annual and monthly discharge summaries
get_flow_statistics("05366800", type = "period_summary")

# Restrict period_summary to a specific date range
get_flow_statistics(
  "05366800",
  type = "period_summary",
  time = c("2010-01-01", "2020-12-31")
)

# Multiple computation types
get_flow_statistics(
  "05366800",
  computation = c("minimum", "median", "maximum")
)

# Overlay on a streamflow time series
flow  <- get_site_streamflow("05366800", time = "P1Y")
stats <- get_flow_statistics("05366800")

## End(Not run)
```

`get_network_cameras` *Find cameras on the same stream network*

Description

Uses the USGS Network Linked Data Index (NLDI) to navigate upstream and/or downstream from a gage site, then returns all flowcam cameras found along the network within a specified distance.

Usage

```
get_network_cameras(
  site_id = NULL,
  cam_id = NULL,
  direction = c("upstream", "downstream", "both"),
  distance_km = 100
)
```

Arguments

`site_id`	Character. A single NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Exactly one of `site_id` or `cam_id` must be provided.
`cam_id`	Character. A camera identifier. If supplied, the function resolves the associated NWIS site ID via `find_cameras()`. Exactly one of `site_id` or `cam_id` must be provided.
`direction`	Character. Navigation direction(s): `"upstream"` (upstream tributary), `"downstream"` (downstream mainstem), or `"both"`. Default `"upstream"`.
`distance_km`	Numeric. How far (in kilometres) to navigate along the stream network from the origin site. Default `100`.

Details

Useful for anticipating flood wave arrival, comparing conditions across a watershed, or discovering monitoring locations on the same river system.

Wraps `dataRetrieval::findNLDI()`.

Value

A tibble with one row per camera (including the origin site) and columns:

`site_id` Bare NWIS site number.

`cam_id` Camera identifier.

`cam_name` Human-readable camera name.

`direction` `"origin"`, `"upstream"`, or `"downstream"`.

`lat` Latitude (decimal degrees, WGS84).

`lng` Longitude (decimal degrees, WGS84).

Returns a zero-row tibble when no networked cameras are found (including the origin site if it has no camera). Requires the dataRetrieval package.

Examples

```
## Not run:
# Cameras upstream of a site (within 100 km)
get_network_cameras("05366800")

# Cameras both upstream and downstream (150 km)
get_network_cameras("05366800", direction = "both", distance_km = 150)

# Via camera ID
get_network_cameras(
  cam_id = "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
  direction = "downstream"
)

## End(Not run)
```

---

get_site_data_availability

Discover available time series at a USGS gage

Description

Returns metadata about every time series recorded at a site — parameter codes, units, and period of record — so you know exactly which codes to pass to `get_site_streamflow()`.

Usage

```
get_site_data_availability(
  site_id = NULL,
  cam_id = NULL,
  parameter_code = NA_character_
)
```

Arguments

`site_id`	Character. A single NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Exactly one of `site_id` or `cam_id` must be provided.
`cam_id`	Character. A camera identifier. If supplied, the function resolves the associated NWIS site ID via `find_cameras()`. Exactly one of `site_id` or `cam_id` must be provided.
`parameter_code`	Character. One or more five-digit USGS parameter codes to filter results (e.g. `"00060"` for discharge). Default `NA` returns all available parameters.

Details

Wraps dataRetrieval::read_waterdata_ts_meta().

Value

A tibble with columns:

`site_id` Bare NWIS site number (without `USGS-` prefix).

`parameter_code` Five-digit parameter code.

`parameter_name` Human-readable parameter name.

`unit_of_measure` Units string (e.g. `"ft3/s"`).

`begin_utc` POSIXct (UTC). Earliest observation in this time series.

`end_utc` POSIXct (UTC). Most recent observation.

`statistic_id` Five-digit statistic code (e.g. `"00003"` for mean).

`time_series_id` Unique identifier for this time series.

Rows are sorted by `parameter_code`. Returns a zero-row tibble when no time series are found. Requires the dataRetrieval package.

Examples

```
## Not run:
# All parameters at a site
get_site_data_availability("05366800")

# Discharge only
get_site_data_availability("05366800", parameter_code = "00060")

# Via camera ID
get_site_data_availability(
  cam_id = "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire"
)

# Chain with get_site_streamflow():
avail <- get_site_data_availability("05366800")
flow  <- get_site_streamflow("05366800",
                             parameter_code = avail$parameter_code[1])

## End(Not run)
```

---

get_site_field_measurements

Retrieve manual field measurements for a USGS gage

Description

Returns the record of in-person discharge measurements made by USGS hydrographers at a gage site. These manual measurements are the "ground truth" used to build stage-discharge rating curves, and they provide direct visual correlates to camera imagery taken at the same location and time.

Usage

```
get_site_field_measurements(
  site_id = NULL,
  cam_id = NULL,
  time = NULL,
  parameter_code = "00060"
)
```

Arguments

`site_id`	Character. A single NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Exactly one of `site_id` or `cam_id` must be provided.
`cam_id`	Character. A camera identifier. If supplied, the function resolves the associated NWIS site ID via `find_cameras()`. Exactly one of `site_id` or `cam_id` must be provided.
`time`	POSIXct, Date, or character vector of length 1 or 2 specifying the time window. Follows the same semantics as `get_site_streamflow()`. `NULL` (default) returns all available measurements.
`parameter_code`	Character. One or more five-digit USGS parameter codes. Default `"00060"` (discharge, ft^3/s). `NULL` returns measurements for all parameters.

Details

Wraps `dataRetrieval::read_waterdata_field_measurements()`.

Value

A tibble with columns:

`site_id` Bare NWIS site number.

`datetime` POSIXct (UTC) of the measurement.

`value` Numeric measured value.

`unit` Units of measure (e.g. `"ft3/s"`).

`parameter_code` Five-digit parameter code.

`approval_status` `"Approved"` or `"Provisional"`.

`measurement_rated` Quality rating of the measurement: `"Excellent"` (2% error), `"Good"` (5%), `"Fair"` (8%), or `"Poor"` (>8%).

`control_condition` Description of channel control conditions during the measurement.

Returns a zero-row tibble (with correct column types) when no measurements are found. Compatible with `get_site_streamflow()` output via `dplyr::bind_rows()` on the shared columns. Requires the dataRetrieval package.

Examples

```
## Not run:
# All discharge measurements at a site
get_site_field_measurements("05366800")

# Measurements during a specific period
get_site_field_measurements(
  "05366800",
  time = c("2020-01-01", "2024-12-31")
)

# Via camera ID
get_site_field_measurements(
  cam_id = "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire"
)

# Compare field measurements against continuous sensor record
field <- get_site_field_measurements("05366800", time = "P2Y")
sensor <- get_site_streamflow("05366800", time = "P2Y")

## End(Not run)
```

---

get_site_streamflow *Retrieve streamflow data for a USGS gage*

Description

Wraps `dataRetrieval::read_waterdata_continuous()` (instantaneous values) or `dataRetrieval::read_waterd` (daily mean values) for the USGS gage associated with a flowcam camera. Returns a clean tibble aligned with flowcam conventions.

Usage

```
get_site_streamflow(
  site_id = NULL,
  cam_id = NULL,
  time = NULL,
  parameter_code = "00060",
  type = c("continuous", "daily"),
  water_year = FALSE
)
```

Arguments

site_id	Character. A single NWIS site number (e.g. "05366800" or "USGS-05366800"). Exactly one of site_id or cam_id must be provided.
cam_id	Character. A camera identifier (e.g. "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire"). If supplied, the function resolves the associated NWIS site ID via find_cameras(). Exactly one of site_id or cam_id must be provided.
time	POSIXct, Date, or character vector of length 1 or 2. Follows the same semantics as the time argument in list_images(): length-1 is treated as a start bound; length-2 sets start and end, with NA for an open bound. ISO 8601 duration strings such as "P7D" (last 7 days) or "P1Y" (last year) are also accepted and forwarded to the API unchanged. When NULL, the API default applies (approximately the last year of data for continuous; all available data for daily).
parameter_code	Character. One or more five-digit USGS parameter codes. Default "00060" is discharge in cubic feet per second. Other common codes: "00010" (water temperature, C), "00065" (gage height, ft), "00095" (specific conductance, µS/cm). When multiple codes are supplied, all parameters are returned in a single tibble distinguished by the parameter_code column.
type	Character. "continuous" (default) returns instantaneous (unit-value) observations; "daily" returns daily mean values.
water_year	Logical. When TRUE, appends a water_year integer column (Oct 1 – Sep 30) to the result using dataRetrieval::calcWaterYear(). Default FALSE.

Value

A tibble with columns:

site_id Bare NWIS site number (without USGS- prefix).

datetime POSIXct (UTC) for type = "continuous"; Date for type = "daily".

value Numeric observation value.

unit Units of measure (e.g. "ft3/s").

parameter_code Five-digit parameter code.

approval_status "Approved" or "Provisional".

water_year Integer water year (only present when water_year = TRUE).

Returns a zero-row tibble (with correct column types) when no data are found. Requires the dataRetrieval package.

Examples

```
## Not run:
# Instantaneous discharge for the last 7 days
get_site_streamflow("05366800", time = "P7D")
```

```
# Daily discharge for a calendar year
get_site_streamflow(
  "05366800",
  time = c("2024-01-01", "2024-12-31"),
  type = "daily"
)

# Water temperature from a camera ID
get_site_streamflow(
  cam_id = "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
  parameter_code = "00010"
)

# Multiple parameters at once
get_site_streamflow("05366800", parameter_code = c("00060", "00065"))

# Include water year column
get_site_streamflow("05366800", time = "P1Y", water_year = TRUE)

# Pair with image timestamps
images <- list_images("05366800", time = c("2024-10-01", "2024-10-07"),
                      raw_item = TRUE)
flow   <- get_site_streamflow("05366800",
                              time = c("2024-10-01", "2024-10-07"))

## End(Not run)
```

`get_timelapse_url` *Get the timelapse video URL for a camera*

Description

Queries the camera's metadata from NIMS and constructs the URL to its timelapse video file (720p MP4). Issues a warning if timelapse is not enabled for the camera.

Usage

```
get_timelapse_url(cam_id)
```

Arguments

`cam_id`	Character. Camera identifier.

Value

A single character string with the full timelapse video URL, or `NULL` (invisibly) if timelapse is not enabled for the camera.

Examples

```
## Not run:
get_timelapse_url("WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire")

## End(Not run)
```

---

list_images *List image filenames for a NIMS camera*

Description

Returns a tibble of image filenames (or raw image metadata) for the specified camera. Use the filenames with `build_image_url()` to construct full image URLs.

Usage

```
list_images(
  cam_id = NULL,
  limit = 1000L,
  recent = TRUE,
  time = NULL,
  raw_item = FALSE,
  site_id = NULL
)
```

Arguments

`cam_id`	Character. Camera identifier. Use `find_cameras()` to look up valid IDs. Cannot be used together with `site_id`.
`limit`	Integer between 1 and 50000. Number of records fetched per API request (page size). Default is `1000`. All matching images are returned via automatic pagination regardless of this value.
`recent`	Logical. If `TRUE` (default), return the most recent images first. If `FALSE`, return the oldest images first.
`time`	POSIXct, Date, or character vector of length 1 or 2 used to filter images by capture time. A length-1 value is treated as a start (on or after). A length-2 vector sets the start and end; use `NA` for an open bound (`c("2025-06-01", NA)` = on or after; `c(NA, "2025-06-02")` = on or before). Character strings are passed through unchanged; the API accepts ISO 8601 (`"2025-12-31T00:00:00"`) and NIMS format (`"2025-12-31T00-00-00Z"`).
`raw_item`	Logical. If `TRUE`, return a tibble with columns `camId`, `filename`, `timestamp`, and `fs` (file size in kb). If `FALSE` (default), return a single-column tibble of filenames.
`site_id`	Character. NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Cannot be used together with `cam_id`. If the site has more than one camera, supply `cam_id` directly.

Details

Identify the camera with either `cam_id` or `site_id` — provide exactly one. `site_id` accepts both bare NWIS numbers (`"05366800"`) and the `USGS-` prefixed form (`"USGS-05366800"`). When a site has multiple cameras, specify the desired camera via `cam_id` instead.

Value

A tibble. When `raw_item = FALSE`, one column: `filename`. When `raw_item = TRUE`, four columns: `camId`, `filename`, `timestamp`, `fs`.

Examples

```
## Not run:
# 10 most recent images by cam_id
list_images("WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire", limit = 10)

# By USGS site ID (bare or prefixed)
list_images(site_id = "05366800", limit = 10)
list_images(site_id = "USGS-05366800", limit = 10)

# Images in a date window
list_images(
  "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
  time = as.POSIXct(c("2025-06-01", "2025-06-02"), tz = "UTC")
)

# Open-ended: on or after June 1
list_images(
  "WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
  time = c("2025-06-01", NA)
)

# With metadata
list_images("WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
            limit = 5, raw_item = TRUE)

## End(Not run)
```

`make_gif` *Assemble camera images into an animated GIF*

Description

Downloads images for a camera over a specified time range and assembles them into an animated GIF using the `gifski` package. Provide either `cam_id` or `site_id` to identify the camera, and use `time` to restrict the range.

Usage

```
make_gif(
  cam_id = NULL,
  site_id = NULL,
  time = NULL,
  output = NULL,
  fps = 2,
  size = "small",
  limit = 1000L,
  dir = NULL,
  one_per_day = FALSE
)
```

Arguments

`cam_id`	Character. Camera identifier. Cannot be used with `site_id`.
`site_id`	Character. NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Cannot be used with `cam_id`.
`time`	POSIXct, Date, or character vector of length 1 or 2. Same semantics as `download_images()`. When `dir` is supplied, timestamps are parsed from the filenames (NIMS format: `<camId>___<timestamp>.jpg`).
`output`	Character. File path for the output GIF. Defaults to `"<cam_id>.gif"` (or `"<site_id>.gif"`, or the directory basename) in the working directory.
`fps`	Positive number. Frames per second. Any positive value is accepted. Default is `2`.
`size`	Image size passed to `download_images()`. One of `"small"` (default), `"overlay"`, or `"thumb"`. Ignored when `dir` is supplied.
`limit`	Integer. Page size for the internal `list_images()` call. Default is `1000`. Ignored when `dir` is supplied.
`dir`	Character. Path to a local directory of already-downloaded images. When supplied, downloads are skipped and JPEG/PNG files are read from this directory. `cam_id`/`site_id` and `time` still apply as filters.
`one_per_day`	Logical. If `TRUE`, reduce frames to one per calendar day by selecting the image whose capture time is closest to noon in the camera's local timezone (from the `tz` field of `find_cameras()`). Default is `FALSE`.

Details

When `dir` is supplied, images are read from that local directory instead of being downloaded. You can still pass `cam_id`/`site_id` to select only files belonging to a particular camera (matched by filename prefix) and `time` to filter by timestamp embedded in the filename — useful when a directory contains images from multiple cameras or a wider date range than needed.

JPEG frames are converted to PNG in a temporary directory before encoding because `gifski` only accepts PNG input. The `jpeg` and `png` packages are required when any frames are JPEG.

Value

The output file path, invisibly.

Examples

```
## Not run:
# Download and assemble images for a date range
make_gif("WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
         time = c("2025-06-01", "2025-06-02"), output = "chippewa.gif")

# One frame per day from a local directory
make_gif(cam_id = "NM_Pecos_Web_Camera_near_Roswell",
         time        = c("2023-08-01", "2023-08-31"),
         dir         = "~/Downloads/Pecos",
         one_per_day = TRUE,
         output      = "pecos_august.gif")

## End(Not run)
```

---

make_video *Assemble camera images into an MP4 video*

Description

Downloads images for a camera over a specified time range and encodes them into an MP4 video using the `av` package. Provide either `cam_id` or `site_id` to identify the camera, and use `time` to restrict the range.

Usage

```
make_video(
  cam_id = NULL,
  site_id = NULL,
  time = NULL,
  output = NULL,
  fps = 2,
  size = "small",
  limit = 1000L,
  dir = NULL,
  one_per_day = FALSE
)
```

Arguments

`cam_id`	Character. Camera identifier. Cannot be used with `site_id`.
`site_id`	Character. NWIS site number (e.g. `"05366800"` or `"USGS-05366800"`). Cannot be used with `cam_id`.

time	POSIXct, Date, or character vector of length 1 or 2. Same semantics as download_images(). When dir is supplied, timestamps are parsed from the filenames (NIMS format: <camId>___<timestamp>.jpg).
output	Character. File path for the output MP4. Defaults to "<cam_id>.mp4" (or "<site_id>.mp4", or the directory basename) in the working directory.
fps	Positive number. Frames per second. Any positive value is accepted. Default is 2.
size	Image size passed to download_images(). One of "small" (default), "overlay", or "thumb". Ignored when dir is supplied.
limit	Integer. Page size for the internal list_images() call. Default is 1000. Ignored when dir is supplied.
dir	Character. Path to a local directory of already-downloaded images. When supplied, downloads are skipped and JPEG/PNG files are read from this directory. cam_id/site_id and time still apply as filters.
one_per_day	Logical. If TRUE, reduce frames to one per calendar day by selecting the image whose capture time is closest to noon in the camera's local timezone (from the tz field of find_cameras()). Default is FALSE.

Details

When dir is supplied, images are read from that local directory instead of being downloaded. You can still pass cam_id/site_id to select only files belonging to a particular camera (matched by filename prefix) and time to filter by timestamp embedded in the filename — useful when a directory contains images from multiple cameras or a wider date range than needed.

Value

The output file path, invisibly.

Examples

```
## Not run:
# Download and assemble images for a date range
make_video("WI_Chippewa_River_at_Grand_Ave_at_Eau_Claire",
           time = c("2025-06-01", "2025-06-02"), output = "chippewa.mp4")

# One frame per day from a local directory
make_video(cam_id = "NM_Pecos_Web_Camera_near_Roswell",
           time        = c("2023-08-01", "2023-08-31"),
           dir         = "~/Downloads/Pecos",
           one_per_day = TRUE,
           output      = "pecos_august.mp4")

## End(Not run)
```

`set_usgs_api_key` *Store your USGS API key in .Renviron*

Description

Writes `API_USGS_PAT` to the user's `~/.Renviron` file and applies it immediately in the current session. The same variable is used by package dataRetrieval, so one key can be shared across both packages.

Usage

```
set_usgs_api_key(key)
```

Arguments

`key`	A single non-empty character string containing your API key.

Details

Register for a free API key at https://api.waterdata.usgs.gov/signup/.

Value

`key`, invisibly.

Examples

```
## Not run:
set_usgs_api_key("my_api_key_here")

## End(Not run)
```

Index

build_image_url, 2
build_image_url(), *4*, *16*

dataRetrieval, *8*, *10*, *11*, *13*, *14*, *21*
dataRetrieval::calcWaterYear(), *14*
dataRetrieval::findNLDI(), *9*
dataRetrieval::read_waterdata_continuous(), *13*
dataRetrieval::read_waterdata_daily(), *13*
dataRetrieval::read_waterdata_field_measurements(), *12*
dataRetrieval::read_waterdata_stats_daterange(), *7*
dataRetrieval::read_waterdata_stats_por(), *7*
dataRetrieval::read_waterdata_ts_meta(), *11*
download_images, 3
download_images(), *18*, *20*

find_cameras, 5
find_cameras(), *3*, *4*, *6*, *7*, *9*, *10*, *12*, *14*, *16*, *18*, *20*
find_gage_cameras, 6

get_flow_statistics, 7
get_network_cameras, 9
get_site_data_availability, 10
get_site_data_availability(), *6*
get_site_field_measurements, 11
get_site_streamflow, 13
get_site_streamflow(), *7*, *10*, *12*, *13*
get_timelapse_url, 15

list_images, 16
list_images(), *2–4*, *14*, *18*, *20*

make_gif, 17
make_video, 19

set_usgs_api_key, 21