

# Package ‘weathR’

May 8, 2026

**Title** Interact with the U.S. National Weather Service API

**Version** 0.1.0

**Description** Enables interaction with the National Weather Service application programming web-interface for fetching of real-time and forecast meteorological data. Users can provide latitude and longitude, Automated Surface Observing System identifier, or Automated Weather Observing System identifier to fetch recent weather observations and recent forecasts for the given location or station. Additionally, auxiliary functions exist to identify stations nearest to a point, convert wind direction from character to degrees, and fetch active warnings. Results are returned as simple feature objects whenever possible.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** dplyr, httr2, janitor, jsonlite, lubridate, lutz, magrittr, purrr, sf, tibble

**Suggests** gt, snakecase, tmap

**URL** <https://github.com/JeffreyFowler/weathR>

**BugReports** <https://github.com/JeffreyFowler/weathR/issues>

**NeedsCompilation** no

**Author** Jeffrey Fowler [aut, cre, cph] (ORCID:  
<<https://orcid.org/0009-0003-9448-0980>>)

**Maintainer** Jeffrey Fowler <JeffreyF6120@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-04-02 09:00:02 UTC

## Contents

..point_forecast . . . . .	2
.point_data . . . . .	3
.point_forecast . . . . .	3
.station_data . . . . .	4

alerts . . . . .	4
dir_as_char . . . . .	5
dir_as_integer . . . . .	5
point_data . . . . .	6
point_forecast . . . . .	6
point_station . . . . .	7
point_today . . . . .	8
point_tomorrow . . . . .	8
safe_collapse . . . . .	9
stations_near . . . . .	10
station_coords . . . . .	10
station_forecast . . . . .	11
station_obs . . . . .	11
station_point . . . . .	12
station_today . . . . .	13
station_tomorrow . . . . .	13
station_tz . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

..point_forecast	<i>Point Forecast Data and Local Timezone</i>
------------------	-----------------------------------------------

---

## Description

Point Forecast Data and Local Timezone

## Usage

```
..point_forecast(lat, lon, timezone = -1, dir_numeric = FALSE)
```

## Arguments

lat	Latitude.
lon	Longitude.
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.

## Value

A list containing point forecast sf and the timezone.

## Examples

```
..point_forecast(33, -80)
```

---

.point\_data                    *Get the JSON Data for a Point*

---

**Description**

Get the JSON Data for a Point

**Usage**

```
.point_data(lat, lon)
```

**Arguments**

lat	Latitude of the point to fetch data for.
lon	Longitude of the point to fetch data for.

**Value**

A nested list containing NWS point data.

**Examples**

```
.point_data(lat = 40.71427000, lon = -74.00597000) %>% data.frame()
```

---

.point\_forecast                *Raw JSON Point Forecast Data*

---

**Description**

Raw JSON Point Forecast Data

**Usage**

```
.point_forecast(lat, lon)
```

**Arguments**

lat	Latitude.
lon	Longitude.

**Value**

Returns the json data as a nested list.

**Examples**

```
.point_forecast(33, -80)
```

---

<code>.station_data</code>	<i>Raw JSON Station Metadata</i>
----------------------------	----------------------------------

---

**Description**

Raw JSON Station Metadata

**Usage**

```
.station_data(station_id)
```

**Arguments**

`station_id` The station identifier (ex: KDEN, KBOS, KNYC, etc).

**Value**

Station data provided for the National Weather Service.

**Examples**

```
.station_data("KDEN")
```

---

<code>alerts</code>	<i>National Weather Service Alerts</i>
---------------------	----------------------------------------

---

**Description**

National Weather Service Alerts

**Usage**

```
alerts()
```

**Value**

Dataframe containing various columns identifying and describing alerts.

**Examples**

```
alerts()
```

---

dir_as_char	<i>Convert Wind Direction from Numeric to Character</i>
-------------	---------------------------------------------------------

---

**Description**

Convert Wind Direction from Numeric to Character

**Usage**

```
dir_as_char(direction)
```

**Arguments**

direction      Numeric degrees clockwise from north, (0 = N, 45 = NE, etc).

**Value**

Character direction (N, NNE, NE, etc).

**Examples**

```
dir_as_char(330)
```

---

dir_as_integer	<i>Convert Wind Direction from a Character to an Integer</i>
----------------	--------------------------------------------------------------

---

**Description**

Convert Wind Direction from a Character to an Integer

**Usage**

```
dir_as_integer(direction)
```

**Arguments**

direction      A direction, as a string (N, NNE, NE, ENE, E, ESE, etc).

**Value**

An integer representing degrees clockwise from north.

**Examples**

```
dir_as_integer("NNW")
```

---

point_data	<i>Get NWS Metadata for a Point</i>
------------	-------------------------------------

---

**Description**

Get NWS Metadata for a Point

**Usage**

```
point_data(lat, lon)
```

**Arguments**

lat	Latitude.
lon	Longitude.

**Value**

A simple features point object with NWS metadata.

**Examples**

```
point_data(lat = 40.71427000, lon = -74.00597000)
```

---

point_forecast	<i>Point Forecast Data</i>
----------------	----------------------------

---

**Description**

Point Forecast Data

**Usage**

```
point_forecast(lat, lon, timezone = -1, dir_numeric = FALSE)
```

**Arguments**

lat	Latitude.
lon	Longitude.
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.

**Value**

Simple features object with forecast meteorological values.

**Examples**

```
point_forecast(lat = 40.71427000, lon = -74.00597000, dir_numeric = TRUE)
```

---

point_station	<i>Find Nearest ASOS/AWOS Station</i>
---------------	---------------------------------------

---

**Description**

Find Nearest ASOS/AWOS Station

**Usage**

```
point_station(lat, lon)
```

**Arguments**

lat	Latitude.
lon	Longitude.

**Value**

A string corresponding to an ASOS or AWOS station.

**Examples**

```
# Gets the observation data as an sf associated with a point  
point_station(lat = 42, lon = -80) %>% station_obs() %>% data.frame()
```

---

point_today	<i>Weather Observed Today at a Point</i>
-------------	------------------------------------------

---

### Description

Weather Observed Today at a Point

### Usage

```
point_today(lat, lon, timezone = -1, dir_numeric = FALSE)
```

### Arguments

lat	Latitude.
lon	Longitude.
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.

### Value

Simple features point corresponding to the given station identifier with recent meteorological forecast values for today.

### Examples

```
# Produce a GT summary of the weather thus far for a given lat/lon

point_today(lat = 33, lon = -80) %>%
  as.data.frame() %>%
  dplyr::mutate(time = as.POSIXct(time) %>% format("%H:%M")) %>%
  dplyr::select(time, temp, dewpoint, humidity, wind_speed)
```

---

point_tomorrow	<i>Weather Forecast for Tomorrow at a Point</i>
----------------	-------------------------------------------------

---

### Description

Weather Forecast for Tomorrow at a Point

### Usage

```
point_tomorrow(lat, lon, timezone = -1, dir_numeric = FALSE, short = TRUE)
```

**Arguments**

lat	Latitude.
lon	Longitude.
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.
short	TRUE for only tomorrow, FALSE for today and tomorrow; defaults to TRUE.

**Value**

Simple features object with forecast meteorological values.

**Examples**

```
point_tomorrow(lat = 33, lon = -80)
```

---

safe_collapse	<i>Safe Collapse</i>
---------------	----------------------

---

**Description**

Safe Collapse

**Usage**

```
safe_collapse(x)
```

**Arguments**

x	A list to collapse.
---	---------------------

**Value**

A comma delimited version of your input list, or NA when the input list is blank.

**Note**

This is a helper function that is used to collapse a list into a string, and is used in building the active warnings dataset.

**Examples**

```
safe_collapse(c("This is one", "And this is another one"))
```

---

stations_near	<i>Find All Stations in a Point's Forecast Zone</i>
---------------	-----------------------------------------------------

---

**Description**

Find All Stations in a Point's Forecast Zone

**Usage**

```
stations_near(lat, lon)
```

**Arguments**

lat	Latitude.
lon	Longitude.

**Value**

An sf object with station identifiers, geometry as coordinates, and their euclidian distance (in miles) to the station provided.

**Examples**

```
# Plot the a station with given points and the nearby stations in a tmap  
stations_near(lat = 33, lon = -80)
```

---

station_coords	<i>Station Coordinates</i>
----------------	----------------------------

---

**Description**

Station Coordinates

**Usage**

```
station_coords(station_id)
```

**Arguments**

station_id	The station identifier (ex: KDEN, KBOS, KNYC, etc).
------------	-----------------------------------------------------

**Value**

Named list with latitude and longitude like: `c("lat" = x, "lon" = y)`.

**Examples**

```
station_coords("KBOS")
```

---

station_forecast	<i>Station Forecast</i>
------------------	-------------------------

---

**Description**

Station Forecast

**Usage**

```
station_forecast(station_id, timezone = -1, dir_numeric = FALSE)
```

**Arguments**

station_id	Station identifier (ex: KDEN, KBOS, KNYC, etc).
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.

**Value**

Simple features point corresponding to the given station identifier with recent meteorological forecast values.

**Examples**

```
station_forecast("KBOS") %>% data.frame() %>% dplyr::select(-geometry)
```

---

station_obs	<i>Station Observations</i>
-------------	-----------------------------

---

**Description**

Station Observations

**Usage**

```
station_obs(station_id, timezone = -1, dir_numeric = FALSE)
```

**Arguments**

station_id	The station identifier (ex: KDEN, KBOS, KNYC, etc).
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.

**Value**

Simple features corresponding to the given station identifier with recent meteorological observations.

**Examples**

```
station_obs("KBOS") %>% data.frame() %>%
  dplyr::select(-geometry) %>%
  dplyr::filter(temp == max(.$temp))
```

---

station_point	<i>Station Coordinates as a Point</i>
---------------	---------------------------------------

---

**Description**

Station Coordinates as a Point

**Usage**

```
station_point(station_id)
```

**Arguments**

station_id	The station identifier (ex: KDEN, KBOS, KNYC, etc).
------------	-----------------------------------------------------

**Value**

Simple features point corresponding to the given station identifier.

**Examples**

```
station_point("KDEN")
```

---

station_today	<i>Weather Observed Today at a Station Identifier</i>
---------------	-------------------------------------------------------

---

**Description**

Weather Observed Today at a Station Identifier

**Usage**

```
station_today(station_id, timezone = -1, dir_numeric = FALSE)
```

**Arguments**

station_id	The station identifier (ex: KDEN, KNYC, etc).
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.

**Value**

Simple features object with timestamps and meteorological values.

**Examples**

```
station_today("KNYC")
```

---

station_tomorrow	<i>Tomorrows Forecast</i>
------------------	---------------------------

---

**Description**

Tomorrows Forecast

**Usage**

```
station_tomorrow(station_id, timezone = -1, dir_numeric = FALSE, short = TRUE)
```

**Arguments**

station_id	Station identifier (ex: KDEN, KBOS, KNYC, etc).
timezone	The nominal timezone for the forecast. One of OlsonNames() or -1 for local time. Defaults to -1.
dir_numeric	TRUE for numeric directions, FALSE for character directions; defaults to FALSE.
short	TRUE for only tomorrow, FALSE for today and tomorrow; defaults to TRUE.

**Value**

Simple features point corresponding to the given station identifier with recent meteorological forecast values.

**Examples**

```
station_tomorrow("KBOS")
```

---

station_tz	<i>Fetch Station Timezone</i>
------------	-------------------------------

---

**Description**

Fetch Station Timezone

**Usage**

```
station_tz(station_id)
```

**Arguments**

station\_id      The station identifier (ex: KDEN, KBOS, KNYC, etc).

**Value**

A character corresponding to a timezone from `OlsonNames()`.

**Examples**

```
Sys.time() %>% lubridate::force_tz(tzone = station_tz("KDEN"))
```

# Index

..point\_forecast, 2  
.point\_data, 3  
.point\_forecast, 3  
.station\_data, 4

alerts, 4

dir\_as\_char, 5  
dir\_as\_integer, 5

point\_data, 6  
point\_forecast, 6  
point\_station, 7  
point\_today, 8  
point\_tomorrow, 8

safe\_collapse, 9  
station\_coords, 10  
station\_forecast, 11  
station\_obs, 11  
station\_point, 12  
station\_today, 13  
station\_tomorrow, 13  
station\_tz, 14  
stations\_near, 10