



VISUAL CROSSING WEATHER

Articles, videos, and documentation to help you get the most from Visual Crossing Weather

JANUARY 27, 2023 BY VISUAL CROSSING

Timeline Weather API

The Timeline Weather API is the simplest and most powerful way to retrieve weather data. You can request data over any time window including windows that span the past, present, and future. The API will take care of the combining historical observations, current 15-day forecasts, and statistical weather forecasts to create a single, consolidated dataset via a single API call.

Looking for the /forecast and /history end point docs?

Our existing Weather API endpoints for /forecast and /history queries are still fully supported. The documentation for these endpoints can be found [here](#).

The Timeline API offers complete, global weather data coverage both geographically and chronologically. It always picks the best available data sources to answer any weather API query. These sources include:

- Current weather conditions
- Daily historical, forecast and statistical forecast data (depending on dates requested)
- Hourly historical observations and 15-day forecast
- Weather alerts
- Astronomical observations including sunrise, sunset and moon phase.

Result data is provided in a common JSON format allowing you to ignore the complex underlying data sources and focus entirely on your weather data use case. You can also request the result in CSV text format if you prefer.

Migrating from Dark Sky

If you are replacing your existing Dark Sky API implementation, read the [How to replace the Dark Sky API with the Timeline Weather API](#) guide.

Timeline Weather API Query Builder

The [Weather Data Services](#) page includes a full, interactive query builder so you can create queries and see the live results directly in your browser.

API Request Types

All requests to the Timeline Weather API use the following the form:

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/[location]/[date1]/[date2]?key=YOUR_API_KEY
```

location (required) – is the address, partial address or latitude,longitude location for which to retrieve weather data. You can also use US ZIP Codes.

date1 (optional) – is the start date for which to retrieve weather data. If a **date2** value is also given, then it represents the first date for which to retrieve weather data. If no **date2** is specified then weather data for a single day is retrieved, and that date is specified in **date1**. All dates and times are in local time of the **location** specified. Dates should be in the format yyyy-MM-dd. For example 2020-10-19 for October 19th, 2020 or 2017-02-03 for February 3rd, 2017.

Instead of an exact date, you can specify a [dynamic date period](#). See below for more details. You may also supply the in “UNIX time”. In this case provide the number of seconds since 1st January 1970 UTC. For example 1612137600 for Midnight on 1st February 2021.

You can also request the information for a specific time for a single date by including time into the date1 field using the format yyyy-MM-ddTHH:mm:ss. For example 2020-10-19T13:00:00.

The results are returned in the ‘currentConditions’ field and are truncated to the hour requested (i.e. 2020-10-19T13:59:00 will return data at 2020-10-19T13:00:00).

date2 (optional) – is the end date for which to retrieve weather data. This value may only be used when a **date1** value is given. When both **date1** and **date2** values are given, the query is inclusive of **date2** and the weather data request period will end on midnight of the **date2** value. All dates and times are in local time of the specified location and should be in the format yyyy-MM-dd.

When no **date1** or **date2** is specified, the request will retrieve the forecast at the requested **location** for the next 15 days.

Forecast Request Example

The following will retrieve the weather forecast for London, United Kingdom for the next 15 days, starting at midnight at the start of the current day (local time of the requested location).

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK?key=YOUR_API_KEY
```

Forecast Request Example using longitude and latitude

You may also pass the location as a “latitude,longitude” value. For example: 38.9697,-77.385

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/38.9697,-77.385?key=YOUR_API_KEY
```

Date Range Request Example

The following will retrieve the weather data for London, UK from October 1st, 2020 to December 31st, 2020 (inclusive).

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/2020-10-01/2020-12-31?key=YOUR_API_KEY
```

Assuming the current date is November 1st, 2020, the result will include historical observations from October 1st to 31st, then 15 days of weather forecast and finally the remaining days will include the statistical forecast based on processing years of historical observations.

Date Range Request Example using UNIX Time (Epoch Time)

The following shows the same query using UNIX format of seconds since the 1970 UNIX time epoch. Note that these times are seconds in the UTC (GMT/Z) time zone.

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/1601510400/1609372800?key=YOUR_API_KEY
```

Specific Time Request Example

The following will retrieve the weather data for London, UK for the 15th December 2020 and will request the current conditions property be populated using the conditions at 13:00 local time (1pm local time).

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/2020-12-15T13:00:00?key=YOUR_API_KEY
```

The above example will include the daily and hourly detail for the day of the current conditions (2020-12-15 in this case). This query cost will therefore be 24. If you only need the specific time data, and don't need the hourly detail, you can reduce the query costing using the include parameter:

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/2020-12-15T13:00:00?key=YOUR_API_KEY&include=current
```

This query cost will be one.

Dynamic period Request Example

Rather than specify a date range, you can also specify a dynamic period. A request based on a dynamic period will automatically adjust based on the period. In this example, we will use the dynamic period value "last30days" to retrieve data for the most recent 30 days.

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/last30days?key=YOUR_API_KEY
```

Other dynamic period values include "today", "yesterday", and "lastyear". For a complete list of options and additional example, please see the dynamic period article.

Using elements list and options parameter to request only daily with limited elements

By default both daily and hourly data is included in the response along with all the response weather data elements (see below). To reduce the size of the result JSON for lower query cost, reduced network transfer and faster client processing, the 'options' and 'elements' parameters may be used.

This example requests only the daily data:

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/last30days?key=YOUR_API_KEY&include=days
```

This example requests the daily data with only temperature elements

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/last30days?key=YOUR_API_KEY&include=days&elements=tempmax,tempmin,temp
```

Degree day elements list example

The following includes the daily 'degree days' information for the requested location and date range (in this case the last 30 days).

```
https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London,UK/last30days?unitGroup=us&key=YOUR_API_KEY&include=days&elements=datetime,tempmax,tempmin,degreedays,accdegreedays&degreeDayTempMaxThreshold=86&degreeDayTempBase=50
```

For more information on our growing, cooling and heating degree day options, please see our [degree day page](#).

Industry elements

The Timeline API can be extended to include advanced agriculture, horticulture and energy elements. These include evapotranspiration, soil temperature & moisture, advanced solar radiation and higher altitude wind speeds and directions.

To include these advanced industry elements, please see our [Agriculture and Horticulture](#) and [Wind and Solar Energy](#) pages.

Additional request parameters

The following are specified as HTTP query parameters (they may also be passed in a POST query).

key (required) – your API key. Sign up for a free account using our [Weather Data Services](#) page.

unitGroup (optional) – The system of units used for the output data.

Supported values are us, uk, metric, base. See [Unit groups and measurement units](#) for more information. Defaults to US system of units.

lang (optional) – Sets the language of the translatable parts of the output such as the conditions field. Available languages include: ar (Arabic), bg (Bulgarian), da (Danish), de (German), el (Greek Modern), en (English), es (Spanish), fa (Farsi), fi (Finnish), fr (French), hu, (Hungarian), it (Italian), ja

(Japanese), ko (Korean), nl (Netherlands), pl (Polish), pt (Portuguese), ru (Russian), se (Swedish), sr (Serbian), tr (Turkish), uk (Ukrainian) and zh (Chinese). In addition passing in 'id' will result in the raw descriptor IDs.

See [How to create or modify language files](#) for more information on how to help add additional languages.

include (optional) – Specifies the sections you would like to include in the result data. This allows you to reduce query cost and latency. Specify this as a comma separated list. For example: &include=obs,fcst to include the historical observations and forecast data. The options are:

- **days** – daily data
- **hours** – hourly data
- **alerts** – weather alerts
- **current** – current conditions or conditions at requested time.
- **events** – [historical events such as a hail, tornadoes, wind damage and earthquakes](#) (not enabled by default)
- **obs** – historical observations from weather stations
- **remote** – historical observations from remote source such as satellite or radar
- **fcst** – forecast based on 16 day models.
- **stats** – historical statistical normals and daily statistical forecast
- **statsfcst** – use the full statistical forecast information for dates in the future beyond the current model forecast. Permits hourly statistical forecast.

elements (optional) – Specifies the specific weather elements you would like to include in the response as a comma separated list. For example &elements=tempmax,tempmin,temp will request the only the tempmax, tempmin and temp response elements. For the full list of available elements, see the response below.

options (optional) – Specifies additional options on the requests to either indicate the type of data or format of the output. Supported values include:

- **nonulls** – remove all null values from the JSON response
- **noheaders** – removes the header row from the CSV response

contentType (optional) – indicates the output format for the API. By default the output is formatted in JSON. You can also set contentType=csv to retrieve CSV formatted data.

Note that CSV format does not support the full JSON output due to the format limitations. You must use the **include** parameter to indicate which section you would like to retrieve. CSV include parameters support are days, hours, alerts, events and current.

iconSet (optional) – used to choose which icons IDs are populated. For more information see [Defining the icon set parameter in the Weather API](#).

timezone (optional) – specifies the timezone of the input and result dates and times. When not specified, all date times are considered local times. If you would like to specify that all dates are entered as UTC dates and times, use `timezone=Z` parameter.

maxDistance (optional) The maximum distance in meters used to search for local weather stations (By default, 50 miles or approximately 80km (80,467m). This setting is combined with the 'maxStations' parameter to find local weather stations.

maxStations (optional) The maximum number of weather stations used to calculate a weather record (default 3). Closer weather stations are weighted significantly more heavily than farther stations.

elevationDifference (optional) The maximum elevation difference in meters between the requested location and weather stations. Any weather station that is either lower or higher than more than this setting will be excluded. Default is turned off.

locationNames (optional) provides alternative name for the location requested. This is typically used by users who need to be able to join the API results back to a dataset. For example, you may pass a store database ID in this parameter so that you can populate the weather forecast for that store.

forecastBasisDate (optional, requires historical forecast license) – specifies the date when the weather forecast model was run. For example, if you specify 2021-06-01, then the data from 2021-06-01 to 2021-06-15 will use the weather forecast produced on June 1st, 2021.

forecastBasisDay (optional, requires historical forecast license) – specifies the forecast day for the `date1` parameter above. For example, if you specify the start date as 2021-06-05 and a `forecastBasisDay` value of 5, then the value for 2021-06-06 will be the 5th day forecast (i.e. it will be based on the weather forecast produced five days earlier (2021-06-01). You can use this parameter to quickly find the forecast that was predicted for a particular day on multiple days (for example the 5, 3 and 1 day forecast).

Typical JSON response format

All Timeline Weather API requests return response JSON in the same format. Here is an example for Reston, VA on 11/12/2020:

```
{
  "latitude" : 38.9697,
  "longitude" : -77.385,
  "resolvedAddress" : "Reston, VA, United States",
  "address" : " Reston,VA",
```

```

"timezone": "America/New_York",
"tzoffset" : -5,
"description": "Cooling down with a chance of rain on Friday.",
"days" : [{ //array of days of weather data objects
  "datetime": "2020-11-12",
  "datetimeEpoch": 1605157200,
  "temp" : 59.6,
  "feelslike" : 59.6,
  ...
  "stations" : {
  },
  "source" : "obs",
  "hours" : [{ //array of hours of weather data objects
    "datetime" : "01:00:00",
    ...
  }, ...]
}, ...],
>alerts" : [{
  "event" : "Flash Flood Watch",
  "description" : "...",
  ...
}]
,
"currentConditions" : {
  "datetime" : "2020-11-11T22:48:35",
  "datetimeEpoch" : 160515291500,
  "temp" : 67.9,
  ...
}
}

```

The response starts with a set of properties describing the location and request including the specific location requested, the resolved address, the latitude and longitude, a text based time zone and a time zone offset in hours. Note that the time zone offset can change within a dataset (based on daylight savings). If this occurs, the a tzoffset property will be found in the day, hour or current conditions weather data object.

The days array is an array of weather data objects for each day requested.

Within each day may be an hours array for the hourly information. Statistical forecast days will not include an hourly array. Typically the hours array will be length 24 but day light savings may cause the array to be 23 or 25 hours long.

If the request includes the current date, the current conditions and any weather alerts available for the location will be included.

Response weather data elements

The following are the list of properties in a day or hourly data object:

cloudcover – how much of the sky is covered in cloud ranging from 0-100%

conditions – textual representation of the weather conditions. See [Weather Data Conditions](#).

description – longer text descriptions suitable for displaying in weather displays. The descriptions combine the main features of the weather for the day such as precipitation or amount of cloud cover. Daily descriptions are provided for historical and forecast days. When the timeline request includes the model forecast period, a seven day outlook description is provided at the root response level.

datetime – ISO 8601 formatted date, time or datetime value indicating the date and time of the weather data in the local time zone of the requested location. See [Dates and Times in the Weather API](#) for more information.

datetimeEpoch – number of seconds since 1st January 1970 in UTC time

tzoffset – the time zone offset in hours. This will only occur in the data object if it is different from the global time zone offset.

dew – dew point temperature

feelslike – what the temperature feels like accounting for heat index or wind chill. Daily values are average values (mean) for the day.

feelslikemax (day only) – maximum feels like temperature at the location.

feelslikemin (day only) – minimum feels like temperature at the location.

hours – array of hourly weather data objects. This is a child of each of the daily weather object when hours are selected.

humidity – relative humidity in %

icon – a fixed, machine readable summary that can be used to display an icon

moonphase – represents the fractional portion through the current moon lunation cycle ranging from 0 (the new moon) to 0.5 (the full moon) and back to 1 (the next new moon). See [How to include sunrise, sunset, moon phase, moonrise and moonset data into your API requests](#)

normal – array of normal weather data values – Each weather data normal is an array of three values representing, in order, the minimum value over the statistical period, the mean value, and the maximum value over the statistical period.

offsetseconds (hourly only) – time zone offset for this weather data object in seconds – This value may change for a location based on daylight saving time observation.

precip – the amount of liquid precipitation that fell or is predicted to fall in the period. This includes the liquid-equivalent amount of any frozen precipitation such as snow or ice.

precipcover (days only) – the proportion of hours where there was non-zero precipitation

precipprob (forecast only) – the likelihood of measurable precipitation ranging from 0% to 100%

precipitype – an array indicating the type(s) of precipitation expected or that occurred. Possible values include rain, snow, freezingrain and ice.

pressure – the sea level atmospheric or barometric pressure in millibars (or hectopascals)

snow – the amount of snow that fell or is predicted to fall

snowdepth – the depth of snow on the ground

source – the type of weather data used for this weather object. – Values include historical observation (“obs”), forecast (“fcst”), historical forecast (“histfcst”) or statistical forecast (“stats”). If multiple types are used in the same day, “comb” is used. Today a combination of historical observations and forecast data.

stations (historical only) – the weather stations used when collecting an historical observation record

sunrise (day only) – The formatted time of the sunrise (For example “2022-05-23T05:50:40”). See [How to include sunrise, sunset, moon phase, moonrise and moonset data into your API requests](#)

sunriseEpoch – sunrise time specified as number of seconds since 1st January 1970 in UTC time

sunset – The formatted time of the sunset (For example “2022-05-23T20:22:29”). See [How to include sunrise, sunset, moon phase, moonrise and moonset data into your API requests](#)

sunsetEpoch – sunset time specified as number of seconds since 1st January 1970 in UTC time

moonrise (day only, optional) – The formatted time of the moonrise (For example “2022-05-23T02:38:10”). See [How to include sunrise, sunset, moon phase, moonrise and moonset data into your API requests](#)

moonriseEpoch (day only, optional) – moonrise time specified as number of seconds since 1st January 1970 in UTC time

moonset (day only, optional) – The formatted time of the moonset (For example “2022-05-23T13:40:07”)

moonsetEpoch (day only, optional) – moonset time specified as number of seconds since 1st January 1970 in UTC time

temp – temperature at the location. Daily values are average values (mean) for the day.

tempmax (day only) – maximum temperature at the location.

tempmin (day only) – minimum temperature at the location.

uvindex – a value between 0 and 10 indicating the level of ultra violet (UV) exposure for that hour or day. 10 represents high level of exposure, and 0 represents no exposure. The UV index is calculated based on amount of short wave solar radiation which in turn is a level the cloudiness, type of cloud, time of day, time of year and location altitude. Daily values represent the maximum value of the hourly values.

visibility – distance at which distant objects are visible

winddir – direction from which the wind is blowing

windgust – instantaneous wind speed at a location – May be empty if it is not significantly higher than the wind speed. Daily values are the maximum hourly value for the day.

windspeed – the sustained wind speed measured as the average windspeed that occurs during the preceding one to two minutes. Daily values are the maximum hourly value for the day.

windspeedmax (day only, optional) – maximum wind speed over the day.

windspeedmean (day only , optional) – average (mean) wind speed over the day.

windspeedmin (day only , optional) – minimum wind speed over the day.

solarradiation – (W/m^2) the solar radiation power at the instantaneous moment of the observation (or forecast prediction). See the [full solar radiation data documentation](#) and [Wind and Solar Energy](#) pages .

solarenergy – (MJ/m^2) indicates the total energy from the sun that builds up over an hour or day. See the [full solar radiation data documentation](#) and [Wind and Solar Energy](#) pages .

severerisk (forecast only) – a value between 0 and 100 representing the risk of convective storms (eg thunderstorms, hail and tornadoes). It is a scaled measure that combines a variety of other fields such as the convective available potential energy (CAPE) and convective inhibition (CIN), predicted rain and wind. Typically a value less than 30 indicates a low risk, between 30 and 70 a moderate risk and above 70 a high risk.

cape (forecast only) – convective available potential energy. This is a numbering indicating amount of energy available to produce thunderstorms. A higher values indicates a more unstable atmosphere capable of creating stronger storms. Values lower than 1000 J/kg indicate generally low instability, between 1000-2500 J/kg medium instability and 2500-4000 J/kg high instability. Values greater than 4000 J/kg indicating an extremely unstable atmosphere.

cin (forecast only) – convective inhibition. A number representing the level of atmospheric tendency to prevent instability and therefore prevent thunderstorms.

degreedays (day only) – optional elements indicating the number of degree days for this date. See the [degree days API](#) for more information on degree days. To turn degree days and degree day accumulation on, use the elements parameter. For example,
elements=datetime,tempmax,tempmin,degreedays,accdgreedays.

To convert existing Dark Sky API parameters to the Timeline Weather API, see [How to replace the Dark Sky API with the Timeline Weather API](#).

Degree days parameters

degreeDayTempFix – The temperature at which to start the degree day season if no season start and end dates are specified. For example, if the fix temperature is 32F (0C), then the cumulative degree days will reset to zero on the last 32F temperature of the winter. This defines the growing season for growing degree days.

degreeDayStartDate – Fixes the start of the degree day season based on fixed dates of the format yyyy-M-d (eg 1990-3-1 for 1st March). Default=not specified.

degreeDayTempMaxThreshold – Defines the maximum temperature that is considered for the calculation. Any temperature above this temperature will be set to the maximum threshold temperature. Default value=not set.

degreeDayTempBase – The degree day base temperature. Default value is set to 10C/50F

degreeDayInverse – Calculate the inverse degree days so colder temperatures contribute more degree days. Used for heating degree days. Default=false.

degreeDayMethod (beta) – The method used for calculating the degree days. Values include average, sine, doublesine, triangle and doubletriangle. Default value is set to average.

Accessing the Timeline Weather API using the OpenAPI Description

The [OpenAPI Initiative](#) standardizes the description of APIs such as the Visual Crossing Weather API. If you are using a tool that supports the OpenAPI standard or Swagger, you can more easily use our API by importing the the specification document below.

<https://www.visualcrossing.com/weather/specs/visualcrossing-weather-api-openapi.json>

<https://www.visualcrossing.com/weather/specs/visualcrossing-weather-api-openapi.yaml>

<https://www.visualcrossing.com/weather/specs/visualcrossing-weather-api-swagger.json>

Questions or need help?

If you have a question or need help, please post on our [actively monitored forum](#) for the fastest replies. You can also contact us via our [support site](#) or drop us an email at support@visualcrossing.com.

 **WEATHER API**

HISTORICAL WEATHER, JSON, WEATHER API, WEATHER DATA

14 Replies to “Timeline Weather API”

Tommy Gombar

DECEMBER 14, 2020 AT 12:53 PM

Hi VisualCrossing,

Is it possible to request only a subset of the weather data elements to be included in the response ? Or would not it be affecting the response time at all, even for periods span over multiple decades ?

Thank you.

Visual Crossing**DECEMBER 14, 2020 AT 11:40 PM**

Tommy

Thank you for your question. We are currently designing an API option to be able to request only some weather data elements. We do not yet have a time frame for the release of the feature however. Please contact support at support@visualcrossing.com if you would like to be notified when the feature is released.

Regards

Visual Crossing Support

Marc**DECEMBER 20, 2020 AT 2:15 PM**

What are the units of rainfall and snow in metric

Bjoern Schaefer**DECEMBER 21, 2020 AT 4:26 PM**

Hi – you find the measures information under:

<https://www.visualcrossing.com/resources/documentation/weather-api/unit-groups-and-measurement-units/>

Jerry Yip**JANUARY 8, 2021 AT 5:03 AM**

Hello, can someone clarify the list of possible “conditions”? For example, I’ve found your documentation on possible icons (i.e. located at: <https://www.visualcrossing.com/resources/documentation/weather-api/defining-icon-set-in-the-weather-api/>), but am unable to find similar documentation for the different weather conditions. The possible conditions I’ve seen appear to be more descriptive and diverse than the icons, so was hoping to get a better understanding of all their possible values.

Thanks!

David Shaske**JANUARY 8, 2021 AT 10:37 PM**

How would I only request one specific hour of historic data, instead of the entire day of historic data?

Visual Crossing

JANUARY 11, 2021 AT 1:14 PM

David

Thank you for your comment. It is currently necessary to request the whole day for a specific hour. We are looking to add the ability to request a particular hour in the near future. Please contact support@visualcrossing.com if you would like more information and timing on that upcoming feature.

Regards

Visual Crossing Support

David Shaske

JANUARY 8, 2021 AT 11:25 PM

I also cannot figure out a way to request less than 15 days of forecast data. What if I only want current conditions? Or 1 day?

Visual Crossing

JANUARY 11, 2021 AT 1:19 PM

David

You may request a single day or a small number of forecast days by supplying a date range or dynamic period. For example this will return the single day of forecast for tomorrow (at this time):

https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/Los%20Angles,CA/2021-01-12?key=YOUR_API_KEY

You can also supply a dynamic date range:

https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/Los%20Angles,CA/tomorrow?key=YOUR_API_KEY

https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/Los%20Angles,CA/next3days?key=YOUR_API_KEY

Regards

Visual Crossing Support

Eric

JANUARY 20, 2021 AT 8:44 AM

Hi,

Looking for a replacement for Darksky and doing some tests against your service.

Why is it that most of your historical data returns null values? For example:

https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London/2021-1-3?unitGroup=metric&key=YOUR_API_KEY&include=hours%2Calerts

Does it mean historical data is not available?

```
{
  "datetime": "2021-01-03",
  "datetimeEpoch": 1609632000,
  "tempmax": 0.0,
  "tempmin": 0.0,
  "temp": null,
  "feelslikemax": 0.0,
  "feelslikemin": 0.0,
  "feelslike": null,
  "dew": null,
  "humidity": null,
  "precip": null,
  "precipprob": null,
  "precipcover": 0.0,
  "preciptype": null,
  "snow": null,
  "snowdepth": null,
  "windgust": null,
  "windspeed": null,
  "winddir": null,
  "pressure": null,
  "cloudcover": null,
  "visibility": null,
  "solarradiation": null,
  "solarenergy": null,
  "sunrise": "08:05:54",
  "sunriseEpoch": 1609661154,
  "sunset": "16:04:36",
  "sunsetEpoch": 1609689876,
  "moonphase": 0.6,
  "conditions": "",
  "icon": "",
  "stations": null,
  "hours": [
    {
      "datetime": "00:00:00",
      "datetimeEpoch": 1609632000,
      "temp": null,
      "feelslike": null,
      "humidity": null,
      "dew": null,
      "precip": null,
      "precipprob": null,
      "snow": null,
      "snowdepth": null,
```



```
“preciptype”: null,  
“windgust”: null,  
“windspeed”: null,  
“winddir”: null,  
“pressure”: null,  
“visibility”: null,  
“cloudcover”: null,  
“solarradiation”: null,  
“solarenergy”: null,  
“conditions”: “”,  
“icon”: “”,  
“stations”: null,  
“moonphase”: 0.0  
},
```

Visual Crossing

JANUARY 20, 2021 AT 2:40 PM

Thank you for your question.

The hourly historical observations are not being included because the ‘includes’ parameters needs to also include the ‘obs’ (for observation) field so that the observation data is populated. To include the historical observations, add ‘obs’ to your includes list. For example:

https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/London/2021-1-3?unitGroup=metric&key=YOUR_API_KEY&include=obs%2Chours%2Calerts

Regards

Visual Crossing Support

R.L.

FEBRUARY 3, 2021 AT 8:07 AM

On a date-based query:

<https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/32.806128%2C-117.027267/2021-2-2?unitGroup=us&key=mykey&options=nonnulls>

Does the returned ‘precip’ field within the first-level ‘days[0]’ element indicate the total precipitation (rain or melted snow I guess) for the day? And, would I then get the same accumulation if I summed all of the ‘precip’ results from each of the hour groups within the ‘days[0]’ element?

R.L.

FEBRUARY 3, 2021 AT 8:11 AM

There is no description for the 'preciptype' field. How is this field used? I've seen it appear once when there was snow in an area, and it appeared to be an array (but of what and how do I properly process that field). How does it relate specifically to the 'precip' and 'snow' fields?

R.L.

FEBRUARY 9, 2021 AT 4:00 PM

I see that the 'preciptype' field has been added:

preciptype (forecast only) – an array indicating the type(s) of precipitation expected. Possible values include rain, snow, freezingrain and ice.

Comments are closed.