Weather API research

AmbeeAPI :

Traits?

• Simple Integration: Regardless of the platform or programming language you're using, the Ambee Weather API is well-documented and simple to incorporate into your application. This shortens the time it takes for your application to go live and streamlines the development process.

• Historical Weather Data: You can analyse previous weather patterns and trends by using the historical weather data that is accessible through the API. Research, planning, and trend analysis in a variety of industries, including tourism, transportation, and agriculture, can benefit from this.

• Scalability and Reliability: Ambee's infrastructure is built to manage high request volumes with little downtime, guaranteeing your application's scalability and dependability. Ambee's infrastructure can provide reliable performance regardless of the size of your user base—it can handle millions or just a few thousand.

• Hyper-local Weather Data: Ambee Weather API provides extremely detailed weather information at a hyper-local level. This might be especially helpful if your application needs accurate weather predictions for particular areas or locales.

• API Customization and Flexibility: Ambee provides customizable API parameters and endpoints so you may alter the data you receive to meet your unique needs. This adaptability can guarantee that you only retrieve the data necessary for your application and speed up development.

• Community and Developer Support: Ambee provides information and developer support to help you integrate their API into your application. Joining a developer community that makes use of the Ambee API can also present chances for learning, sharing, and teamwork.

CALL API?

- Endpoint Structure: The Ambee Weather API offers a range of endpoints for obtaining several kinds of meteorological information. Every endpoint is associated with a particular kind of information or feature, like the present or historical weather, or weather projections.

- HTTP Methods: Common HTTP methods including GET, POST, PUT, and DELETE are supported by the API. However, the GET approach is usually used for the majority of data retrieval tasks, such as obtaining weather information.

  Authentication: Developers must authenticate their requests prior to making API calls Usually, this entails adding an API key to the query parameters or request header. The

API key guarantees safe access to the data and acts as a distinctive identification for the application requesting access to the API.

- Request Parameters: Developers may need to add particular parameters in their API requests, depending on the endpoint they are accessing. These parameters may contain information about the location for which meteorological data is needed, the time frame for historical data, preferred language, units of measurement, etc.

  Response Format: JSON (JavaScript Object Notation) is the most prevalent structured format that the API returns responses in. Developers may easily interpret and retrieve the necessary data from the API answers in their apps thanks to this format.

  Rate Limiting: Ambee may set rate restrictions on API calls in order to guard against misuse and guarantee equitable use of the API. It is imperative for developers to acknowledge these limitations and incorporate elegant techniques for handling rate limitation in their systems, such as caching mechanisms or logic.

- Error Handling: In the event of invalid queries, unsuccessful authentication attempts, or other problems, the Ambee Weather API, like any other API, may produce error answers. It is recommended that developers incorporate error handling features into their programs to effectively handle such instances and furnish users with insightful feedback.

  Asynchronous Operations: Certain API calls may be handled asynchronously, particularly if they involve big datasets or intricate calculations. In these situations, developers might have to deal with asynchronous responses or make use of the callback methods the API offers to get the results as soon as they're ready.

YOUTUBE LINK FOR HOW TO SEND API REQUEST:
https://www.youtube.com/watch?v=7w8r5tGzVow&ab_channel=Ambee

DOCUMENTATION?

The technical documentation from Ambee is a straightforward manual that explains how to utilize and integrate an API efficiently. It includes all the details about functions, classes, return types, arguments, and more that are needed to operate with the API. This will make it easier for developers and users of Ambee's environmental APIs to learn more about the APIs and acquire the skills required to use the platform.

LINK FOR DOCUMENTATION:

https://docs.ambeedata.com/DeveloperTools/?refId=api-intro

Open-meteo Weather API:

Through the weather API offered by Open-meteo, developers can obtain weather forecast data for numerous global locations. The API provides forecasts for up to 14 days ahead of time in addition to the present weather.

Traits:

- Global Coverage: The API is appropriate for applications that need weather data for a variety of geographic regions because it provides weather forecasts for sites all over the world.
- Hourly Forecast: Developers can get comprehensive weather predictions for particular periods of the day with the help of Open-meteo.com's hourly weather forecasts.
- Multiple Data Formats: The API allows developers to select the format that best meets their needs by offering weather data in JSON, XML, and CSV, among other formats.
- Historical Weather Data: Open-meteo.com provides access to historical weather data in addition to current conditions and forecasts. Analyzing historical weather patterns and trends can benefit from this.
- Easy to Use: The API has been created with ease of use in mind, featuring simple endpoints and parameters. Weather data may be easily integrated into applications by developers without requiring a lot of setup or preparation.
- Free Tier: Open-meteo.com provides developers with a free tier with limited usage so they may test the API and incorporate weather data into their apps without having to pay anything. For testing or tiny projects, this can be useful.
- Developer assistance: To help developers successfully integrate the API into their applications, Open-meteo.com offers developer documentation and assistance. This consists of support forums, code samples, and thorough documentation.

Overall, with features like hourly forecasts, historical data, and support for several data formats, the Open-meteo.com weather API provides a practical way to get weather forecast data for numerous places worldwide.

DATA SOURCE

The Open-Meteo weather prediction Weather models from several national weather suppliers are used by APIs. The best models will be integrated to get the best forecast possible for any area on Earth.

Different weather variables are provided by weather models, which cover different geographic areas at different resolutions. Depending on the model, not all weather variables are accessible or the data have been interpolated to hourly values.

Weather Model	National Weather Provider	Origin Country	Resolution	Forecast Length	Update frequency
ICON	Deutscher Wetterdienst (DWD)	Germany	2 - 11 km	7.5 days	Every 3 hours
GFS & HRRR	NOAA	United States	3 - 25 km	16 days	Every hour
ARPEGE & AROME	Météo-France	France	1 - 25 km	4 days	Every hour
IFS & AIFS	ECMWF	European Union	25 km	7 days	Every 6 hours
MSM & GSM	JMA	Japan	5 - 55 km	11 days	Every 3 hours
MET Nordic	MET Norway	Norway	1 km	2.5 days	Every hour
GEM	Canadian Weather Service	Canada	2.5 km	10 days	Every 6 hours
GFS GRAPES	China Meteorological Administration (CMA)	China	15 km	10 days	Every 6 hours
ACCESS-G	Australian Bureau of Meteorology (BOM)	Australia	15 km	10 days	Every 6 hours
COSMO 2I & 5M	AM ARPAE ARPAP	Italy	2 km	3 days	Every 3 hours

Figure 1: Weather models and Variables.

DOCUMENTATION

The Open-meteo.com weather API's /v1/forecast endpoint enables users to obtain hourly weather forecast information for a given geographic coordinate. Starting at 0:00 today and spanning up to seven days, the forecast offers information for each of the twenty-four hours of a day.

Geographical Coordinate: This option indicates the place where information about the weather prediction is needed. Coordinates for latitude and longitude are usually included :
latitude: The latitude coordinate of the location for which weather forecast data is requested.
longitude: The longitude coordinate of the location for which weather forecast data is requested.

List of Weather Variables: The weather variables that the API ought to incorporate into the forecast answer are specified by this option. A few examples of weather variables are temperature, humidity, wind direction, and precipitation. Which variables developers are interested in getting can be specified.

prediction Days: The number of days of prediction data to be returned is specified by this parameter, which is denoted by &forecast_days=16. When this parameter is set to 16, the API can return up to 16 days of forecast data instead of the typical 7 days.

With this data, we can create API queries to the /v1/forecast endpoint that obtain hourly weather forecast data for the chosen place. The requests should include the required geographic coordinates, weather variables, and forecast duration. A JSON object with the forecast data for the specified time period will be returned by the API.

So now we can put this together in a URL to request :https://api.open-meteo.com/v1/forecast?latitude=XX.X&longitude=YY.Y&variables=temperature,humidity,wind_speed,precipitation&forecast_days=7

Expected Return from JSON :


```
{
  "latitude": XX.XXXX,
  "longitude": YY.YYYY,
  "timezone": "UTC",
  "forecast_days": 7,
  "forecast": [
    {
      "time": "2024-03-17T00:00:00Z",
      "temperature": 20.5,
      "humidity": 65,
      "wind_speed": 10.2,
      "precipitation": 0.0
    },
    // Additional hourly forecast data for the next 167 hours...
  ]
}
```



```
  "latitude": 52.52,
  "longitude": 13.419,
  "elevation": 44.812,
  "generationtime_ms": 2.2119,
  "utc_offset_seconds": 0,
  "timezone": "Europe/Berlin",
  "timezone_abbreviation": "CEST",
  "hourly": {
    "time": ["2022-07-01T00:00", "2022-07-01T01:00", "2022-07-01T02:00", ...],
    "temperature_2m": [13, 12.7, 12.7, 12.5, 12.5, 12.8, 13, 12.9, 13.3, ...]
  },
  "hourly_units": {
    "temperature_2m": "°C"
  }
```


Figure 2:Success fom request

HISTORICAL WEATHER API:

Open-Meteo's Historical Weather API provides a comprehensive archive of historical weather conditions derived from reanalysis datasets. Numerous sources of data, such as weather stations, airplanes, buoys, radar, and satellite observations, are combined in these datasets. These databases interpolate and estimate weather variables by using mathematical models, which makes it possible to create comprehensive historical weather records.

Data Sources and Resolution

The European Centre for Medium-Range Weather Forecasts Integrated Forecast System (ECMWF IFS) dataset is the main source of historical weather data, while it is derived from a number of databases. This dataset uses 9 km spatial resolution and simulation runs at 0 and 12 z. It was painstakingly assembled by Open-Meteo. This high resolution makes sure that historical weather conditions are accurately depicted, which is especially helpful for locations with difficult topography or coastal areas.

It is advised to use the ERA5 or ERA5-Land datasets for decades-long climate studies in order to maintain consistency and avoid unintentional changes brought about by modifications in weather model updates. With a resolution of 0.1° (~11 km), ERA5-Land gives even more detail than ERA5's 0.25° (~25 km) spatial resolution. Hourly data availability from 1940 onward is provided by both datasets, albeit five days after the fact.

Data accessibility and frequency of updates

There is a wealth of historical weather data available, with the ERA5 and ERA5-Land datasets going back to 1940 and the ECMWF IFS to 2017. For the ERA5 and ERA5-Land datasets, there is a 5-day delay and a 2-day delay for the ECMWF IFS dataset. Every day, data updates are carried out to guarantee that customers have access to the most recent historical weather data.

It is important to remember that different sets of meteorological variables may be included in the datasets of different reanalysis models. For example, ERA5-Land concentrates on surface factors such as temperature, humidity, soil temperature, and soil moisture, whereas ERA5 covers all weather variables. Comprehending the variables incorporated in every model facilitates the understanding of data constraints and possible biases.

API DOCUMENTATION

Users can obtain comprehensive historical weather information for particular places and time periods by utilizing the Historical Weather API's /v1/archive endpoint. Users can retrieve extensive historical weather records by supplying time intervals, geographic coordinates, and relevant weather variables. Researchers, analysts, and applications needing access to comprehensive historical weather data for a variety of uses will find this endpoint to be a useful resource.

Parameters for the URL:

- latitude: The location's latitude coordinate for which historical weather information is needed.
- longitude: The location's longitude coordinate for which historical weather information is needed.
- start_date: The beginning date of the time frame for which you are requesting historical weather information. The format that it should be in is YYYY-MM-DD.
- end_date: The conclusion date of the period frame for which you are requesting historical meteorological data. The format that it should be in is YYYY-MM-DD.
- variables: A list of meteorological factors that the historical weather data response from the API should contain. Variables including temperature, humidity, wind speed, precipitation, etc. can be specified by users.

Lets put together:
/v1/archive?latitude=XX.X&longitude=YY.Y&start_date=2022-01-01&end_date=2022-01-07&variables=temperature,humidity,wind_speed,precipitation

On success a JSON object will be returned.


```
"latitude": 52.52,
"longitude": 13.419,
"generationtime_ms": 2.2119,
"timezone": "Europe/Berlin",
"timezone_abbreviation": "CEST",
"hourly": {
  "time": ["2022-07-01T00:00", "2022-07-01T01:00", "2022-07-01T02:00", ...],
  "temperature_2m": [13, 12.7, 12.7, 12.5, 12.5, 12.8, 13, 12.9, 13.3, ...]
},
"hourly_units": {
  "temperature_2m": "°C"
},
```


Figure 3:Output from a successful request

Errors

In case an error occurs, for example a URL parameter is not correctly specified, a JSON error object is returned with a HTTP 400 status code.


```
"error": true,
"reason": "Cannot initialize WeatherVariable from invalid String value tempeture_2m for key hourly"
```


Figure 4: Error