

# WeatherWear API Documentation

## Overview

WeatherWear provides RESTful APIs for retrieving weather data and performing geocoding services. All endpoints return JSON responses and adhere to standard HTTP status codes. This documentation outlines the available endpoints, their parameters, expected responses, and best practices.

## Base URL

- Local environment: `http://localhost:3000/api`
- Production environment (example): `https://weatherwear.com/api`

**Note:** Update the base URL as appropriate based on your environment.

## Authentication

All weather-related endpoints require a valid RapidAPI key. Include your RapidAPI key in the request headers. For local development or testing, ensure your key is configured in environment variables.

Example header:

X-RapidAPI-Key: YOUR\_API\_KEY

## Endpoints

### 1. Current Weather

**GET** `/weather`

Retrieve the current weather conditions for a given location.

**Query Parameters**

Parameter	Type	Required	Description
lat	number	*	Latitude (-90 to 90).
lon	number	*	Longitude (-180 to 180).
city	string	*	City name (alternative to providing lat/lon).
country	string	No	ISO 3166-1 alpha-2 country code (default: "US").

*You must provide either a (lat, lon) pair or a city name.*

#### Response Example

```
{  
  "city": "New York",  
  "country": "US",  
  "temperature": {  
    "current": 75.2,  
    "feels_like": 77.0,  
    "min": 70.0,  
    "max": 80.1  
  },  
  "humidity": 60,  
  "wind": {  
    "speed": 5.2,  
    "degree": 180  
  },  
}
```

```
"weather": {  
  "main": "Clouds",  
  "description": "broken clouds",  
  "icon": "04d"  
},  
"pressure": 1012,  
"visibility": 10,  
"timestamp": 1697043600  
}
```

### Example Request

```
curl "http://localhost:3000/api/weather?city=New%20York" \  
-H "X-RapidAPI-Key: YOUR_API_KEY"
```

## 2. Weather Forecast (5-Day)

**GET** /weather/forecast

Retrieve a 5-day forecast for a given location, returned in 3-hour intervals aggregated by day.

### Query Parameters

Parameter	Type	Required	Description
lat	number	Yes	Latitude (-90 to 90).
lon	number	Yes	Longitude (-180 to 180).

## Response Example

```
{  
  "city": "New York",  
  "country": "US",  
  "forecast": [  
    {  
      "temperature": {  
        "current": 76.3,  
        "feels_like": 78.0,  
        "min": 68.0,  
        "max": 79.5  
      },  
      "humidity": 55,  
      "wind": {  
        "speed": 4.5,  
        "degree": 200  
      },  
      "weather": {  
        "main": "Rain",  
        "description": "moderate rain",  
        "icon": "10d"  
      },  
      "pressure": 1015,  
      "visibility": 8,  
      "timestamp": 1697050800,
```

```
    "dt_txt": "2024-10-11 15:00:00",  
    "pop": 0.7  
  }  
]  
}
```

### Example Request

```
curl "http://localhost:3000/api/weather/forecast?lat=40.7128&lon=-74.0060" \  
-H "X-RapidAPI-Key: YOUR_API_KEY"
```

## 3. Geocoding

**GET** /geocoding

Search for locations by name or address. The response includes up to 5 results, each containing latitude and longitude coordinates.

### Query Parameters

Parameter	Type	Required	Description
search	string	Yes	Location search term (minimum 2 characters)

### Response Example

```
[  
  {  
    "name": "New York",  
    "state": "New York",  
    "country": "US",
```

```
"lat": 40.7128,  
"lon": -74.0060,  
"fullAddress": "New York, New York, US"  
}  
]
```

### Example Request

```
curl "http://localhost:3000/api/geocoding?search=New%20York" \  
-H "X-RapidAPI-Key: YOUR_API_KEY"
```

## Error Handling

All endpoints return errors in a consistent JSON format:

```
{  
  "error": "Location not found. Please check the spelling and try again",  
  "status": 404  
}
```

### Common HTTP Status Codes

- **200:** Success
- **400:** Bad Request (e.g., invalid parameters)
- **404:** Location not found
- **408:** Request timeout
- **429:** Rate limit exceeded
- **500:** Internal server error
- **502:** Invalid response from weather service
- **503:** Weather service not properly configured

### Common Error Messages

- "Location not found. Please check the spelling and try again"
- "Invalid coordinates provided"
- "Either city name or coordinates (lat/lon) are required"
- "Rate limit exceeded. Please try again later"
- "Weather service temporarily unavailable"

## Rate Limiting

The API uses RapidAPI's rate limiting. Consider implementing client-side caching to reduce unnecessary requests. Weather data is cached server-side for 5 minutes.

## Data Freshness and Timeouts

- Weather data is cached for 5 minutes (revalidate: 300).
- All API requests have a 5-second timeout.

## Best Practices

1. Always handle error responses gracefully.
2. Implement exponential backoff for retries to avoid hitting rate limits.
3. Cache responses to improve performance and reduce overhead.
4. Validate coordinates before making requests.
5. Use HTTPS in production environments to ensure data security.

## Units and Formats

- **Temperature:** Fahrenheit
- **Wind Speed:** Meters per second (m/s)
- **Wind Direction:** Degrees (0-360)
- **Visibility:** Kilometers
- **Pressure:** Hectopascals (hPa)
- **Timestamps:** Unix timestamp (seconds)

## Notes

- The API converts temperatures from Kelvin (as provided by upstream providers) to Fahrenheit.
- Weather icons returned are codes compatible with the OpenWeather icon set.

- Geocoding results are limited to 5 entries per request.
- Forecast data is provided in 3-hour increments and aggregated by day.

weatherwear