@mostajs/weather

> Weather module for @mostajs — OpenWeatherMap, flyability criteria, alerts, demo mode.

Install

```
npm install @mostajs/weather
```

Usage

Fetch weather data

```
import { fetchWeather } from '@mostajs/weather'

const data = await fetchWeather({
  apiKey: process.env.OPENWEATHER_KEY,
  location: { lat: 48.8566, lon: 2.3522, label: 'Paris' },
})

console.log(data.current.temp)     // 22
console.log(data.forecast[0].flyable) // true
```

Demo mode (no API key)

```
const data = await fetchWeather({
  location: { lat: 36.75, lon: 3.06, label: 'Alger' },
})
// Returns mock data with isRealData: false
```

Fetch by place name (composes `@mostajs/geo`)

Resolve a place name to coordinates via `@mostajs/geo`, then fetch the weather — no need to know lat/lon.

```
import { fetchWeatherByPlace } from '@mostajs/weather'

// Real geocoding via OpenStreetMap (default), real weather via OWM key
const data = await fetchWeatherByPlace('Alger', {
  apiKey: process.env.OPENWEATHER_KEY,
  geo: { provider: 'osm', lang: 'fr' },
})

// Fully offline (demo geocoder + demo weather)
const demo = await fetchWeatherByPlace('Alger', { geo: { provider: 'demo' } })
demo.location // 'Alger'  ·  demo.isRealData // false
```

The classic `fetchWeather({ location: { lat, lon } })` API is **unchanged** (backward compatible — see `test-scripts/non-regression.test.mjs`).

React component

```
import { WeatherPage } from '@mostajs/weather'

<WeatherPage
  config={{
    apiKey: process.env.NEXT_PUBLIC_OPENWEATHER_KEY,
    location: { lat: 36.7538, lon: 3.0588, label: 'Alger — Site de decollage' },
    flyabilityCriteria: { maxWindKmh: 25, minVisibilityKm: 3, forbiddenConditions: ['Rain',
'Thunderstorm', 'Snow'] },
  }}
  onBack={() => router.back()}
/>
```

Flyability check

```
import { isFlyable, msToKmh } from '@mostajs/weather'

const windKmh = msToKmh(7.2) // 26 km/h
isFlyable(windKmh, 'Clear')  // false (> 25 km/h)
isFlyable(15, 'Rain')        // false (Rain is forbidden)
isFlyable(15, 'Clear')       // true
```

API

Export	Description
`fetchWeather(config)`	Fetch current + 5-day forecast (or mock data)
`fetchWeatherByPlace(query, config?)`	Geocode a place via `@mostajs/geo` then fetch its weather
`resolveLabel(location, geo?)`	Fill a missing label via reverse-geocoding (`@mostajs/geo`)
`isFlyable(wind, condition, criteria?)`	Check if conditions allow flight
`msToKmh(ms)`	Convert m/s to km/h
`getConditionIcon(condition)`	Get emoji for weather condition
`getMockWeather(config)`	Generate mock weather data
`WeatherPage`	React component for weather display
`DEFAULT_CRITERIA`	Default flyability criteria
`CONDITION_ICONS`	Condition-to-emoji mapping

License