# User Guide: alerts.json Configuration File

## Introduction

The alerts.json configuration file defines weather alerts based on real-time data collected from weather stations. This guide explains the structure, available parameters, and best practices for configuring alerts effectively.

## Alert Configuration Structure

Each alert is defined as a JSON object within the alerts array. Below is an example structure:

"alerts": [

{

"name": "Data update",

"weather\_field": "WC\_temp",

"weather\_unit": "mn",

"alert\_type": "no\_update",

"threshold": 10,

"time\_ago": "0m",

"enabled": true,

"message": "This is extremely important regarding the connection between weather station and Database!",

"cooldown": 120,

"stations": ["db1", "db2"],

"recipients": ["user@example.com", "alert@example.com"]

}

]

## Field Descriptions

| Field | Type | Description |
| --- | --- | --- |
| name | String | The name of the alert. |
| weather\_field | String | The database field being monitored (e.g., temperature, precipitation). |
| weather\_unit | String | The unit of measurement (e.g., °C, mm, mn). |
| alert\_type | String | The type of alert (increase, decrease, or no\_update). |
| threshold | Number | The threshold value that triggers the alert. |
| time\_ago | String | The time period for comparison (e.g., 1h for 1 hour ago). |
| enabled | Boolean | Whether the alert is active (true) or disabled (false). |
| message | String | Custom message displayed in the alert email. Supports placeholders like {threshold} and {time\_ago}. |
| cooldown | Number | The minimum time (in minutes) before the same alert can be triggered again. |
| stations | Array | List of weather station database IDs to monitor. |
| recipients | Array | List of email addresses that will receive the alert notification. |

## Alert Types

* increase: Triggers when the value rises above the threshold within the specified time\_ago.
* decrease: Triggers when the value drops below the threshold within the specified time\_ago.
* no\_update (New): Triggers when no data update has been received for more than threshold minutes.

## Example Alert Scenarios

1. **Detect missing data updates**

{

"name": "No Data Update",

"weather\_field": "WC\_temp",

"weather\_unit": "mn",

"alert\_type": "no\_update",

"threshold": 15,

"time\_ago": "0m",

"enabled": true,

"message": "No new data received in the last 15 minutes!",

"cooldown": 120,

"stations": ["db1"],

"recipients": ["admin@example.com"]

}

1. **Rapid temperature increase alert**

{

"name": "Rapid Temperature Rise",

"weather\_field": "WC\_temp",

"weather\_unit": "°C",

"alert\_type": "increase",

"threshold": 5,

"time\_ago": "1h",

"enabled": true,

"message": "Temperature increased by more than {threshold}°C in the last {time\_ago}!",

"cooldown": 60,

"stations": ["db1", "db2"],

"recipients": ["alerts@example.com"]

}

1. **Heavy Rainfall Warning**

{

"name": "Very Heavy Rainfall",

"weather\_field": "WC\_precipRate",

"weather\_unit": "mm",

"alert\_type": "increase",

"threshold": 30,

"time\_ago": "0m",

"enabled": true,

"message": "Rainfall intensity exceeded {threshold}mm!",

"cooldown": 60,

"stations": ["db1"],

"recipients": ["weatherteam@example.com"]

}

## Tips & Tricks

### 1️⃣ Use no\_update Alerts for Data Integrity

* Ensure your weather station is always sending data by setting a no\_update alert.
* Example: If no data arrives for **10 minutes**, an alert is sent.

### 2️⃣ Utilize message Placeholders

* Dynamic placeholders like {threshold} and {time\_ago} personalize alert messages.
* Example: "Temperature dropped by more than {threshold}°C in the last {time\_ago}!"

### 3️⃣ Exclude Specific Time Ranges

* Use exclude\_hours to ignore fluctuations in specific time frames (e.g., sunrise or evening cooling periods).

### 4️⃣ Adjust cooldown to Prevent Spam

* Prevent multiple alerts in short periods by setting a reasonable cooldown.

### 5️⃣ Monitor Multiple Weather Stations

* Assign alerts to multiple stations to track various locations simultaneously.

## Conclusion

The alerts.json file is a powerful tool to monitor weather conditions and ensure timely notifications. Proper configuration allows for efficient alerting, reducing unnecessary notifications while maintaining data reliability.

🚀 **Stay ahead of weather changes and keep your monitoring system efficient!**