JUMBO BLOCK Public API

Version: 8.2.0
Base URL / current environment: https://jumboblock.net
Full endpoint: https://jumboblock.net/api/public/sensors
As of: 2026-11-24

> *jumboblock.net is the current environment under which this API is provided. It is the same address for both testing and production use — there is no separate test environment with a different URL.*

A public, authentication-free API for municipalities, authorities, emergency services, and state/federal government bodies.

The data hierarchy (Sensor → Zone → Plant) is stable and will not change. New fields may be added; existing fields remain unchanged.

Endpoint

GET /api/public/sensors

Returns the current state of all sensors, zones and the overall plant, the water level indicator, weather data and the status of data anchoring.

Response (example, shortened)

```
{
  "ok": true,
  "ts": "2026-11-23T08:15:02.114Z",
  "dashboardVersion": "8.2.0",
  "sensors": [
    { "uid": "JBS-XXXX-XXXX-XXXX-XXXX",
      "fillM3": 5.004, "levelMm": 900, "levelIndicator": "gelb",
      "updatedAt": "2026-11-23T08:13:40.902Z",
      "ageSeconds": 81, "isStale": false }
  ],
  "zones": [
    { "id": "G1", "label": "Zone A", "blocks": 20,
      "capacityNetM3": 278, "capacityGrossM3": 312.5,
      "fillNetM3": 100.08, "fillGrossM3": 112.5, "fillPercent": 36 }
  ],
  "plant": {
    "uid": "JBA-XXXX-XXXX-XXXX-XXXX",
    "levelMm": 900,
    "levelIndicator": { "code": "gelb", "color": "#ffc107",
      "label": { "de": "Beobachten", "en": "Watch" }, "reason": null },
    "rise": { "mmPerMin": 0.8, "minutesToOverflow": 1625,
      "uplinkIntervalMin": 5, "checkActive": true, "horizonMin": 15 },
    "capacityNetM3": 278, "capacityGrossM3": 312.5,
    "fillNetM3": 100.08, "fillGrossM3": 112.5, "fillPercent": 36
  },
  "levelScale": [ { "code": "leer", "fromMm": 0, "toMm": 50, ... }, ... ],
  "hydraulics": { "overflowMm": 2200, "drainOutletMm": 1000,
    "criticalFromMm": 1900, "criticalBasis": "rain15",
    "designRain15": { "lPerSecHa": 200, "returnPeriodYears": 5 } },
  "weather": { "current": { ... }, "nextHours": [ ... ],
    "fetchedAt": "2026-11-23T08:10:11.020Z", "isStale": false },
  "anchor": { "spec": "JB-ANCHOR-V1", "network": "mainnet",
    "lastConfirmed": { "seq": 112, "digest": "...", ... },
    "isStale": false, ... }
}
```

> *Example values. Parts shortened with "…" are complete in the real response.*

Error Response (example)

If a server-side problem occurs, the endpoint responds with HTTP status 500:

```
{
  "ok": false,
  "error": "public_sensors_error"
}
```

Field Reference

Field	Type	Description
ok	boolean	true on a successful request
ts	string (ISO 8601)	Timestamp of the response (UTC)
dashboardVersion	string	Version of the dashboard software
sensors[]	array	Individual sensors
↳ uid	string	Permanent, random sensor identifier (JBS-…); key for time series
↳ fillM3 / levelMm	number	Fill level per block in m^3 / water level in mm
↳ levelIndicator	string	Indicator level of the sensor (code)
↳ updatedAt / ageSeconds / isStale	string / number / boolean	Time and age of the last measurement; isStale = no current measurement
zones[]	array	Zones: id, label, blocks, capacityNetM3, capacityGrossM3, fillNetM3, fillGrossM3, fillPercent
plant	object	Overall plant (sum of all zones)
↳ uid	string	Permanent plant identifier
↳ capacityNetM3 / capacityGrossM3	number	Net / gross capacity in m^3
↳ fillNetM3 / fillGrossM3	number	Actual fill (net) / total fill (gross) in m^3
↳ fillPercent	number	Fill level in percent, relative to net capacity
↳ levelMm	number	Highest current water level in mm
↳ levelIndicator	object	Water level indicator: code, color, label (de/en), reason
↳ rise	object	Rise in mm/min and expected time to overflow
levelScale[]	array	The six indicator levels of this plant with range in mm
hydraulics	object	Design basis of the indicator (overflow height, start of “Critical”, design rain)
weather	object	Current weather and precipitation for the next hours
anchor	object	Last confirmed anchor of data anchoring on IOTA

> *Fields may be null if no valid value is available or a part has not (yet) been configured. Integrations should treat null as “not available”, not as 0.*
>
> *The number of entries in sensors[] and zones[] adjusts to the plant configuration.*

Water Level Indicator

The indicator assigns the water level to one of six levels. The thresholds are plant-specific and are delivered with every response in levelScale. The “Critical” level takes the plant’s hydraulic design and, with a sufficiently short measurement interval, the current rise rate into account.

Code	Level	Meaning
leer	Empty	No significant water in the plant
normal	Normal	Regular operation
gelb	Watch	Elevated fill level
orange	Caution	High fill level
rot	Critical	Overflow possible with further heavy rain or continued rise
purpur	Overflowing	Overflow height reached

If plant.levelIndicator is null, no current measurement is available. This means “no data”, not “empty”.

> *The indicator is information based on measured values. It is not an official warning and does not replace the plant operator’s alert level. Assessment and any resulting measures are the responsibility of the recipient.*

Weather and Data Anchoring

weather contains weather data at the plant without location (source: Open-Meteo, CC BY 4.0). Its times are local time; all other times in the API are UTC.

Measurements are continuously anchored on the IOTA Mainnet using the JB-ANCHOR-V1 method. anchor.lastConfirmed.digest refers to the corresponding IOTA transaction and can be verified in an IOTA explorer.

How Authorities Can Integrate the API

No authentication, no API key, no registration. CORS is enabled (Access-Control-Allow-Origin: *); the API can be queried directly from web applications on third-party domains.

cURL

```
curl -s https://jumboblock.net/api/public/sensors
```

JavaScript

```
fetch('https://jumboblock.net/api/public/sensors')
  .then(r => r.json())
  .then(data => {
    const p = data.plant;
    console.log('Indicator:', p.levelIndicator ? p.levelIndicator.label.en : 'no data');
    console.log('Fill level:', p.fillPercent + '%');
  });
```

Python

```
import requests
p = requests.get('https://jumboblock.net/api/public/sensors', timeout=10).json()['plant']
level = p['levelIndicator']['label']['en'] if p['levelIndicator'] else 'no data'
print(f"Indicator: {level} · Fill level: {p['fillPercent']}%")
```

Typical Use Cases

Audience	Typical use	Recommended polling interval
Municipal dispatch center	Fill level and indicator in situational map / GIS	every 30–60 seconds
Fire / emergency services	Rapid assessment during heavy rainfall	as needed / every 1–2 min
Water management authority	Monitoring & reporting	every 5–15 minutes
State situation center	Overview of multiple plants	every 1–5 minutes
Geoportal / dashboard	Map layer showing fill level and indicator	every 60 seconds

Recommendations

- Use plant.levelIndicator.code for traffic-light logic; the codes are fixed, names and colors may change.
- Retain timestamps and isStale so the freshness of the data remains visible.
- On ok: false or an HTTP status other than 200, display the last known state with a notice.
- Handle null values; do not interpret them as 0.
- Poll no more than every 30–60 seconds; new measurements arrive at the sensors' interval.
- For integration into situational awareness systems, SCADA or GIS, a periodic HTTP GET is sufficient. We are happy to provide an example script on request.