

JUMBO BLOCK Public API

Version: 1.0.0

Base URL / current environment: <https://jumboblock.net>

Full endpoint: <https://jumboblock.net/api/public/sensors>

As of: 2026-08-07

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.

Endpoint

GET /api/public/sensors

Returns the current state of all sensors, zones, and the overall plant.

Response (example)

```
{
  "ok": true,
  "ts": "2026-08-07T17:03:35.340Z",
  "sensors": [
    { "id": "jb-001", "fillM3": 8, "updatedAt": "2026-08-07T16:56:03.901Z" }
  ],
  "zones": [
    {
      "id": "G1", "label": "Zone A", "blocks": 100,
      "capacityNetM3": 1390, "capacityGrossM3": 1562.5,
      "fillNetM3": 800, "fillGrossM3": 899.28, "fillPercent": 57.6
    }
  ],
  "plant": {
    "capacityNetM3": 2780, "capacityGrossM3": 3125,
    "fillNetM3": 1600, "fillGrossM3": 1798.56, "fillPercent": 57.6
  }
}
```

Error Response (example)

If a server-side problem occurs (e.g. the plant model cannot be read), 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 API response (UTC)
sensors[]	array	Individual sensors
↳ id	string	Unique sensor ID
↳ fillM3	number	Current fill level in m ³
↳ updatedAt	string (ISO 8601)	Time of the last measurement (UTC)
zones[]	array	Zones (groups of blocks)
↳ id	string	Zone ID
↳ label	string	Display name of the zone
↳ blocks	number	Number of JUMBO BLOCKs in the zone
↳ capacityNetM3	number	Net storage capacity in m ³
↳ capacityGrossM3	number	Gross storage capacity in m ³
↳ fillNetM3	number	Current net fill level in m ³
↳ fillGrossM3	number	Current gross fill level in m ³
↳ fillPercent	number	Fill level in percent (0–100), relative to net capacity
plant	object	Overall plant (sum of all zones)
↳ capacityNetM3	number	Total net capacity in m ³
↳ capacityGrossM3	number	Total gross capacity in m ³
↳ fillNetM3	number	Total net fill level in m ³
↳ fillGrossM3	number	Total gross fill level in m ³
↳ fillPercent	number	Overall fill level in percent

Numeric fields may be null if the underlying capacity has not (yet) been configured, or if no valid sensor reading is available. Integrations should treat null as "not available", not as 0.

Rounding: m³ values in zones[] and plant are rounded to 2 decimal places, fillM3 in sensors[] to 3 decimal places, and fillPercent to 1 decimal place.

The number of entries in sensors[] and zones[] scales automatically with the plant configuration held in the Setup-Center — there is no fixed upper limit.

How Authorities Can Integrate the API

The API is deliberately kept simple and requires no authentication, no API key, and no registration. It can be queried directly by dispatch centers, municipal geoportals, situation centers, and third-party systems — including directly from web applications (see the CORS note below).

1. Simplest option – Browser or Excel

- Open the URL in a browser: <https://jumboblock.net/api/public/sensors>
- The JSON response can be copied directly or imported into Excel/Power Query.

2. Automated polling (recommended for dispatch centers)

cURL example

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

JavaScript (browser or Node.js)

```
fetch('https://jumboblock.net/api/public/sensors')
  .then(r => r.json())
  .then(data => {
    console.log('Overall fill level:', data.plant.fillPercent + '%');
    console.log('Net fill volume:', data.plant.fillNetM3 + ' m³');
  });
```

Python

```
import requests
r = requests.get('https://jumboblock.net/api/public/sensors')
data = r.json()
print(f'Overall plant fill level: {data["plant"]["fillPercent"]}%%')
```

3. Typical Use Cases

Audience	Typical use	Recommended polling interval
Municipal dispatch center	Live fill level 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	Aggregated overview of multiple plants	every 1–5 minutes
Geoportal / dashboard	Map layer showing fill level	every 60 seconds

4. Recommendations for Production Use

- Read only the plant and zones sections if individual sensor data is not needed → lower data volume.
- On ok: false or an HTTP status other than 200, display the last known state and show a notice.
- The fillPercent and fillNetM3 fields are sufficient for traffic-light logic (green / yellow / red) in most cases.
- Always retain the timestamps (ts and updatedAt) so the freshness of the data remains visible.
- CORS is enabled (Access-Control-Allow-Origin: *) — the API can be called directly from web applications on third-party domains via fetch(), without needing your own server-side proxy.
- Handle null values (see field reference) when parsing; do not interpret them as 0.

5. Further Integration

If an authority wishes to bring the data into an existing system (e.g. a situational awareness system, SCADA, or GIS), the following is generally sufficient:

- Periodic HTTP GET requests to the endpoint
- Parse the JSON
- Write the values into your own database / dashboard

If needed, we are happy to provide a small example script (Python, Node.js, or PowerShell) tailored to the relevant system landscape.

Notes

- All volume figures are given in cubic meters (m³).
- Percentage figures (fillPercent) are relative to net capacity.
- The Sensor → Zone → Plant hierarchy is fixed and will not change.
- No authentication is required (public API).
- CORS is enabled — direct access from any web application is possible.
- The number of entries in sensors[] and zones[] adjusts automatically to the plant configuration.
- Recommended polling interval: no more than every 30–60 seconds.