INTEGRATIONS

REST API and webhooks

Pull events, totals and hourly counts from the REST API, and receive every event in real time by webhook.

Updated 5 November 2026

Which to use

- Webhooks: every event, as it happens, posted to your server. The best choice for real-time work.
- Report alerts: a live occupancy, queue or crossing number posted to your server when it passes a threshold, or every minute.
- Report and event feeds: CSV or JSON addresses for spreadsheets and BI tools.
- The REST API: query locations, cameras, lines, events, totals and hourly counts when you need them.

API keys

Create keys in the portal under Settings › API keys. Each key has a public part and a secret part. Keep the secret out of code repositories and URLs, and disable the key if it may have leaked. Keys can be limited to the IP addresses you name.

Endpoints

The base address is `https://cloud.crowdsense.events/service/api/v1`. Every request is a GET and returns JSON.

- `/locations` and `/locations/status`: your computers, their app versions and time zones, and whether each location and camera is online.
- `/channels`: the cameras (feeds) on a location.
- `/triggers`: the lines and zones on each camera.
- `/events`: individual events, filtered by location, camera, trigger, type, class and time (`since`, `until`), up to 1,000 a page with `limit` and `starting_after`.

- `/totals/triggers` and `/totals/types`: event counts grouped by line or zone, or by event type.
- `/histograms`: hourly counting data per trigger.

Signing a request

Every request carries a `nonce` query parameter that must be larger than the last one you sent (a Unix timestamp works), your public key in the `X-Key` header, and in `X-Sign` an HMAC-SHA512 of the request path and query, signed with your secret. URL-encode any dates first.

```
import crypto from 'node:crypto';

const since = encodeURIComponent('2026-11-01T00:00:00Z');
const path = `/service/api/v1/totals/triggers?
nonce=${Math.floor(Date.now() / 1000)}&since=${since}`;
const sign = crypto.createHmac('sha512',
process.env.CS_SECRET).update(path).digest('hex');

const res = await fetch('https://cloud.crowdsense.events' + path, {
  headers: { 'X-Key': process.env.CS_PUBLIC_KEY, 'X-Sign': sign },
});
console.log(await res.json()); // { code: 0, msg: "success", result:
[...] }
```

> Send no more than one request a second. Faster than that returns HTTP 429 or 503.

Webhooks

In Settings › Webhooks, give an address for a single camera or for a whole location (every camera on it). Each event is posted as JSON, in a list, with much the same fields as the API: the event type, the location, camera and trigger, the object number and class (person, or vehicle type), the age band and gender on a faces camera, and the time in UTC. Your endpoint should answer within 10 seconds.

Event types in the data use system names: `TripwireCrossed` (line crossed), `Tailgating`, `RegionJoin` (zone joined), `MotionInRegionOn` (motion started), `Loitering` (dwell), `CrowdOn` (crowd appeared) and `Sabotage` (camera obstructed).

AI assistants

The portal can also give compatible AI assistants read-only access to your analytics over the Model Context Protocol (MCP). Enable it on an API key in Settings › API keys; connection details are in your portal, and support@crowdsense.events can help.

Still stuck?

Email **support@crowdsense.events** with your account reference (it starts CS-D-), the computer's name and a screenshot of what you see. For sales questions, **contact us** (crowdsense.events/contact).