EMV Gate Heartbeat – Technical Specification (EN)

EMV Gate Heartbeat – Technical Specification (EN)

1) Overview This document specifies a lightweight HTTP-based heartbeat for EMV Gate devices and the desktop monitor app (Electron + React). Devices periodically POST their status to a local HTTP server running inside the app. The app also watches a device list file and merges that with the latest heartbeat.

Default port: `3070` (configurable via `config.json` → `heartbeatPort`, if present)
Device file path: `device-communication.json` (absolute or resolved from `deviceCommunicationPath` in `config.json`).

---

2) JSON Payload ```jsonc { "id": "G1-01", // Unique device id "ip": "192.168.1.101", // Device IP "status": "online", // "online" | "offline" | "fault" | "maintenance" "ts": "2025-08-14T13:31:52Z"// ISO-8601 UTC (Z) } ```

Required fields
- **id**: string (unique ID known by the monitor app)
- **ip**: IPv4/IPv6 string
- **status**: enum: `online`, `offline`, `fault`, `maintenance`
- **ts**: ISO-8601 UTC timestamp, e.g. `2025-08-14T13:31:52Z`

> If `ts` is omitted, the server may fallback to current server time.

---

3) HTTP Endpoints

3.1 POST `/hb`
Accepts a single heartbeat event.

Request headers
```
Content-Type: application/json
```

Request body
See JSON payload above.

Response 200
```json
{ "ok": true }
```

Response 400
```json
{ "ok": false, "error": "Invalid payload" }
```

3.2 GET `/devices`
Returns the current device list with last known heartbeats merged.

Response 200
```json
{
"ok": true,
"devices": [
{
"id": "G1-01",
"name": "Entry Reader 01",
"side": "north",
"gateId": "G1",
"deviceIp": "192.168.1.101",
"status": "online",
"lastHeartbeat": "2025-08-14T13:31:52Z"
}
]
}
```

> The app also watches the `device-communication.json` file. If that file exists and is an array of device objects, fields are merged and then updated by the latest heartbeat.

---

4) Device File (`device-communication.json`) **Format:** JSON array of devices.

```
json [ { "id": "G1-01", "name": "Entry Reader 01", "side": "north", "gateId": "G1", "deviceIp": "192.168.1.101", "status": "online", "lastHeartbeat": "2025-08-14T13:31:52Z" } ]
```

If the app sees an *object* with `"devices": [...]`, it will fallback to the inner array; otherwise it expects a top-level array. Always prefer the **array** format.

---

5) Effective Status Logic The UI shows **effective status** (derived from heartbeat + TCP probe):

- **Input**
- Heartbeat status → `online | offline | stale` (stale = no heartbeat for `staleMs`, offline = no heartbeat for `offlineMs`).
- TCP probe (`probeTcp(host, port, timeout)`): `reachable: true|false` on port `22` by default.
- **Decision (recommended “A” mapping)**
- If device status is `maintenance` → **maintenance**.
- If heartbeat is `offline` → probe reachable? **fault** : **offline**.
- If heartbeat is `stale` → probe reachable? **online** : **offline**.
- If heartbeat is `online` → probe not reachable? **fault** : **online**.

Defaults: `staleMs=60000`, `offlineMs=300000`, `tcpPort=22`, `timeoutMs=1200`.

---

6) Testing

6.1 cURL
```bash
# Send one heartbeat (local dev server)
```

curl -sS -X POST http://127.0.0.1:3070/hb -H 'Content-Type: application/json' -d '{"id":"G1-01","ip":"192.168.1.101","status":"online","ts":"2025-08-14T13:31:52Z"}'

Query merged devices curl -sS http://127.0.0.1:3070/devices | jq ```

6.2 TCP checks (manual)
```bash
# TCP reachability
nc -vz -w 2 192.168.1.101 22 ; echo $?
# SSH attempt (expect auth error but proves reachability)
ssh -o ConnectTimeout=2 -o BatchMode=yes -p 22 user@192.168.1.101 exit
```

6.3 Postman
1. Import the provided **Collection** and **Environment** files.
2. Set `{{baseUrl}}` to `http://127.0.0.1:3070` or your server IP.
3. Run `HB: POST /hb` then `GET /devices` to verify merge.

---

7) Error Handling - Invalid JSON → `400 { "ok": false, "error": "Invalid payload" }` - Missing fields → `400 { "ok": false, "error": "Missing id/ip/status/ts" }` - Internal errors → `500 { "ok": false, "error": "" }`

---

8) Security Notes - The heartbeat server is intended for **trusted LAN** only. - If needed, implement IP allowlist or a shared secret in a header. - Consider rotating logs and not storing credentials in plain text.

---

9) Appendix — Sample `config.json` ```json { "environment": "development", "logsPath": "./logs", "logsRetentionDays": 14, "logLevel": "info", "deviceCommunicationPath": "./data/device-communication.json", "heartbeatPort": 3070, "fullScreen": false } ```