

Config Server API

 最後更新:2026-05-24 |  對應版本:v2.2.10 |  負責人:KC

 雲端: `https://opta.smms.com.tw` (`SERVER_ROLE=cloud`) | 地端: `http://<SERVER_IP>:8888`
(`SERVER_ROLE=local`)

本文件涵蓋 MES I/O Gateway Config Server 的全部 REST API。Server 採 **單一 codebase + 角色模式**；以

`SERVER_ROLE` 環境變數切換：

- `local`：地端備份 / 還原伺服器 (檔案系統 + SQLite，無 PG、無 license、無 Cloud Bridge)
- `cloud`：雲端授權 + OTA 派送 (**PostgreSQL 必要**，啟用 license 模組與 Cloud Bridge)

標  的端點僅 `SERVER_ROLE=cloud` 時可用。

01

認證機制

`middleware/auth.js` 採三軌制：

順序	Token 型態	來源	注入 <code>req.user</code>
1	JWT	<code>POST /api/auth/login</code> 簽發 (12h 有效)	<code>{ id, email, role, displayName }</code>
2	Legacy DEVICE_TOKEN	環境變數 <code>DEVICE_TOKEN</code> (預設 <code>admin-token</code>)	<code>{ role: 'device', id: 0 }</code>
3	License Token	雲端 license 模組簽發給已啟用 device	<code>{ role: 'device', uid }</code>

呼叫方式：

```
Authorization: Bearer <token>
```

未帶 Bearer header → `401 Unauthorized {"error": "Unauthorized", "message": "Missing Authorization header"}`。失敗事件會打 `auth.reject` log，使用者回報 401/403 時用 `grep auth.reject` 立刻看到原因。

角色 (role)

Role	來源	權限
admin	DB user role	所有 /api/admin/*、user/license/firmware 管理
local_admin	DB user role	等同 admin 但限地端
customer	DB user role	Tenant-scoped: /api/devices 只看自己擁有的
device	Legacy / License Token	Device 端 heartbeat、config sync、OTA fetch

adminOnly middleware 額外把關到 role ∈ {admin, local_admin}。

端點總覽

共用 API (local + cloud)

#	Method	Path	認證	呼叫方	用途
0	GET	/api/health	Public	All	健康檢查 (含 role / version / device 數)
1	POST	/api/auth/login	Public	User	登入取得 JWT
2	GET	/api/auth/me	JWT	User	查目前登入者
3	POST	/api/auth/password	JWT (非 device)	User	自助改密碼
4	POST	/api/devices/claim/generate	JWT	User Dashboard	產生 6 位數 OTP 綁定 device
5	POST	/api/devices/claim	Public	Device	Device 用 OTP 完成綁定
6	POST	/api/devices/:id/heartbeat	Optional	Device	Device 心跳 (含 firmware/license/OTA 回路)
7	GET	/api/devices	Required	Dashboard	列出所有設備 (tenant filter)
8	DELETE	/api/devices/:id	Required	Dashboard	刪除設備 (記憶體)
9	POST	/api/admin/devices/cleanup-offline	Admin	Dashboard	批次刪除離線設備
10	GET	/api/devices/:id/config	Required	Device	下載設備設定
11	POST	/api/devices/:id/config	Required	Device	上傳設備設定 (自動備份 + snapshot)
12	GET	/api/devices/:id/backups	Required	Dashboard	Backup 清單 (含 diff summary)
13	GET	/api/devices/:id/backups/:filename	Required	Dashboard	取單一 backup 全文
14	GET	/api/devices/:id/diff	Required	Dashboard	Backup vs current diff
15	POST	/api/devices/:id/restore	Required	Dashboard	還原至 backup
16	GET	/api/devices/:id/snapshots	Required	Dashboard	自動 snapshot 列表 (最近 50 / 30 天)
17	GET	/api/devices/:id/snapshots/:filename	Required	Dashboard	取單一 snapshot

#	Method	Path	認證	呼叫方	用途
18	POST	/api/devices/:id/snapshots/:filename/rollback	Required	Dashboard	Rollback 到 snapshot
19	POST	/api/devices/:id/web	Required	Device	上傳 web bundle (CRC 比對省流量)
20	GET	/api/devices/:id/web/crc	Required	Device	查 web bundle CRC
21	GET	/api/firmware/latest	Public	Device	依 channel 查最新韌體 (auto-update)
22	GET	/api/firmware/download/:filename	Public	Device	下載韌體 .bin
23	POST	/api/firmware/upload	Admin	Dashboard	上傳韌體
24	GET	/api/firmware	Admin	Dashboard	列出韌體 (含每版本在 線 device 數)
25	GET	/api/firmware/:filename/devices	Admin	Dashboard	看哪些 device 在跑這 版
26	GET	/api/firmware/:filename/channel-history	Admin	Dashboard	Promote/demote 歷 史
27	POST	/api/admin/firmware/:filename/channel	Admin	Dashboard	改 channel (dev/beta/stable) — nginx 不擋
28	PATCH	/api/admin/firmware/:filename/channel	Admin	Dashboard	同上
29	DELETE	/api/admin/firmware/:filename	Admin	Dashboard	刪韌體 (擋 stable, 可 ?force=1)
30	POST	/api/firmware/deploy/:deviceId	Admin	Dashboard	Queue OTA 給單一 device
31	POST	/api/devices/:id/ota (deprecated)	Optional	-	舊 .mesb 上傳 → 一律 410 Gone
32	GET	/api/devices/:id/ota/latest (deprecated)	Optional	-	舊 .mesb 查詢 → 一律 410 Gone
33	GET	/api/customers	Required	Dashboard	客戶列表 (含 device/online 計數)
34	GET	/api/customers/:id	Required	Dashboard	客戶詳細
35	GET	/api/customers/:id/devices	Required	Dashboard	該客戶旗下設備
36	POST	/api/customers	Required	Dashboard	建立客戶
37	PATCH	/api/customers/:id	Required	Dashboard	修改客戶

#	Method	Path	認證	呼叫方	用途
38	DELETE	/api/customers/:id	Required	Dashboard	刪除客戶 (device 改未指派而非刪除)
39	POST	/api/devices/:uid/assign-customer	Required	Dashboard	指派 / 解除 device 對客戶歸屬
40	GET	/api/admin/users	Admin	Dashboard	使用者列表
41	POST	/api/admin/users	Admin	Dashboard	新增使用者
42	PUT	/api/admin/users/:id	Admin	Dashboard	修改使用者
43	PUT	/api/admin/users/:id/password	Admin	Dashboard	Admin 重置別人密碼
44	GET	/api/audit-logs	Admin	Dashboard	稽核日誌 (分頁)
45	GET	/api/admin/backup-status	Admin	Dashboard	pg_dump 健康度 (v2.2.6+)
46	GET	/api/cloud/status	Public	Dashboard	Cloud Bridge 狀態 (local 模式回 disabled)

Cloud-only

僅當 `SERVER_ROLE=cloud` 時掛載 (`server.js:302-309`) :

#	Method	Path	認證	說明
C1	POST	/api/license/activate	Public	Device 申請啟用 (whitelist 直接 active)
C2	GET	/api/license/status/:uid	Public	Device 查狀態
C3	GET	/api/admin/license/list	Admin	全部 license (含統計)
C4	POST	/api/admin/license/import	Admin	批次匯入 UID 白名單
C5	POST	/api/admin/license/approve/:uid	Admin	審核通過 (pending → active)
C6	POST	/api/admin/license/reject/:uid	Admin	拒絕 (device 可重新申請回 pending)
C7	POST	/api/admin/license/deactivate/:uid	Admin	強制停權 (active → rejected, 無 grace period)
C8	DELETE	/api/admin/license/:uid	Admin	硬刪除 (寫 tombstone, 90 天不可重新 activate)
C9	DELETE	/api/admin/license-revoked/:uid	Admin	清掉 tombstone (讓被刪 UID 可重新 activate)

03

0. 健康檢查

```
GET /api/health
```

200 OK :

```
{
  "status": "ok",
  "role": "cloud",
  "version": "2.2.10",
  "uptime": 3600,
  "timestamp": "2026-05-24T06:00:00.000Z",
  "deviceCount": 12
}
```

JSON

```
curl https://opta.smms.com.tw/api/health
```

BASH

04

1. Auth

POST /api/auth/login

```
{ "email": "kwei.chen@gmail.com", "password": "..."} 
```

JSON

200 OK :

```
{
  "ok": true,
  "token": "eyJhbGciOi...",
  "user": { "id": 1, "email": "kwei.chen@gmail.com", "role": "admin", "displayName": "KC" }
}
```

JSON

401 : {"error": "Unauthorized", "message": "invalid credentials"}

JWT TTL 12 小時，逾期後須重新 login。

GET /api/auth/me

回目前 JWT 對應的 user 物件。供前端 reload 後重新水合 user state。

POST /api/auth/password

```
{ "currentPassword": "old", "newPassword": "new12345" }
```

JSON

newPassword 至少 8 字元。Device token 一律 **403** (不准改 **device** user)。

05

2. Device Claim

OTP-based 雙向綁定 (Sprint 21 WI-070)：

POST /api/devices/claim/generate (user 端)

需 JWT。產生 6 位數 OTP，10 分鐘有效。

200 OK：

```
{
  "ok": true,
  "claimToken": "493217",
  "expiresAt": "2026-05-24T06:10:00.000Z",
  "instruction": "在設備頁面輸入此 6 位數驗證碼以完成綁定"
}
```

JSON

POST /api/devices/claim (device 端)

```
{ "deviceId": "opta-AA-BB-CC", "claimToken": "493217" }
```

JSON

200 OK：{"ok": true, "message": "Device claimed", "deviceId": "...", "userId": 1}

410：OTP 失效

06

3. Heartbeat

POST /api/devices/:id/heartbeat

最熱門 endpoint。Device 約 30 秒一次回報；server 回 OTA / license verdict / 待派送 web upload。

掛載於 auth middleware 之前 → token 為 optional (若帶會解，缺也不擋)。

Request body:

```
{
  "deviceName": "opta-line-3",
  "ip": "192.168.0.100",
  "uptime": 123456,
  "configVersion": 17,
  "uid": "AA:BB:CC:DD:EE:FF",
  "firmwareVersion": "5.9.130",
  "firmwareVariant": "standard",
  "otaProgress": null
}
```

JSON

Response 200 OK:

```
{
  "status": "ok",
  "license": { "status": "active", "licenseId": "LIC-2026-..." },
  "ota": {
    "url": "https://opta.smms.com.tw/api/firmware/download/opta_v5.9.131.bin",
    "version": "5.9.131",
    "force": false
  }
}
```

JSON

可能的 status:

- ok — 一切正常
- ota_available — 有更新可拉
- download_update — 後端要求立刻拉 config + web (例如 restore / rollback 完成)

Side effects:

1. Persist 至 PG devices 表(uid PK) (cloud 模式;v2.2.10 dff2c2d)

2. 若 configVersion 變化 → 觸發 auto-snapshot (source= auto-heartbeat)

3. 派送 pending OTA / web upload 給 device
4. 呼叫 `cloud-bridge.onHeartbeat` (Arduino Cloud 同步)
5. License upsert: 未見過的 UID 會建 `unseen` 紀錄

07

4. Devices

GET /api/devices

Auth required。Cloud 模式回傳 PG `devices` 表 + in-memory map 的 union (uid 去重) ; Local 模式 fallback 只看 in-memory。

`customer` role 會被 tenant filter (只看自己擁有的 device IDs)。

```
[
  {
    "id": "opta-AA-BB-CC",
    "uid": "AA:BB:CC:DD:EE:FF",
    "deviceName": "opta-line-3",
    "ip": "192.168.0.100",
    "firmwareVersion": "5.9.130",
    "firmwareVariant": "standard",
    "lastSeen": "2026-05-24T05:59:30.000Z",
    "uptime": 123456,
    "configVersion": 17,
    "otaProgress": null,
    "pendingCommand": null,
    "registered": true,
    "status": "online",
    "customerId": 5,
    "customerName": "Foxconn Tier-1"
  }
]
```

JSON

v2.2.10 (commit dff2c2d): 原本 `/api/devices` 用 in-memory `deviceId` 為 key, 導致同樣 `deviceName` 但不同 UID 的 device 互蓋。改用 PG 的 `uid` PK 後, 每顆 Opta 都有唯一 row; 舊資料沒 uid 者標 `registered: false`。

DELETE /api/devices/:id

刪除 in-memory map 上的 device (不會動 PG row)。

POST /api/admin/devices/cleanup-offline

```
{ "olderThanSec": 86400 }
```

JSON

或 `?olderThanSec=86400` query。預設 86400 (24h)。

200 OK :

```
{ "success": true, "count": 3, "devices": [...], "olderThanSec": 86400 }
```

JSON

08

5. Device Config

GET /api/devices/:id/config

回 device 的最新 config JSON (剝掉 `crc32` 欄位)。

POST /api/devices/:id/config

```
{ "deviceName": "opta-line-3", "mqtt": { ... }, ... }
```

JSON

Side effects:

1. 寫 `backups/config.bak.{ts}.json` 並 prune 7 天以上舊檔
2. 觸發 `saveSnapshotIfChanged` (source= `auto-push` ;v2.2.3+)
3. 取消任何 pending `download_update` command

200 OK : {"success": true, "message": "Config updated"}

09

6. Backups & Restore

GET /api/devices/:id/backups

```
[
  {
    "filename": "config.bak.1748044800.json",
    "timestamp": 1748044800,
    "date": "2026-05-24T00:00:00.000Z",
    "sizeBytes": 8421,
    "summary": { "added": 1, "removed": 0, "changed": 3 }
  }
]
```

JSON

GET /api/devices/:id/backups/:filename

`filename` 需符合 `config.bak.\d+.json`，否則 `400 Bad Request`。回完整 backup JSON。

GET /api/devices/:id/diff?file=config.bak.{ts}.json

```
{
  "added": { "mqtt.useTls": true },
  "removed": { "mqtt.legacyField": "..." },
  "changed": { "mqtt.broker": { "from": "...", "to": "..." } }
}
```

JSON

POST /api/devices/:id/restore

```
{ "filename": "config.bak.1748044800.json" }
```

JSON

設置 `pendingCommand=download_update` + `pendingWebUpload=true`，下次 heartbeat device 會收到指令並主動拉新 config。

10

7. Snapshots（自動觸發）

不同於 manual backup，snapshot 由系統在 `configVersion` 變化時自動產生。`backups/snapshot.{ts}.v{version}.json`。最多保留 50 個 / 30 天。

GET /api/devices/:id/snapshots?limit=20

```
[
  {
    "filename": "snapshot.1748044800.v17.json",
    "timestamp": 1748044800,
    "date": "2026-05-24T00:00:00.000Z",
    "configVersion": 17,
    "sizeBytes": 8421,
    "source": "auto-heartbeat",
    "deviceName": "opta-line-3",
    "summary": { "added": 1, "removed": 0, "changed": 3 }
  }
]
```

JSON

source 可能值:

- **auto-push** — 由 **POST /api/devices/:id/config** 觸發
- **auto-heartbeat** — heartbeat 偵測 **configVersion** 變化觸發

GET /api/devices/:id/snapshots/:filename

回完整 snapshot:

```
{
  "_snapshot": { "timestamp": 1748044800, "configVersion": 17, "deviceName": "...", "source": "auto-pu
  "deviceName": "...",
  "mqtt": { ... }
}
```

JSON

POST /api/devices/:id/snapshots/:filename/rollback

設 **pendingCommand=download_update** 並把 snapshot 內容寫回當前 config。

```
{ "success": true, "message": "Rolled back to snapshot", "newConfigVersion": 18 }
```

JSON

8. Web Upload

Device firmware 把 inline 的 `WebPage.h` 算 CRC32 上傳; server 比對 CRC, 若無變更則跳過寫檔省流量。

POST /api/devices/:id/web

Raw binary body (`Content-Type: application/octet-stream` , 最大 2 MB)。

200 OK :

```
{ "success": true, "bytes": 23456, "crc": 3735928559 }
```

JSON

未變更:

```
{ "success": true, "unchanged": true, "crc": 3735928559 }
```

JSON

GET /api/devices/:id/web/crc

```
{ "crc": 3735928559 }
```

JSON

無紀錄回 `{"crc": 0}`。

12

9. Firmware Management

Public — Device 用

GET /api/firmware/latest?channel=stable

`channel` 可選 `stable` / `beta` / `dev` / `all` , 預設 `stable` 。

JSON

```
{
  "filename": "opta_v5.9.131.bin",
  "version": "5.9.131",
  "variant": "standard",
  "releaseChannel": "stable",
  "size": 781234,
  "sha256": "ab12...",
  "changelog": "- Fix MQTTS handshake\n- Add ...",
  "downloadUrl": "/api/firmware/download/opta_v5.9.131.bin"
}
```

無符合: {"error": "No firmware in channel", "hint": "Try ?channel=all"}

GET /api/firmware/download/:filename

回 `application/octet-stream`。 `filename` 必須是 `.bin`，否則 400。

Admin — Dashboard 用

POST /api/firmware/upload

Raw `.bin` body (最大 4 MB) ◦ Headers:

Header	必要	說明
<code>X-Firmware-Version</code>	✓	<code>M.m.p</code> 格式 (如 <code>5.9.131</code>)
<code>X-Firmware-Changelog-B64</code> 或 <code>X-Firmware-Changelog</code>	✓	≥ 10 字元 (純文字者直接放, URL-safe 者放 base64)
<code>X-Firmware-Channel</code>		<code>dev</code> / <code>beta</code> / <code>stable</code> (預設 <code>dev</code>)
<code>X-Firmware-No-License</code>		<code>1</code> = no-license variant (變體欄位)

200 OK:

JSON

```
{
  "ok": true,
  "filename": "opta_v5.9.131.bin",
  "size": 781234,
  "sha256": "ab12...",
  "releaseChannel": "dev",
  "changelogLength": 142
}
```

GET /api/firmware?channel=stable&variant=all

回 channel 下韌體列表, 附「目前在線跑這版的 device 數」統計:

```
[
  {
    "filename": "opta_v5.9.131.bin",
    "version": "5.9.131",
    "variant": "standard",
    "releaseChannel": "stable",
    "size": 781234,
    "sha256": "ab12...",
    "devices": { "total": 8, "online": 5, "byVersion": { "5.9.131": 3, "5.9.130": 2 } }
  }
]
```

GET /api/firmware/:filename/devices

GET /api/firmware/:filename/channel-history

回 promote/demote 歷史, 每筆含 { timestamp, fromChannel, toChannel, changedBy, reason }。

POST /api/admin/firmware/:filename/channel (或 **PATCH**)

```
{ "channel": "stable", "reason": "Passed regression suite 2026-05-24" }
```

雙寫 POST + PATCH 是因為前端 nginx CDN 預設擋 PATCH。後端兩個 verb 都接到同一 handler。

200 OK :

```
{ "ok": true, "filename": "opta_v5.9.131.bin", "releaseChannel": "stable", "previousChannel": "beta" }
```

DELETE /api/admin/firmware/:filename?force=1

- 若該 firmware 在 **stable** channel → **409 Conflict** (必須先 demote)
- 若有在線 device 跑這版 → **409 Conflict**, 加 **?force=1** 強制
- 若有 in-flight OTA → **409 Conflict**

POST /api/firmware/deploy/:deviceId

```
{ "filename": "opta_v5.9.131.bin" }
```

Query **?force=1**: device 收到 heartbeat 回應後立刻自動執行 (不等使用者按 update)。

JSON

```
{ "ok": true, "message": "OTA queued", "targetVersion": "5.9.131", "targetDevice": "opta-line-3", "for
```

13

10. OTA（已廢棄 `.mesb` 格式）

```
POST /api/devices/:id/ota
GET /api/devices/:id/ota/latest
```

兩個一律回 `410 Gone`，body 內附遷移指引：

JSON

```
{
  "error": "Gone",
  "message": "The .mesb OTA API has been removed. Use /api/firmware/upload and /api/firmware/deploy/:c
  "migration": "https://docs.smms.com.tw/migration/mesb-to-firmware-v2"
}
```

14

11. Customers（多租戶）

GET /api/customers?tier=...&active=true

```
[
  {
    "id": 5,
    "name": "Foxconn Tier-1",
    "short_code": "FXN",
    "tier": "enterprise",
    "email": "...",
    "contactPerson": "...",
    "status": "active",
    "createdAt": "2026-04-01T00:00:00.000Z",
    "deviceCount": 12,
    "onlineCount": 8
  }
]
```

JSON

POST /api/customers

```
{ "name": "...", "short_code": "...", "tier": "smb", "email": "...", "contactPerson": "...", "status": ... }
```

JSON

201 Created ◦

PATCH /api/customers/:id

部分更新◦

DELETE /api/customers/:id

不會刪 device，只把該 customer 旗下 device 改為「未指派」。

POST /api/devices/:uid/assign-customer

指派：

```
{ "customerId": 5 }
```

JSON

解除指派：

```
{ "customerId": null }
```

JSON

Find-or-create (沒有對應 customer 自動建一個)：

```
{ "customerName": "新客戶 X" }
```

JSON

15

12. Admin Users (Sprint 21)

GET /api/admin/users

```
[  
  { "id": 1, "email": "kc@...", "role": "admin", "displayName": "KC", "is_active": true }  
]
```

JSON

POST /api/admin/users

```
{ "email": "ops@example.com", "password": "min8chars", "role": "customer", "displayName": "Ops" }
```

JSON

PUT /api/admin/users/:id/password

```
{ "newPassword": "min8chars" }
```

JSON

PUT /api/admin/users/:id

```
{ "role": "customer", "displayName": "...", "isActive": false }
```

JSON

16

13. Audit Logs

GET /api/audit-logs?limit=50&offset=0

limit 最大 200°

JSON

```
{
  "logs": [
    {
      "timestamp": "2026-05-24T06:00:00.000Z",
      "action": "license.approve",
      "userId": 1,
      "objectId": "AA:BB:CC:DD:EE:FF",
      "details": { "previousStatus": "pending", "newStatus": "active" },
      "ip": "1.2.3.4"
    }
  ],
  "total": 4123,
  "limit": 50,
  "offset": 0
}
```

17

14. PG Backup Status (v2.2.6+)

GET /api/admin/backup-status

讀 **/var/backups/pg** mount 點，回傳每日 **pg_dump** 健康度：

JSON

```
{
  "mounted": true,
  "dumpDir": "/var/backups/pg",
  "dumpCount": 14,
  "lastDumpFile": "pg_dump_2026-05-23.sql.gz",
  "lastDumpAt": "2026-05-23T02:00:00.000Z",
  "hoursSinceLastDump": 28,
  "oldestDumpAt": "2026-05-10T02:00:00.000Z",
  "totalSize": 1583492000,
  "healthy": false
}
```

healthy: false 表示最後一份 dump 距今 > 26 小時——需要排查 cronjob / mount。

18

15. Cloud Bridge

GET /api/cloud/status

Local 模式：

JSON

```
{ "enabled": false, "reason": "SERVER_ROLE=local" }
```

Cloud 模式：

JSON

```
{
  "enabled": true,
  "started": "2026-05-20T14:00:00.000Z",
  "mqtt": { "connected": true, "host": "...", "port": 1883 },
  "arduinoCloud": { "connected": true, "thingsActive": 12 },
  "devices": 12
}
```

19

☁ License API (Cloud-only)

License 採七段狀態機，由 device-side activate / heartbeat 與 admin-side 審核共同推進：

State	進入條件	退出條件
<code>unseen</code>	Heartbeat 看到新 UID 自動建立	Device 顯式 <code>activate</code> → <code>pending</code>
<code>pending</code>	Device <code>activate</code> 或 admin 把 unseen 提升	Admin <code>approve</code> → <code>active</code> ; <code>reject</code> 維持 pending
<code>whitelisted</code>	Admin <code>import</code> 批次預核	Device <code>activate</code> 直接到 <code>active</code>
<code>active</code>	<code>approve</code> 或 whitelist + activate	Admin <code>deactivate</code> → <code>rejected</code>
<code>rejected</code>	Admin <code>reject</code> / <code>deactivate</code>	Device 重新 <code>activate</code> → <code>pending</code>
<code>revoked</code>	Admin <code>DELETE</code> (90 天 tombstone)	Admin <code>DELETE /api/admin/license-revoked</code> 清除

POST /api/license/activate

```
{ "uid": "AA:BB:CC:DD:EE:FF", "deviceName": "opta-line-3", "firmwareVersion": "5.9.130", "agreed": true }
```

200 OK (whitelisted) :

```
{ "status": "active", "message": "License activated", "licenseId": "LIC-2026-001", "activatedAt": "2026-01-01T00:00:00Z" }
```

200 OK (未預核) :

```
{ "status": "pending", "message": "License pending admin approval" }
```

403 (revoked tombstone 仍有效) :

```
{ "status": "rejected", "message": "UID revoked. Contact admin." }
```

GET /api/license/status/:uid

```
{
  "uid": "AA:BB:CC:DD:EE:FF",
  "status": "active",
  "activatedAt": "2026-05-24T06:00:00.000Z",
  "deviceName": "opta-line-3",
  "expiresAt": null
}
```

JSON

Admin

GET /api/admin/license/list?status=pending

```
{
  "stats": { "total": 142, "active": 98, "pending": 12, "whitelisted": 20, "rejected": 8, "unseen": 4 },
  "licenses": [ { "uid": "...", "status": "pending", "deviceName": "...", "firstSeenAt": "..." } ],
  "revoked": [ { "uid": "...", "revokedAt": "..." } ]
}
```

JSON

POST /api/admin/license/import

```
{ "uids": ["AA:BB:CC:DD:EE:01", "AA:BB:CC:DD:EE:02"] }
```

JSON

{ "success": true, "imported": 1, "skipped": 1, "total": 2 } — pending 的會升級 whitelisted; 已 active 的 skip。

POST /api/admin/license/approve/:uid

Pending → active, 發出 `licenseId`。

POST /api/admin/license/reject/:uid

任意狀態 → rejected。Device 可再次 activate 推回 pending——這跟 deactivate 不同。

POST /api/admin/license/deactivate/:uid

Active → rejected, 無 **grace period**——下次 heartbeat 立刻通知 device 失權。

DELETE /api/admin/license/:uid

硬刪 license row + 寫 90 天 tombstone。Device 之後 activate 會直接拒絕。

```
{
  "success": true,
  "deleted": { "uid": "...", "status": "active" },
  "tombstoneTtlDays": 90,
  "hint": "To allow re-activation, call DELETE /api/admin/license-revoked/:uid"
}
```

DELETE /api/admin/license-revoked/:uid

清掉 tombstone (讓被刪 UID 可再 activate)。

20

HTTP 狀態碼

Code	意義
200	成功
201	Created (POST <code>/api/customers</code> 、 <code>/api/admin/users</code>)
400	Bad Request (validation 失敗、filename 格式錯)
401	Unauthorized (缺 Bearer / token 失效)
403	Forbidden (role 不足; rejected/revoked license)
404	Not Found
409	Conflict (韌體在用、stable channel 不可刪)
410	Gone (<code>.mesb</code> API 已廢; claim OTP 過期)
500	Server error

統一錯誤 shape:

```
{ "error": "Unauthorized", "message": "Missing Authorization header" }
```

21

Body Size Limits

Endpoint	Limit	設定處
<code>express.json()</code> 預設	1 MB	<code>server.js:37</code>
<code>/api/devices/:id/web</code>	2 MB	<code>server.js:39</code>
<code>/api/devices/:id/ota</code>	2 MB	<code>server.js:41</code> (廢)
<code>/api/firmware/upload</code>	4 MB	<code>server.js:43</code>

22

CORS

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Allow-Headers: Content-Type, Authorization
```

OPTIONS 一律回 `200`。

23

Legacy / Migration

Method	Path	行為
<code>GET</code>	<code>/config</code>	<code>400 - Migration: Use /api/devices/:id/config instead.</code>
<code>POST</code>	<code>/config</code>	同上
<code>PUT</code>	<code>/config</code>	同上

24

近期變更 (v2.2.x highlights)

Version	變更
v2.2.10 (dff2c2d)	<code>/api/devices</code> 改用 PG (<code>uid</code> PK) 列設備, 修同 <code>deviceId</code> 互蓋 bug。
v2.2.6	新增 <code>/api/admin/backup-status</code> , 監控 daily <code>pg_dump</code> 。
v2.2.4	Self-serve <code>POST /api/auth/password</code> + admin <code>PUT /api/admin/users/:id/password</code> 。
v2.2.3	<code>POST /api/devices/:id/config</code> 觸發 <code>auto-push</code> snapshot, 補上 heartbeat-only 走過的縫。
Sprint 21	WI-065 JWT auth、WI-070 device claim OTP、WI-071 user CRUD。
ADR-008	單一 codebase + <code>SERVER_ROLE</code> 切換 local/cloud; license 模組僅 cloud 載入。
WI-077	MQTT credentials 在 env 改用 base64-encoded default, 避免明文密碼進 git。

25

實作備註

1. **Auth 順序**: JWT → DEVICE_TOKEN → License Token。任一吻合就 `next()`, 全失敗才回 401。失敗事件打 `auth.reject` log。
2. **Mount 順序很關鍵**: `server.js:103-117` 先掛 public router (auth/firmware-public/deviceClaim/heartbeat), 再 `app.use('/api/devices', authMiddleware)` 等於把 `/api/devices/*` 全段保護起來——但 heartbeat 已先掛載, 所以它仍是 optional auth。
3. **Tenant filter**: `customer` role 在 `GET /api/devices` handler 內查 user → device 對應關係後過濾結果。Admin role 看全部。
4. **Heartbeat 是 cloud sync 的主軸**: device → server 推狀態、server → device 回授權與 OTA 指令; 任何 admin 動作 (approve / deploy / restore) 都是設 pending flag, 等下一次 heartbeat 才生效。
5. **Snapshot vs Backup**: Backup 是手動觸發 `POST /api/devices/:id/config` 寫的 timestamped 副本 (7 天輪替); Snapshot 是系統在 configVersion 改變時自動寫的 (50 個 / 30 天)。Rollback 走 snapshot。
5. **Firmware 刪除安全網**: Stable channel 必須先 demote 才能刪; 有 in-flight OTA 或在線 device 跑該版時要 `force=1` 才放行。
7. **License revoked tombstone**: 硬刪後 90 天內 device activate 會被拒 (避免使用者隨意拔 PG row 後 device 自動復活); 要復活先 `DELETE /api/admin/license-revoked/:uid`。
3. **PATCH 雙寫 POST**: `/api/admin/firmware/:filename/channel` 同時掛 POST 跟 PATCH——前端 nginx CDN 預設擋 PATCH, POST 是 escape hatch。