

04 開發者 / 整合

裝置 REST API

17 最後更新:2026-06-16 | 對應韌體:v5.9.249 | 負責人:KC

Opta (MES Gateway) 韌體在 TCP 埠 80 上以 HTTP/1.1 提供約 50+ 個 REST endpoint。所有路由都是在 `src/main.cpp` 的網路 loop 內以 `firstLine.indexOf(...)` 字串比對方式分派; handler 實作分佈於 `ApiHandlers_Core.cpp`、`ApiHandlers_Connectivity.cpp`、`ApiHandlers_Modbus.cpp`、`ApiHandlers_Logic.cpp`、`OtaHandler.cpp`。

v5.1 起為 **Unified Network Architecture**——同一份韌體同時支援乙太網路與 WiFi。v5.6 整合 TCP IO Channel 與非同步 RS485 掃描。v5.9.125+ 新增 MQTTs (`useTls` / `caCert`) 支援。

01

認證機制

韌體採用 Token-based RBAC。Token 由 `POST /api/tokens` 建立，可透過下列任一方式攜帶：

- HTTP Header: `Authorization: Bearer <token>`
- Query 參數: `?token=<token>`

僅 `/api/system` 在 dispatcher 層強制檢查 (v5.9 之後因 login modal 用此 endpoint 驗證 token)，其餘 endpoint 由各自 handler 內呼叫 `checkAuth(req, PERM_*)` 把關，未通過則回 401 Unauthorized JSON：

```
{"error": "Unauthorized"}
```

JSON

權限等級

等級	數值	可做的事
PERM_READONLY	0	只讀的狀態查詢 (IO、tokens 列表、config server)。
PERM_CONTROL	1	現場控制: DO 切換、Modbus write、converter 預覽。
PERM_ADMIN	2	設定變更: config、rules、tokens、license、OTA、factory reset。

- 表示 dispatcher 直接放行 (如 /api/io、/api/config GET、/、/favicon.ico 等)。

02

Endpoint 總覽

系統 / 診斷

Endpoint	Method	Permission	說明
/	GET	-	主控台 HTML (QSPI 或內嵌 fallback)
/favicon.ico	GET	-	favicon
/api/system	GET	READONLY	系統健康狀態 (RAM/Flash/uptime/MAC/license)
/api/system/reboot	POST	ADMIN	延遲 reboot (先回 200 再 reset)
/api/system/factory-reset	POST	ADMIN	Factory reset (清除設定)
/api/poll	GET	-	合併 system / io / virtual / stats 的 bulk JSON
/api/config/integrity	GET	READONLY	Config checksum / 完整性驗證

設定

Endpoint	Method	Permission	說明
/api/config	GET	-	取得完整節點設定
/api/config	POST	ADMIN	更新節點設定 (partial update)
/api/io-name	POST	ADMIN	命名單一 IO 通道
/api/time/sync	POST	ADMIN	從瀏覽器同步 RTC (WI-079)

I/O

Endpoint	Method	Permission	說明
/api/io	GET	-	DI / DO / AI / 擴充模組即時狀態
/api/io/do	POST	CONTROL	切換 DO (單顆或多顆)
/api/expansions	GET	-	擴充模組列表 (型號、IO 數)
/api/expansions/name	POST	ADMIN	命名擴充模組
/api/analytics	GET	-	Sparkline 用 ring buffer 樣本

規則 / 訊號 / 動作 / 虛擬通道

Endpoint	Method	Permission	說明
/api/rules	GET	-	列出所有 trigger rule
/api/rules	POST	ADMIN	新增 / 修改 / 刪除 rule (依 body)
/api/rules/clear	POST	ADMIN	清空所有 rule (WI-103 FR-H1-02)
/api/signals	GET / POST	ADMIN *	Sensor Group (📡) 定義
/api/actions	GET / POST	ADMIN *	Actuator Group (⚙️) 序列
/api/virtual	GET	-	虛擬通道狀態
/api/virtual	POST	ADMIN	更新虛擬通道
/api/history	GET	-	最近 100 筆事件

*GET 不檢查 auth, POST 需 ADMIN。

Modbus / RS485

Endpoint	Method	Permission	說明
/api/modbus	GET	-	列出 Modbus 裝置與暫存器
/api/modbus	POST	ADMIN	批次更新 Modbus 設定
/api/modbus-devices	POST	ADMIN	新增 / 更新單顆 RS485 device
/api/modbus-devices	DELETE	ADMIN	刪除 RS485 device
/api/modbus/scan	GET / POST / DELETE	ADMIN	RS485 非同步掃描 (POST 啟動、GET 輪詢、DELETE 取消)
/api/modbus/write	POST	CONTROL	手動寫 Modbus 暫存器 (FC06 / FC16)
/api/modbus/lock	POST	ADMIN	鎖匯流排，阻止 polling (用於手動 IO)
/api/modbus/probe	POST	CONTROL	Probe 單一裝置
/api/modbus/sync-baud	POST	ADMIN	跟 slave 同步鮑率
/api/rs485/{busId}/.../write	POST	CONTROL	對指定 RS485 device 寫入 (WI-118)
/api/modbus-tcp/targets	GET	-	Modbus TCP 遠端讀取設定
/api/modbus-tcp/targets	POST	ADMIN	更新 Modbus TCP targets
/api/modbus-tcp/write	POST	CONTROL	寫入遠端 Modbus TCP

TCP IO / Converter（轉換器）

Endpoint	Method	Permission	說明
<code>/api/tcpio</code>	GET	-	列出 TCP IO 通道
<code>/api/tcpio</code>	POST	ADMIN	新增 / 更新
<code>/api/tcpio</code>	DELETE	ADMIN	刪除
<code>/api/tcpio/clear</code>	POST	ADMIN	清空所有 (WI-103 FR-H1-02)
<code>/api/converter/template</code>	GET / POST	ADMIN *	Outbound assembly template
<code>/api/converter/preview</code>	POST	CONTROL	Template 變數展開預覽

連線 / MQTT / WiFi

Endpoint	Method	Permission	說明
<code>/api/mqtt/test</code>	POST	ADMIN	測試 MQTT broker 連線並發測試訊息
<code>/api/mqtt/publish</code>	POST	CONTROL	通用 MQTT 發佈 (工作流程「觸發」用, WI-128) 。body <code>{"topic","payload","qos"?}</code> ;topic 限本機 prefix 命名空間、禁萬用字元、payload ≤127B
<code>/api/wifi/status</code>	GET	-	目前 WiFi 連線狀態
<code>/api/wifi/scan</code>	GET	-	掃描可用 SSID
<code>/api/wifi/connect</code>	POST	ADMIN	連線指定 SSID
<code>/api/wifi/disconnect</code>	POST	ADMIN	中斷 WiFi
<code>/api/wifi/save</code>	POST	ADMIN	儲存憑證至 Flash (provisioning)
<code>/api/network/apmode</code>	POST	ADMIN	切換到 AP (soft AP) 模式重開機

WiFi 與 `/api/network/apmode` 僅在編譯時定義 `USE_WIFI` 才會啟用。

雲端 / Config Server

Endpoint	Method	Permission	說明
<code>/api/configserver</code>	GET	READONLY	取得 cloud 同步設定 (URL / token / enabled)
<code>/api/configserver</code>	POST	ADMIN	更新 cloud 設定
<code>/api/configserver/test</code>	POST	ADMIN	觸發非同步連線測試
<code>/api/configserver/test</code>	GET	READONLY	輪詢測試結果
<code>/api/configserver/push</code>	POST	ADMIN	強制立刻推送 config 到雲
<code>/api/debug/configserver</code>	GET	READONLY	上述 GET 的 legacy alias
<code>/api/telemetry-token</code>	GET	READONLY	回快取的 scoped telemetry token；無 / 過舊則觸發主 loop 向雲端換取並回 <code>202 fetching</code> (ADR-017 高頻遙測)

Token 與認證

Endpoint	Method	Permission	說明
<code>/api/tokens</code>	GET	READONLY	列出 token (已遮罩 secret)
<code>/api/tokens</code>	POST	ADMIN	建立新 token (WI-112, 僅此回傳明文)
<code>/api/tokens</code>	DELETE	ADMIN	撤銷 token

統計與電力

Endpoint	Method	Permission	說明
<code>/api/stats</code>	GET	-	生產計數器與 uptime (/fs/stats.bin 持久化)
<code>/api/stats/reset</code>	POST	ADMIN	重置 IO 統計並刪除 /fs/stats.bin
<code>/api/power</code>	GET	-	PowerMonitor 電力資料

License

Endpoint	Method	Permission	說明
<code>/api/license</code>	GET	-	License 狀態與授權功能
<code>/api/license/activate</code>	POST	ADMIN	申請授權 (async, 由雲端 heartbeat 套用 verdict)
<code>/api/license/renew</code>	POST	ADMIN	續約
<code>/api/license/reset</code>	POST	ADMIN	清空所有 license 狀態 (WI-103 FR-H1-01)
<code>/api/license/token</code>	POST	ADMIN	灌入 base64 簽章 token → 驗簽+比 UID+比到期 → 存 <code>/cfg/license.tok</code> (WI-155 瀏覽器 inbound 授權, 救 outbound/MQTT 壞的設備如 8310)
<code>/api/license/token</code>	GET	-	讀回 <code>/cfg/license.tok</code> 驗證回報 (<code>{present:false}</code> 或 token 資訊)
<code>/api/license/enforce</code>	POST	ADMIN	切換 enforcement 旗標 <code>{"on":true false}</code> (WI-145 rollout/測 試)

OTA（空中更新）

Endpoint	Method	Permission	說明
<code>/api/ota/versions</code>	POST	ADMIN	觸發向雲端拉取版本列表 (async)
<code>/api/ota/versions</code>	GET	-	輪詢列表結果
<code>/api/ota/status</code>	GET	-	目前版本 + 是否有可用更新
<code>/api/ota/start</code>	GET	-	直接啟動下載最近 heartbeat 廣播的更新
<code>/api/ota/trigger</code>	POST	ADMIN	用 body 指定 URL / version 觸發 OTA
<code>/api/ota/upload</code>	POST	ADMIN	直接上傳 raw firmware <code>.bin</code> (串流; 偵測到 <code>.mesb</code> 會拒絕, 要走 <code>/api/ota/bundle</code>)
<code>/api/ota/bundle</code>	POST	ADMIN	上傳 <code>.mesb</code> 單一檔 (firmware + 全部 web 資產, 串流) → <code>handleMesbUpload</code> : fw→QSPI P2、資產→P4, 重開。拓荒/發佈核心(WI-125)
<code>/api/upload</code>	POST	ADMIN	上傳 web UI 至 QSPI (multipart 串流)
<code>/api/web?name=<檔名></code>	POST	ADMIN	串流上傳單一 web 資產到 QSPI <code>/cfg/web/<檔名></code> (CSS/語系/app.js, 免重燒迭代 UI, WI-124)

Debug

Endpoint	Method	Permission	說明
<code>/api/debug/tcptest</code>	GET	-	測試 EthernetClient.connect() 兩種寫法
<code>/api/debug/exp</code>	GET	-	Dump <code>expDiCache</code> 與 signal sources

詳細範例

系統健康狀態 — GET /api/system

此 endpoint 是前端 heartbeat / login 驗證用。從 v5.9 (WI-115k) 起 dispatcher 強制

PERM_READONLY —— login modal 直接拿這個端點驗 token 是否合法，否則先前任意密碼都回 200 會被誤判成功。

回應主要欄位：

```
{
  "apMode": false,
  "version": "5.9.130",
  "board": "Arduino Opta",
  "uptime": 123456,
  "memory": {
    "totalRAM": 523624,
    "freeRAM": 453880,
    "usedRAM": 69744,
    "usagePercent": 13.4
  },
  "flash": {
    "totalFlash": 786432,
    "usedFlash": 781000,
    "qspiFlash": 16777216
  },
  "mac": "AA:BB:CC:DD:EE:FF",
  "licensed": true,
  "licensedFeatures": ["rule_engine", "mqtt", "tcp_io"],
  "bootTries": 1,
  "systemStatus": "OK"
}
```

JSON

⚠ `flash.usedFlash` 接近 `786432` (partition 上限) 時要警覺——v5.9.113 之後 build 已穩定 ≥ 99.3% 使用率, 新功能加進來前要先評估 binary size, 否則 OTA 會被 `/api/ota/trigger` 的 `413 Payload Too Large` 擋下 (見 WI-115g)。

`apMode = true` 是前端切換到 **Provisioning Wizard** 的主要 trigger。

設定 — `GET /api/config`

完整節點設定 dump, 主要 block:

```
{
  "firmwareVersion": "5.9.130",
  "deviceName": "opta-line-3",
  "mac": "AA:BB:CC:DD:EE:FF",
  "network": {
    "ethDhcp": true,
    "ethIp": "192.168.1.100",
    "ethGateway": "192.168.1.1",
    "ethSubnet": "255.255.255.0",
    "wifiDhcp": true,
    "wifiIp": "...",
    "wifiGateway": "...",
    "wifiSubnet": "...",
    "interfaceMode": 0
  },
  "wifi": { "ssid": "MyAP", "passwordLen": 16 },
  "tcp": { "port": 80 },
  "mqtt": {
    "enabled": true,
    "broker": "mqtt.example.com",
    "port": 8883,
    "clientId": "opta-abc",
    "username": "device",
    "password": "*****",
    "topicPrefix": "factory/lineA/",
    "publishPeriodicIo": true,
    "useTls": true,
    "caCert": "-----BEGIN CERTIFICATE-----\nMIIDxz...==\n-----END CERTIFICATE-----",
  },
  "rs485": { "baudRate": 9600, "masterMode": true, "slaveId": 1 },
  "inputMode": [0, 0, 0, 0, 0, 0, 0, 0],
  "inputOp": [0, 0, 0, 0, 0, 0, 0, 0],
  "inputThreshold": [3.3, 3.3, 3.3, 3.3, 3.3, 3.3, 3.3, 3.3],
}
```

```



```

v5.9.125+: MQTT block 新增 `useTls` 與 `caCert`。 `caCert` 在 wire format 上把實際換行字元轉成字面 `\n`，前端送回時必須維持 JSON-escape 形式。

設定 — `POST /api/config`

需 `PERM_ADMIN`。只送要改的欄位，handler 內以 `indexOf("\"key\":value")` 字串比對；JSON 必須是 **compact 格式**(key/value 之間沒有空格)，否則會靜默失敗。

```

{
  "deviceName": "opta-line-3",
  "mqtt": {
    "enabled": true,
    "broker": "mqtt.example.com",
    "port": 8883,
    "useTls": true,
    "caCert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n"
  }
}

```

回應：

- `200 OK`: `{"success": true, "message": "Config updated and saved"}`

- 400 Bad Request : {"error": "No valid fields to update"}
- 401 Unauthorized : {"error": "Unauthorized"}

寫入時透過 `NetworkStateManager::requestSafeSave()` 延後到 WiFi state machine 安全空檔，避免跟 supplicant transition 衝突。MQTT 相關欄位改動會自動重連 broker。

切換數位輸出 — `POST /api/io/do`

需 `PERM_CONTROL`。

```
{ "index": 0, "value": true }
```

JSON

或同時設定多顆(舊版)：

```
{ "values": [true, false, false, true] }
```

JSON

Pulse 模式 (`pulse: true, duration: <ms>`) 目前 handler 預留但尚未實作。

回應：

- 200 OK : {"success": true, "index": 0, "value": true}
- 400 Bad Request : index 越界 {"error": "invalid index"}

Modbus 掃描 — `POST /api/modbus/scan`

需 `PERM_ADMIN`。RS485 掃描是 **非同步** 的：POST 啟動 → 持續 GET 輪詢結果 → 必要時 DELETE 取消。

```
# 啟動
curl -X POST -H "Authorization: Bearer $TOKEN" \
      http://opta.local/api/modbus/scan -d '{"busId":0}'
# → 202 {"status":"scanning"}

# 輪詢
curl -H "Authorization: Bearer $TOKEN" http://opta.local/api/modbus/scan
# 完成: {"devices":[{"slaveId":1,"name":"TDA-08B","registers":[...]}]}

# 取消
curl -X DELETE -H "Authorization: Bearer $TOKEN" http://opta.local/api/modbus/scan
```

掃描期間 RS485 bus 會被自動 lock 阻止 polling，掃完自動釋放。也可手動 **POST /api/modbus/lock** 在做手動寫入時暫鎖匯流排。

MQTT 測試 — **POST /api/mqtt/test**

需 **PERM_ADMIN**。可帶 body 用指定設定測試；省略則用目前設定。

```
{
  "broker": "mqtt.example.com",
  "port": 8883,
  "username": "test",
  "password": "test",
  "useTls": true,
  "caCert": "-----BEGIN CERTIFICATE-----\\n...\\n"
}
```

回應：

- **200 OK** : {"success": true, "message": "Test message published"}
- **400 / 500** : {"success": false, "error": "..."}

MQTT 自簽 broker 注意 (v5.9.125~130) : mbedTLS 不認 IP-type SAN, broker cert 必須同時放 `IP:x.x.x.x` 與 `DNS:x.x.x.x`, 否則 handshake 會在 verify 階段失敗。

Token 管理

列表 (`GET /api/tokens`) — `PERM_READONLY` :

```
[
  { "index": 0, "name": "frontend", "permission": "ADMIN", "createdAt": 17370... },
  { "index": 1, "name": "factory-rd", "permission": "CONTROL", "createdAt": 17370... },
]
```

建立 (`POST /api/tokens`) — `PERM_ADMIN` :

```
{ "name": "ops-readonly", "permission": "READONLY" }
```

回應 僅此一次 包含明文 token:

```
{ "success": true, "token": "tok_...", "name": "ops-readonly", "permission": "RE...
```

撤銷 (`DELETE /api/tokens`) — `PERM_ADMIN` :

```
{ "index": 1 }
```

License 啟用流程

License 採 **device** → **cloud** → **device** 三段式:

1. `POST /api/license/activate` → 標記 pending, 回 `{"status": "pending"}`
2. 雲端 (opta.smms.com.tw) 約 30 秒一次 heartbeat 收 verdict
3. `GET /api/license` 取得當下狀態:

```
{
  "status": "active",
  "licenseId": "LIC-2026-xxx",
  "features": ["rule_engine", "mqtt", "tcp_io"],
  "expiryDate": 1764547200,
  "lastKnownTime": 1745000000
}
```

JSON

重要: `/api/license` GET handler 不會主動 POST activate, 避免跟 server「rejected → pending 重申」邏輯衝突 (前端 polling 會把 rejected 推回 pending 永遠跑不出 PENDING 迴圈)。Verdict pull 是 cloud → device, 使用者顯式 activate 才是 device → cloud。

OTA 觸發 — `POST /api/ota/trigger`

需 `PERM_ADMIN`。Body 帶 url (必要) 與 version (建議, 作為 guard):

```
{ "url": "https://opta.smms.com.tw/firmware/opta_v5.9.131.bin", "version": "5.9.131" }
```

JSON

回應:

HTTP	條件	Body
200 OK	URL 有效 + version ≠ current	<code>{"ok":true,"message":"OTA download started","version":"..."}</code>
409 Conflict	version == FW_VERSION (避免 same- version brick)	<code>{"error":"Same version","message":"...","current":"...","target":"..."}</code>
413 Payload Too Large	binary > 780,000 bytes	<code>{"error":"Firmware too large for safe OTA","size":...,"limit":780000}</code>
400 Bad Request	缺 URL	<code>{"error":"No OTA URL specified"}</code>

WI-115c: **拒絕 same-version OTA**——重複 swap 同一份 firmware 會踩到 bootloader fallback 邏輯, 被丟去 DFU。

WI-115g: partition 上限 786432 bytes, 留 6KB 緩衝設 **780,000 bytes** 為硬上限。超過請改用 USB DFU。

時間同步 — `POST /api/time/sync`

需 `PERM_ADMIN`, 從瀏覽器同步 RTC (WI-079)。

```
{ "timestamp": 1748044800 }
```

JSON

回應:

- 200 OK: `{"status":"success","timestamp":1748044800}`
- 400 Bad Request: timestamp 比 `LastKnownTime` 還舊會被拒。

非同步 Config Server 測試

`POST /api/configserver/test` 觸發後立刻回 `{"status": "testing"}`; client socket 關閉後韌體才在 `loop()` 跑實際連線 (避免佔用 outbound socket)。輪詢 `GET /api/configserver/test` :

```
{
  "status": "done",
  "success": true,
  "latency": 142,
  "version": "v2.2.10",
  "error": ""
}
```

JSON

04

批次輪詢 — `GET /api/poll`

合併 `system + io + virtual + stats` 為單一 response，減少前端輪詢成本。內部依序呼叫各 handler 並省略 HTTP header (`printHeader = false`)，最後組成單一 JSON 物件，欄位即四個子 handler 的合併。

05

HTTP 狀態碼

Code	意義
200	成功
202	已接受 (async 任務排隊; 輪詢另一 endpoint 取結果)
400	Bad Request (JSON 解析失敗、必填缺、值越界)
401	Unauthorized (token 缺或權限不足)
404	Not Found (unknown API path 或 QSPI 檔不存在)
409	Conflict (OTA same version)
413	Payload Too Large (OTA binary > 780KB)
500	Internal Server Error (config 未載入等)

錯誤一律走 JSON:

```
{ "error": "register_not_found", "message": "registerIdx out of range for this device" }
```

Modbus handler 另有 `sendErr(httpCode, statusText, errorCode, message)` 統一包裝，常見 `errorCode`：

- `register_not_found`
- `register_not_writable` (Register `mode != 1`)
- `device_not_found`
- `bus_locked`

近期變更 (v5.9.x 重點)

Version	變更
v5.9.130	MQTTS handshake 修正——chain 父類別 <code>setRootCA</code> + <code>threshold=1</code> ，子類 <code>MqttTlsDebug</code> 印 cert source/len 與 mbedTLS debug。
v5.9.125	<code>/api/config</code> GET/POST 新增 <code>mqtt.useTls</code> 與 <code>mqtt.caCert</code> (PEM，wire 上 <code>\n</code> escape)。
v5.9.x WI-118	新增 <code>/api/rs485/{busId}/.../write</code> ——以 bus path 對指定 RS485 device 寫入。
v5.9.x WI-115k	<code>/api/system</code> dispatcher 強制 <code>PERM_READONLY</code> ，修補登入時假成功問題。
v5.9.x WI-115g	OTA 加入 780KB size guard；超過直接 <code>413</code> 擋下。
v5.9.x WI-115c	OTA 拒絕 same-version trigger，避免 bootloader fallback。
v5.9.x WI-115f	主控台 HTML 加 ETag (值 = <code>FW_VERSION</code>)，減少重複下載。
v5.9.x WI-115d	移除 legacy <code>GET /app.js</code> ，bundle 直接 inline 進 <code>WebPage.h</code> 。
v5.9.x WI-112	Token 管理： <code>POST</code> / <code>DELETE /api/tokens</code> 。
v5.9.x WI-103	FR-H1-01 <code>/api/license/reset</code> 、FR-H1-02 <code>/api/rules/clear</code> + <code>/api/tcpio/clear</code> 。
v5.9.x WI-079	<code>/api/time/sync</code> ：瀏覽器同步 RTC。
v5.9.x WI-085	DO 索引由 0-based → 1-based (DO1–DO4)，影響 <code>/api/io</code> 回傳與 <code>/api/io/do</code> index。

實作備註

1. **無 router framework**: 所有 endpoint 在 `main.cpp:1657-2071` 以 `else if` (`firstLine.indexOf("METHOD /path") >= 0`) 串接, 順序決定優先權——較長的路徑 (例如 `/api/configserver/test`) 必須排在較短前綴 (`/api/configserver`) 之前。新增 endpoint 時注意這個順序陷阱。
2. **Auth 分佈**: 除 `/api/system` 在 dispatcher 即驗, 其餘 endpoint 在各 handler 一進入就呼叫 `checkAuth(req, PERM_*)`, 失敗回 `401`。
3. **Streaming uploads**: `/api/upload` 與 `/api/ota/upload` 不把 body 載進記憶體, 邊收邊寫 QSPI / partition; OTA upload 期間 `localConfig.unmount()` → 寫入 → `remount()`。
4. **Async pattern**: Config server test、OTA version list、Modbus scan、License activate 都用「先回 202/pending → 在 `loop()` 跑工作 → 後續 GET 輪詢結果」模式, 避免阻塞單一 client socket。
5. **JSON 寫回**: 所有 JSON response 都標 `Content-Type: application/json`、`Connection: close`, 多數不帶 `Content-Length` (chunked-style, 由 FIN 標 EOF), client 必須讀到 socket 關閉才算完整。
6. **Config POST 字串比對**: handler 用 `indexOf("\"key\":value")`, JSON 必須 compact 無空格——Python `json.dumps` 預設帶空格會靜默失敗, 務必加 `separators=(",", ":",")`。
7. **USE_WIFI 編譯切換**: WiFi 與 AP mode 相關 endpoint 用 `#ifdef USE_WIFI` 包住; Ethernet-only build 不會註冊這些路由, request 會回 404。
8. **Content-Length 用 byte 數**: handler 依 `Content-Length` 讀 body, 中文 UTF-8 每字 3 bytes; client 若用字元數會截斷 body (POST 回 success 但欄位沒套用)。
9. **保留 MQTT topic** `{prefix}/cmd/_ping` (v5.9.130+): 韌體 self-ping 健康檢查專用, 外部請勿佔用。
10. **MQTT-Chain 工作流**: `/api/signals`、`/api/actions`、`/api/rules`、`/api/tcpio`、`/api/converter/*` 可組成有狀態工作流 (state 經 MQTT loopback)。完整 schema、欄位、雷區見 `../specs/spec-mqtt-chain-workflow.md`。

延伸閱讀

主題	文件
MQTT-Chain 工作流引擎 + API schema	<code>../specs/spec-mqtt-chain-workflow.md</code>
TDA08B 校正引導精靈操作手冊	<code>../guides/guide-calibration-wizard.md</code>
備援機制 + MQTT Watchdog 部署	<code>../guides/guide-failover-resilience.md</code>
Config Server API	<code>api-config-server.md</code>