دليل واجهة EPORDO Connect (العربية)

البند	المحتوى
إصدار المستند	**1.1.5**
تاريخ النشر	2026-11-11
القرّاء	مطورو الطرف المدمج، الوسطاء / المكاملون، فرق تسليم المشاريع
المنتج	**EPORDO Connect** (منصة سحابية لربط أجهزة الطرف الثالث)
شكل التسليم	مصدر Markdown هذا + **PDF** قابل للتنزيل من البوابة (المساعدة → تنزيل دليل الواجهة)

> يصف هذا المستند **واجهة REST الخاصة بالمستأجر** (بادئة المسار `/connect/v1`). يتولى Connect الاتصال بين الجهاز والسحابة؛ **لا يحتاج** الطرف المدمج إلى تحليل أو تنفيذ البروتوكول الثنائي / النصي الأصلي للجهاز.

> تستخدم الأمثلة في النص عنوان Base URL بأسلوب الإنتاج؛ عند الربط التجريبي استبدلوه بالعنوان الذي يقدّمه فريق التسليم.

> **اصطلاح:** جميع المسارات أدناه نسبية إلى Base URL. أسماء حقول JSON وفق هذا المستند (مثلاً حقل جسم الحدث التجاري هو **`payload_json`** ، ويجب عليكم `JSON.parse`).

سجل المراجعات

الإصدار	التاريخ	الوصف
1.1.5	2026-11-11	§5.3.1: `door_mode` و `door_opened` في اللغات الخمس؛ `photo_status` هو `absent` أو `stored` أو `omitted_too_large` أو `omitted_invalid`
1.1.4	2026-11-10	§5.3: الأنواع الحالية `door.remote.recorded` و `device.presence` ؛ نافذة الاتصال 10 دقائق، ومسح انقطاع الاتصال كل 5 دقائق تقريبًا، والأسماء القديمة تطابق الحافة المقابلة فقط
1.1.3	2026-11-08	تحسين: رأس جدول §5.3 يصبح `event_type` وإضافة `door.open.recorded` ؛ الملحق D: `GET /commands/{id}` + `POST .../cancel`
1.1.2	2026-11-08	تصحيحات: حقل الحدث هو `event_type` (وليس `type`)؛ واجهة اللقطة تُرجع JSON/ `data_base64` (وليس تدفق صورة مباشر)؛ لا يوجد `GET /webhooks/{id}` ؛ لا يوجد `DELETE`

		`/device-groups/{id}` (استخدم `status=DISABLED`)؛ توضيح أيام الاحتفاظ بما يوافق 3–14 يوماً
1.1.1	2026-11-08	مطابقة مع البيئة الحية: استكمال مسارات وأمثلة API Key / المجموعات؛ التحقق من توقيع Webhook وoutbox؛ عيّنات `payload_json` للأحداث وتوضيح الاحتفاظ؛ مثال fan-out لإرسال الأشخاص؛ فهرس المحتويات
1.1.0	2026-11-08	إضافة نظرة عامة على النظام وعلاقة الاتصالات؛ استكمال أمثلة cURL وسيناريوهات في كل فصل؛ إضافة فك ربط الجهاز؛ إضافة سجل المراجعات؛ تسليم PDF
1.0.0	2026-11-07	الإصدار العلني الأول: المطالبة، الأحداث، Webhook/MQTT، الأوامر القياسية، fan-out، ملخص رموز الأخطاء

لاحقاً، إذا اقتصر التصحيح على أخطاء إملائية تُضاف نسخة رقعة في السجل؛ وإذا تغيّر سلوك الواجهة يُرفع الإصدار الثانوي ويُذكر في الفصل المعني.

المحتويات

1. نظرة عامة على النظام
2. البيئة والمصادقة (بما في ذلك إدارة API Key)
3. صيغة الخطأ الموحدة
4. الأجهزة
5. الأحداث
6. Webhook / MQTT
7. الأوامر القياسية
8. دفع جماعي fan-out حسب الأرقام التسلسلية
9. مجموعات الأجهزة وScoped Key
10. تحديد المعدل والحصة اليومية
11. مسار الربط الموصى به

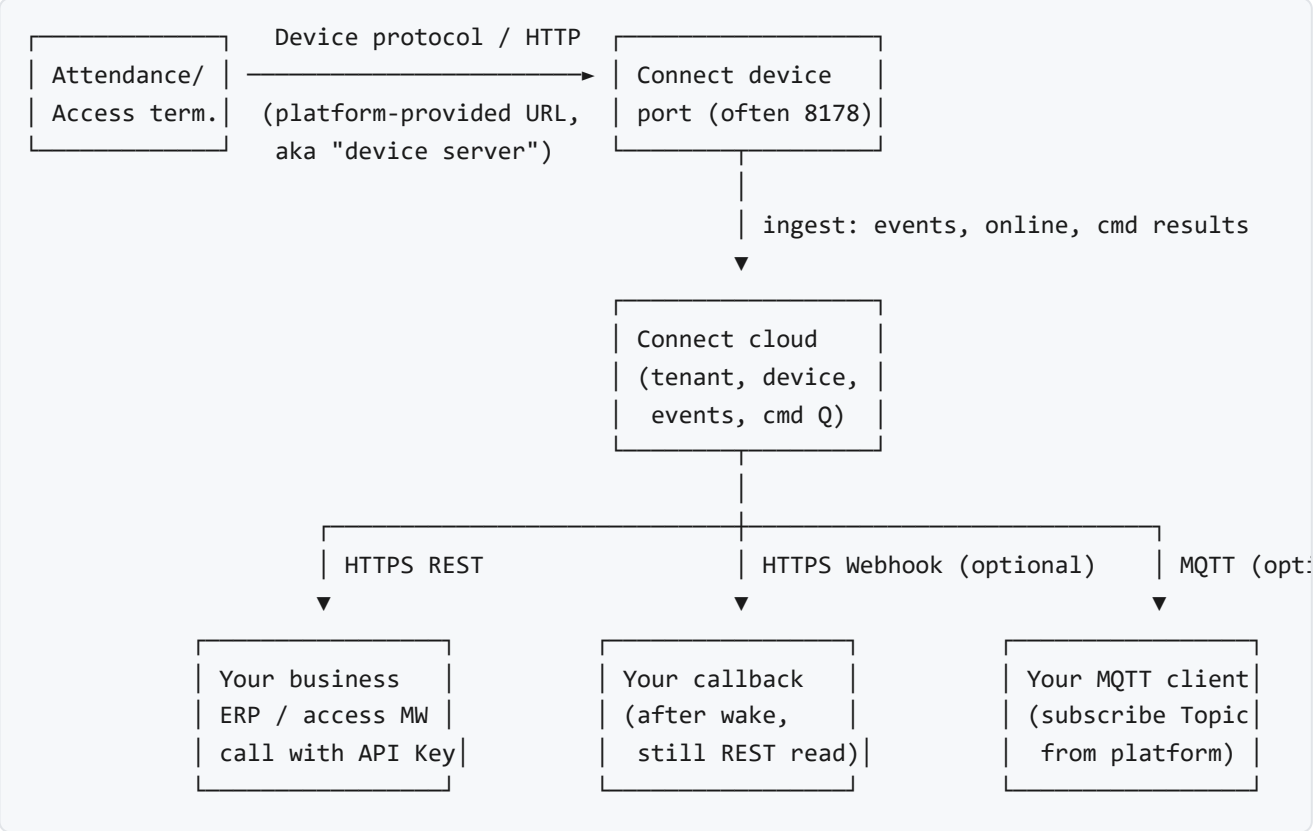
- الملحق أ رموز الأخطاء · الملحق ب أنواع kind لبيانات الاعتماد · الملحق ج الدعم · الملحق د مرجع سريع للواجهات

---

1. نظرة عامة على النظام (اقرأ هذا القسم أولاً)

1.1 موقع Connect في البنية الكلية

يمكن فهمه كـ«طبقة تكييف سحابية بين الأجهزة ونظام أعمالكم»:



لا تخلطوا بين الخطّين:

الخط	من يتصل بمن	ماذا عليكم أن تفعلوا
خط الأجهزة	الطرفيات ← خادم أجهزة Connect (مثال: `http://api.bnetlink.com:8178/`)	املأوا في قائمة الجهاز العنوان الذي تقدّمه المنصة؛ **لا تحتاجون** إلى تركيب رسائل الجهاز في كودكم
خط الأعمال	خادمكم ← Connect REST (مثال: `https://api.bnetlink.com/connect/v1`)	استدعوا واجهات هذا المستند بـ **Bearer API Key**

البوابة (المتصفح) تخدم موظفي الأعمال للمطالبة وعرض الأحداث وتجربة الأوامر؛ **للربط الرسمي استخدموا خلفيتكم + API Key**، ولا تضعوا المفتاح في صفحات عامة.

1.2 بروتوكول اتصال الأجهزة (من منظور الطرف المدمج)

تتواصل الطرفيات مع **خادم الأجهزة** في Connect عبر بروتوكول الجهاز المتفق عليه من الشركة المصنّعة (مثل الجهاز الديناميكي BS، الجهاز العادي FK، إلخ). يتولى Connect:

1. التعرّف على الرقم التسلسلي وانتماء المستأجر؛
2. تنظيم تقارير تسجيل الحضور / التسجيل وغيرها في **JSON حدث موحّد**؛
3. ترجمة **الأوامر القياسية** التي ترسلونها عبر REST إلى رسائل يمكن للجهاز تنفيذها، مع صف الانتظار وكتابة النتائج.

لذلك تظهر في وثائق الربط أسماء قياسية مثل `set-user` و`delete-user`، **وليست** سلاسل رموز الأوامر الداخلية للجهاز. يكفيكم:

- المطالبة بالرقم التسلسلي `sn` ؛
- سحب الأحداث / استقبال Webhook أو MQTT؛
- استدعاء الأوامر القياسية واستطلاع حالة `command_id` .

1.3 دورة حياة نموذجية (مثال)

مثال: متجر يركّب جهازاً للوجه جديداً ويوصله بوسيط الحضور لديكم.

1. عند الخروج من المصنع تكون الشركة المصنّعة قد أدخلت الرقم التسلسلي في «المخزون القابل للبيع» وربطته بمستأجركم.
2. أنتم (أو البوابة) تستدعون `POST /devices` ، `sn=ABC123` → **نجاح المطالبة**.
3. في إعدادات الجهاز تملؤون عنوان خادم الأجهزة وتربطون الشبكة → يبدأ نبض القلب / الإبلاغ.
4. الموظف يمسح الوجه → ينشئ Connect حدث `punch.recorded` → يمكنكم سحبه بـ `GET /events` ، أو الاستيقاظ أولاً عبر Webhook/MQTT ثم السحب.
5. من ERP ترسلون الموظف «1001»: `POST /devices/ABC123/commands/set-user` .
6. تستطلعون `GET /commands/{command_id}` حتى `SUCCEEDED` أو `FAILED` .
7. عند سحب الجهاز من المتجر: `DELETE /devices/ABC123` **لفك الربط**؛ يعود الرقم التسلسلي إلى حالة غير مفتوحة ويمكن المطالبة به لاحقاً؛ أحداث الحضور التاريخية تبقى محفوظة.

1.4 ماذا يفعل Connect / وماذا لا يفعل

يفعل:

- المطالبة / الإيقاف / التفعيل / فك ربط الأجهزة؛
- استقبال أحداث الحضور وتغييرات التسجيل وغيرها؛
- إرسال الأوامر القياسية (مزامنة الوقت، فتح الباب، الأشخاص، فترات المرور، إلخ)؛
- Webhook اختياري، ودفع MQTT اختياري.

لا يفعل:

- الجدولة، حساب الرواتب، محرك قواعد الحضور (ذلك شأن نظام الأعمال أو منتجات أخرى)؛
- حفظ «أي حدث وصلتم إليه» كمؤشر قراءة نيابة عنكم (احفظوه في قاعدتكم)؛
- التحويل التلقائي لصيغ قوالب الوجه / البصمة عبر البرامج الثابتة المختلفة.

1.5 ثلاث كتل يهتم بها الطرف المدمج يومياً

القدرة	الوصف
الأجهزة	المطالبة بـ `sn` ، القائمة، الإيقاف / التفعيل، فك الربط
الأحداث	السحب؛ اختيارياً Webhook و/أو MQTT للإيقاظ أو المسار الجانبي
الأوامر	مزامنة الوقت، الاستعلام عن الحالة، فتح الباب، الأشخاص، الفترات؛ يمكن الدفع الجماعي حسب قائمة الأرقام التسلسلية (fan-out)

---

2. البيئة والمصادقة

2.1 Base URL

البيئة	بادئة REST (مثال)
الإنتاج	يقدّمها العقد / التسليم، بالشكل `https://api.bnetlink.com/connect/v1`
الربط المحلي (إن فُتح)	`http://127.0.0.1:8080/connect/v1`

جميع المسارات أدناه نسبية إلى هذه البادئة. مثلاً `GET {base}/me` = `GET /me`.

مثال خادم الأجهزة (يُملأ في الطرفية): `http://api.bnetlink.com:8178/` (وفق التسليم).

2.2 المصادقة

باستثناء `GET /health`، يجب أن يحمل رأس الطلب:

```
Authorization: Bearer {API_KEY}
Content-Type: application/json
Accept: application/json
```

ملاحظات:

- مفتاح API واحد يقابل **مستأجراً** واحداً، ولا يمكنه الوصول إلا إلى بيانات هذا المستأجر.
- يُنشأ المفتاح في صفحة «المفاتيح» في البوابة، أو يُمنح عند فتح الحساب؛ **النص الصريح يظهر مرة واحدة فقط**، فاحفظوه جيداً.
- **مفتاح توقيع Webhook** و**كلمة مرور اشتراك MQTT** **ليسا** نفس سلسلة API Key.
- حساب تسجيل دخول البوابة (اسم المستخدم + كلمة المرور) للمتصفح فقط، و**لا يمكنه** استبدال API Key لاستدعاء REST.

2.3 فحص الصحة (مثال)

```
curl -s "https://api.bnetlink.com/connect/v1/health"
```

عند النجاح:

```
{ "status": "ok" }
```

2.4 المستأجر الحالي (مثال)

```
curl -s "https://api.bnetlink.com/connect/v1/me" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

عند النجاح تحتوي الاستجابة على `tenant_id` والباقة ونطاق المفتاح وغيرها (الحقول وفق الواقع).

مثال: إذا كان `scope_type` المُرجَع نطاق مجموعة (`GROUPS`)، فإن بعض واجهات مستوى الحساب (مثل المطالبة، فك الربط، إدارة Webhook / المفاتيح / المجموعات) تُرجع `403` `forbidden_scope` .

2.5 إدارة API Key (REST)

صفحة «المفاتيح» في البوابة تعتمد على الواجهات التالية. عند الإنشاء يُعاد النص الصريح **مرة واحدة فقط**.

```
# قائمة (بدون نص صريح)
curl -s "https://api.bnetlink.com/connect/v1/credentials" \
  -H "Authorization: Bearer YOUR_API_KEY"

# إنشاء مفتاح على مستوى الحساب بالكامل
curl -s -X POST "https://api.bnetlink.com/connect/v1/credentials" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"name\":\"erp-prod\",\"scope_type\":\"ACCOUNT\"}"

# إنشاء مفتاح محدود بمجموعات (group_ids يجب أن تكون مجموعات ACTIVE لهذا المستأجر)
curl -s -X POST "https://api.bnetlink.com/connect/v1/credentials" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"name\":\"store-a\",\"scope_type\":\"GROUPS\",\"group_ids\":[1,2]}"

# إيقاف / إعادة تفعيل (status: REVOKED | ACTIVE)
curl -s -X PUT "https://api.bnetlink.com/connect/v1/credentials/2" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"status\":\"REVOKED\"}"
```

مثال نجاح الإنشاء (احفظوا `api_key` فوراً):

```
{
  "api_key": "epc_live_…",
  "key_prefix": "epc_live_…",
  "scope_type": "ACCOUNT",
  "warning": "Store api_key now; it will not be shown again."
}
```

ملاحظة: لا يمكن إيقاف **آخر** مفتاح `ACTIVE` للمستأجر (`409` `last_active_api_key`). مفتاح المجموعة **لا يمكنه** تسجيل الدخول إلى البوابة.

---

3. صيغة الخطأ الموحدة

```
{
  "error": {
    "code": "device_not_found",
    "message_key": "connect.error.device_not_found",
    "message": "device_not_found"
  }
}
```

HTTP	المعنى (ملخص)
400	معاملات غير صالحة
401	لم يُرسل المفتاح / المفتاح غير صالح أو موقوف
403	صلاحية غير كافية (مثلاً مفتاح مجموعة استدعى واجهة على مستوى الحساب)
404	المورد غير موجود أو لا ينتمي إلى هذا المستأجر
409	تعارض (مطالبة مكررة، جهاز موقوف، تعارض idempotency، إلخ)
429	بلوغ حد المعدل بالدقيقة أو الحصة اليومية للأوامر

مثال: إرسال فتح باب إلى جهاز موقوف → `409` HTTP، و`code` = `device_disabled`.

رموز `code` الشائعة في الملحق أ.

---

4. الأجهزة

4.1 مطالبة الجهاز

```
POST /devices
```

```
{ "sn": "TEST-SN-001", "alias": "بوابة الاستقبال" }
```

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"sn\":\"TEST-SN-001\",\"alias\":\"بوابة الاستقبال\"}"
```

مثال النجاح:

```
{
  "sn": "TEST-SN-001",
  "connect_device_id": 12,
  "register_status": "ACTIVE",
  "protocol_track": "BS",
  "model_code": "P5",
  "firmware_version": "1.02"
}
```

ملاحظات:

- يمكن المطالبة فقط بأرقام تسلسلية **موجودة في مخزون الشركة المصنّعة، مربوطة بهذا المستأجر، ولم تُفتح بعد**.
- البروتوكول / الطراز / البرنامج الثابت من إعدادات الشركة المصنّعة المسبقة؛ جسم الطلب **لا يمكنه** تزويرها.
  طراز البيع هو الاسم القصير `P5` أو `dev_model` . `P7` في الترويسة لا يغيّره.
- رقم تسلسلي غير معروف → `device_not_in_stock` `404` .
- مخصّص لمستأجر آخر → `device_not_for_tenant` `403` .
- سبق أن طالب به هذا المستأجر → `device_already_registered` `409` .
- اسم الحقل الخارجي دائماً `sn` .

مثال: ملصق العلبة يحمل `B515…` ، لكن الشركة المصنّعة لم تُدخل الرقم بعد في مخزونكم ← المطالبة تُرجع `device_not_in_stock` ؛ اتصلوا بالشركة لإعداده مسبقاً ثم أعيدوا المحاولة.

4.2 قائمة الأجهزة / التفاصيل

```
curl -s "https://api.bnetlink.com/connect/v1/devices?limit=50" \
  -H "Authorization: Bearer YOUR_API_KEY"

curl -s "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

القائمة تدعم المؤشر: `next_cursor` غير شفاف؛ للصفحة التالية ضعوه كما هو في `cursor=` .

4.3 الإيقاف / التفعيل

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/disable" \
  -H "Authorization: Bearer YOUR_API_KEY"

curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/enable" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

- **الإيقاف**: يبقى تابعاً لهذا الحساب، لكن لا يمكن إرسال أوامر بعد ذلك (`device_disabled` `409`). أحداث إبلاغ الجهاز عادةً ما تزال تُدخل إلى المخزن (وفق registry).
- **التفعيل**: يستعيد إمكانية إرسال الأوامر.
- الإيقاف ≠ فك الربط.

4.4 فك ربط الجهاز

```
curl -s -X DELETE "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

مثال النجاح:

```
{ "sn": "TEST-SN-001", "claim_status": "UNCLAIMED" }
```

الأثر:

- يُزال الجهاز من هذا الحساب؛ يعود مخزون الشركة المصنّعة إلى «غير مفتوح»، و**يمكن المطالبة به مجدداً**؛
- تُمسح سجلات الأشخاص على المنصة لهذا الجهاز؛ تُلغى الأوامر قيد الانتظار؛
- **أحداث الحضور التاريخية تبقى**؛
- المفاتيح الفرعية بنطاق المجموعة **لا يمكنها** الاستدعاء (`403`).

4.5 الأشخاص وبيانات الاعتماد على الجهاز (التسجيل الحالي على المنصة)

الطريقة	المسار	الوصف	
GET	`/devices/{sn}/users`	قائمة الأشخاص؛ اختيارياً `q` ، `compatible` \scope=device`	
GET	`/devices/{sn}/users/{userId}`	تفاصيل الشخص وبيانات وصفية لبيانات الاعتماد	
GET	`/devices/{sn}/users/{userId}/credentials/{kind}/{slot}`	جسم بيانات الاعتماد (`data_base64`)	

```
curl -s "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/users?limit=50" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

`kind` مسموح فقط: `password` ، `card` ، `palm` ، `fingerprint` ، `face` ، `photo` (انظر الملحق ب).

---

5. الأحداث

Connect **لا يحفظ** مؤشر قراءتكم. سجّلوا في قاعدتكم «آخر `event_id` / وقت تمت معالجته».

5.1 سحب الأحداث

```
curl -s "https://api.bnetlink.com/connect/v1/events?limit=50" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

سحب آخر يوم حسب وقت الإدخال (مثال؛ استبدلوا بـ ISO-8601 حقيقي):

```
curl -s "https://api.bnetlink.com/connect/v1/events?received_from=2026-11-07T00:00:00Z&recei
  -H "Authorization: Bearer YOUR_API_KEY"
```

ملخص القواعد:

- معاملات الوقت بصيغة ISO-8601؛ عند اقتران البداية والنهاية لا يتجاوز المدى **7 أيام**.
- `from` / `to` = وقت حدوث العمل؛ `received_from` / `received_to` = وقت إدخال المنصة.
- عنصر القائمة يحتوي **سلسلة** `payload_json` (شكل نصي لـ JSON العمل)، و**لا يحتوي** صورة اللقطة الكبيرة.
- تحتفظ المنصة بالأحداث الساخنة وفق إعداد المستأجر: الافتراضي حوالي **7 أيام**، والمدى القابل للضبط حوالي **3–14 يوماً** (وفق العقد / إعداد المنصة)؛ الساعة وفق `received_at`. بعد انتهاء مدة الاحتفاظ قد يصبح `GET /events/{id}` غير موجود. خزّنوا البيانات في الوقت المناسب.
- في استعلام القائمة «كل زوج من نوافذ الوقت بحد أقصى 7 أيام» قيد **نافذة الاستعلام**، وهو أمر مختلف عن أيام الاحتفاظ على القرص.

شكل عنصر القائمة (توضيحي؛ أسماء الحقول مطابقة للبيئة الحية):

```
{
  "event_id": "evt_a1b2c3d4e5f6478990abcdef12345678",
  "event_type": "punch.recorded",
  "occurred_at": "2026-10-29T14:30:00",
  "received_at": "2026-10-29T14:30:01.123+00:00",
  "connect_device_id": 12,
  "sn": "TEST-SN-001",
  "payload_json": "{\"event_id\":\"evt_a1b2…\",\"sn\":\"TEST-SN-001\",\"user_id\":\"1001\",\'
}
```

نفّذوا `JSON.parse` مرة واحدة على `payload_json`. لاحظوا أن الحقل الخارجي هو `event_type`، وليس `type`. جسم HTTP الذي يدفعه Webhook / MQTT هو **JSON العمل بعد تحليل** `payload_json` (وليس غلاف صف القائمة بالكامل).

مثال: فشل Webhook ليلاً وفُقدت 20 رسالة. في الصباح اسحبوا بالصفحات مع `received_from` يغطي فترة الليلة الماضية، وأزيلوا التكرار حسب `event_id` عند الكتابة إلى قاعدة الأعمال لتعويض الفقد.

5.2 حدث مفرد واللقطة

```
curl -s "https://api.bnetlink.com/connect/v1/events/{eventId}" \
  -H "Authorization: Bearer YOUR_API_KEY"

curl -s "https://api.bnetlink.com/connect/v1/events/{eventId}/photo" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

واجهة الصورة متاحة فقط عندما تكون لقطة الحضور قد أُدخلت (غالباً `photo_status=stored`). الاستجابة هي **JSON**، وليست تدفق بايتات صورة مباشرة، مثلاً:

```
{
  "event_id": "evt_…",
  "content_type": "image/jpeg",
  "byte_length": 12345,
  "sha256": "…",
  "data_base64": "/9j/4AAQ…"
}
```

فكّوا ترميز `data_base64` ثم احفظوه على القرص. بلا صورة ← `photo_not_found` `404`؛ الحدث غير موجود ← `event_not_found` `404`. القائمة / التفاصيل نفسها لا تحمل بايتات الصورة.

5.3 أنواع الأحداث الشائعة

event_type	المعنى
`punch.recorded`	تسجيل حضور
`enroll.data.received`	ملخص تسجيل الشخص / تغيّر بيانات الاعتماد
`command.completed` / `command.failed`	الحالة النهائية للأمر
`door.state.changed`	تغيّر حالة مغناطيس الباب (إبلاغ الجهاز النشط)
`door.remote.recorded`	إجراء باب عن بُعد (`mode`: pulse/hold/close؛ ليس تسجيل حضور)
`device.presence`	حافة اتصال / انقطاع (`online`: 1/0)

الصفوف الجديدة تستخدم الأنواع أعلاه فقط. قيمة `online` في `GET /devices` هي `1` إذا كان `last_seen_at` خلال **10 دقائق**، وإلا `0`. يُكتب `device.presence` عند الحافة فقط: أي رسالة صاعدة تكتب `online=1`؛ ومسح كل **5 دقائق** تقريبًا يكتب `online=0`، فيظهر حدث الانقطاع بعد نحو **10–15 دقيقة** من آخر حزمة. `occurred_at` هو آخر رسالة صاعدة، و`received_at` هو وقت تسجيل الحافة. الأسماء القديمة / `device.online` و`device.offline` / `door.close.recorded` / `door.open.recorded` ما زالت تُستخدم في التصفية والاشتراك، لكن للحافة المطابقة فقط.

5.3.1 حقول `payload_json` لتسجيل الحضور (بعد التحليل)

الحقل	النوع	الوصف
`event_id`	string	معرّف الحدث، مفتاح إزالة التكرار
`sn`	string	الرقم التسلسلي للجهاز
`user_id`	string	رقم الشخص
`occurred_at`	number	وقت حدوث العمل، طابع زمني **بالمللي ثانية**
`device_time`	string	سلسلة الوقت المحلي للجهاز (غالباً `yyyyMMddHHmmss`)
`verify_mode`	string	وصف طريقة التحقق (مثل Face / Card)
`door_mode`	string	اختياري. نص الباب من الجهاز (مثل `open` / `hand_open`)
`door_opened`	boolean	**P5 / BS فقط.** `true` إذا أرسل الجهاز `doorMode` غير فارغ (تم تحريك القفل)؛ و `false` إذا كان مفقوداً أو فارغاً. النتيجة فقط، بلا سبب رفض. / `remote_open` `remote_close` ليسا تسجيل حضور
`has_log_photo`	boolean	هل أُعلن عن وجود لقطة
`photo_status`	string	`absent` أو `stored` أو `omitted_too_large` أو `omitted_invalid`

5.4 ملخص أحدث الأحداث (اختياري)

```
curl -s "https://api.bnetlink.com/connect/v1/sync/status" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

للنظر السريع إن كانت هناك أحداث جديدة؛ **ليس** مؤشر علامة مائية (watermark)، ولا يغني عن نقطة الاستمرار الخاصة بكم.

---

6. Webhook (اختياري)

بعد تسجيل عنوان HTTPS للاستدعاء العكسي، يرسل Connect عند حدوث الأحداث طلباً POST إليكم (بتوقيع).

موصى به: استخدموا Webhook **للإيقاظ فقط**، وابقوا البيانات المرجعية على استمرار القراءة عبر `GET /events` .

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/webhooks" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"url\":\"https://example.com/hooks/connect\",\"secret\":\"at-least-16-chars!\",\"ever
```

أخرى: `GET /webhooks` (قائمة)، `POST /webhooks/{id}/test` ، `PUT /webhooks/{id}` .

لا يوجد `GET /webhooks/{id}` منفصل: خذوا حقول التفاصيل من نتيجة القائمة حسب `id` .

يمكن لـ `PUT` تغيير `status` (`ACTIVE` | `DISABLED`) / `event_types` / `url`)؛ عند تغيير `secret` املأوا القيمة الجديدة، و**إغفال الحقل يبقي السر السابق**. `event_types` فارغ يعني الاشتراك في كل الأنواع.

نقاط أساسية:

- عنوان الإنتاج العام يجب أن يكون **HTTPS**، ولا يجوز أن يشير إلى شبكة خاصة / حلقة محلية.
- `secret` من 16 إلى 128 حرفاً، من اختياركم؛ **لا تستخدموا** API Key كـ Webhook secret.
- القائمة / التفاصيل **لم تعد تُرجع** النص الصريح لـ secret؛ احفظوه عند الإنشاء.
- جسم HTTP للاستدعاء = JSON عمل الحدث (نفس ما بعد تحليل `payload_json`)، **وليس** صورة.

التحقق من التوقيع (إلزامي)

يستخدم Connect HMAC-SHA256:

```
سلسلة التوقيع = timestamp + "." + raw_body
X-Epordo-Signature = hex_lower( HMAC_SHA256(secret, سلسلة التوقيع) )
```

رؤوس الطلب تشمل أيضاً: `X-Epordo-Event-Type` ، `X-Epordo-Event-Id` ، `X-Epordo-Timestamp` .

يجب على المستقبل التحقق بالتوقيع على **بايتات الجسم الخام**؛ أي **2xx** يعني النجاح؛ عند الفشل تُعاد المحاولة، وقد ينتهي الأمر بـ `DEAD` .

مثال (كود وهمي):

```
expected = hmac_sha256_hex(secret, header["X-Epordo-Timestamp"] + "." + rawBody)
if expected != header["X-Epordo-Signature"]: reject
else: upsert_by_event_id(JSON.parse(rawBody)); return 200
```

قائمة التسليم (لتشخيص الأعطال)

الطريقة	المسار	الوصف
GET	`/webhooks/outbox`	صفوف القائمة: `DEAD` / `SENT` / `FAILED` / `PENDING`
GET	`/webhooks/deliveries`	تدقيق كل محاولة (رمز الحالة، المدة، ملخص الخطأ)
POST	`/webhooks/outbox/{id}/replay`	يمكن إعادة التشغيل فقط لـ `DEAD` / `FAILED`

مثال: بعد استلام Webhook سجّلوا فقط «هناك بيانات جديدة»، ثم استدعوا فوراً `GET /events` بنافذة الوقت المحفوظة سابقاً؛ هكذا حتى لو فُقد استدعاء واحد لن تُفقد بيانات الأعمال.

6.1 MQTT (اختياري، يمكن فتحه مع Webhook معاً)

يمكن لـ Connect إرسال نفس JSON الحدث الذي يُرسل إلى Webhook إلى Topic ثابت. يمكن تشغيله **بالتوازي** مع Webhook؛ أزيلوا التكرار بـ `event_id` .

البند	الوصف

Topic	`epc/{tenant_id}/events/{event_type}` (ثابت من المنصة)
الحمولة	نفس JSON العمل للحدث / Webhook
QoS	0 أو 1 (الافتراضي 1)
Broker	تقدّمه المنصة؛ في صفحة MQTT بالبوابة **أصدروا حساب اشتراك** للحصول على اسم المستخدم / كلمة المرور
الاتصال	الخوادم غالباً TCP؛ المتصفحات غالباً WSS (العنوان المحدد في تلميح البوابة)

```
# تسجيل الأحداث المطلوب دفعها (سجل واحد لكل مستأجر)
curl -s -X POST "https://api.bnetlink.com/connect/v1/mqtt" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"event_types\":[],\"qos\":1}"

# إصدار حساب اشتراك (كلمة المرور تُعرض مرة واحدة فقط)
curl -s -X POST "https://api.bnetlink.com/connect/v1/mqtt/credentials" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

ملاحظة: حساب الاشتراك **ليس** API Key، و**ليس** Webhook secret. اتصال الأجهزة ما يزال عبر خادم الأجهزة، و**لن** يتحوّل إلى اتصال MQTT مباشر من الجهاز.

---

7. الأوامر القياسية

7.1 الإنشاء والاستعلام

```
# الإرسال (مزامنة الوقت كمثال)
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/set-time" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{}"

# استعلام الحالة
curl -s "https://api.bnetlink.com/connect/v1/commands/{command_id}" \
  -H "Authorization: Bearer YOUR_API_KEY"

# الإلغاء ممكن فقط في QUEUED
curl -s -X POST "https://api.bnetlink.com/connect/v1/commands/{command_id}/cancel" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

حقول شائعة عند نجاح الإنشاء:

```
{
  "command_id": "cmd_a1b2c3d4e5f6",
  "sn": "TEST-SN-001",
  "command": "set-time",
  "state": "QUEUED"
}
```

الحقل	الوصف
`command_id`	مثل `cmd_` + ست عشري
`state`	`QUEUED` → `SENT` → `SUCCEEDED` / `FAILED` / `CANCELLED`
`result`	عند الحالة النهائية قد يوجد كائن نتيجة (حسب الأمر)

- رأس اختياري (128≤) `X-Idempotency-Key`: نفس المفتاح + نفس الجسم → يُعاد الأمر الأصلي؛ نفس المفتاح + جسم مختلف → `409` `idempotency_conflict`.
- عند اتصال الجهاز: الأوامر العادية تفشل تقريباً بعد 3 دقائق بلا تقدّم؛ خمول بين حزم القراءة المجزأة حوالي 10 دقائق. على الجهاز نفسه يُسمح بحد أقصى بأمر `SENT` واحد في الوقت ذاته.
- عند انقطاع الجهاز يبقى الأمر في `QUEUED` ويُرسل بعد الاتصال (ما يزال خاضعاً للمهلة والحصة).

مثال: بسبب اهتزاز الشبكة أعدتم فتح الباب مرتين. باستخدام نفس `X-Idempotency-Key` ونفس JSON تدخل القائمة مرة واحدة فقط، فلا يُفتح الباب مرتين.

7.2 قائمة الأوامر

الاسم القياسي	لاحقة المسار	الدور
`set-time`	`set-time`	مزامنة الوقت
`query-status`	`query-status`	استعلام حالة الجهاز
`unlock`	`unlock`	فتح الباب عن بُعد / فتح مستمر / إغلاق
`set-user`	`set-user`	إرسال تسجيل شخص واحد (مع قوالب اختيارية)
`delete-user`	`delete-user`	حذف شخص واحد أو أكثر
`list-user-ids`	`list-user-ids`	قراءة قائمة أرقام الأشخاص
`get-user`	`get-user`	قراءة الشخص وتسجيله على المنصة
`set-access-schedule`	`set-access-schedule`	فترات المرور
`clear-managers`	`clear-managers`	مسح علامة المدير (دون حذف الأشخاص)
`pull-logs`	`pull-logs`	سحب تكميلي لسجلات الحضور حسب نافذة الوقت

حالياً **غير متوفر**: واجهة التسجيل عن بُعد، إعدادات الرنين، إلخ. لا تربطوا بها.

7.3 `set-time`

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/set-time" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"time\":\"20261108120000\"}"
```

`time` اختياري، بالصيغة `yyyyMMddHHmmss` ؛ عند الإغفال تُستخدم ساعة الخادم المحلية وقت أخذ الأمر.

7.4 `query-status`

بلا جسم. عند النجاح يكون `result.model` هو `dev_model` لهذه الرسالة (P5 و P7 يرسلان `P5`)، وليس طراز البيع `model_code` . ويضم البرنامج الثابت والسعة وغيرها (البنود التي لا يدعمها الجهاز قد تكون `null`).

7.5 `unlock`

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/unlock" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"mode\":\"pulse\"}"
```

mode	المعنى
`pulse` (افتراضي)	فتحة واحدة ثم إغلاق حسب تأخير الجهاز
`hold`	إبقاء الباب مفتوحاً
`close`	إغلاق الباب

`SUCCEEDED` يعني فقط أن الجهاز أكّد التنفيذ، و**لا يعني** أن مغناطيس الباب قد فُتح.

7.6 `set-user` (شخص واحد · جهاز واحد)

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/set-user" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"user_id\":\"1001\",\"name\":\"Zhang San\",\"role\":\"user\",\"card\":\"123456\"}"
```

مثال حقول كامل:

```
{
  "user_id": "1001",
  "name": "Zhang San",
  "role": "user",
  "card": "123456",
  "password": "666",
  "valid_from": "20260101",
  "valid_to": "20261231",
  "replace": false,
  "face": "<Base64>",
  "photo": "<Base64>",
  "fingerprint": ["<Base64>"],
  "palm": ["<Base64>"]
}
```

- `user_id` إلزامي؛ `palm` / `fingerprint` بحد أقصى 10 فتحات، و**المصفوفة الفارغة تعني مسح ذلك النمط**.
- Connect **لا يقوم** بتحويل صيغة القوالب.
- لفترات المرور استخدموا `set-access-schedule`، ولا تضعوها في هذا الأمر.

7.7 `delete-user` (يمكن لعدة أشخاص · جهاز واحد)

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/delete-user
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"user_ids\":[\"1001\",\"1002\"]}"
```

- أمر واحد يمكنه حذف عدة أشخاص (بحد أقصى 100)؛ لا ترسلوا `user_id` و `user_ids` معاً.
- **لا يوجد** «مسح كل الأشخاص على الجهاز بالكامل».

7.8 `get-user` / 7.9 `list-user-ids`

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/list-user-i
  -H "Authorization: Bearer YOUR_API_KEY" -H "Content-Type: application/json" -d "{}"

curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/get-user" \
  -H "Authorization: Bearer YOUR_API_KEY" -H "Content-Type: application/json" \
  -d "{\"user_ids\":[\"1001\"]}"
```

- يمكن لـ `get-user` أيضاً `{"all":true}` (تخزين صامت لتجنب عاصفة الأحداث).
- بعد تخزين معرّفات محددة يظهر عادةً `enroll.data.received`.

7.10 `set-access-schedule`

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/devices/TEST-SN-001/commands/set-access-
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"time_groups\":{\"mon\":[\"09001200\",\"14001800\"],\"tue\":[\"09001800\"]},\"user_id
```

الحقل	الوصف
`time_groups`	إلزامي. المفاتيح `sat` … `sun` ؛ القيم مصفوفات سلاسل **1–6** ، `HHmmHHmm` **مقاطع يومياً**
`user_ids`	اختياري؛ الإغفال / الفارغ = كل الأشخاص على الجهاز

الفترات موجودة على الجهاز فقط. للتحقق يمكن إعادة إرسال `get-user` ومراجعة `access_schedules` .

مثال: الإثنين صباحاً 09:00–12:00 ومساءً 14:00–18:00 → `"mon": ["09001200", "14001800"]` .

7.11 `clear-managers` / `pull-logs`

- `clear-managers` : بلا جسم؛ مسح علامة المدير.
- `pull-logs` : `{"from":"…","to":"…"}` (8 أو 14 رقماً، المدى ≥31 يوماً).

---

8. دفع جماعي حسب قائمة الأرقام التسلسلية (fan-out)

```
POST /commands/fan-out/set-user
POST /commands/fan-out/delete-user
POST /commands/fan-out/set-access-schedule
```

حقل مشترك: `sns` (مصفوفة سلاسل غير فارغة).

8.1 المبادئ

على الجهاز نفسه، ادمجوا العملية ذاتها قدر الإمكان في أمر **واحد** (بما يوافق قدرة الجهاز)، وليس «طلب HTTP واحد لكل شخص».

الواجهة	الصيغة الموصى بها	سلوك القائمة
إرسال الأشخاص	`users:[{user_id,…}]`	**أمر واحد لكل SN**
حذف الأشخاص	`user_ids:[…]`	**أمر واحد لكل SN**
فترات المرور	`time_groups` مشتركة + `user_ids` اختياري	**أمر واحد لكل SN**

8.2 الحدود (افتراضي)

البند	الافتراضي
عدد `sn` المختلفة	60 (قد ينخفض إلى 30 مع القوالب الكبيرة)
عدد معرّفات الأشخاص	100
عدد أوامر القائمة الناتجة	200

التجاوز ← `400` `fan_out_too_large`.

8.3 إرسال أشخاص إلى أجهزة متعددة (مثال)

يُفضّل استخدام `users[]` (لكل شخص اسم / بطاقة / قالب مستقل)، و**لكل SN يدخل أمر واحد فقط** إلى القائمة:

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/commands/fan-out/set-user" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"sns\":[\"SN-A\",\"SN-B\"],\"users\":[{\"user_id\":\"1001\",\"name\":\"Zhang San\",\'
```

قوالب الوجه الكبيرة وغيرها قد تتجاوز مخزن الجهاز الواحد؛ قسّموا الصفحات أو قلّلوا القوالب ثم أرسلوا.

8.4 حذف عدة أشخاص على أجهزة متعددة (مثال)

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/commands/fan-out/delete-user" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d "{\"sns\":[\"SN-A\",\"SN-B\"],\"user_ids\":[\"1001\",\"1002\"]}"
```

شكل الاستجابة:

```
{
  "command": "delete-user",
  "queued": [
    { "sn": "SN-A", "user_ids": ["1001", "1002"], "command_id": "cmd_…" }
  ],
  "skipped": [
    { "sn": "SN-X", "reason": "device_disabled" }
  ]
}
```

استطلعوا كل `command_id` على حدة. قيم `skipped.reason` الشائعة: `device_disabled`، `device_not_found`، وما يتعلق بالحصة.

8.5 نفس مجموعة فترات المرور على أجهزة متعددة (مثال)

```
{
  "sns": ["SN-001", "SN-002"],
  "time_groups": {
    "mon": ["09001200", "14001800"],
    "tue": ["09001800"]
  },
  "user_ids": ["1001", "1002", "1003"]
}
```

---

9. مجموعات الأجهزة و Scoped Key

بادئة المسار: `/device-groups` . إنشاء / إعادة تسمية / تفعيل أو إيقاف المجموعة يحتاج مفتاح ACCOUNT على مستوى الحساب بالكامل.

لا يوجد `DELETE /device-groups/{id}` : للإيقاف استخدموا `PUT` وضبطوا `status` إلى `DISABLED` .

الطريقة	المسار	الوصف	
GET	`/device-groups`	قائمة	
POST	`/device-groups`	إنشاء: `{"code":"store-a","name":"متجر أ"}`	
GET	`/device-groups/{id}`	التفاصيل + أجهزة الأعضاء	
PUT	`/device-groups/{id}`	تغيير `name` / `status` (`ACTIVE` \ `DISABLED` ؛ لا يمكن تغيير `code`)	
PUT	`/device-groups/{id}/devices/{sn}`	إضافة جهاز مُطالَب به	
DELETE	`/device-groups/{id}/devices/{sn}`	إزالة عضو	
POST	`/device-groups/{id}/commands/set-time`	مزامنة وقت جماعية داخل المجموعة	
POST	`/device-groups/{id}/commands/query-status`	استعلام حالة جماعي داخل المجموعة	
POST	`/device-groups/{id}/commands/unlock`	فتح باب جماعي داخل المجموعة (`mode` كما في الجهاز الواحد)	

```
curl -s -X POST "https://api.bnetlink.com/connect/v1/device-groups" \
  -H "Authorization: Bearer YOUR_API_KEY" -H "Content-Type: application/json" \
  -d "{\"code\":\"store-a\",\"name\":\"متجر أ\"}"

curl -s -X PUT "https://api.bnetlink.com/connect/v1/device-groups/1/devices/TEST-SN-001" \
  -H "Authorization: Bearer YOUR_API_KEY"

curl -s -X POST "https://api.bnetlink.com/connect/v1/device-groups/1/commands/query-status" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

استجابة الدفع الجماعي على مستوى المجموعة دائماً **200** (يُسمح بنجاح جزئي)، والشكل مشابه لـ fan-out: `skipped[]` / `queued[]`.

ملاحظات:

- الأشخاص / الحذف / فترات المرور **ليس لها** دفع جماعي على مستوى المجموعة؛ استخدموا fan-out §8 (`sns` صريحة).
- يمكن إصدار API Key بـ `scope_type=GROUPS` لمجموعة (انظر §2.5).
- مفتاح المجموعة **لا يمكنه** تسجيل الدخول إلى البوابة، ولا المطالبة / فك الربط / إدارة Webhook أو MQTT أو المفاتيح أو المجموعات نفسها.
- عندما تكون حالة المجموعة `DISABLED`، قد يُرجع الإرسال على مستوى المجموعة `group_disabled` `409`.

---

10. تحديد المعدل والحصة اليومية (ملخص)

- **مستوى الدقيقة**: معدل الطلبات حسب المفتاح؛ عند البلوغ `rate_limited` `429` (قد تحمل الاستجابة `Retry-After`).
- **الحصة اليومية (يوم UTC)**: فتح الباب، القراءة الكاملة، السحب التكميلي، إرسال / حذف الأشخاص، الفترات وغيرها من «الأوامر المكلفة»؛ الاستخدام الشاذ قد يوقف المفتاح تلقائياً (`quota_exceeded_key_revoked`).
- إعادة التشغيل idempotent عادةً **لا تستهلك** الحصة مجدداً. الأرقام الدقيقة وفق العقد / إعداد المنصة.

---

11. مسار الربط الموصى به

11.1 الحلقة الدنيا (الحضور)

1. احصلوا على API Key، وأكّدوا بـ `GET /me`.
2. `POST /devices` للمطالبة؛ املأوا عنوان خادم الأجهزة على الجهاز ووصّلوه.
3. (اختياري) سجّلوا Webhook أو MQTT.
4. دورياً أو بعد الإيقاظ استمروا بالقراءة عبر `GET /events`؛ احفظوا التقدّم بأنفسكم.
5. عند الحاجة إلى اللقطة: `GET /events/{id}/photo`.

11.2 إرسال الأشخاص

1. حافظوا على الأشخاص والقوالب في جانب الأعمال.
2. جهاز واحد: `set-user` ؛ أجهزة متعددة: `fan-out/set-user` .
3. نفس قواعد المرور: `set-access-schedule` أو fan-out مع `time_groups` مشتركة.
4. التحقق: `get-user` / `list-user-ids` .

11.3 أسطول من نفس السلسلة

- تبقى موديلات البيع منفصلة (P5 و P7).
- لا تنفّذوا `get-user` كاملاً على كل جهاز.
- اجعلوا ERP أو جهازاً ممثلاً واحداً هو المرجع؛ على الأجهزة الأخرى استخدموا `list-user-ids` للمطابقة ثم سدّوا الفجوات.
- قائمة `GET /devices/{sn}/users?scope=compatible` تعبر الأجهزة فقط إذا عيّن المصنع نفس العائلة وكان البرنامج الثابت متطابقاً وغير فارغ. `dev_model` في الترويسة لا يغيّر موديل البيع. البرنامج الفارغ لا يعبر الأجهزة.

الملحق أ. رموز الأخطاء الشائعة

code	سيناريو نموذجي
`missing_bearer` / `empty_key` / `invalid_key`	فشل المصادقة
`missing_sn`	المطالبة بلا رقم تسلسلي
`device_not_in_stock`	الرقم التسلسلي غير موجود في مخزون الشركة المصنّعة
`device_not_for_tenant`	المخزون غير مخصّص لهذا الحساب
`device_not_found`	الجهاز غير موجود أو خارج الصلاحية
`device_already_registered` / `sn_already_registered`	تسجيل مكرر
`device_disabled`	الجهاز موقوف
`invalid_set_user` / `invalid_delete_user` / `invalid_access_schedule`	جسم الأمر غير صالح
`invalid_fan_out` / `fan_out_too_large`	خطأ في معاملات الدفع الجماعي أو تجاوز الحد
`command_not_found` / `command_not_cancelable`	استعلام / إلغاء الأمر
`idempotency_conflict`	تعارض مفتاح idempotency
`rate_limited` / `quota_exceeded` / `quota_exceeded_key_revoked`	تحديد المعدل / الحصة

`invalid_webhook_secret` / `invalid_webhook_url`	معاملات Webhook
`forbidden_scope`	مفتاح مجموعة استدعى واجهة على مستوى الحساب
`last_active_api_key`	محاولة إيقاف آخر مفتاح فعّال
`api_key_not_found`	معرّف المفتاح غير موجود
`mqtt_already_exists`	المستأجر سجّل بالفعل خروجاً لـ MQTT؛ لا تكرروا POST
`outbox_not_replayable`	حالة قائمة التسليم غير قابلة لإعادة التشغيل
`group_disabled` / `group_not_found`	المجموعة غير موجودة أو موقوفة

الملحق ب. أنواع **kind** لبيانات الاعتماد

kind	الفتحة	الوصف
`photo`	0	صورة ضوء مرئي
`face`	0	قالب الوجه
`fingerprint`	0–9	بصمة الإصبع
`palm`	0–9	بصمة الكف / وريد الكف
`card`	0	رقم البطاقة (UTF-8)
`password`	0	كلمة مرور الجهاز (UTF-8)

أي كتلة بعد فك الترميز تتجاوز حوالي **1MB** تُرفض. المنصة لا تحوّل القوالب عبر البرامج الثابتة المختلفة.

الملحق ج. الوثائق والدعم

المادة	الوصف
هذا المستند / PDF	دليل الواجهة العلني بالعربية (يمكن تنزيل PDF من «المساعدة» في البوابة)
البوابة	المطالبة من المتصفح، تجربة الأوامر، إصدار حساب اشتراك MQTT، إلخ
ApiPost / Postman	مجموعة استيراد يقدّمها تسليم المشروع (مرفق تقني)
ملاحظات المشكلات	قنوات الدعم التجاري / التقني

الملحق د. مرجع سريع للواجهات

التصنيف	الطريقة والمسار					
فحص الحياة / الهوية	`GET` · `GET /me` · `GET /health`	POST /credentials · PUT `/credentials/{id}`				
الأجهزة	`POST`	GET /devices · GET	DELETE /devices/{sn} · POST .../disable	enable · GET `...../users`		
الأحداث	`GET /events` · `GET /events/{id}` · `GET /events/{id}/photo` · `GET /sync/status`					
Webhook	`GET`	POST /webhooks · PUT /webhooks/{id} · POST .../test · GET .../outbox	POST deliveries · `.../outbox/{id}/replay` (لا يوجد GET منفصل بالمعرّف)			
MQTT	`GET`	POST /mqtt · PUT	DELETE /mqtt/{id} · GET · POST .../test	POST	`LETE als` · `eplay`	
الأوامر	`POST /devices/{sn}/commands/{name}` · `GET /commands/{id}` · `POST /commands/{id}/cancel`					
الدفع الجماعي	`POST /commands/fan-out/set-user`	delete-user	`set-access-schedule`			
المجموعات	`GET`	POST /device-groups · GET	PUT /device-groups/{id} · `PUT/DELETE أعضاء على مستوى المجموعة` · set-time / query-status / `unlock` (لا يوجد DELETE للمجموعة نفسها)			

أسماء الأوامر القياسية `list-user-ids` ، `get-user` ، `set-access-schedule` ، `clear-managers` ، `pull-logs` ، `delete-user` ، `set-user` ، `unlock` ، `query-status` ، `set-time` : `{name}`.

نهاية المستند · دليل واجهة EPORDO Connect الإصدار v1.1.5