

Dış API Entegrasyon Kılavuzu

Plaka • Guard • Varlık • İş Emri • Fire Al

SGE Vision verilerine Zenon SCADA gibi üçüncü taraf sistemlerden HTTPS üzerinden, anahtar korumalı REST API ile erişim sağlanır. Yetkili anahtarlarla alarm onayı, iş emri, abone ve varlık işlemleri gerçekleştirilebilir.

DOKÜMAN BİLGİLERİ

Tarih	07 Ekim 2026
Hazırlayan	Sultan ERKOC
API	HTTPS / REST API / JSON
Temel adres	https://<SUNUCU-ADRESİ>/api/ext/v1

2. Ortak kurallar

Aşağıdaki kurallar tüm modüllerde aynıdır.

Yanıt biçimi

Başarılı yanıtlar aynı zarfla gelir; liste uç noktalarında ayrıca `paging` alanı bulunur.

```
{
  "success": true,
  "generated_at": "2026-10-07T12:00:00+03:00",
  "data": [ ... ],
  "paging": { "limit": 100, "offset": 0, "total": 250, "has_more": true }
}
```

Hatalı yanıtlar `success: false` ile gelir:

```
{ "success": false, "error": { "code": "invalid_parameter", "message":
  "'limit' 1 ile 1000 arasında olmalı" } }
```

Zaman ve aralık

- Tüm zamanlar ISO 8601 ve saat dilimi farkıyla döner (`2026-10-05T08:02:11+03:00`).
- `from` ve `to` parametreleri yalnızca tarih (`2026-10-05`) ya da tarih-saat (`2026-10-05T08:00:00`) olabilir. Yalnızca tarih verilirse `to` o günü **dahil eder**.
- Aralık verilmezse varsayılan **son 24 saattir** (PDF plaka raporunda son 30 gün).
- Tek istekte en fazla **93 günlük** aralık sorgulanır; daha uzunlu için aralığı bölün.

Sayfalama

`limit` (sayfa başına kayıt) ve `offset` (atlanacak kayıt) kullanılır. Varsayılan `limit` 100, üst sınır uç noktaya göre 200 ile 1000 arasındadır. Yanıttaki `paging.has_more` değeri `true` ise `offset`'i `limit` kadar artırarak sonraki sayfayı alın. Guard tespitleri istisnadır; `paging.next_cursor` değeri `cursor=` olarak gönderilir.

İstek sınırı

Her anahtarın dakikada istek sınırı vardır. Yanıtlardaki `X-RateLimit-Limit` ve `X-RateLimit-Remaining` başlıkları kalan hakkı gösterir. Sınır aşılsa `429` döner ve `Retry-After` başlığındaki saniye kadar beklenir.

Hata kodları

HTTP	code	Anlamı
400	<code>invalid_parameter</code> , <code>range_too_large</code> , <code>invalid_json</code>	Parametre ya da gövde hatalı; aralık çok uzun
401	<code>missing_key</code> , <code>invalid_key</code> , <code>key_revoked</code> , <code>key_expired</code>	Anahtar yok, yanlış, pasifleştirilmiş ya da süresi dolmuş
403	<code>ip_not_allowed</code> , <code>insufficient_scope</code> , <code>license_inactive</code> , <code>module_not_licensed</code>	IP izinli değil; yetki yok; lisans etkin değil ya da modül lisanslı değil
404	<code>not_found</code>	Kayıt, fotoğraf ya da PDF bulunamadı
409	<code>already_exists</code> , <code>conflict</code> , <code>confirmation_required</code>	Yazma işleminde çakışma (örn. plaka/seri no zaten kayıtlı)
429	<code>rate_limited</code> , <code>too_many_failures</code>	İstek sınırı aşıldı; çok fazla yanlış anahtar denemesi
502 / 503	<code>guard_error</code> , <code>guard_unavailable</code> , <code>storage_unavailable</code>	Guard servisine ya da veri deposuna ulaşılamıyor; biraz sonra tekrar deneyin

3. Yetkiler (scope)

Bir anahtar yalnızca kendisine verilen yetkilerin uç noktalarına erişir. Yetkiler yönetici panelinde modül başlıkları altında işaretlenir; **yazma yetkileri kırmızı** gösterilir ve okuma yetkisi yazma yetkisi vermez. Kişisel ve hassas veriler ayrı yetkilerle korunur; yetki yoksa ilgili alanlar yanıtta çıkarılır.

Modül	Yetki	Verdiği erişim
Plaka	<code>plate:read</code>	Giriş/çıkış, okumalar, araç, geçitler, PDF rapor
Plaka	<code>plate:person</code>	Abone adı, telefon, personel/birim no; <code>person_no</code> ile arama
Plaka	<code>plate:image</code>	Araç fotoğrafı
Guard	<code>guard:read</code>	Tespitler, arama, İSG alarmları, kameralar, alarm raporu PDF
Guard	<code>guard:image</code>	Tespit ve alarm fotoğrafları

Modül	Yetki	Verdiği erişim
Guard	<code>guard:sensitive</code>	Cinsiyet, yaş gibi hassas özellikler (KVKK)
Varlık	<code>assets:read</code>	Varlık, tip, servis geçmişi, uyarılar, PDF
Varlık	<code>assets:sensitive</code>	IP/MAC adresi, tedarikçi, servis maliyeti, notlar
Operasyon	<code>ops:read</code>	İş emirleri, özet, PDF
Operasyon	<code>ops:person</code>	Personel adları, müdahale kullanıcıları ve notlar
Fire AI	<code>fire_ai:read</code>	Ateş/duman alarmları, olaylar, kameralar
Fire AI	<code>fire_ai:image</code>	Alarm ve olay fotoğrafları
Yazma	<code>guard:ack</code> , <code>fire_ai:ack</code>	Alarm onayı
Yazma	<code>plate:manage</code>	Abone ekleme, güncelleme, aktif/pasif yapma
Yazma	<code>assets:write</code> , <code>assets:delete</code>	Varlık ve servis kaydı ekleme/güncelleme; varlık silme
Yazma	<code>ops:write</code> , <code>ops:approve</code>	İş emri açma, müdahale notu; çözüm onayı/reddi

Bir yetki kendi temel okuma yetkisini gerektiriyorsa (örneğin `plate:image` için `plate:read`), yönetici panelinde seçildiğinde temel yetki otomatik eklenir.

4. Plaka Tanıma API

Temel adres: `/api/ext/v1/plate`. Plaka tanıma sisteminin geçit (bariyer/kamera) kayıtlarını, giriş/çıkış ziyaretlerini ve kayıtlı araç/kişi bilgilerini verir. Puantaj, ziyaretçi takibi ve içeride kalan araç listesi için kullanılır.

Yol (GET)	Ne verir	Parametreler	Yetki
<code>/visits</code>	Giriş/çıkış kayıtları: geçit, zaman, kalış süresi, araç ve kullanıcı	<code>from</code> , <code>to</code> , <code>by</code> , <code>status</code> , <code>plate</code> , <code>match</code> , <code>gate_id</code> , <code>person_no</code> , <code>order</code> , <code>limit</code> , <code>offset</code>	<code>plate:read</code> (<code>person_no</code> : <code>plate:person</code>)

Yol (GET)	Ne verir	Parametreler	Yetki
/inside	Şu an içeride olan araçlar ve ne kadardır içeride oldukları	limit , offset	plate:read
/events	Ham plaka okumaları (her okuma, güven oranı, geçit)	from , to , plate , match , gate_id , direction , color , order , limit , offset	plate:read
/vehicles/{plaka}	Tek aracın bilgisi: tip, renk, kullanıcı, son görülme, son 10 ziyaret	—	plate:read
/gates	Geçit listesi: ad, konum, yön	—	plate:read
/subscribers	Kayıtlı abone/araç listesi: ad, personel no, telefon, araç tipi	active , q , person_no , limit , offset	plate:person
/events/{id}/image	Okuma anındaki araç fotoğrafı (JPEG)	—	plate:image
/report.pdf	Plaka raporu PDF (bkz. Bölüm 9)	plate , from , to	plate:read

Parametre açıklamaları

Parametre	Değerler	Açıklama
by	any (varsayılan), entry , exit	Aralığa hangi zamana göre bakılacağı. any , girişi veya çıkışı aralıkta olan kayıtları getirir (gece vardiyası kaçmaz)
status	all , inside , exited , exit_only	exit_only : girişi okunamamış çıkışlar
match	exact (varsayılan), starts , ends , contains	Plaka eşleşme biçimi. Plaka büyük/küçük harf ve boşluğa duyarlıdır (34 abc 123 = 34ABC123)

Parametre	Değerler	Açıklama
direction	entry, exit	Yalnızca giriş ya da çıkış geçidinde yapılan okumalar
person_no	metin	Kullanıcı kaydındaki Daire / Blok / Personel No alanı

Örnek: bir kişinin aylık giriş/çıkışları (puantaj)

```
curl -H "Authorization: Bearer ANAHTAR" \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/plate/visits?from=2026-10-01&to=2026-10-31&person_no=P102&order=asc"
```

```
{
  "success": true,
  "data": [
    {
      "id": 4821,
      "plate": "34ABC123",
      "status": "exited",
      "entry": { "time": "2026-10-05T08:02:11+03:00", "gate": { "id": 1,
"name": "Giriş Kapısı", "direction": "entry" }, "confidence": 0.94,
"image_url": "/api/ext/v1/plate/events/90211/image" },
      "exit": { "time": "2026-10-05T17:41:50+03:00", "gate": { "id": 2,
"name": "Çıkış Kapısı", "direction": "exit" }, "confidence": 0.91 },
      "duration_minutes": 579,
      "vehicle": { "plate": "34ABC123", "type": "BMW 320i", "color": "Beyaz",
"registered": true },
      "person": { "name": "Ayşe Yılmaz", "unit_or_personnel_no": "P102",
"phone": "0532 000 00 00", "active": true }
    }
  ],
  "paging": { "limit": 100, "offset": 0, "total": 1, "has_more": false }
}
```

person bloğu yalnızca plate:person yetkisi olan anahtarlara gelir. image_url yalnızca plate:image yetkisi varsa gelir.

Puantaj için notlar

- Personel eşlemesi:** unit_or_personnel_no alanı ile person_no parametresi aynı değerdir; personel kaydında bu alanın doldurulmuş olması gerekir.

- **Eksik kayıtlar:** `status=exit_only` girişi okunamayan çıkışları (`entry: null`), `status=inside` hâlâ içeride görünenleri verir. Bunlar puantajda ayrıca ele alınmalıdır.
- **Doğruluk:** her okumada `confidence` (0-1) bulunur; düşük güvenli okumalar yanlış plaka olabilir.

5. SGE Guard API

Temel adres: `/api/ext/v1/guard`. Kamera görüntülerinden çıkarılan **kişi ve araç tespitlerini** (üst/alt renk, kıyafet, araç tipi, plaka vb.) ve **İSG alarmlarını** (baretsiz, yeleksiz, telefon, yatan kişi, yasak bölge, sigara) verir.

Yol (GET)	Ne verir	Parametreler	Yetki
<code>/detections</code>	Kişi/araç tespitleri; özelliğe göre arama	<code>from, to, type, camera_id, attr. <özellik>, vehicle_class, plate, text, similar_to, min_similarity, limit (en çok 200), cursor</code>	<code>guard:read</code>
<code>/detections/{id}</code>	Tek tespitin ayrıntısı	—	<code>guard:read</code>
<code>/alarms</code>	İSG alarmları	<code>from, to, status, types, camera_id, limit (en çok 1000)</code>	<code>guard:read</code>
<code>/catalog</code>	Aranabilir özellikler ve izinli değerler (Türkçe adlarıyla; açılır liste için)	—	<code>guard:read</code>
<code>/cameras</code>	Kamera listesi	—	<code>guard:read</code>
<code>/image/{yol}</code>	Fotoğraf (JPEG); ? <code>format=base64</code> ile JSON içinde base64	<code>format</code>	<code>guard:image</code>
<code>/alarms/report.pdf</code>	Alarm raporu PDF (bkz. Bölüm 9)	<code>from, to, status, types, camera_id, limit, images</code>	<code>guard:read</code>

Arama mantığı

- `type`: `person` ya da `vehicle`.
- `attr.<özellik>=değer`: aynı özellikte birden çok değer **VEYA** anlamındadır (`attr.upper_color=red,blue`); farklı özellikler **VE** anlamındadır. Özellik adı ve değerleri `/catalog` uç noktasından alınır.
- `plate`: tespit edilen araç plakası; `34*` gibi örnek araması desteklenir.
- `text`: serbest metinle arama. `similar_to=<tespit no>`: aynı kişi/araca benzeyen tespitleri bulur (`min_similarity` ile eşik).
- Alarm türleri (`types`): `no_helmet`, `no_vest`, `phone`, `lying`, `zone`, `smoking`; `status`: `open` ya da `acked`.

Örnek: kırmızı üstlü kişileri bul

```
curl -H "Authorization: Bearer ANAHTAR" \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/guard/detections?
type=person&attr.upper_color=red&from=2026-10-07&limit=50"
```

```
{
  "success": true,
  "data": [
    {
      "id": 120345,
      "time": "2026-10-07T12:00:00+03:00",
      "camera": { "id": 2, "name": "Kapı Önü" },
      "type": "person",
      "confidence": 0.91,
      "attributes": { "upper_color": "red", "lower_type": "trousers" },
      "display": { "upper_color": { "label": "Üst renk", "value": "red",
"value_tr": "Kırmızı" } },
      "image_url": "/api/ext/v1/guard/image/2026/10/07/120345.jpg"
    }
  ],
  "paging": { "limit": 50, "next_cursor": "eyJ0Ijo...", "has_more": true }
}
```

Sonraki sayfa için `paging.next_cursor` değeri `cursor=` olarak gönderilir.

Fotoğraf ve hassas özellikler

`image_url` yalnızca `guard:image` yetkisi varsa gelir ve aynı anahtarla, başlıkla çağrılır. Cinsiyet ve yaş gibi hassas özellikler (KVKK kapsamı) `guard:sensitive` yetkisi yoksa yanıtta çıkarılır ve bunlarla arama yapılamaz.

Örnek: açık alarmlar

```
curl -H "Authorization: Bearer ANAHTAR" \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/guard/alarms?
  status=open&types=no_helmet,no_vest&from=2026-10-07"
```

6. Varlık Yönetimi API

Temel adres: `/api/ext/v1/assets` . Cihaz ve ekipman envanterini (seri no, tip, marka/model, konum, durum, garanti), servis geçmişini ve yaklaşan servis/garanti uyarılarını verir.

Yol (GET)	Ne verir	Parametreler	Yetki
<code>/items</code>	Varlık listesi	<code>type_id</code> , <code>status</code> , <code>q</code> , <code>brand</code> , <code>model</code> , <code>location</code> , <code>parent_id</code> , <code>warranty_before</code> , <code>updated_since</code> , <code>order</code> , <code>dir</code> , <code>limit</code> (en çok 500), <code>offset</code>	<code>assets:read</code>
<code>/items/{id}</code>	Tek varlık: Üst/alt varlıklar, son 10 servis, son 10 arıza raporu, sonraki servis tarihi	—	<code>assets:read</code>
<code>/items/serial/{seri_no}</code>	Seri numarasıyla aynı ayrıntı (barkod/QR okutma için)	—	<code>assets:read</code>
<code>/items/{id}/services</code>	Servis geçmişinin tamamı	<code>limit</code> , <code>offset</code>	<code>assets:read</code>
<code>/due</code>	Uyarı listesi: servisi gelen/geçen ve garantisi bitecek varlıklar	<code>days</code> (varsayılan 30)	<code>assets:read</code>

Yol (GET)	Ne verir	Parametreler	Yetki
/summary	Özet: duruma/tipe göre adet, garanti ve servis uyarı sayıları	—	assets:read
/types	Varlık tipleri ve her tipteki adet	—	assets:read
/items/{id}/pdf	Varlık PDF'i (bkz. Bölüm 9)	—	assets:read

Parametre açıklamaları

Parametre	Değerler	Açıklama
status	active, maintenance, faulty, retired	Varıl durumu: Aktif, Bakımda, Arızalı, Hizmet dışı
type_id	sayı listesi (1, 2)	/types uç noktasındaki tip numaraları
q	metin	Ad, seri no ve benzeri alanlarda serbest arama (Türkçe karakterlere duyarlı)
warranty_before	tarih	Garantisi bu tarihten önce biten varlıklar
updated_since	tarih/zaman	Bu andan sonra değişenler (değişiklik aktarımı için)
order	name, serial, created, updated, warranty	Sıralama alanı; dir parametresi yönü belirler (asc: artan, desc: azalan)

Örnek: arızalı varlıklar

```
curl -H "Authorization: Bearer ANAHTAR" \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/assets/items?status=faulty&limit=50"
```

```
{
  "success": true,
  "data": [
    {
      "id": 12,
      "serial_no": "SW-0412",
      "name": "Pano A Switch",
      "type": { "id": 1, "name": "Switch" },
      "brand": "Cisco", "model": "C1000",
      "location": "Pano A",
      "status": "faulty", "status_tr": "Arızalı",
      "parent_id": null,
      "purchase_date": "2025-03-01",
      "warranty": { "end": "2026-10-20", "days_left": 13, "expired": false }
    }
  ],
  "paging": { "limit": 50, "offset": 0, "total": 1, "has_more": false }
}
```

`assets:sensitive` yetkisi yoksa IP ve MAC adresi, tedarikçi, servis maliyeti ve notlar yanıtta çıkarılır. SCADA uyarı panosu için `/due?days=30` ve `/summary` yeterlidir; tek bir cihazın ayrıntısı için `/items/serial/{seri_no}` kullanın.

7. Operasyon / İş Emri API

Temel adres: `/api/ext/v1/ops`. Operasyon raporlarını (arıza kayıtları / iş emirleri), müdahale zaman çizelgesini, bağlı varlıkları ve çözüm onay bilgisini verir.

Yol (GET)	Ne verir	Parametreler	Yetki
<code>/orders</code>	İş emri listesi	<code>from</code> , <code>to</code> , <code>state</code> , <code>status</code> , <code>equipment_type</code> , <code>q</code> , <code>asset_id</code> , <code>technician</code> , <code>supervisor</code> , <code>order</code> , <code>dir</code> , <code>limit</code> (en çok 500), <code>offset</code>	<code>ops:read</code> (<code>technician</code> , <code>supervisor</code> : <code>ops:person</code>)
<code>/orders/{id}</code>	Tek iş emri: müdahale zaman çizelgesi, bağlı varlıklar, onay bilgisi	—	<code>ops:read</code>

Yol (GET)	Ne verir	Parametreler	Yetki
/orders/wo/{iş_emri_no}	İş emri numarasıyla aynı ayrıntı (örn. IS-041)	—	ops:read
/summary	Özet: durum sayıları, açık iş emirleri (ekipmana göre), ortalama çözüm süresi, en eski açık iş emri	days (varsayılan 30)	ops:read
/equipment-types	Ekipman tipleri ve kayıt sayıları	—	ops:read
/orders/{id}/pdf	İş emri PDF'i (bkz. Bölüm 9)	—	ops:read

Parametre açıklamaları

Parametre	Değerler	Açıklama
state	open , closed , all	Açık ya da kapalı iş emirleri
status	arizali , calisiyor , bakimda , cozum_onay , cozuldu	Ayrıntılı durum; virgülle birden çok değer verilir. cozum_onay : çözüldü, yönetici onayı bekliyor
from , to	tarih	İş emrinin açılış zamanına göre aralık
asset_id	sayı	Bir varlığın arıza geçmişi (varlık numarası = Varlık API'sindeki id)
order	opened , updated , solved , id	Sıralama alanı

Örnek: açık iş emirleri

```
curl -H "Authorization: Bearer ANAHTAR" \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/ops/orders?state=open&limit=50"
```

```
{
  "success": true,
  "data": [
    {
      "id": 41,
      "work_order_no": "IS-041",
      "opened_at": "2026-10-06T09:00:00+03:00",
      "equipment": { "type": "ups", "type_tr": "UPS", "name": "UPS-2" },
      "status": "arizali", "status_tr": "Arızalı (açık)", "is_open": true,
      "fault_reason": "Akü arızası",
      "action_taken": null,
      "solved_at": null, "resolution_minutes": null, "age_minutes": 1984,
      "approval": { "status": null, "requested_at": null, "approved_at": null
    }
  ],
  "paging": { "limit": 50, "offset": 0, "total": 1, "has_more": false }
}
```

Süreler dakika cinsindendir. `ops:person` yetkisi yoksa raporu açan, teknisyen, saha sorumlusu, onaylayan adları, müdahale kullanıcıları ve notlar yanıtta çıkarılır. SCADA'da açık iş emri sayısı için `/summary`, liste için `/orders?state=open` kullanılır.

8. Fire AI (ateş / duman) API

Temel adres: `/api/ext/v1/fire_ai`. Kameralardaki yapay zekâ ateş ve duman alarmlarını verir. Bu modül **lisanslıdır**: lisansta `fire_ai` yoksa `403 module_not_licensed` döner.

Yol (GET)	Ne verir	Parametreler	Yetki
<code>/alarms</code>	Ateş/duman alarmları (kamera başına birleşmiş; operatör onaylayana kadar açık)	<code>from</code> , <code>to</code> , <code>status</code> , <code>verdict</code> , <code>camera_id</code> , <code>class</code> , <code>dir</code> , <code>limit</code> (en çok 1000), <code>offset</code>	<code>fire_ai:read</code>
<code>/alarms/{id}</code>	Tek alarm	—	<code>fire_ai:read</code>
<code>/alarms/{id}/image</code>	Alarm fotoğrafı (JPEG, kutu işaretli)	<code>which</code> , <code>format</code>	<code>fire_ai:image</code>
<code>/summary</code>	Özet: açık alarm sayısı ve listesi, son N saatte alarm/gerçek/yanlış sayıları	<code>hours</code> (varsayılan 24)	<code>fire_ai:read</code>

Yol (GET)	Ne verir	Parametreler	Yetki
/cameras	Kameralar: AI izleme açık mı, tarama aralığı, güven eşiği, açık alarm sayısı (RTSP bilgisi verilmez)	enabled	fire_ai:read
/events	Ham tespitler (her kare tespiti). Aralık verilmezse son 24 saat	from, to, camera_id, class, dir, limit, offset	fire_ai:read
/events/{id}/image	Ham tespit fotoğrafı	format	fire_ai:image

Parametre açıklamaları

Parametre	Değerler	Açıklama
status	open, acked	Açık (onay bekliyor) ya da onaylanmış alarmlar
verdict	real, false, none	Operatör kararı: gerçek alarm, yanlış alarm, karar verilmemiş
class	fire, smoke	Ateş ya da duman
which	first, last	Alarm fotoğrafı için ilk ya da son kare
format	base64	Fotoğrafı JSON içinde base64 olarak verir

Örnek: açık alarmlar

```
curl -H "Authorization: Bearer ANAHTAR" \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/fire_ai/alarms?status=open"
```

```
{
  "success": true,
  "data": [
    {
      "id": 17,
      "camera": { "id": 3, "name": "Depo Girişi" },
      "classes": [ { "name": "fire", "name_tr": "Ateş" }, { "name": "smoke",
"name_tr": "Duman" } ],
      "status": "open", "status_tr": "Açık (onay bekliyor)",
      "first_detected_at": "2026-10-07T10:00:00+03:00",
      "last_detected_at": "2026-10-07T10:02:00+03:00",
    }
  ]
}
```

```

    "duration_seconds": 120,
    "detection_count": 5,
    "max_confidence": 0.81,
    "verdict": null,
    "acknowledged": null,
    "image_first_url": "/api/ext/v1/fire_ai/alarms/17/image?which=first",
    "image_last_url": "/api/ext/v1/fire_ai/alarms/17/image?which=last"
  }
],
"paging": { "limit": 100, "offset": 0, "total": 1, "has_more": false }
}

```

`verdict` operatörün kararıdır; `null` karar verilmemiş ya da henüz onaylanmamış demektir. SCADA'da açık alarm sayısı için `/summary` içindeki `open_alarms` değerini birkaç saniyede bir sorgulayın; sıfırdan büyükse kırmızı uyarı gösterip `image_last_url` ile fotoğrafı açın. Alarmin onaylanması için Bölüm 10'a bakın.

9. PDF çıktıları

Aşağıdaki uç noktalar hazır bir PDF raporu döner. Yeni yetki gerekmez; ilgili modülün **okuma** yetkisi yeter. Varlık ve iş emri PDF'leri web arayüzündekiyle aynı şablonla üretilir.

Yol (GET)	Rapor	Parametreler	Yetki
<code>/plate/report.pdf?plate=34ABC123</code>	Plaka raporu: araç bilgisi, giriş/çıkış kayıtları, okumalar. <code>plate</code> verilmezse dönem özeti (toplam okuma, en sık plakalar, geçit dağılımı)	<code>plate</code> , <code>from</code> , <code>to</code> (varsayılan son 30 gün)	<code>plate:read</code>
<code>/plate/vehicles/{plaka}/pdf</code>	Aynı plaka raporu, adres biçiminde	<code>from</code> , <code>to</code>	<code>plate:read</code>

Yol (GET)	Rapor	Parametreler	Yetki
/assets/items/{id}/pdf ve /assets/items/serial/{seri_no}/pdf	Varlık PDF'i (kart ve servis geçmişi)	—	assets:read
/ops/orders/{id}/pdf ve /ops/orders/wo/{iş_emri_no}/pdf	İş emri PDF'i	—	ops:read
/guard/alarms/report.pdf	Guard alarm raporu (özet ve alarm listesi)	from, to (varsayılan son 24 saat), status, types, camera_id, limit (en çok 500), images	guard:read (images=1 : guard:image)

Yanıt biçimleri

- **Varsayılan:** application/pdf olarak gelir ve tarayıcıda açılır.
- ?download=1 : dosya indirme olarak gelir.
- ?format=base64 : PDF, JSON zarfı içinde base64 alanıyla gelir (filename , size , content_type ile birlikte). Başlık gönderemeyen SCADA bileşenleri için kullanılır.

```
curl -H "Authorization: Bearer ANAHTAR" -o plaka.pdf \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/plate/report.pdf?
plate=34ABC123&from=2026-09-01"
```

```
curl -H "Authorization: Bearer ANAHTAR" -o is_emri.pdf \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/ops/orders/wo/IS-041/pdf"
```

Gizlilik kuralları

PDF'ler JSON API ile aynı yetki kurallarına uyar: plate:person yoksa abone adı, telefon ve personel no; assets:sensitive yoksa IP, MAC, tedarikçi, notlar ve maliyet; ops:person yoksa personel adları ve notlar PDF'te görünmez. Alarm karelerini eklemek (images=1 , en fazla 12 kare) için guard:image gerekir; yoksa 403 döner.

10. Yazma işlemleri

Zenon gibi sistemlerden alarm onayı, iş emri açma, abone ve varlık işlemleri yapılabilir. Yazma uç noktaları **ayrı yetkilerle** açılır; okuma yetkisi yazma yetkisi vermez. İstek gövdesi JSON'dur ve `Content-Type: application/json` başlığı gerekir. Yapan kişi tüm kayıtlarda **"API: <anahtar adı>"** olarak görünür ve her yazma işlemi ayrıca bir yazma günlüğüne işlenir.

Alarm onayı

İstek	Ne yapar	Gövde	Yetki
POST /guard/alarms/ack ve POST /guard/alarms/{id}/ack	Guard İSG alarmlarını onaylar	{"ids":[12,13]}	guard:ack
POST /fire_ai/alarms/ack ve POST /fire_ai/alarms/{id}/ack	Ateş/duman alarmını onaylar; gerçek/yanlış kararını kaydeder	{"ids": [7],"verdict":"real","note":"..."} (verdict: real ya da false)	fire_ai:ack

Onay yanıtı `requested` , `acknowledged` ve zaten onaylı/bulunamayan sayısını (`already_acknowledged_or_missing`) döner. Aynı isteklerin tekrarı zararsızdır.

Plaka aboneleri

İstek	Ne yapar	Gövde	Yetki
POST /plate/subscribers	Abone ekler	{"plate":"34 ABC 123","name":"...","unit_no":"...","phone":"... ","vehicle_type":"...","vehicle_color":"..." ,"subscriber_type":"resident","active":true} (subscriber_type: resident ya da company)	plate:manage
PUT /plate/subscribers/{id}	Abone bilgisini günceller (plaka değişmez)	{"phone":"...","active":false}	plate:manage
POST /plate/subscribers/{id}/deactivate ve .../activate	Aboneyi pasif / yeniden aktif yapar	—	plate:manage

Abone **silme yoktur**; abone geçmişini korunur, bunun yerine pasifleştirilir. Aynı plaka ikinci kez eklenirse `409 already_exists` döner.

Varlık işlemleri

İstek	Ne yapar	Gövde	Yetki
POST /assets/items	Varlık ekler	<code>{"type_id":1,"serial_no":"...", "name":"...", "brand":"...", "model":"...", "location":"...", "status":"active", "warranty_end":"2027-01-31", "metadata":{}}</code>	assets:write
PUT /assets/items/{id}	Varlığı günceller (yalnızca gönderilen alanlar)	<code>{"status":"faulty", "location":"..."}</code>	assets:write
POST /assets/items/{id}/services	Servis kaydı ekler	<code>{"type":"Bakım", "description":"...", "cost":0, "next_service_date":"2027-01-01"}</code>	assets:write
DELETE /assets/items/{id}?confirm={seri_no}	Varlığı kalıcı olarak siler (servis geçmişiyile). Yanlış silmeye karşı seri numarası onayı şarttır	—	assets:delete

Seri numarası benzersizdir (büyük/küçük harf farkı yoktur); çakışırsa 409 already_exists döner. confirm eksik ya da yanlışsa 409 confirmation_required döner.

İş emri işlemleri

İstek	Ne yapar	Gövde	Yetki
POST /ops/orders	İş emri (arıza kaydı) açar; e-posta bildirimini arayüzdeki gibi gönderir	<code>{"equipment_type":"switch", "equipment_name":"...", "fault_reason":"...", "assigned_technician":"...", "assigned_supervisor":"...", "asset_ids":[1], "notify":true}</code>	ops:write
POST /ops/orders/{id}/activities	Müdahale notu ekler	<code>{"note":"..."}</code>	ops:write
POST /ops/orders/{id}/request-approval	İş emrini "çözüldü, onay bekliyor" durumuna alır	<code>{"note":"..."}</code>	ops:write
POST /ops/orders/{id}/approval	Çözümü onaylar (iş emrini kapatır) ya da reddeder. Yalnızca onay bekleyen iş emri için	<code>{"action":"approve", "note":"..."}</code> ya da <code>{"action":"reject", "reason":"..."}</code>	ops:approve

İş emri bu işlem için uygun durumda değilse 409 conflict döner (örneğin onay bekleyen olmayan bir iş emrine onay verilemez).

Başarı kodları ve çift kayıttan korunma

Başarılı oluşturma 201 , güncelleme ve onay 200 döner ve güncel kaydı verir. Oluşturma isteklerine Idempotency-Key: <benzersiz değer> başlığı eklenirse, aynı istek tekrarlandığında (yeniden deneme, çift tıklama) ikinci kayıt açılmaz; ilk yanıt aynı şekilde döner. Bu koruma **48 saat** geçerlidir.

```
curl -X POST -H "Authorization: Bearer ANAHTAR" -H "Content-Type: application/json" \
  -H "Idempotency-Key: zenon-alarm-17" \
  -d '{"ids":[17],"verdict":"false","note":"Kaynak işi, yanlış alarm"}' \
  "https://<SUNUCU-ADRESİ>/api/ext/v1/fire_ai/alarms/ack"
```

11. Entegrasyon önerileri ve güvenlik

Zenon / SCADA için önerilen kullanım

API sorgulama (polling) tabanlıdır; sistem kendiliğinden bildirim göndermez. Aşağıdaki sıklıklar öneridir; toplam istek sayısı anahtarın dakikadaki sınırını (varsayılan 60) aşmamalıdır.

Veri	Önerilen uç nokta	Sıklık
Açık ateş/duman alarm sayısı	/fire_ai/summary	5-10 saniye
Açık İSG alarmları	/guard/alarms?status=open	10-30 saniye
Açık iş emri sayısı	/ops/summary	1 dakika
Varlık uyarı panosu	/assets/due ve /assets/summary	15 dakikada bir ya da saatlik
İçeride kalan araçlar	/plate/inside	30-60 saniye
Puantaj / geçmiş aktarımı	/plate/visits	Günde bir, önceki günün aralığı

Akıllı sorgulama

- Yeni kayıtlar:** from parametresine en son aldığınız kaydın zamanını verin; her seferinde büyük aralık çekmeyin.
- Değişenler:** varlıklarda updated_since , iş emirlerinde order=updated ile yalnızca değişenleri alın.

- **Başlık gönderemeyen bileşenler:** fotoğraf ve PDF uç noktalarında `format=base64` kullanın.
- **Hata yönetimi:** 429 ve 503 yanıtlarında bekleyip yeniden deneyin (`Retry-After` başlığına uyun); 401 ve 403 yanıtlarında yeniden denemek yerine anahtar ve yetkileri kontrol edin.
- **Yazma işlemleri:** oluşturma isteklerinde her zaman `Idempotency-Key` gönderin; ağ kesintisinde aynı istek güvenle tekrarlanabilir.

Güvenlik kuralları

- **Her dış sistem için ayrı anahtar** üretin; örneğin Zenon, puantaj sistemi ve bir raporlama aracı ayrı anahtar kullansın. Böylece biri iptal edildiğinde diğerleri etkilenmez.
- **En az yetki ilkesi:** yalnızca gereken yetkileri verin. Kişisel ve hassas veri yetkilerini (`plate:person` , `guard:sensitive` , `assets:sensitive` , `ops:person`) ve yazma yetkilerini gerekmedikçe vermeyin.
- **IP kısıtlaması:** anahtarı yalnızca dış sistemin sabit IP adresine bağlayın.
- **Son kullanma tarihi ve sınır:** geçici entegrasyonlara son kullanma tarihi ve düşük dakika sınırı verin.
- **HTTPS:** anahtarı yalnızca şifreli bağlantıyla gönderin; kaynak koda, e-postaya ya da adrese yazmayın.
- **Anahtar yenileme:** anahtarın ele geçirildiğinden kuşku duyulursa yönetici panelinden **Yenile** ile yeni anahtar üretin; eski anahtar hemen geçersiz olur. Artık gerekmeyen anahtarı silin ya da pasifleştirin.
- **İzleme:** her anahtarın son istekleri (yol, durum kodu, IP, süre) yönetici panelindeki **Kayıt** ekranında görülür. Çok sayıda geçersiz anahtar denemesinden sonra istekler 1 dakika boyunca reddedilir (hata kodu `too_many_failures`).
- **Kişisel veri:** plaka, kişi ve fotoğraf verileri kişisel veri niteliğindedir (KVKK); üretilen PDF ve fotoğrafların saklanması ve paylaşılması alıcı kurumun sorumluluğundadır.

Bağlantı testi özeti

Entegrasyona başlarken önce `/ping` uç noktasıyla anahtarı doğrulayın (401 anahtar, 403 yetki ya da IP sorunu gösterir); sonra her modülün `/summary` ya da liste uç noktasını `limit=5` ile deneyin.