

# NanoNet API

REST API Dokümantasyonu — OpenAPI 3.1.0

YMH354 Web Tasarım ve Programlama

Final Projesi

**Hazırlayan:** Kael Valen

**Tarih:** 2026

**Base URL:** /api/v1

**Auth:** JWT Bearer Token

## İçindekiler

- Genel Bilgiler
- Kimlik Doğrulama
- Auth Endpoint'leri
  - POST /auth/register · POST /auth/login · POST /auth/logout · POST /auth/refresh · GET /auth/me · PUT /auth/password
- Servis Yönetimi (Services)
  - GET/POST /services · GET/PUT/DELETE /services/{id} · POST /services/{id}/ping,restart,stop,start,exec,scale
- Metrikler (Metrics)
  - GET /services/{id}/metrics · /metrics/aggregated · /metrics/uptime
- Uyarılar (Alerts)
  - GET /alerts · GET /services/{id}/alerts · POST /alerts/{alertId}/resolve
- AI Analiz
  - POST /services/{id}/analyze · GET /services/{id}/insights
- Komut Geçmişi (Commands)
- Ayarlar (Settings)
- Audit Log
- Şema Referansları
- Hata Kodları

## 1. Genel Bilgiler

Parametre	Değer
Base URL (Yerel)	<a href="http://localhost:8080">http://localhost:8080</a>
Base URL (Production)	<a href="https://api.nanonet.example.com">https://api.nanonet.example.com</a>
API Prefix	/api/v1
Auth Yöntemi	JWT Bearer Token
İçerik Tipi	application/json
Rate Limit (Genel)	100 istek / dakika
Rate Limit (Auth)	10 istek / dakika
Rate Limit (Komutlar)	10 istek / dakika (sıkı)
Rate Limit (AI)	10 istek / dakika / kullanıcı

**Not:** Tüm başarılı yanıtlar `{"status": "success", "data": {...}}` formatında döner. Hata yanıtları ise `{"status": "error", "message": "..."}` formatındadır.

## 2. Kimlik Doğrulama

NanoNet JWT (JSON Web Token) tabanlı kimlik doğrulama kullanır.

### Access Token

15 dakika geçerli. API isteklerinde `Authorization: Bearer <token>` header'ı ile gönderilir.

### Refresh Token

7 gün geçerli. Access token yenileme için `/auth/refresh` endpoint'ine gönderilir.

**Dikkat:** Logout işlemi sonrasında access token blacklist'e alınır. Bu token ile yapılan sonraki istekler **HTTP 401** ile reddedilir.

## 3. Auth Endpoint'leri

Kayıt, giriş, çıkış ve token yönetimi

POST

/api/v1/auth/register

Yeni kullanıcı kaydı

E-posta ve parola ile yeni hesap oluşturur. Başarılıysa kullanıcı bilgisi ve JWT token çifti döner. Aynı e-posta ile ikinci kayıt HTTP 409 döndürür.

### Request Body

```
{  
  "email": "user@example.com",      // zorunlu, RFC5322 format  
  "password": "Secure1234"          // zorunlu, min 8 karakter  
}
```

### Response 201

```
{  
  "status": "success",  
  "data": {  
    "user": { "id": "uuid", "email": "user@example.com", "created_at": "2026-01-01T00:00:00+00:00" },  
    "tokens": { "access_token": "eyJ...", "refresh_token": "eyJ..." }  
  }  
}
```

201

Kayıt başarılı

400

Validasyon hatası (eksik alan, geçersiz e-posta vb.)

409

E-posta zaten kayıtlı

**POST****/api/v1/auth/login**

Giriş yap

E-posta ve parola ile giriş. Rate limit: 10 istek/dk. Başarılı girişte access\_token (15dk) ve refresh\_token (7gün) döner.

### Request Body

```
{ "email": "user@example.com", "password": "Secure1234" }
```

### Response 200

```
{
  "status": "success",
  "data": {
    "user": { "id": "uuid", "email": "user@example.com", "created_at": "..." },
    "tokens": { "access_token": "eyJ...", "refresh_token": "eyJ..." }
  }
}
```

**200** Giriş başarılı**401** Geçersiz kimlik bilgileri**429** Rate limit aşındı**POST****/api/v1/auth/logout**

Çıkış yap



Mevcut access token'ı blacklist'e ekler. Sonraki isteklerde bu token geçersiz sayılır.

### Headers

```
Authorization: Bearer <access_token>
```

**200** Çıkış başarılı**401** Token eksik veya geçersiz

**POST****/api/v1/auth/refresh**

Access token yenile

Süresi dolmak üzere olan access token'ı yenilemek için refresh token kullanılır.

**Request Body**

{ "refresh\_token": "eyJ..." }

**Response 200**

{ "status": "success", "data": { "access\_token": "eyJ...", "refresh\_token": "eyJ..." } }

**200** Token yenilendi**401** Geçersiz veya süresi dolmuş refresh token**GET****/api/v1/auth/me**

Mevcut kullanıcı bilgisi



Access token'dan decode edilen kullanıcının profil bilgilerini döner.

**Response 200**

{ "status": "success", "data": { "id": "uuid", "email": "user@example.com", "created\_at": "2024-01-15T12:00:00Z", "updated\_at": "2024-01-15T12:00:00Z" } }

**200** Kullanıcı bilgisi**401** Token gerekli**PUT****/api/v1/auth/password**

Parola değiştir

**Request Body**

{ "current\_password": "OldPass123", "new\_password": "NewPass456" }

**200** Parola güncellendi**400** Mevcut parola hatalı

**POST****/api/v1/auth/forgot-password**

Şifre sıfırlama e-postası

E-posta adresine sıfırlama linki gönderir. E-posta kayıtlı olmasa bile HTTP 200 döner (güvenlik — kullanıcı keşfi önleme).

**Request Body**

{ "email": "user@example.com" }

**200**

İstek alındı (e-posta gönderildi veya göz ardı edildi)

**POST****/api/v1/auth/reset-password**

Parola sıfırla

**Request Body**

{ "token": "sifirlamaTokni123", "new\_password": "NewSecure789" }

**200**

Parola sıfırlandı

**400**

Token geçersiz veya süresi dolmuş (1 saat)

**POST****/api/v1/auth/agent-token**

Agent token üret



İzlenecek sunucuya deploy edilecek Rust agent binary'sinin authentication için kullanacağı token üretir.

**Response 200**

{ "status": "success", "data": { "agent\_token": "nanonet\_agt\_..." } }

**200**

Agent token oluşturuldu

## 4. Servis Yönetimi

İzlenen servis CRUD ve kontrol operasyonları — Tüm endpoint'ler JWT gerektirir

**GET** /api/v1/services

Servisleri listele



Kullanıcının tüm kayıtlı servislerini anlık durumlarıyla listeler.

### Response 200

```
{  
  "status": "success",  
  "data": {  
    "services": [  
      {  
        "id": "uuid", "name": "auth-service", "host": "10.0.0.5",  
        "port": 8001, "health_endpoint": "/health",  
        "poll_interval_sec": 10, "status": "healthy",  
        "agent_id": "uuid or null",  
        "created_at": "2026-01-01T00:00:00Z",  
        "updated_at": "2026-01-01T00:00:00Z"  
      }  
    ]  
  }  
}
```

**200** Servis listesi

**POST** /api/v1/services

Yeni servis ekle



### Request Body

```
{  
  "name": "auth-service",          // zorunlu, 2-100 karakter  
  "host": "192.168.1.50",         // zorunlu  
  "port": 8001,                   // zorunlu, 1-65535  
  "health_endpoint": "/health",    // zorunlu  
  "poll_interval_sec": 10         // zorunlu, 5-300  
}
```

**201** Servis oluşturuldu (body: Service objesi)

**400** Validasyon hatası

**GET****/api/v1/services/{id}**

Servis detayı



### Path Parametreleri

**id****uuid**Servis UUID'si **zorunlu****200**

Servis objesi

**404**

Servis bulunamadı

**PUT****/api/v1/services/{id}**

Servis güncelle



Kısmi güncelleme desteklenir; yalnızca gönderilen alanlar güncellenir.

### Request Body (Tüm alanlar opsiyonel)

```
{ "name": "new-name", "host": "10.0.0.6", "port": 8002, "poll_interval_sec": 15 }
```

**200**

Güncellendi

**404**

Servis bulunamadı

**DELETE****/api/v1/services/{id}**

Servis sil



Servisi ve ilişkili metrikleri, alertleri, komut loglarını siler. Geri alınamaz.

**200**

Silindi

**404**

Servis bulunamadı

**POST****/api/v1/services/{id}/ping**

Servisi ping'le



Health endpoint'e anlık istek gönderir ve latency + HTTP status'unu döner.

### Response 200

```
{ "status": "success", "data": { "latency_ms": 87.3, "status": "healthy" } }
```

**POST****/api/v1/services/{id}/restart**

Yeniden başlat



Bağlı agent'a restart komutu gönderir. **Rate limit: 10/dk (sıkı)**. Agent bağlı değilse hata döner.

**200**

Komut agent'a iletildi

**429**

Rate limit aşındırıldı

**500**

Agent bağlı değil

**POST****/api/v1/services/{id}/stop | /start**

Servis durdur / başlat



Sırasıyla stop ve start komutlarını agent'a ileter. Aynı rate limit kuralları geçerlidir.

**200**

Komut gönderildi

**POST****/api/v1/services/{id}/exec**

Komut çalıştır



### Request Body

```
{ "command": "ps aux | grep node" }
```

**200**

Komut gönderildi

## 5. Metrikler

Zaman serisi metrik sorgulama — Tüm endpoint'ler JWT gerektirir

GET

/api/v1/services/{id}/metrics

Metrik geçmiş



TimescaleDB hypertable'dan ham zaman serisi verisi döner. Grafik render için kullanılır.

### Query Parametreleri

Parametre	Tip	Default	Açıklama
window	string	1h	1h   6h   24h   7d
limit	integer	100	Maksimum nokta sayısı (max: 1000)

### Response 200

```
{
  "status": "success",
  "data": {
    "metrics": [
      {
        "time": "2026-01-01T12:00:00Z",
        "cpu_percent": 45.2,
        "memory_used_mb": 512.3,
        "latency_ms": 87.0,
        "error_rate": 0.0,
        "disk_used_gb": 15.7,
        "status": "healthy"
      }
    ]
  }
}
```

**GET****/api/v1/services/{id}/metrics/aggregated**

Agregat metrikler

 **JWT**

Ortalama, maksimum, P95 gibi istatistiksel özet. Dashboard widget'larında kullanılır.

### Response 200

```
{  
  "status": "success",  
  "data": {  
    "avg_cpu": 38.5,  
    "max_cpu": 92.1,  
    "avg_memory": 487.3,  
    "avg_latency": 112.4,  
    "p95_latency": 287.6,  
    "error_rate": 0.2,  
    "sample_count": 360  
  }  
}
```

**GET****/api/v1/services/{id}/metrics/uptime**

Uptime hesabi

 **JWT**

Belirtilen saatlik pencerede servisin sağlıklı olduğu sürenin yüzdesi.

### Query Parametreleri

**hours****integer**

Analiz penceresi (saat). Default: 24

### Response 200

```
{  
  "status": "success",  
  "data": { "uptime_percent": 99.72, "total_checks": 360, "healthy_checks": 359, "window_hours": 24 }  
}
```

## 6. Uyarılar (Alerts)

Alert listeleme ve çözümleme — Tüm endpoint'ler JWT gerektirir

**GET** /api/v1/alerts

Aktif alertleri listele



Kullanıcının tüm servislerindeki çözümlenmemiş alertleri döner.

### Response 200

```
{  
  "status": "success",  
  "data": {  
    "alerts": [  
      {  
        "id": "uuid", "service_id": "uuid",  
        "type": "cpu_threshold", "severity": "warn",  
        "message": "CPU %87 - eşik aşıldı",  
        "triggered_at": "2026-01-01T12:00:00Z", "resolved_at": null  
      }  
    ]  
  }  
}
```

**GET**

/api/v1/services/{id}/alerts

Servise ait alertler



### Query Parametreleri

**resolved**

**boolean**

true ise çözümlenmiş alertler de dahil edilir. Default: false

**POST**

/api/v1/alerts/{alertId}/resolve

Alert'i çözümle



Alert'in `resolved_at` alanını şimdiki zamanla günceller.

**200**

Alert çözümlendi

**404**

Alert bulunamadı veya kullanıcıya ait değil

## 7. AI Analiz (Claude API Entegrasyonu)

Anthropic Claude API ile anomali tespiti ve kök neden analizi

**Güvenlik:** Claude API anahtarı yalnızca backend ortam değişkeninde saklanır ( CLAUDE\_API\_KEY ). API çağrıları tamamen server-side gerçekleşir; anahtar ve ham API yanıtı frontend'e iletilmez.

POST

/api/v1/services/{id}/analyze

AI anomali analizi

JWT

Belirtilen zaman penceresindeki metrikleri (ortalama, P95, max, spike sayısı) ve cross-servis korelasyon bilgisini yapılandırılmış prompt olarak Claude API'ye gönderir. Yanıt JSON formatında parse edilip veritabanına kaydedilir ve döndürülür.

**Rate limit:** Kullanıcı başına dakikada 10 istek.

### Request Body

```
{  
    "window_minutes": 30,      // opsiyonel, default 30, min 5, max 1440  
    "deep_analysis": false    // opsiyonel; true → claude-sonnet-3, false → claude-haiku-3  
}
```

### Response 200

```
{  
    "status": "success",  
    "data": {  
        "insight": {  
            "summary": "CPU tüketimi son 30dk içinde %15 artış gösterdi...",  
            "root_cause": "Memory leak şüphesi – GC durma süresi yüksek",  
            "risk_level": "medium",  
            "recommendations": [  
                { "action": "Heap dump alarak memory profilini inceleyin", "priority": "high" },  
                { "action": "JVM parametrelerini gözden geçirin", "priority": "medium" }  
            ],  
            "anomalies": ["P95 latency normalin 2.3x üzerinde"]  
        }  
    }  
}
```

200

AI analiz sonucu

429

AI rate limit aşındı (1 dakika bekleyin)

500

Claude API hatası veya yanıt parse edilemedi

GET

/api/v1/services/{id}/insights

AI analiz geçmişi



### Query Parametreleri

Parametre	Default	Açıklama
page	1	Sayfa numarası
limit	20	Sayfa başına kayıt (max: 100)

### Response 200

```
{  
  "status": "success",  
  "data": {  
    "insights": [ { "id": "uuid", "model": "claude-haiku-3", "summary": "...", "created_at": "2023-10-01T12:00:00Z", "total": 47, "page": 1 } ]  
  }  
}
```

## 8. Komut Geçmişi

Servise gönderilmiş komutların tarihçesi — JWT gerektirir

GET

/api/v1/services/{id}/commands

Komut geçmişi



### Response 200

```
{  
    "status": "success",  
    "data": {  
        "commands": [  
            { "id": "uuid", "type": "restart", "status": "success", "requested_at": "2026-01-01T00:00:00Z", "completed_at": "2026-01-01T00:00:15Z" }  
        ]  
    }  
}
```

Status değerleri: pending | success | failed

## 9. Kullanıcı Ayarları

Alert eşik değerleri ve bildirim tercihleri — JWT gerektirir

**GET** /api/v1/settings

Ayarları getir



### Response 200

```
{  
    "status": "success",  
    "data": {  
        "cpu_threshold": 80.0,  
        "memory_threshold": 85.0,  
        "latency_threshold": 1000.0,  
        "error_rate_threshold": 5.0,  
        "email_notifications": false  
    }  
}
```

**PUT** /api/v1/settings

Ayarları güncelle



### Request Body

```
{ "cpu_threshold": 75.0, "latency_threshold": 800.0, "email_notifications": true }
```

**200** Ayarlar güncellendi

## 10. Audit Log

Güvenlik ve işlem günlüğü — JWT gerektirir

GET /api/v1/audit

Audit logları



### Query Parametreleri

Parametre	Default	Açıklama
page	1	Sayfa numarası
limit	50	Kayıt sayısı
action	-	Eylem filtresi: login, service_create, command_exec vb.

### Response 200

```
{  
    "status": "success",  
    "data": {  
        "logs": [  
            {  
                "id": "uuid", "user_id": "uuid",  
                "action": "service_create",  
                "resource": "services/uuid",  
                "ip_address": "203.0.113.1",  
                "created_at": "2026-01-01T12:00:00Z"  
            }  
        ]  
    }  
}
```

## 11. Şema Referansları

Anahtar veri modelleri

### Service

<b>id</b>	<code>uuid</code>	Benzersiz servis kimliği
<b>user_id</b>	<code>uuid</code>	Sahibi kullanıcı
<b>name</b>	<code>string</code>	Servis adı (2-100 karakter)
<b>host</b>	<code>string</code>	IP adresi veya hostname
<b>port</b>	<code>integer</code>	Port numarası (1-65535)
<b>health_endpoint</b>	<code>string</code>	Health check yolu (ör. /health)
<b>poll_interval_sec</b>	<code>integer</code>	Poll aralığı saniye (5-300)
<b>status</b>	<code>string</code>	healthy   unhealthy   degraded   unknown
<b>agent_id</b>	<code>uuid null</code>	Bağılı agent UUID'si

### MetricSnapshot

<b>time</b>	<code>datetime</code>	Ölçüm zaman damgası (UTC)
<b>cpu_percent</b>	<code>float</code>	CPU kullanımı yüzdesi (0-100)
<b>memory_used_mb</b>	<code>float</code>	Kullanılan bellek (MB)
<b>latency_ms</b>	<code>float</code>	Health endpoint yanıt süresi (ms)
<b>error_rate</b>	<code>float</code>	Hata oranı yüzdesi (0-100)
<b>disk_used_gb</b>	<code>float</code>	Disk kullanımı (GB)
<b>status</b>	<code>string</code>	Anlık servis durumu

### Alert

<b>type</b>	<code>string</code>	cpu_threshold   memory_threshold   latency_threshold   error_rate   ai_analysis
<b>severity</b>	<code>string</code>	info   warn   crit
<b>resolved_at</b>	<code>datetime null</code>	Çözümlenme zamanı; null ise aktif alert

### AnalysisResult (AI Yanıtı)

<b>summary</b>	<code>string</code>	Servis durumunun kısa özeti
<b>root_cause</b>	<code>string</code>	Tespit edilen sorunun kök nedeni
<b>risk_level</b>	<code>string</code>	low   medium   high   critical
<b>recommendations</b>	<code>array</code>	[{action: string, priority: low medium high}]

**anomalies**

**string[]**

Tespit edilen anomalilerin açıklamaları

## 12. Hata Kodları

HTTP Kodu	Durum	Açıklama
200	OK	İstek başarılı
201	Created	Kaynak oluşturuldu
400	Bad Request	Validasyon hatası; request body geçersiz
401	Unauthorized	Token eksik, geçersiz veya süresi dolmuş
403	Forbidden	Kaynağa erişim yetkisi yok
404	Not Found	Kaynak bulunamadı
409	Conflict	Çakışma (ör. duplicate e-posta)
429	Too Many Requests	Rate limit aşındı
500	Internal Server Error	Sunucu hatası (AI hatası dahil)

### Hata Yanıt Formatı

```
{  
  "status": "error",  
  "message": "bu email adresi zaten kullanılıyor"  
}
```

**YAML Dosyası:** Bu dokümantasyonun makine-okunabilir OpenAPI 3.1.0 formatındaki tam versiyonu `nanonet-api.yaml` dosyasında sunulmaktadır. Swagger UI veya Redoc ile interaktif olarak incelenebilir.