

# SmartIoT – Web Servis Dokümantasyonu (v1)

Bu doküman, SmartIoT arka uç (backend) servislerinin REST API kullanımını açıklar. Detaylı şema ve canlı denemeler için Swagger UI kullanılabilir.

## Genel Bilgiler

Base URL (dev): <http://localhost:8080/api>

Swagger (canlı test): <http://localhost:8080/swagger-ui.html>

İçerik Tipi: application/json

Kimlik Doğrulama: Şimdilik yok (JWT entegrasyonu planlanabilir).

## Kapsam

- Kullanıcı uçları (register, login, deactivate)
- Cihaz Kataloğu (Device Management)
- Kullanıcıya atanmış cihazlar (User Device)
- Kullanıcı odaları (User Room)
- Cihaz kontrol & sensör/ MQTT tetikleyicileri

## 1) Kullanıcı Uçları

### 1.1 Kayıt Ol – POST /register

İstek Gövdesi

```
{  
  "email": "user@example.com",  
  "password": "123456",  
  "firstName": "Mustafa",  
  "lastName": "Çetin"  
}
```

200 OK (Örnek Yanıt)

```
{  
  "id": 7,  
  "email": "user@example.com",  
  "password": "123456",  
  "firstName": "Mustafa",  
  "lastName": "Çetin",  
  "active": true  
}
```

curl

```
curl -X POST http://localhost:8080/api/register -H "Content-Type: application/json" -d '{"email":"user@example.com", "password":"123456", "firstName":"Mustafa", "lastName":"Çetin"}'
```

### 1.2 Giriş Yap – POST /login

İstek Gövdesi

```
{ "email": "user@example.com", "password": "123456" }
```

Yanıtlar

200 OK: Kullanıcı JSON

401 Unauthorized: "Invalid credentials"

403 Forbidden: "User is deactivated. Please create a new account."

### 1.3 Kullanıcı Pasifleştir – PUT /users/{id}/deactivate

Yol Parametresi: id (Long)

200 OK: "[] User, rooms, user\_devices and related devices deactivated (safe mode)."

404 Not Found

## 2) Cihaz Yönetimi (Katalog)

### 2.1 Cihazları Listele - GET /device

200 OK (Örnek Yanıt)

```
[
  {
    "id": 101,
    "deviceUid": "ESP32-ABCD",
    "deviceName": "Salon Lamba 1",
    "deviceModel": "LED_WS2812B",
    "active": true,
    "type": "LED"
  }
]
```

### 2.2 Cihaz Ekle (Kataloga) - POST /device

İstek Gövdesi

```
{
  "deviceUid": "ESP32-NEW",
  "deviceName": "Yeni Cihaz",
  "deviceModel": "LED_WS2812B",
  "active": true
}
```

### 2.3 Cihaz Sil - DELETE /device/{id}

Yol Parametresi: id (Long)

### 2.4 Cihaz Aktiflik Güncelle - PUT /device/{id}/status?active=true|false

## 3) Kullanıcıya Atanmış Cihazlar (User Device)

### 3.1 Kullanıcının Cihazlarını Getir - GET /user-devices/user/{userId}

### 3.2 Aktiflik Güncelle - PUT /user-devices/{id}/status?active=true|false

### 3.3 Pasifleştir - PUT /user-devices/{id}/deactivate

### 3.4 Sil - DELETE /user-devices/{id}

### 3.5 Tüm User-Device Listesi - GET /user-devices

### 3.6 Eski Havuz - Sahip Olunan Cihaz Kaydet - POST /user-devices/register

İstek Gövdesi

```
{
  "userId": 7,
  "deviceModel": "LED_WS2812B",
  "deviceName": "Salon Lamba",
  "deviceUid": "ESP32-3F:AB:9C:11",
  "alias": "lamba1"
}
```

### 3.7 Havuz Listele - GET /user-devices/pool/{userId}

### 3.8 Havuzdaki Cihazı Odaya Ata - PUT

/user-devices/{deviceId}/assign

İstek Gövdesi

```
{ "userId": 7, "userRoomId": 12, "assignedName": "Salon Lamba 1" }
```

### 3.9 Katalogtan Doğrudan Odaya Ekle - POST /user-devices/add

İstek Gövdesi

```
{ "userId": 7, "deviceId": 101, "userRoomId": 12, "assignedName": "Salon Lamba" }
```

## 4) Kullanıcı Odaları (User Room)

### 4.1 Aktif Odalar - GET /user-rooms/by-user/{userId}

### 4.2 Oda Oluştur - POST /user-rooms

İstek Gövdesi

```
{ "userId": 7, "roomName": "Salon" }
```

### 4.3 Oda Pasifleştir - PATCH /user-rooms/{id}/deactivate?userId=7

## 5) Cihaz Kontrol & Sensör

### 5.1 LED Toggle (Genel) - POST /control/led

### 5.2 Renkli LED (MQTT) - POST /led/toggle?color=red&state=true

200 OK (Örnek):

```
"LED komutu gönderildi: red_on"
```

### 5.3 Servo Açı Ver - POST /servo-control

İstek Gövdesi

```
{ "servo": "servo1", "angle": 90 }
```

### 5.4 Buzzer - POST /buzzer/beep

### 5.5 Sensör Değerlerini Elle Güncelle - POST /sensor?temp=23.4&hum;=45.2

### 5.6 RFID Güncelle - POST /rfid?cardId=A1B2C3D4

### 5.7 Durum Çek - GET /status

200 OK (Örnek Yanıt):

```
{
  "ledOn": true,
  "servo1Open": false,
  "servo2Open": true,
  "temperature": 24.1,
  "humidity": 47.0,
  "lastCardID": "A1B2C3D4"
}
```

## 6) Hata Yanıtları (Özet)

400 Bad Request - Eksik/yanlış parametre

401 Unauthorized - Kimlik doğrulama başarısız (JWT eklendiğinde)

403 Forbidden - Kullanıcı pasif  
404 Not Found - Kayıt bulunamadı  
500 Internal Server Error - Sunucu hatası

## 7) Örnek İş Akışı

- 1) /register → Kullanıcı oluştur
- 2) /login → Oturum aç
- 3) /device → Kataloğa cihaz ekle (opsiyonel, admin akışı)
- 4) /user-rooms → Oda oluştur
- 5) /user-devices/add → Katalogtan odaya cihaz ekle
- 6) /led/toggle, /servo-control, /status → Kontrol & izleme

## 8) Notlar ve Öneriler

- Swagger UI ile canlı test mümkündür. Bu doküman insan-okur rehber niteliğindedir. Her endpoint'in teknik şeması ve örnek gövdeleri Swagger'da da görülebilir.
- Üretim ortamında JWT tabanlı kimlik doğrulama önerilir (Swagger'a Authorize butonu eklenebilir).
- Tutarlı hata modeli için ortak bir ApiError şeması ilerleyen aşamalarda eklenebilir.