

# Video Processing API - Language Examples

## Desteklenen Diller

API 99 farklı dil desteği sunar. İşte popüler diller:

| Kod | Dil        | Kod | Dil        |
|-----|------------|-----|------------|
| tr  | Türkçe     | es  | İspanyolca |
| fr  | Fransızca  | de  | Almanca    |
| it  | İtalyanca  | pt  | Portekizce |
| ru  | Rusça      | ja  | Japonca    |
| ko  | Korece     | zh  | Çince      |
| ar  | Arapça     | hi  | Hintçe     |
| nl  | Hollandaca | sv  | İsveççe    |
| no  | Norveççe   | da  | Danca      |
| pl  | Lehçe      | cs  | Çekçe      |

## Desteklenen Dilleri Listele

```
curl -X GET "http://localhost:8082/api/v1/upload/languages"
```

## Video Yükleme (Dil Seçimi ile)

Türkçe'ye Çeviri

```
curl -X POST "http://localhost:8082/api/v1/upload/" \
-H "accept: application/json" \
-H "Content-Type: multipart/form-data" \
-F "file=@video.mp4" \
-F "target_language=tr"
```

İspanyolca'ya Çeviri

```
curl -X POST "http://localhost:8082/api/v1/upload/" \
-H "accept: application/json" \
-H "Content-Type: multipart/form-data" \
-F "file=@video.mp4" \
-F "target_language=es"
```

## Fransızca'ya Çeviri

```
curl -X POST "http://localhost:8082/api/v1/upload/" \
  -H "accept: application/json" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@video.mp4" \
  -F "target_language=fr"
```

## Almanca'ya Çeviri

```
curl -X POST "http://localhost:8082/api/v1/upload/" \
  -H "accept: application/json" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@video.mp4" \
  -F "target_language=de"
```

## Japonca'ya Çeviri

```
curl -X POST "http://localhost:8082/api/v1/upload/" \
  -H "accept: application/json" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@video.mp4" \
  -F "target_language=ja"
```

## Çince'ye Çeviri

```
curl -X POST "http://localhost:8082/api/v1/upload/" \
  -H "accept: application/json" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@video.mp4" \
  -F "target_language=zh"
```



## Durum Kontrolü

```
curl -X GET "http://localhost:8082/api/v1/status/{video_id}"
```



## Video İndirme

```
curl -X GET "http://localhost:8082/api/v1/status/download/{video_id}" \
-o "processed_video.mp4"
```

## 🔑 Sistem ve Yönetim Endpointleri

### Sistem Kaynakları (CPU, RAM, Disk, GPU)

Sistemin anlık kaynak kullanımını ve GPU durumunu döner.

```
curl -X GET "http://localhost:8082/api/v1/status/system/resources"
```

Yanıt örneği:

```
{
  "cpu_percent": 12.5,
  "cpu_cores": 8,
  "memory_percent": 45.2,
  "memory_used_gb": 7.2,
  "memory_total_gb": 16.0,
  "disk_percent": 60.1,
  "disk_free_gb": 120.5,
  "disk_total_gb": 256.0,
  "gpu": {
    "available": true,
    "gpus": [
      {
        "name": "NVIDIA RTX 3060",
        "memory_total_mb": 8192,
        "memory_used_mb": 1024,
        "utilization_percent": 15,
        "temperature_celsius": 45
      }
    ]
  }
}
```

### Sistem Sağlığı

API, Celery worker ve Redis servislerinin sağlık durumunu döner.

```
curl -X GET "http://localhost:8082/api/v1/status/system/health"
```

### Depolama Bilgisi

Uploads, outputs, storage ve logs klasörlerinin toplam boyutunu ve dosya sayısını döner.

```
curl -X GET "http://localhost:8082/api/v1/status/system/storage"
```

## Performans Metrikleri

Son 100 log satırından upload, işleme, hata ve başarı sayısını döner.

```
curl -X GET "http://localhost:8082/api/v1/status/system/performance"
```

---

## Toplu Temizlik (Cleanup)

Uploads ve outputs klasörlerindeki tüm video dosyalarını ve ilgili geçici dosyaları temizler.

```
curl -X POST "http://localhost:8082/api/v1/status/system/cleanup"
```

Yanıt örneği:

```
{
  "message": "Cleanup completed for 3 videos.",
  "cleaned": ["videoid1", "videoid2"],
  "failed": [{"video_id": "videoid3", "error": "Açıklama"}]
}
```

Not: Bu endpoint, status klasörüne bağlı kalmadan uploads ve outputs klasörlerindeki tüm video\_id'leri otomatik olarak tespit eder ve temizler.

---

## Dinamik Konfigürasyon Yönetimi

Tüm Ayarları Görüntüle

```
curl -X GET "http://localhost:8082/api/v1/config"
```

Ayar Güncelle

```
curl -X POST "http://localhost:8082/api/v1/config/update" \
-H "Content-Type: application/json" \
-d '{"setting_path": "timeouts.process_timeout", "value": 600}'
```

## Python Örneği

```
import requests

# Video yükleme (Türkçe'ye çeviri)
files = {'file': open('video.mp4', 'rb')}
data = {'target_language': 'tr'}

response = requests.post(
    'http://localhost:8082/api/v1/upload/',
    files=files,
    data=data
)

if response.status_code == 202:
    result = response.json()
    video_id = result['video_id']
    print(f"Video ID: {video_id}")
    print(f"Target Language: {result['language_name']}")
```

## JavaScript/Fetch Örneği

```
// Video yükleme (İspanyolca'ya çeviri)
const formData = new FormData();
formData.append('file', videoFile);
formData.append('target_language', 'es');

fetch('http://localhost:8082/api/v1/upload/', {
  method: 'POST',
  body: formData
})
.then(response => response.json())
.then(data => {
  console.log('Video ID:', data.video_id);
  console.log('Language:', data.language_name);
});
```

## Yanıt Örnekleri

### Başarılı Yükleme

```
{
  "video_id": "550e8400-e29b-41d4-a716-446655440000",
  "filename": "video.mp4",
  "target_language": "tr",
  "language_name": "Turkish",
  "status": "processing",
```

```
"task_id": "task-123",  
"message": "Video uploaded and processing started for Turkish translation"  
}
```

## Durum Kontrolü

```
{  
  "video_id": "550e8400-e29b-41d4-a716-446655440000",  
  "status": "translating",  
  "progress": 40  
}
```

## Tamamlanmış İşlem

```
{  
  "video_id": "550e8400-e29b-41d4-a716-446655440000",  
  "status": "completed",  
  "progress": 100,  
  "result": {  
    "video_id": "550e8400-e29b-41d4-a716-446655440000",  
    "filename": "video_subtitled.mp4",  
    "size_mb": 45.2,  
    "storage_url": "/storage/550e8400-e29b-41d4-a716-446655440000/video_subtitled.mp4",  
    "download_url": "/api/v1/download/550e8400-e29b-41d4-a716-446655440000",  
    "status": "uploaded"  
  }  
}
```

## ⚠ Hata Durumları

### Geçersiz Dil Kodu

```
{  
  "detail": "Invalid language code: invalid_lang. Supported languages: af, am, ar, as, az, ba, be, bg, bn, bo..."  
}
```

### Geçersiz Dosya Formatı

```
{  
  "detail": "Invalid file format or size. Allowed: .mp4, .avi, .mov, .mkv, .wmv, .flv, .webm, Max size: 500MB"  
}
```

## İşlem Hatası

```
{
  "video_id": "550e8400-e29b-41d4-a716-446655440000",
  "status": "failed",
  "error": "Transcription failed: Video file not found"
}
```

## API Teknik Referans (Input/Output ve Açıklamalar)

### /api/v1/upload/languages [GET]

**Açıklama:**

Desteklenen tüm dilleri ve kodlarını listeler.

**Girdi:**

Yok

**Çıktı:**

```
{
  "languages": {
    "tr": "Türkçe",
    "en": "English",
    ...
  },
  "count": 99,
  "default": "tr"
}
```

### /api/v1/upload/ [POST]

**Açıklama:**

Bir video dosyasını yükler ve işleme başlatır (altıyazı çıkarma, çeviri, hardsub).

**Girdi (multipart/form-data):**

- file** (zorunlu): Video dosyası (örn: .mp4)
- target\_language** (zorunlu): Hedef dil kodu (örn: "tr")

**Çıktı (202 Accepted):**

```
{
  "video_id": "uuid",
  "filename": "video.mp4",
  "target_language": "tr",
  "language_name": "Turkish",
  "status": "uploaded",
  "task_id": "celery-task-id"
}
```

**Hata Durumları:**

- 400: Geçersiz dosya veya dil kodu
- 413: Dosya boyutu limiti aşıldı
- 500: Sunucu hatası

---

## /api/v1/status/{video\_id} [GET]

**Açıklama:**

Belirli bir video işinin durumunu döner.

**Girdi (path):**

- **video\_id**: Video işinin UUID'si

**Çıktı:**

```
{
  "video_id": "uuid",
  "status": "processing|completed|failed",
  "progress": 40,
  "error": "Açıklama (opsiyonel)"
}
```

---

## /api/v1/status/download/{video\_id} [GET]

**Açıklama:**

İşlenmiş videoyu indirir.

**Girdi (path):**

- **video\_id**: Video işinin UUID'si

**Çıktı:**

- Başarılı: video/mp4 dosyası
- Hata: 404 veya 500 JSON hata mesajı



## /api/v1/status/system/resources [GET]

### Açıklama:

Sistem kaynaklarını (CPU, RAM, Disk, GPU) döner.

### Girdi:

Yok

### Çıktı:

```
{
  "cpu_percent": 12.5,
  "cpu_cores": 8,
  "memory_percent": 45.2,
  "memory_used_gb": 7.2,
  "memory_total_gb": 16.0,
  "disk_percent": 60.1,
  "disk_free_gb": 120.5,
  "disk_total_gb": 256.0,
  "gpu": {
    "available": true,
    "gpus": [
      {
        "name": "NVIDIA RTX 3060",
        "memory_total_mb": 8192,
        "memory_used_mb": 1024,
        "utilization_percent": 15,
        "temperature_celsius": 45
      }
    ]
  }
}
```

---

## /api/v1/status/system/health [GET]

### Açıklama:

API, Celery worker ve Redis servislerinin sağlık durumunu döner.

### Girdi:

Yok

### Çıktı:

```
{
  "overall_status": "healthy|degraded",
  "services": {
    "api": {"status": "healthy"},
    "celery": {"status": "healthy|unhealthy", "workers": ["worker1", ...]},
    "redis": {"status": "healthy|unhealthy"}
  }
}
```

```
}  
}
```

---

## /api/v1/status/system/storage [GET]

### Açıklama:

Uploads, outputs, storage ve logs klasörlerinin toplam boyutunu ve dosya sayısını döner.

### Girdi:

Yok

### Çıktı:

```
{  
  "uploads": {"size_mb": 100.5, "file_count": 10, "exists": true},  
  "outputs": {"size_mb": 200.1, "file_count": 20, "exists": true},  
  "storage": {"size_mb": 500.0, "file_count": 30, "exists": true},  
  "logs": {"size_mb": 5.0, "file_count": 2, "exists": true}  
}
```

---

## /api/v1/status/system/performance [GET]

### Açıklama:

Son 100 log satırından upload, işleme, hata ve başarı sayısını döner.

### Girdi:

Yok

### Çıktı:

```
{  
  "upload_count": 5,  
  "processing_count": 3,  
  "error_count": 1,  
  "success_count": 2,  
  "total_operations": 100  
}
```

---

## /api/v1/status/system/cleanup [POST]

### Açıklama:

Uploads ve outputs klasörlerindeki tüm video dosyalarını ve ilgili geçici dosyaları temizler.

**Girdi:**

Yok

**Çıktı:**

```
{
  "message": "Cleanup completed for 3 videos.",
  "cleaned": ["videoid1", "videoid2"],
  "failed": [{"video_id": "videoid3", "error": "Açıklama"}]
}
```

---

## /api/v1/status/tasks/active [GET]

**Açıklama:**

Aktif (devam eden) arka plan görevlerini listeler.

**Girdi:**

Yok

**Çıktı:**

```
{
  "active_tasks": [
    {
      "task_id": "celery-task-id",
      "name": "process_video_task",
      "worker": "worker1",
      "args": [],
      "kwargs": {},
      "time_start": "2024-06-22T12:00:00"
    }
  ],
  "count": 1
}
```

---

## /api/v1/status/tasks/{task\_id} [DELETE]

**Açıklama:**

Belirli bir arka plan görevini iptal eder.

**Girdi (path):**

- `task_id`: Celery task id

**Çıktı:**

```
{  
  "message": "Task {task_id} cancelled successfully"  
}
```

---

## /api/v1/config [GET]

### Açıklama:

Tüm konfigürasyon ayarlarını döner.

### Girdi:

Yok

### Çıktı:

Tüm ayarları içeren bir JSON nesnesi.

---

## /api/v1/config/update [POST]

### Açıklama:

Bir konfigürasyon ayarını dinamik olarak günceller.

### Girdi (application/json):

```
{  
  "setting_path": "timeouts.process_timeout",  
  "value": 600  
}
```

### Çıktı:

```
{  
  "message": "Setting timeouts.process_timeout updated successfully",  
  "config": { ... }  
}
```

---