

# REST API Dokumentation

**DOKUMENTATION** · DOC-2025-001

**Projekt:** Backend API  
**Kunde:** Mustermann IT-Services  
**Version:** 1.0  
**Status:** **FINAL**  
**Erstellt:** 2025-01-22  
**Autoren:** Max Mustermann

API

REST

Backend

JSON

# Inhaltsverzeichnis

1 Übersicht

2 Authentifizierung

2.1 API-Schlüssel anfordern

3 Endpunkte

3.1 Projekte

3.1.1 GET /v1/projects

3.1.2 POST /v1/projects

3.1.3 GET /v1/projects/{id}

3.2 Dokumente

3.2.1 GET /v1/documents

3.2.2 POST /v1/documents

4 Fehlerbehandlung

5 Rate Limiting

6 Webhooks

6.1 Webhook einrichten

6.2 Verfügbare Events

6.3 Webhook-Payload

7 Changelog

7.1 Version 1.0 (22.01.2025)

3

3

3

3

3

3

4

4

5

5

5

5

6

6

7

7

7

# 1 Übersicht

Diese Dokumentation beschreibt die REST API der Mustermann Plattform Version 1.0. Die API ermöglicht die Integration externer Systeme und bietet Zugriff auf alle Kernfunktionen.

**Basis-URL:** `https://api.example.com/v1`

**Formate:** JSON (Content-Type: application/json)

## 2 Authentifizierung

Alle API-Anfragen erfordern einen gültigen API-Schlüssel. Dieser wird im Header `X-API-Key` übergeben:

```
curl -H "X-API-Key: your-api-key" https://api.example.com/v1/projects
```

### 2.1 API-Schlüssel anfordern

API-Schlüssel können über das Dashboard unter Einstellungen → API erstellt werden. Jeder Schlüssel hat ein konfigurierbares Berechtigungsprofil.

Schlüsseltyp	Beschreibung
Read-Only	Nur lesender Zugriff auf alle Ressourcen
Standard	Lesen und Schreiben von Projekten und Dokumenten
Admin	Vollzugriff inkl. Benutzerverwaltung

## 3 Endpunkte

### 3.1 Projekte

#### 3.1.1 GET /v1/projects

Listet alle Projekte auf.

**Parameter:**

- `status` (optional): Filter nach Status ( `active`, `archived`, `draft` )
- `page` (optional): Seitennummer (Standard: 1)
- `limit` (optional): Anzahl Ergebnisse pro Seite (Standard: 50, Max: 100)

**Beispiel-Anfrage:**

```
curl -H "X-API-Key: your-api-key" \
  "https://api.example.com/v1/projects?status=active&limit=10"
```

**Antwort:**

```
{
  "data": [
    {
      "id": "proj_123",
      "name": "Website Relaunch",
      "status": "active",
      "created_at": "2025-01-15T10:00:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 10,
    "total": 42
  }
}
```

### 3.1.2 POST /v1/projects

Erstellt ein neues Projekt.

**Request Body:**

```
{
  "name": "Neues Projekt",
  "description": "Projektbeschreibung",
  "client_id": "client_456",
  "status": "draft"
}
```

**Antwort (201 Created):**

```
{
  "id": "proj_789",
  "name": "Neues Projekt",
  "status": "draft",
  "created_at": "2025-01-22T14:30:00Z"
}
```

### 3.1.3 GET /v1/projects/{id}

Ruft ein einzelnes Projekt ab.

**Antwort:**

```
{
  "id": "proj_123",
  "name": "Website Relaunch",
  "description": "Kompletter Relaunch der Unternehmenswebsite",
  "status": "active",
  "client": {
    "id": "client_456",

```

```
  "name": "Beispiel GmbH"
},
"created_at": "2025-01-15T10:00:00Z",
"updated_at": "2025-01-20T16:45:00Z"
}
```

## 3.2 Dokumente

### 3.2.1 GET /v1/documents

Listet alle Dokumente auf.

**Parameter:**

- `project_id` (optional): Filter nach Projekt
- `type` (optional): Filter nach Typ ( `invoice` , `offer` , `concept` , `documentation` )
- `from_date` (optional): Von-Datum (ISO 8601)
- `to_date` (optional): Bis-Datum (ISO 8601)

### 3.2.2 POST /v1/documents

Erstellt ein neues Dokument.

**Request Body:**

```
{
  "project_id": "proj_123",
  "type": "invoice",
  "title": "Rechnung Januar 2025",
  "data": {
    "items": [...],
    "totals": {...}
  }
}
```

## 4 Fehlerbehandlung

Die API verwendet Standard-HTTP-Statuscodes:

Code	Bedeutung
200 OK	Erfolgreiche Anfrage
201 Created	Ressource erfolgreich erstellt
400 Bad Request	Ungültige Anfrage (z.B. fehlende Pflichtfelder)
401 Unauthorized	Fehlende oder ungültige Authentifizierung
403 Forbidden	Keine Berechtigung für diese Ressource
404 Not Found	Ressource nicht gefunden
422 Unprocessable Entity	Validierungsfehler

429 Too Many Requests	Rate Limit überschritten
500 Internal Server Error	Serverfehler

Fehlerantwort-Format:

```
{
  "error": {
    "code": "validation_error",
    "message": "Das Feld 'name' ist erforderlich",
    "details": [
      {
        "field": "name",
        "reason": "required",
        "message": "Dieses Feld ist erforderlich"
      }
    ]
  }
}
```

## 5 Rate Limiting

Die API ist auf **1000 Anfragen pro Stunde** begrenzt. Der aktuelle Status wird in den Response-Headern zurückgegeben:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 847
X-RateLimit-Reset: 1642694400
```

Bei Überschreitung des Limits wird ein `429 Too Many Requests` zurückgegeben. Die Antwort enthält einen `Retry-After` Header mit der Wartezeit in Sekunden.

## 6 Webhooks

Die API unterstützt Webhooks für Echtzeit-Benachrichtigungen bei Änderungen.

### 6.1 Webhook einrichten

```
curl -X POST \
  -H "X-API-Key: your-api-key" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://your-server.com/webhook",
    "events": ["project.created", "document.updated"],
    "secret": "your-webhook-secret"
  }' \
  https://api.example.com/v1/webhooks
```

## 6.2 Verfügbare Events

- `project.created` - Neues Projekt erstellt
- `project.updated` - Projekt aktualisiert
- `project.deleted` - Projekt gelöscht
- `document.created` - Neues Dokument erstellt
- `document.updated` - Dokument aktualisiert
- `document.deleted` - Dokument gelöscht

## 6.3 Webhook-Payload

```
{
  "event": "project.created",
  "timestamp": "2025-01-22T14:30:00Z",
  "data": {
    "id": "proj_789",
    "name": "Neues Projekt",
    "status": "draft"
  }
}
```

# 7 Changelog

## 7.1 Version 1.0 (22.01.2025)

- Initiale API-Version
- Projekte: CRUD-Operationen
- Dokumente: CRUD-Operationen
- Authentifizierung via API-Key
- Rate Limiting (1000 req/h)
- Webhook-Support