

# cflux API Dokumentation

---

Version 1.0

Base URL: <http://localhost:3001/api>

Authentication: JWT Bearer Token

---

## Inhaltsverzeichnis

1. [Authentifizierung](#)
  2. [Benutzer](#)
  3. [Zeiterfassung](#)
  4. [Projekte](#)
  5. [Abwesenheiten](#)
  6. [Reports](#)
  7. [Rechnungen](#)
  8. [Fehlerbehandlung](#)
- 

## Authentifizierung

Registrierung

**Endpunkt:** [POST /api/auth/register](#)

**Beschreibung:** Erstellt einen neuen Benutzer-Account

**Request Body:**

```
{  
  "email": "user@example.com",  
  "password": "securePassword123",  
  "firstName": "Max",  
  "lastName": "Mustermann"  
}
```

**Response:** [201 Created](#)

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "user": {  
    "id": "123e4567-e89b-12d3-a456-426614174000",  
    "email": "user@example.com",  
    "firstName": "Max",  
    "lastName": "Mustermann",  
    "role": "USER",  
    "active": true,  
    "lastLogin": null  
  }  
}
```

```
"vacationDays": 25,  
"createdAt": "2025-01-01T10:00:00.000Z"  
}  
}
```

## Validierung:

- Email: Muss gültige E-Mail-Adresse sein
- Password: Mindestens 8 Zeichen
- firstName/lastName: Nicht leer

## Fehler:

- **400 Bad Request**: Validierungsfehler
- **409 Conflict**: E-Mail bereits registriert

## Login

**Endpunkt:** POST /api/auth/login

**Beschreibung:** Authentifiziert einen Benutzer und gibt JWT Token zurück

### Request Body:

```
{  
  "email": "user@example.com",  
  "password": "securePassword123"  
}
```

**Response:** 200 OK

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "user": {  
    "id": "123e4567-e89b-12d3-a456-426614174000",  
    "email": "user@example.com",  
    "firstName": "Max",  
    "lastName": "Mustermann",  
    "role": "USER"  
  }  
}
```

## Fehler:

- **400 Bad Request**: Fehlende Credentials
- **401 Unauthorized**: Ungültige Credentials
- **403 Forbidden**: Account deaktiviert

**Rate Limiting:** 5 Versuche pro 15 Minuten

---

## Benutzer

**Alle Endpunkte erfordern Authentifizierung (Bearer Token)**

Eigenes Profil abrufen

**Endpunkt:** GET /api/users/me

**Beschreibung:** Gibt das Profil des aktuell eingeloggten Benutzers zurück

**Headers:**

```
Authorization: Bearer <token>
```

**Response:** 200 OK

```
{
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "email": "user@example.com",
  "firstName": "Max",
  "lastName": "Mustermann",
  "role": "USER",
  "active": true,
  "vacationDays": 25,
  "vacationDaysUsed": 5,
  "vacationDaysRemaining": 20,
  "createdAt": "2025-01-01T10:00:00.000Z",
  "updatedAt": "2025-01-15T14:30:00.000Z"
}
```

---

Alle Benutzer abrufen

**Endpunkt:** GET /api/users

**Beschreibung:** Liste aller Benutzer (nur ADMIN)

**Headers:**

```
Authorization: Bearer <admin-token>
```

**Query Parameters:**

- **active** (optional): true | false - Filtert nach aktivem Status

- **role** (optional): **USER | ADMIN** - Filtert nach Rolle
- **page** (optional): Seitennummer (Standard: 1)
- **limit** (optional): Einträge pro Seite (Standard: 50)

**Beispiel:** GET /api/users?active=true&role=USER&page=1&limit=20

**Response:** 200 OK

```
{
  "users": [
    {
      "id": "123e4567-e89b-12d3-a456-426614174000",
      "email": "user1@example.com",
      "firstName": "Max",
      "lastName": "Mustermann",
      "role": "USER",
      "active": true,
      "vacationDays": 25,
      "createdAt": "2025-01-01T10:00:00.000Z"
    },
    {
      "id": "223e4567-e89b-12d3-a456-426614174001",
      "email": "user2@example.com",
      "firstName": "Anna",
      "lastName": "Schmidt",
      "role": "ADMIN",
      "active": true,
      "vacationDays": 30,
      "createdAt": "2025-01-02T11:00:00.000Z"
    }
  ],
  "pagination": {
    "total": 45,
    "page": 1,
    "limit": 20,
    "totalPages": 3
  }
}
```

## Fehler:

- 403 Forbidden: Keine Admin-Rechte

## Benutzer Details abrufen

**Endpunkt:** GET /api/users/:id

**Beschreibung:** Details eines spezifischen Benutzers (nur ADMIN)

**Response:** 200 OK

```
{
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "email": "user@example.com",
  "firstName": "Max",
  "lastName": "Mustermann",
  "role": "USER",
  "active": true,
  "vacationDays": 25,
  "projects": [
    {
      "id": "proj-1",
      "name": "Website Redesign",
      "active": true
    }
  ],
  "statistics": {
    "totalHoursThisMonth": 160,
    "totalHoursThisYear": 320,
    "currentlyClocked": false
  }
}
```

**Fehler:**

- **403 Forbidden:** Keine Admin-Rechte
  - **404 Not Found:** Benutzer existiert nicht
- 

## Benutzer aktualisieren

**Endpunkt:** PUT /api/users/:id**Beschreibung:** Aktualisiert Benutzer-Daten (nur ADMIN)**Request Body:**

```
{
  "firstName": "Maximilian",
  "lastName": "Mustermann",
  "email": "new.email@example.com",
  "role": "ADMIN",
  "active": true,
  "vacationDays": 30
}
```

**Response:** 200 OK

```
{
  "id": "123e4567-e89b-12d3-a456-426614174000",
```

```
"email": "new.email@example.com",
"firstName": "Maximilian",
"lastName": "Mustermann",
"role": "ADMIN",
"active": true,
"vacationDays": 30,
"updatedAt": "2025-01-15T14:30:00.000Z"
}
```

**Fehler:**

- **400 Bad Request**: Validierungsfehler
- **403 Forbidden**: Keine Admin-Rechte
- **404 Not Found**: Benutzer existiert nicht
- **409 Conflict**: E-Mail bereits vergeben

## Benutzer löschen

**Endpunkt:** `DELETE /api/users/:id`

**Beschreibung:** Löscht einen Benutzer (Soft Delete, nur ADMIN)

**Response:** `204 No Content`

**Hinweis:** Dies ist ein Soft Delete - der Benutzer wird auf `active: false` gesetzt, aber nicht physisch gelöscht.

**Fehler:**

- **403 Forbidden**: Keine Admin-Rechte
- **404 Not Found**: Benutzer existiert nicht
- **409 Conflict**: Benutzer hat aktive Zeitbuchungen

## Zeiterfassung

### Einstempeln

**Endpunkt:** `POST /api/time/clock-in`

**Beschreibung:** Startet eine neue Zeiterfassung

**Request Body:**

```
{
  "projectId": "proj-123",
  "description": "Working on feature X"
}
```

**Response:** 201 Created

```
{  
  "id": "time-entry-456",  
  "userId": "user-123",  
  "projectId": "proj-123",  
  "clockIn": "2025-01-15T08:00:00.000Z",  
  "clockOut": null,  
  "status": "CLOCKED_IN",  
  "description": "Working on feature X"  
}
```

**Business Rules:**

- Nur ein aktiver TimeEntry pro Benutzer erlaubt
- Projekt muss existieren und Benutzer zugewiesen sein
- Einstempeln in die Zukunft nicht möglich

**Fehler:**

- 400 Bad Request: Bereits eingestempelt oder ungültiges Projekt
- 404 Not Found: Projekt existiert nicht

---

**Ausstempeln****Endpunkt:** POST /api/time/clock-out**Beschreibung:** Beendet die aktuelle Zeiterfassung**Request Body:**

```
{  
  "description": "Completed feature X implementation"  
}
```

**Response:** 200 OK

```
{  
  "id": "time-entry-456",  
  "userId": "user-123",  
  "projectId": "proj-123",  
  "clockIn": "2025-01-15T08:00:00.000Z",  
  "clockOut": "2025-01-15T17:00:00.000Z",  
  "status": "CLOCKED_OUT",  
  "duration": 32400, // Sekunden (9 Stunden)  
  "durationFormatted": "09:00:00",  
}
```

```
        "description": "Completed feature X implementation"
    }
```

## Business Rules:

- Benutzer muss eingestempelt sein
- clockOut kann nicht vor clockIn liegen
- Maximale Arbeitszeit: 10 Stunden (konfigurierbar)

## Fehler:

- **400 Bad Request**: Nicht eingestempelt oder ungültige Zeit
- **422 Unprocessable Entity**: Zeitdauer überschreitet Maximum

## Aktuellen Zeiteintrag abrufen

**Endpunkt:** GET /api/time/current

**Beschreibung:** Gibt den aktuellen aktiven Zeiteintrag zurück

**Response:** 200 OK

```
{
  "id": "time-entry-456",
  "userId": "user-123",
  "projectId": "proj-123",
  "project": {
    "id": "proj-123",
    "name": "Website Redesign"
  },
  "clockIn": "2025-01-15T08:00:00.000Z",
  "clockOut": null,
  "status": "CLOCKED_IN",
  "currentDuration": 14400, // Sekunden seit clockIn
  "currentDurationFormatted": "04:00:00",
  "description": "Working on feature X"
}
```

**Response (nicht eingestempelt):** 200 OK

```
{
  "currentEntry": null,
  "status": "CLOCKED_OUT"
}
```

## Eigene Zeiteinträge abrufen

**Endpunkt:** GET /api/time/my-entries

**Beschreibung:** Liste der eigenen Zeiteinträge

**Query Parameters:**

- `startDate` (optional): ISO Date - Von-Datum (inkl.)
- `endDate` (optional): ISO Date - Bis-Datum (inkl.)
- `projectId` (optional): UUID - Filtert nach Projekt
- `status` (optional): `CLOCKED_IN` | `CLOCKED_OUT`
- `page` (optional): Seitennummer
- `limit` (optional): Einträge pro Seite (max 100)

**Beispiel:** GET /api/time/my-entries?startDate=2025-01-01&endDate=2025-01-31&projectId=proj-123

**Response:** 200 OK

```
{  
  "entries": [  
    {  
      "id": "time-entry-456",  
      "projectId": "proj-123",  
      "project": {  
        "id": "proj-123",  
        "name": "Website Redesign"  
      },  
      "clockIn": "2025-01-15T08:00:00.000Z",  
      "clockOut": "2025-01-15T17:00:00.000Z",  
      "status": "CLOCKED_OUT",  
      "duration": 32400,  
      "durationFormatted": "09:00:00",  
      "description": "Completed feature X"  
    }  
  ],  
  "summary": {  
    "totalEntries": 20,  
    "totalDuration": 648000, // Sekunden  
    "totalDurationFormatted": "180:00:00", // HH:MM:SS  
    "totalHours": 180  
  },  
  "pagination": {  
    "total": 20,  
    "page": 1,  
    "limit": 50,  
    "totalPages": 1  
  }  
}
```

**Endpunkt:** GET /api/time/user/:userId

**Beschreibung:** Zeiteinträge eines bestimmten Benutzers (nur ADMIN)

**Query Parameters:** Gleich wie /my-entries

**Response:** Gleiche Struktur wie /my-entries

**Fehler:**

- 403 Forbidden: Keine Admin-Rechte
- 404 Not Found: Benutzer existiert nicht

---

Zeiteintrag korrigieren

**Endpunkt:** PUT /api/time/:id

**Beschreibung:** Korrigiert einen Zeiteintrag (nur ADMIN)

**Request Body:**

```
{  
  "clockIn": "2025-01-15T08:00:00.000Z",  
  "clockOut": "2025-01-15T17:30:00.000Z",  
  "projectId": "proj-123",  
  "description": "Updated description",  
  "status": "CLOCKED_OUT"  
}
```

**Response:** 200 OK

```
{  
  "id": "time-entry-456",  
  "userId": "user-123",  
  "projectId": "proj-123",  
  "clockIn": "2025-01-15T08:00:00.000Z",  
  "clockOut": "2025-01-15T17:30:00.000Z",  
  "status": "CLOCKED_OUT",  
  "duration": 34200,  
  "durationFormatted": "09:30:00",  
  "description": "Updated description",  
  "updatedAt": "2025-01-16T10:00:00.000Z"  
}
```

**Fehler:**

- 403 Forbidden: Keine Admin-Rechte
- 404 Not Found: Zeiteintrag existiert nicht
- 400 Bad Request: Ungültige Zeitangaben

## Zeiteintrag löschen

**Endpunkt:** `DELETE /api/time/:id`

**Beschreibung:** Löscht einen Zeiteintrag (nur ADMIN)

**Response:** `204 No Content`

**Fehler:**

- `403 Forbidden`: Keine Admin-Rechte
- `404 Not Found`: Zeiteintrag existiert nicht

## Projekte

Alle Projekte abrufen

**Endpunkt:** `GET /api/projects`

**Beschreibung:** Liste aller Projekte (ADMIN: alle, USER: nur zugewiesene)

**Query Parameters:**

- `active` (optional): `true` | `false`
- `page`, `limit`: Pagination

**Response (ADMIN):** `200 OK`

```
{
  "projects": [
    {
      "id": "proj-123",
      "name": "Website Redesign",
      "description": "Complete redesign of company website",
      "active": true,
      "createdAt": "2025-01-01T00:00:00.000Z",
      "assignedUsers": [
        {
          "id": "user-123",
          "firstName": "Max",
          "lastName": "Mustermann"
        }
      ],
      "statistics": {
        "totalHours": 120,
        "totalUsers": 3
      }
    },
    {
      "id": "proj-456",
      "name": "Backend API Development",
      "description": "Implementation of new API endpoints for user management",
      "active": false,
      "createdAt": "2024-12-31T23:59:59.999Z",
      "assignedUsers": [
        {
          "id": "user-456",
          "firstName": "Sarah",
          "lastName": "Schmidt"
        }
      ],
      "statistics": {
        "totalHours": 80,
        "totalUsers": 2
      }
    }
  ],
  "pagination": {
    "total": 10,
    "current": 1,
    "perPage": 5
  }
}
```

```
    "page": 1,
    "limit": 50,
    "totalPages": 1
}
}
```

**Response (USER):** Nur Projekte wo Benutzer zugewiesen ist

---

Eigene Projekte abrufen

**Endpunkt:** GET /api/projects/my-projects

**Beschreibung:** Projekte des aktuellen Benutzers

**Response:** 200 OK

```
{
  "projects": [
    {
      "id": "proj-123",
      "name": "Website Redesign",
      "description": "Complete redesign of company website",
      "active": true,
      "myHours": 45.5,
      "lastClockIn": "2025-01-15T08:00:00.000Z"
    }
  ]
}
```

---

Projekt erstellen

**Endpunkt:** POST /api/projects

**Beschreibung:** Erstellt ein neues Projekt (nur ADMIN)

**Request Body:**

```
{
  "name": "Mobile App Development",
  "description": "Develop iOS and Android app",
  "active": true
}
```

**Response:** 201 Created

```
{  
  "id": "proj-456",  
  "name": "Mobile App Development",  
  "description": "Develop iOS and Android app",  
  "active": true,  
  "createdAt": "2025-01-15T10:00:00.000Z"  
}
```

**Fehler:**

- **403 Forbidden**: Keine Admin-Rechte
- **400 Bad Request**: Name ist erforderlich
- **409 Conflict**: Projekt mit diesem Namen existiert bereits

---

**Projekt aktualisieren****Endpunkt:** PUT /api/projects/:id**Beschreibung:** Aktualisiert Projekt-Daten (nur ADMIN)**Request Body:**

```
{  
  "name": "Mobile App Development (Updated)",  
  "description": "Updated description",  
  "active": false  
}
```

**Response:** 200 OK

```
{  
  "id": "proj-456",  
  "name": "Mobile App Development (Updated)",  
  "description": "Updated description",  
  "active": false,  
  "updatedAt": "2025-01-15T11:00:00.000Z"  
}
```

---

**Projekt löschen****Endpunkt:** DELETE /api/projects/:id**Beschreibung:** Löscht ein Projekt (Soft Delete, nur ADMIN)**Response:** 204 No Content

**Hinweis:** Projekt wird auf `active: false` gesetzt. Bestehende Zeitbuchungen bleiben erhalten.

### Fehler:

- `403 Forbidden`: Keine Admin-Rechte
- `404 Not Found`: Projekt existiert nicht

---

Benutzer zu Projekt zuweisen

**Endpunkt:** `POST /api/projects/:id/assign`

**Beschreibung:** Weist einen Benutzer einem Projekt zu (nur ADMIN)

### Request Body:

```
{  
  "userId": "user-123"  
}
```

**Response:** `200 OK`

```
{  
  "project": {  
    "id": "proj-456",  
    "name": "Mobile App Development"  
  },  
  "user": {  
    "id": "user-123",  
    "firstName": "Max",  
    "lastName": "Mustermann"  
  },  
  "assignedAt": "2025-01-15T12:00:00.000Z"  
}
```

### Fehler:

- `403 Forbidden`: Keine Admin-Rechte
- `404 Not Found`: Projekt oder Benutzer existiert nicht
- `409 Conflict`: Benutzer bereits zugewiesen

---

Benutzer von Projekt entfernen

**Endpunkt:** `DELETE /api/projects/:id/unassign/:userId`

**Beschreibung:** Entfernt einen Benutzer von einem Projekt (nur ADMIN)

**Response:** `204 No Content`

**Fehler:**

- **403 Forbidden**: Keine Admin-Rechte
  - **404 Not Found**: Projekt, Benutzer oder Zuweisung existiert nicht
- 

## Abwesenheiten

Abwesenheitsantrag erstellen

**Endpunkt:** POST /api/absences

**Beschreibung:** Erstellt einen neuen Abwesenheitsantrag

**Request Body:**

```
{  
  "type": "VACATION",  
  "startDate": "2025-02-01",  
  "endDate": "2025-02-07",  
  "reason": "Family vacation",  
  "halfDay": false  
}
```

### Abwesenheits-Typen:

- **VACATION** - Urlaub
- **SICK** - Krankheit
- **PERSONAL** - Persönliche Gründe
- **TRAINING** - Weiterbildung
- **HOME\_OFFICE** - Home Office
- **OTHER** - Sonstige

**Response:** 201 Created

```
{  
  "id": "absence-789",  
  "userId": "user-123",  
  "type": "VACATION",  
  "startDate": "2025-02-01T00:00:00.000Z",  
  "endDate": "2025-02-07T23:59:59.999Z",  
  "reason": "Family vacation",  
  "halfDay": false,  
  "status": "PENDING",  
  "days": 5, // Arbeitstage (ohne Wochenende)  
  "createdAt": "2025-01-15T13:00:00.000Z"  
}
```

### Business Rules:

- Urlaub: Verfügbare Urlaubstage werden geprüft
  - startDate muss vor oder gleich endDate sein
  - Keine überlappenden Anträge erlaubt
  - Minimale Vorlaufzeit: 2 Tage (konfigurierbar)

## Fehler:

- **400 Bad Request:** Ungültige Daten oder nicht genug Urlaubstage
  - **409 Conflict:** Überlappende Abwesenheit existiert

#### Eigene Abwesenheitsanträge abrufen

**Endpunkt:** GET /api/absences/my-requests

**Beschreibung:** Liste der eigenen Abwesenheitsanträge

## Query Parameters:

- **status** (optional): PENDING | APPROVED | REJECTED
  - **type** (optional): Abwesenheitstyp
  - **year** (optional): Jahr (z.B. 2025)
  - **page, limit**: Pagination

## Response: 200 OK

```
{
  "absences": [
    {
      "id": "absence-789",
      "type": "VACATION",
      "startDate": "2025-02-01T00:00:00.000Z",
      "endDate": "2025-02-07T23:59:59.999Z",
      "reason": "Family vacation",
      "status": "APPROVED",
      "days": 5,
      "approvedBy": {
        "id": "admin-123",
        "firstName": "Admin",
        "lastName": "User"
      },
      "approvedAt": "2025-01-16T09:00:00.000Z",
      "createdAt": "2025-01-15T13:00:00.000Z"
    }
  ],
  "summary": {
    "totalPending": 0,
    "totalApproved": 5,
    "totalRejected": 1,
    "vacationDaysUsed": 5,
    "vacationDaysRemaining": 20
  }
},
```

```

"pagination": {
    "total": 6,
    "page": 1,
    "limit": 50,
    "totalPages": 1
}
}

```

## Alle Abwesenheitsanträge abrufen

**Endpunkt:** GET /api/absences

**Beschreibung:** Liste aller Abwesenheitsanträge (nur ADMIN)

**Query Parameters:** Gleich wie /my-requests plus:

- **userId** (optional): Filtert nach Benutzer

**Response:** Gleiche Struktur wie /my-requests, aber mit User-Info:

```

{
  "absences": [
    {
      "id": "absence-789",
      "user": {
        "id": "user-123",
        "firstName": "Max",
        "lastName": "Mustermann",
        "email": "max@example.com"
      },
      "type": "VACATION",
      "startDate": "2025-02-01T00:00:00.000Z",
      "endDate": "2025-02-07T23:59:59.999Z",
      "reason": "Family vacation",
      "status": "PENDING",
      "days": 5,
      "createdAt": "2025-01-15T13:00:00.000Z"
    }
  ]
}

```

## Abwesenheitsantrag genehmigen

**Endpunkt:** PUT /api/absences/:id/approve

**Beschreibung:** Genehmigt einen Abwesenheitsantrag (nur ADMIN)

**Request Body (optional):**

```
{  
    "comment": "Genehmigt, viel Spaß!"  
}
```

**Response:** 200 OK

```
{  
    "id": "absence-789",  
    "status": "APPROVED",  
    "approvedBy": {  
        "id": "admin-123",  
        "firstName": "Admin",  
        "lastName": "User"  
    },  
    "approvedAt": "2025-01-16T09:00:00.000Z",  
    "comment": "Genehmigt, viel Spaß!"  
}
```

### Business Rules:

- Bei VACATION: Urlaubstage werden vom Kontingent abgezogen
- E-Mail-Benachrichtigung an Benutzer (falls konfiguriert)
- Antrag kann nur einmal genehmigt werden

### Fehler:

- 403 Forbidden: Keine Admin-Rechte
- 404 Not Found: Antrag existiert nicht
- 409 Conflict: Antrag bereits genehmigt/abgelehnt

---

Abwesenheitsantrag ablehnen

**Endpunkt:** PUT /api/absences/:id/reject

**Beschreibung:** Lehnt einen Abwesenheitsantrag ab (nur ADMIN)

### Request Body:

```
{  
    "comment": "Konflikt mit Projekttermin"  
}
```

**Response:** 200 OK

```
{  
    "id": "absence-789",  
    "status": "REJECTED",  
    "rejectedBy": {  
        "id": "admin-123",  
        "firstName": "Admin",  
        "lastName": "User"  
    },  
    "rejectedAt": "2025-01-16T09:00:00.000Z",  
    "comment": "Konflikt mit Projekttermin"  
}
```

---

## Abwesenheitsantrag löschen

**Endpunkt:** `DELETE /api/absences/:id`

**Beschreibung:** Löscht einen Abwesenheitsantrag (nur ADMIN oder eigener PENDING Antrag)

**Response:** `204 No Content`

**Business Rules:**

- Benutzer können nur eigene PENDING Anträge löschen
- Admin kann alle Anträge löschen
- APPROVED Urlaubsanträge: Urlaubstage werden zurückgebucht

**Fehler:**

- `403 Forbidden`: Keine Berechtigung
- `404 Not Found`: Antrag existiert nicht

---

## Reports

### Eigene Zusammenfassung

**Endpunkt:** `GET /api/reports/my-summary`

**Beschreibung:** Persönliche Arbeitszeit-Zusammenfassung

**Query Parameters:**

- `startDate` (optional): ISO Date
- `endDate` (optional): ISO Date
- `groupBy` (optional): `day` | `week` | `month` | `project`

**Beispiel:** `GET /api/reports/my-summary?startDate=2025-01-01&endDate=2025-01-31&groupBy=project`

**Response:** `200 OK`

```
{  
  "summary": {  
    "totalHours": 160,  
    "totalDays": 20,  
    "averageHoursPerDay": 8,  
    "period": {  
      "start": "2025-01-01T00:00:00.000Z",  
      "end": "2025-01-31T23:59:59.999Z"  
    }  
  },  
  "breakdown": {  
    "byProject": [  
      {  
        "projectId": "proj-123",  
        " projectName": "Website Redesign",  
        "hours": 80,  
        "percentage": 50  
      },  
      {  
        "projectId": "proj-456",  
        " projectName": "Mobile App",  
        "hours": 80,  
        "percentage": 50  
      }  
    ],  
    "byDay": [  
      {  
        "date": "2025-01-01",  
        "hours": 8,  
        "entries": 1  
      },  
      {  
        "date": "2025-01-02",  
        "hours": 8.5,  
        "entries": 2  
      }  
    ]  
  },  
  "absences": {  
    "totalDays": 2,  
    "byType": [  
      {  
        "type": "VACATION",  
        "days": 2  
      }  
    ]  
  }  
}
```

**Endpunkt:** GET /api/reports/user-summary/:userId

**Beschreibung:** Arbeitszeit-Zusammenfassung für einen Benutzer (nur ADMIN)

**Query Parameters:** Gleich wie /my-summary

**Response:** Gleiche Struktur wie /my-summary plus User-Info

**Fehler:**

- 403 Forbidden: Keine Admin-Rechte
- 404 Not Found: Benutzer existiert nicht

---

## Alle Benutzer Zusammenfassung

**Endpunkt:** GET /api/reports/all-users-summary

**Beschreibung:** Übersicht aller Benutzer (nur ADMIN)

**Query Parameters:**

- startDate, endDate: Zeitraum
- sortBy: hours | name | department
- order: asc | desc

**Response:** 200 OK

```
{
  "period": {
    "start": "2025-01-01T00:00:00.000Z",
    "end": "2025-01-31T23:59:59.999Z"
  },
  "users": [
    {
      "userId": "user-123",
      "firstName": "Max",
      "lastName": "Mustermann",
      "totalHours": 160,
      "totalDays": 20,
      "vacationDays": 2,
      "sickDays": 0,
      "projects": [
        {
          "projectId": "proj-123",
          " projectName": "Website Redesign",
          "hours": 160
        }
      ]
    }
  ],
  "totals": {
    "totalHours": 3200,
```

```
        "totalUsers": 20,  
        "averageHoursPerUser": 160  
    }  
}
```

---

## Projekt-Zusammenfassung

**Endpunkt:** GET /api/reports/project-summary/:projectId

**Beschreibung:** Arbeitszeit-Zusammenfassung für ein Projekt (nur ADMIN)

**Query Parameters:**

- `startDate, endDate`: Zeitraum
- `includeInactive: true | false` - Inaktive Benutzer einbeziehen

**Response:** 200 OK

```
{  
  "project": {  
    "id": "proj-123",  
    "name": "Website Redesign",  
    "description": "Complete redesign",  
    "active": true  
  },  
  "period": {  
    "start": "2025-01-01T00:00:00.000Z",  
    "end": "2025-01-31T23:59:59.999Z"  
  },  
  "summary": {  
    "totalHours": 320,  
    "totalUsers": 4,  
    "averageHoursPerUser": 80  
  },  
  "userBreakdown": [  
    {  
      "userId": "user-123",  
      "firstName": "Max",  
      "lastName": "Mustermann",  
      "hours": 80,  
      "percentage": 25,  
      "lastEntry": "2025-01-31T17:00:00.000Z"  
    }  
  ],  
  "timeline": [  
    {  
      "week": "2025-W01",  
      "hours": 80  
    },  
    {  
      "week": "2025-W02",  
      "hours": 80  
    }  
  ]  
}
```

```
        "hours": 80
    }
]
}
```

## Rechnungen

### Rechnung erstellen

**Endpunkt:** POST /api/invoices

**Beschreibung:** Erstellt eine neue Rechnung (nur ADMIN)

**Request Body:**

```
{
  "customerId": "customer-123",
  "projectId": "proj-123",
  "invoiceNumber": "RE-2025-001",
  "invoiceDate": "2025-01-31",
  "dueDate": "2025-02-28",
  "items": [
    {
      "description": "Entwicklung Feature X",
      "quantity": 40,
      "unit": "Stunden",
      "unitPrice": 120,
      "vatRate": 8.1
    },
    {
      "description": "Projektmanagement",
      "quantity": 10,
      "unit": "Stunden",
      "unitPrice": 150,
      "vatRate": 8.1
    }
  ],
  "notes": "Zahlbar innerhalb 30 Tagen",
  "vatIncluded": false
}
```

**Response:** 201 Created

```
{
  "id": "invoice-999",
  "invoiceNumber": "RE-2025-001",
  "status": "DRAFT",
  "customerId": "customer-123",
```

```

"projectId": "proj-123",
"invoiceDate": "2025-01-31T00:00:00.000Z",
"dueDate": "2025-02-28T00:00:00.000Z",
"items": [
  {
    "id": "item-1",
    "description": "Entwicklung Feature X",
    "quantity": 40,
    "unit": "Stunden",
    "unitPrice": 120,
    "subtotal": 4800,
    "vatRate": 8.1,
    "vatAmount": 388.80,
    "total": 5188.80
  },
  {
    "id": "item-2",
    "description": "Projektmanagement",
    "quantity": 10,
    "unit": "Stunden",
    "unitPrice": 150,
    "subtotal": 1500,
    "vatRate": 8.1,
    "vatAmount": 121.50,
    "total": 1621.50
  }
],
"subtotal": 6300,
"vatTotal": 510.30,
"grandTotal": 6810.30,
"currency": "CHF",
"qrCode": "base64-encoded-qr-code",
"createdAt": "2025-01-31T10:00:00.000Z"
}

```

## Swiss QR Code Fields:

- IBAN
- Rechnungsbetrag
- Währung (CHF/EUR)
- Referenznummer

## Rechnungen abrufen

**Endpunkt:** GET /api/invoices

**Beschreibung:** Liste aller Rechnungen (nur ADMIN)

### Query Parameters:

- **status:** DRAFT | SENT | PAID | OVERDUE | CANCELLED

- `customerId`: Filter nach Kunde
- `projectId`: Filter nach Projekt
- `year`: Jahr
- `page, limit`: Pagination

**Response:** 200 OK

```
{  
  "invoices": [  
    {  
      "id": "invoice-999",  
      "invoiceNumber": "RE-2025-001",  
      "status": "SENT",  
      "customer": {  
        "id": "customer-123",  
        "name": "Acme Corp"  
      },  
      "invoiceDate": "2025-01-31T00:00:00.000Z",  
      "dueDate": "2025-02-28T00:00:00.000Z",  
      "grandTotal": 6810.30,  
      "currency": "CHF",  
      "isPaid": false,  
      "isOverdue": false  
    },  
    ],  
    "summary": {  
      "totalDraft": 2,  
      "totalSent": 5,  
      "totalPaid": 10,  
      "totalOverdue": 1,  
      "totalAmount": 68103.00,  
      "totalPaidAmount": 50000.00,  
      "totalOutstanding": 18103.00  
    }  
}
```

---

Rechnung als PDF exportieren

**Endpunkt:** GET /api/invoices/:id/pdf

**Beschreibung:** Generiert PDF der Rechnung

**Response:** 200 OK - Content-Type: application/pdf

**Headers:**

```
Content-Type: application/pdf  
Content-Disposition: attachment; filename="RE-2025-001.pdf"
```

---

## Rechnungsstatus ändern

**Endpunkt:** PUT /api/invoices/:id/status

**Beschreibung:** Ändert den Status einer Rechnung

**Request Body:**

```
{  
  "status": "PAID",  
  "paidDate": "2025-02-15",  
  "paymentMethod": "Bank Transfer",  
  "notes": "Received payment"  
}
```

**Response:** 200 OK

```
{  
  "id": "invoice-999",  
  "status": "PAID",  
  "paidDate": "2025-02-15T00:00:00.000Z",  
  "paymentMethod": "Bank Transfer",  
  "updatedAt": "2025-02-15T10:00:00.000Z"  
}
```

---

## Fehlerbehandlung

Standard Fehler-Format

Alle API-Fehler folgen diesem Format:

```
{  
  "error": {  
    "code": "ERROR_CODE",  
    "message": "Benutzerfreundliche Fehlermeldung",  
    "details": [  
      {  
        "field": "email",  
        "message": "Email ist ungültig"  
      }  
    ],  
    "timestamp": "2025-01-15T10:00:00.000Z",  
    "path": "/api/users/register"  
  }  
}
```

## HTTP Status Codes

<b>Code</b>	<b>Bedeutung</b>	<b>Verwendung</b>
200	OK	Erfolgreiche GET, PUT Anfragen
201	Created	Erfolgreiche POST Anfragen (Resource erstellt)
204	No Content	Erfolgreiche DELETE Anfragen
400	Bad Request	Validierungsfehler, ungültige Eingaben
401	Unauthorized	Fehlende oder ungültige Authentication
403	Forbidden	Fehlende Berechtigung
404	Not Found	Resource nicht gefunden
409	Conflict	Konflikt (z.B. E-Mail bereits registriert)
422	Unprocessable Entity	Business Logic Fehler
429	Too Many Requests	Rate Limit überschritten
500	Internal Server Error	Serverfehler

## Validierungsfehler

```
{
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Eingabe-Validierung fehlgeschlagen",
    "details": [
      {
        "field": "email",
        "message": "Email ist erforderlich"
      },
      {
        "field": "password",
        "message": "Passwort muss mindestens 8 Zeichen haben"
      }
    ]
  }
}
```

## Authentication Fehler

```
{
  "error": {
    "code": "AUTHENTICATION_FAILED",
    "message": "Ungültige Credentials"
```

```
    }
}
```

## Authorization Fehler

```
{
  "error": {
    "code": "INSUFFICIENT_PERMISSIONS",
    "message": "Administrator-Rechte erforderlich"
  }
}
```

## Anhang

### Beispiel Workflow: Vollständiger Arbeitsablauf

```
// 1. Login
POST /api/auth/login
{
  "email": "max@example.com",
  "password": "secure123"
}
→ Token: "eyJhbGci..."  
  

// 2. Einstempeln
POST /api/time/clock-in
Headers: { Authorization: "Bearer eyJhbGci..." }
{
  "projectId": "proj-123",
  "description": "Working on feature X"
}  
  

// 3. Arbeit erledigen...  
  

// 4. Ausstempeln
POST /api/time/clock-out
{
  "description": "Completed feature X"
}  
  

// 5. Übersicht abrufen
GET /api/time/my-entries?startDate=2025-01-01&endDate=2025-01-31  
  

// 6. Urlaubsantrag
POST /api/absences
{
  "type": "VACATION",
  "startDate": "2025-02-01",
```

```
"endDate": "2025-02-07",
"reason": "Family vacation"
}
```

## Rate Limiting

Endpunkt	Limit	Zeitfenster
/api/auth/login	5	15 Minuten
/api/auth/register	3	1 Stunde
Alle anderen	100	15 Minuten

## API Versioning

Aktuell: v1 (im URL-Path implizit)

Zukünftig: [/api/v2/...](#) bei Breaking Changes

---

**API Dokumentation Ende**