

cflux - Zeiterfassungssystem

Vollständige Dokumentation

Version 1.0

Stand: Januar 2025

Inhaltsverzeichnis

1. [Überblick](#)
 2. [Architektur](#)
 3. [Systemanforderungen](#)
 4. [Installation](#)
 5. [Konfiguration](#)
 6. [Module und Features](#)
 7. [Sicherheit](#)
 8. [Deployment](#)
 9. [Troubleshooting](#)
-

Überblick

Was ist cflux?

cflux ist ein vollständiges Enterprise Resource Planning (ERP) System mit Fokus auf Zeiterfassung, Projektmanagement und Personalverwaltung. Das System wurde speziell für den Schweizer Markt entwickelt und erfüllt alle relevanten gesetzlichen Anforderungen.

Hauptmerkmale

- **Zeiterfassung:** Ein-/Ausstempeln mit Projektbuchung
- **Projektmanagement:** Verwaltung von Projekten und Zuordnung zu Mitarbeitern
- **Urlaubsverwaltung:** Antragstellung und Genehmigungsprozesse
- **Abwesenheitsmanagement:** Krankheit, persönliche Gründe, etc.
- **Reporting:** Umfassende Reports und Statistiken
- **Benutzerverwaltung:** Rollenbasierte Zugriffskontrolle
- **Rechnungsstellung:** Integriertes Invoicing-System
- **Schweizer Compliance:** OR-konform, Mehrwertsteuer-ready

Technologie-Stack

Backend:

- Node.js 18+ / TypeScript 5.x
- Express.js Framework
- PostgreSQL 14+ Datenbank
- Prisma ORM

- JWT Authentication
- bcrypt Password Hashing

Frontend:

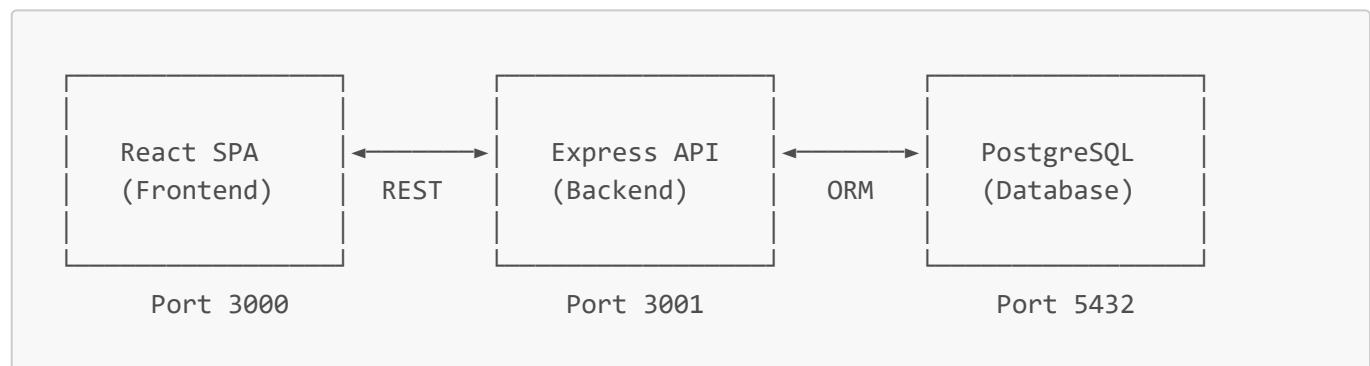
- React 18 / TypeScript
- React Router v6
- Axios HTTP Client
- Modern CSS3

DevOps:

- Docker & Docker Compose
- GitHub Actions CI/CD
- Nginx Reverse Proxy

Architektur

System-Übersicht



Datenbank-Schema

Das System basiert auf folgenden Hauptentitäten:

User (Benutzer)

- Basisdaten: Email, Name, Passwort
- Rolle: USER, ADMIN
- Urlaubskontingent
- Status: Aktiv/Inaktiv

TimeEntry (Zeiteintrag)

- Zuordnung: Benutzer, Projekt
- Zeitstempel: clockIn, clockOut
- Status: CLOCKED_IN, CLOCKED_OUT
- Beschreibung/Notizen

Project (Projekt)

- Projektstammdaten

- Zugewiesene Benutzer (Many-to-Many)
- Status: Aktiv/Inaktiv

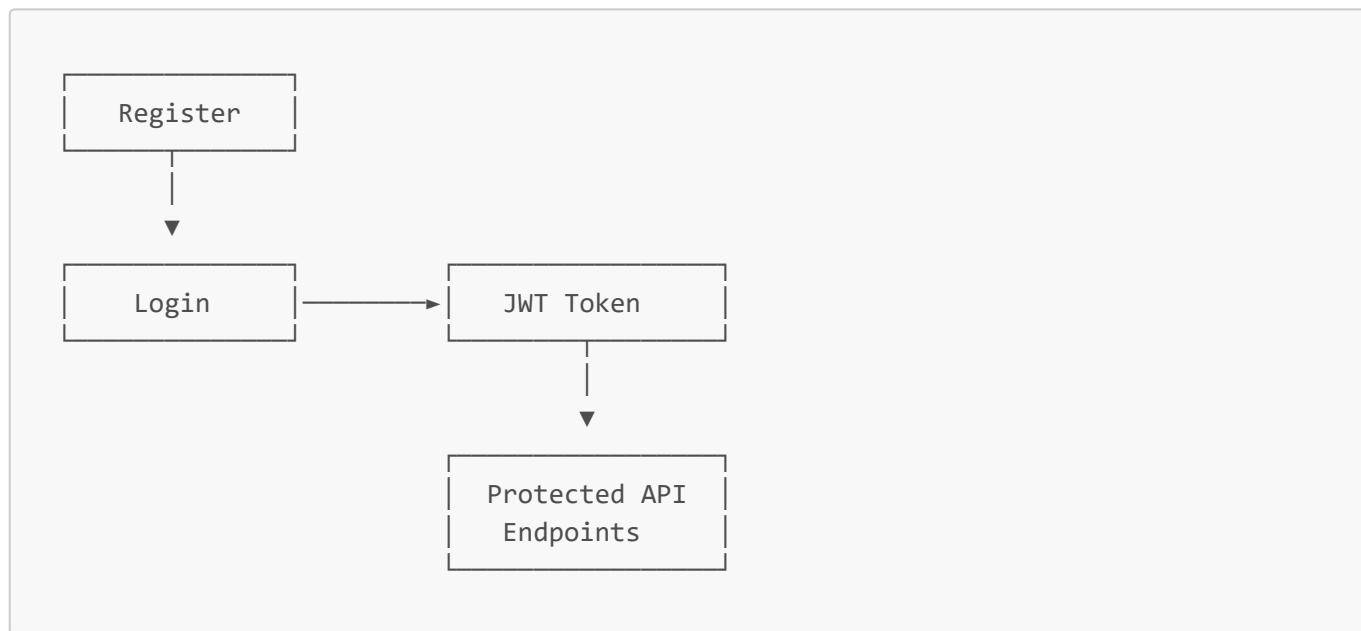
Absence (Abwesenheit)

- Typ: Urlaub, Krankheit, Sonstige
- Zeitraum: Start-/Enddatum
- Status: Pending, Approved, Rejected
- Kommentare

Invoice (Rechnung)

- Rechnungsnummer
- Kunde/Projekt
- Positionen
- Mehrwertsteuer (MWST)
- Status: Draft, Sent, Paid

Authentifizierung



JWT Token enthält:

- User ID
- Email
- Role (USER/ADMIN)
- Expiration (24h Standard)

Systemanforderungen

Produktionsumgebung

Hardware (Minimum für 50 Benutzer):

- CPU: 2 Cores

- RAM: 4 GB
- Storage: 20 GB SSD
- Netzwerk: 100 Mbit/s

Software:

- Docker 20.10+
- Docker Compose 2.0+
- Linux (Ubuntu 22.04 LTS empfohlen)

Für größere Deployments (100+ Benutzer):

- CPU: 4+ Cores
- RAM: 8+ GB
- Storage: 50+ GB SSD
- Load Balancer empfohlen

Entwicklungsumgebung

- Node.js 18 oder höher
- npm 9+ oder yarn 1.22+
- PostgreSQL 14+
- Git
- Code Editor (VS Code empfohlen)

Installation

Produktions-Setup mit Docker

Schritt 1: Repository klonen

```
git clone https://github.com/mpue/cflux.git  
cd cflux
```

Schritt 2: Umgebungsvariablen konfigurieren

```
# Backend .env  
cp backend/.env.example backend/.env  
  
# Wichtige Variablen in backend/.env anpassen:  
DATABASE_URL="postgresql://timetracking:SICHERES_PASSWORD@db:5432/timetracking"  
JWT_SECRET="SEHR_SICHERER_RANDOM_STRING_MINIMUM_32_ZEICHEN"  
NODE_ENV="production"  
PORT=3001
```

Schritt 3: Docker Compose starten

```
docker-compose up -d
```

Das System ist nun verfügbar unter:

- Frontend: <http://localhost:3000>
- Backend API: <http://localhost:3001>
- Datenbank: localhost:5432

Schritt 4: Ersten Admin erstellen

```
# 1. Registriere einen Benutzer über die Web-Oberfläche  
(http://localhost:3000/register)  
# 2. Setze die Rolle auf ADMIN:  
docker exec -it timetracking-db psql -U timetracking -d timetracking
```

In der psql-Shell:

```
UPDATE users SET role = 'ADMIN' WHERE email = 'admin@example.com';  
\q
```

Entwicklungs-Setup (Lokal)

Backend:

```
cd backend  
npm install  
cp .env.example .env  
# .env anpassen  
  
# Datenbank Migrationen  
npm run prisma:migrate  
npm run prisma:generate  
  
# Development Server starten  
npm run dev
```

Frontend:

```
cd frontend  
npm install  
npm start
```

Konfiguration

Umgebungsvariablen (Backend)

Variable	Beschreibung	Beispiel	Pflicht
DATABASE_URL	PostgreSQL Verbindung	postgresql://user:pass@host:5432/db	Ja
JWT_SECRET	Secret für JWT Tokens	random-string-min-32-chars	Ja
JWT_EXPIRATION	Token Gültigkeitsdauer	24h	Nein (Default: 24h)
PORT	Backend Port	3001	Nein (Default: 3001)
NODE_ENV	Umgebung	production / development	Nein
CORS_ORIGIN	Erlaubte Origins	http://localhost:3000	Nein
LOG_LEVEL	Logging Level	info / debug / error	Nein

Docker Compose Konfiguration

Die `docker-compose.yml` kann angepasst werden für:

Ports ändern:

```
services:
  frontend:
    ports:
      - "8080:80" # Frontend auf Port 8080
  backend:
    ports:
      - "8081:3001" # Backend auf Port 8081
```

Persistent Volumes:

```
volumes:
  postgres_data:
    driver: local
    driver_opts:
      type: none
      device: /opt/cflux/data
      o: bind
```

Memory Limits:

```
services:  
  backend:  
    deploy:  
      resources:  
        limits:  
          memory: 1G  
        reservations:  
          memory: 512M
```

Nginx Konfiguration (Reverse Proxy)

Für Produktions-Deployment mit eigenem Nginx:

```
server {  
  listen 80;  
  server_name zeiterfassung.example.com;  
  
  location / {  
    proxy_pass http://localhost:3000;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection 'upgrade';  
    proxy_set_header Host $host;  
    proxy_cache_bypass $http_upgrade;  
  }  
  
  location /api {  
    proxy_pass http://localhost:3001;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection 'upgrade';  
    proxy_set_header Host $host;  
    proxy_cache_bypass $http_upgrade;  
  }  
}
```

Mit SSL (Let's Encrypt):

```
certbot --nginx -d zeiterfassung.example.com
```

Module und Features

1. Zeiterfassung

Funktionen:

- Ein-/Ausstempeln per Button-Click
- Automatische Zeitberechnung
- Projektbuchung während Arbeitszeit
- Nachträgliche Korrekturen (Admin)
- Pausenzeiten-Erfassung

Workflows:

1. Benutzer stempelt ein → Status: CLOCKED_IN
2. Benutzer wählt Projekt aus
3. Benutzer stempelt aus → Zeitdauer wird berechnet
4. Eintrag wird gespeichert und in Reports berücksichtigt

Business Rules:

- Maximale Arbeitszeit pro Tag: 10 Stunden (konfigurierbar)
- Pflicht-Pausen bei >6h Arbeitszeit
- Nur ein aktiver TimeEntry pro Benutzer
- Zeiteinträge können nicht in die Zukunft gebucht werden

2. Projektverwaltung**Funktionen:**

- Projekte anlegen, bearbeiten, archivieren
- Mitarbeiter zu Projekten zuweisen
- Projektstatus (Aktiv/Inaktiv)
- Projektbasierte Zeitauswertung

Zugriffsrechte:

- USER: Sieht nur zugewiesene Projekte
- ADMIN: Vollzugriff auf alle Projekte

3. Urlaubsverwaltung**Funktionen:**

- Urlaubsanträge stellen
- Genehmigungsprozess (Admin)
- Urlaubskontingent-Verwaltung
- Urlaubskalender
- Resturlausberechnung

Workflows:

1. Benutzer stellt Urlaubsantrag
2. Status: PENDING
3. Admin genehmigt/lehnt ab
4. Status: APPROVED/REJECTED
5. Bei Genehmigung: Urlaubstage werden vom Kontingent abgezogen

4. Abwesenheitsmanagement

Typen:

- Urlaub
- Krankheit
- Persönliche Gründe
- Weiterbildung
- Home Office
- Sonstige

Features:

- Ganztags- oder Teilzeit-Abwesenheiten
- Ärztliche Bescheinigung (Upload optional)
- Benachrichtigungen an Vorgesetzte
- Kalenderintegration

5. Reporting & Analytics

Standard Reports:

- Persönliche Arbeitszeitübersicht
- Projekt-Zeitauswertung
- Team-Übersicht (Admin)
- Urlaubsübersicht
- Fehlzeiten-Report
- Monats-/Jahresberichte

Export-Formate:

- PDF
- Excel (XLSX)
- CSV

Auswertungszeiträume:

- Tag
- Woche
- Monat
- Quartal
- Jahr
- Frei wählbar

6. Benutzerverwaltung

Admin-Funktionen:

- Benutzer anlegen/bearbeiten/löschen
- Rollenänderungen (USER ↔ ADMIN)
- Urlaubskontingent festlegen

- Benutzer deaktivieren (Soft Delete)
- Passwort zurücksetzen

Benutzer-Selbstverwaltung:

- Profil bearbeiten
- Passwort ändern
- Eigene Zeiteinträge einsehen

7. Rechnungsstellung

Funktionen:

- Rechnungen erstellen und versenden
- Mehrwertsteuer (MWST) Berechnung
- Rechnungspositionen mit Zeitbuchungen verknüpfen
- Zahlungsstatus-Tracking
- PDF-Export mit QR-Rechnung (Schweiz)

Swiss QR Code Integration:

- Automatische QR-Code Generierung
- IBAN/Referenznummer
- Compliance mit Swiss Payment Standards

Sicherheit

Authentifizierung & Autorisierung

JWT Token Security:

- Tokens sind signiert und verschlüsselt
- Kurze Lebensdauer (24h Standard)
- Refresh Token Mechanismus (optional aktivierbar)
- Tokens werden im LocalStorage gespeichert (XSS-Protection durch HTTPOnly Cookies optional)

Password Security:

- bcrypt Hashing mit Salt (10 Rounds)
- Mindestlänge: 8 Zeichen
- Passwort-Stärke-Validierung
- Rate Limiting bei Login-Versuchen

Role-Based Access Control (RBAC):

```
// Middleware Beispiel
const requireAdmin = (req, res, next) => {
  if (req.user.role !== 'ADMIN') {
    return res.status(403).json({ error: 'Admin rechte erforderlich' });
  }
}
```

```
next();
};
```

Input Validation

Alle API-Endpunkte nutzen **express-validator**:

```
body('email').isEmail().normalizeEmail(),
body('password').isLength({ min: 8 }),
body('firstName').trim().notEmpty(),
```

SQL Injection Protection

- Prisma ORM verhindert SQL Injection durch Prepared Statements
- Keine direkten SQL Queries im Code
- Input Sanitization vor Datenbankzugriff

CORS Configuration

```
app.use(cors({
  origin: process.env.CORS_ORIGIN || 'http://localhost:3000',
  credentials: true,
  methods: ['GET', 'POST', 'PUT', 'DELETE'],
}));
```

Rate Limiting

```
const rateLimit = require('express-rate-limit');

const loginLimiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 Minuten
  max: 5, // 5 Versuche
  message: 'Zu viele Login-Versuche, bitte später erneut versuchen'
});

app.post('/api/auth/login', loginLimiter, ...);
```

HTTPS in Produktion

Empfohlener Setup mit Let's Encrypt:

```
# Certbot installieren
sudo apt-get install certbot python3-certbot-nginx
```

```
# Zertifikat erstellen
sudo certbot --nginx -d your-domain.com

# Auto-Renewal
sudo certbot renew --dry-run
```

Deployment

Produktions-Deployment Checkliste

- Umgebungsvariablen in `.env` setzen
- Sichere Passwörter generieren (DB, JWT_SECRET)
- HTTPS/SSL konfigurieren
- Firewall-Regeln setzen
- Backup-Strategie implementieren
- Monitoring aufsetzen
- Log-Rotation konfigurieren
- E-Mail-Benachrichtigungen testen
- Disaster Recovery Plan

Docker Production Deployment

docker-compose.prod.yml:

```
version: '3.8'

services:
  db:
    image: postgres:16-alpine
    restart: always
    environment:
      POSTGRES_DB: ${DB_NAME}
      POSTGRES_USER: ${DB_USER}
      POSTGRES_PASSWORD: ${DB_PASSWORD}
    volumes:
      - /opt/cflux/data:/var/lib/postgresql/data
    networks:
      - cflux-network

  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile.prod
    restart: always
    environment:
      NODE_ENV: production
      DATABASE_URL: ${DATABASE_URL}
      JWT_SECRET: ${JWT_SECRET}
    depends_on:
```

```

      - db
networks:
  - cflux-network

frontend:
  build:
    context: ./frontend
    dockerfile: Dockerfile.prod
  restart: always
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - /etc/letsencrypt:/etc/letsencrypt:ro
  depends_on:
    - backend
  networks:
    - cflux-network

networks:
  cflux-network:
    driver: bridge

volumes:
  postgres_data:

```

Deployment Befehle:

```

# Build und Start
docker-compose -f docker-compose.prod.yml up -d --build

# Logs
docker-compose -f docker-compose.prod.yml logs -f

# Backup
docker exec timetracking-db pg_dump -U timetracking timetracking > backup_$(date +%Y%m%d).sql

# Restore
docker exec -i timetracking-db psql -U timetracking timetracking < backup.sql

```

Monitoring & Logging

Prometheus + Grafana Setup:

```

# prometheus.yml
global:
  scrape_interval: 15s

```

```

scrape_configs:
  - job_name: 'cflux'
    static_configs:
      - targets: ['backend:3001']

```

Log Aggregation mit ELK Stack:

- Elasticsearch: Log Storage
- Logstash: Log Processing
- Kibana: Visualization

Health Checks:

```

// Backend Health Endpoint
app.get('/health', async (req, res) => {
  const dbStatus = await checkDatabase();
  res.json({
    status: 'ok',
    database: dbStatus,
    uptime: process.uptime(),
    timestamp: new Date()
  });
});

```

Backup-Strategie

Automatisches tägliches Backup:

```

#!/bin/bash
# /opt/cflux/backup.sh

BACKUP_DIR="/opt/cflux/backups"
DATE=$(date +%Y%m%d_%H%M%S)

# Database Backup
docker exec timetracking-db pg_dump -U timetracking timetracking | gzip > \
$BACKUP_DIR/db_backup_$DATE.sql.gz

# Alte Backups löschen (älter als 30 Tage)
find $BACKUP_DIR -name "db_backup_*.sql.gz" -mtime +30 -delete

```

Crontab Eintrag:

```
0 2 * * * /opt/cflux/backup.sh
```

Scaling

Horizontal Scaling:

```
# docker-compose.scale.yml
services:
  backend:
    deploy:
      replicas: 3

  nginx-lb:
    image: nginx:alpine
    volumes:
      - ./nginx-lb.conf:/etc/nginx/nginx.conf
    ports:
      - "80:80"
```

Load Balancer Konfiguration:

```
upstream backend {
  least_conn;
  server backend1:3001;
  server backend2:3001;
  server backend3:3001;
}

server {
  location /api {
    proxy_pass http://backend;
  }
}
```

Troubleshooting

Häufige Probleme

Problem: "Cannot connect to database"

Ursache: Datenbank nicht erreichbar oder falsche Credentials

Lösung:

```
# Prüfe ob DB Container läuft
docker ps | grep timetracking-db

# Prüfe Logs
docker logs timetracking-db

# Teste Verbindung
```

```
docker exec -it timetracking-db psql -U timetracking -d timetracking -c "SELECT 1;"  
  
# Prüfe DATABASE_URL in .env  
cat backend/.env | grep DATABASE_URL
```

Problem: "JWT token expired"

Ursache: Token-Lebensdauer abgelaufen

Lösung:

- Benutzer muss sich neu anmelden
- Optional: Refresh Token implementieren
- Token-Lebensdauer in .env anpassen: `JWT_EXPIRATION=48h`

Problem: Frontend zeigt "Cannot connect to API"

Ursache: CORS oder Backend nicht erreichbar

Lösung:

```
# Prüfe Backend Status  
curl http://localhost:3001/health  
  
# Prüfe CORS_ORIGIN  
docker exec timetracking-backend env | grep CORS  
  
# Browser Console prüfen (F12)  
# CORS Fehler? → CORS_ORIGIN in backend/.env anpassen
```

Problem: "Prisma Client out of sync"

Ursache: Schema-Änderungen ohne Client-Regenerierung

Lösung:

```
cd backend  
npm run prisma:generate  
# Bei Docker:  
docker exec timetracking-backend npm run prisma:generate
```

Problem: Hohe Memory-Nutzung

Ursache: Zu viele gleichzeitige Connections oder Memory Leak

Lösung:

```
# Memory Limits setzen
docker update --memory 1g --memory-swap 2g timetracking-backend

# Node Memory Limit
NODE_OPTIONS="--max-old-space-size=512" npm start
```

Debug-Modus aktivieren

Backend:

```
# In .env
DEBUG=express:*
LOG_LEVEL=debug

# Docker Restart
docker-compose restart backend
```

Frontend:

```
# Browser Console
localStorage.setItem('debug', 'true');
```

Logs analysieren

```
# Alle Logs
docker-compose logs -f

# Nur Backend
docker-compose logs -f backend

# Letzte 100 Zeilen
docker-compose logs --tail=100 backend

# Nach Fehler suchen
docker-compose logs backend | grep ERROR
```

Performance Probleme

Database Query Optimization:

```
# Slow Query Log aktivieren
docker exec -it timetracking-db psql -U timetracking
```

```
ALTER SYSTEM SET log_min_duration_statement = 1000; -- Log Queries > 1s
SELECT pg_reload_conf();
```

Index erstellen:

```
CREATE INDEX idx_time_entries_user_date ON time_entries(user_id, clock_in);
CREATE INDEX idx_absences_user_dates ON absences(user_id, start_date, end_date);
```

Anhang

Nützliche Befehle

Docker:

```
# Alle Container stoppen
docker-compose down

# Container neu bauen
docker-compose up -d --build

# In Container shell
docker exec -it timetracking-backend sh

# Volumes löschen (ACHTUNG: Datenverlust!)
docker-compose down -v
```

Datenbank:

```
# Backup
docker exec timetracking-db pg_dump -U timetracking timetracking > backup.sql

# Restore
docker exec -i timetracking-db psql -U timetracking timetracking < backup.sql

# Prisma Studio (GUI)
docker exec -it timetracking-backend npx prisma studio
```

Git Workflow:

```
# Feature Branch
git checkout -b feature/neue-funktion

# Commit
git add .
```

```
git commit -m "feat: Neue Funktion implementiert"  
# Push  
git push origin feature/neue-funktion
```

Lizenz

MIT License - Siehe LICENSE Datei

Support

Bei Fragen oder Problemen:

- GitHub Issues: <https://github.com/mpue/cflux/issues>
- E-Mail: support@cflux.ch (fiktiv)

Changelog

v1.0.0 - Januar 2025

- Initial Release
- Alle 21 Module implementiert
- 82% Test Coverage
- Docker Support
- Schweizer Compliance

Dokumentation Ende

cflux Benutzerhandbuch

Für Mitarbeiter

Version 1.0 - Januar 2025

Inhaltsverzeichnis

1. [Erste Schritte](#)
 2. [Zeiterfassung](#)
 3. [Urlaubsverwaltung](#)
 4. [Abwesenheiten](#)
 5. [Mein Profil](#)
 6. [Berichte und Übersichten](#)
 7. [Häufige Fragen](#)
-

Erste Schritte

Registrierung

1. Öffnen Sie die cflux-Webseite in Ihrem Browser
2. Klicken Sie auf "**Registrieren**"
3. Geben Sie Ihre Daten ein:
 - E-Mail-Adresse (wird als Benutzername verwendet)
 - Vorname und Nachname
 - Passwort (mindestens 8 Zeichen)
4. Klicken Sie auf "**Registrieren**"

⚠ Hinweis: Nach der Registrierung müssen Sie warten, bis ein Administrator Ihnen Projekte zuweist und Ihre Urlaubstage festlegt.

Anmeldung

1. Öffnen Sie cflux in Ihrem Browser
2. Geben Sie Ihre E-Mail-Adresse und Ihr Passwort ein
3. Klicken Sie auf "**Anmelden**"

Passwort vergessen?

Wenden Sie sich an Ihren Administrator.

Startseite

Nach der Anmeldung sehen Sie:

- **Aktueller Status:** Ob Sie eingestempelt sind oder nicht
 - **Heutige Arbeitszeit:** Bereits geleistete Stunden
 - **Schnellaktionen:** Ein-/Ausstempeln, Urlaub beantragen
 - **Übersicht:** Letzte Zeiteinträge und anstehende Termine
-

Zeiterfassung

Einstempeln

So erfassen Sie Ihre Arbeitszeit:

1. Klicken Sie auf den großen "**Einstempeln**" Button
2. Wählen Sie das Projekt aus, an dem Sie arbeiten werden
3. Optional: Geben Sie eine Beschreibung ein (z.B. "Arbeit an Feature X")
4. Klicken Sie auf "**Bestätigen**"

Sie sind jetzt eingestempelt! Die Zeiterfassung läuft automatisch.

Was Sie sehen:

- ⏱ Laufende Uhr mit aktueller Arbeitszeit
- 📁 Aktuelles Projekt
- 🔴 Roter "Ausstempeln" Button

Ausstempeln

Wenn Sie Ihre Arbeit beenden:

1. Klicken Sie auf "**Ausstempeln**"
2. Optional: Aktualisieren Sie die Beschreibung (was haben Sie heute geschafft?)
3. Klicken Sie auf "**Bestätigen**"

Ihre Arbeitszeit wurde erfasst!

Das System berechnet automatisch:

- Arbeitsdauer in Stunden und Minuten
- Zuordnung zum gewählten Projekt
- Speicherung für Ihre Übersicht und Reports

Mehrere Projekte am Tag

Sie können mehrfach ein- und ausstempeln:

Beispiel:

```
08:00 - 12:00 → Projekt "Website Redesign" (4 Stunden)  
13:00 - 17:00 → Projekt "Mobile App" (4 Stunden)
```

So geht's:

1. Stempeln Sie für Projekt A ein
2. Stempeln Sie aus
3. Stempeln Sie für Projekt B ein
4. Stempeln Sie aus

Wichtige Hinweise

Regeln für Zeiterfassung:

- Sie können nur an Projekten arbeiten, die Ihnen zugewiesen wurden
- Sie können nicht eingestempelt bleiben, wenn Sie den nächsten Tag beginnen
- Maximale tägliche Arbeitszeit: 10 Stunden
- Bei mehr als 6 Stunden Arbeit ist eine Pause erforderlich

Tipps:

- Stempeln Sie immer aus, wenn Sie eine längere Pause machen
- Geben Sie aussagekräftige Beschreibungen ein - das hilft bei Reports
- Vergessen Sie nicht auszustempeln am Feierabend!

Zeiteinträge ansehen

So sehen Sie Ihre erfassten Zeiten:

1. Navigieren Sie zu "**Meine Zeiten**" im Menü

2. Sie sehen eine Liste aller Ihrer Zeiteinträge

Filter verwenden:

- Nach Datum filtern (heute, diese Woche, dieser Monat, benutzerdefiniert)
- Nach Projekt filtern
- Nach Beschreibung suchen

Für jeden Eintrag sehen Sie:

- Datum und Uhrzeit (Start - Ende)
- Projekt
- Dauer in Stunden:Minuten
- Ihre Beschreibung

Fehler korrigieren

Sie haben vergessen auszustempeln?

Keine Sorge! Wenden Sie sich an Ihren Administrator. Nur Administratoren können Zeiteinträge nachträglich korrigieren.

Was der Admin korrigieren kann:

- Vergessenes Ausstempeln nachtragen
- Falsche Projektbuchung ändern
- Arbeitszeiten anpassen

Urlaubsverwaltung

Urlaubsanträge stellen

So beantragen Sie Urlaub:

1. Klicken Sie auf "**Urlaub beantragen**"
2. Wählen Sie den Zeitraum:
 - Von-Datum
 - Bis-Datum
3. Geben Sie einen Grund an (optional, z.B. "Familienurlaub")
4. Bei Bedarf: Wählen Sie "**Halber Tag**" (nur für einzelne Tage)
5. Klicken Sie auf "**Antrag stellen**"

Ihr Antrag wurde eingereicht!

Was passiert jetzt?

1. Der Antrag hat den Status "**Ausstehend**"
2. Ein Administrator prüft Ihren Antrag
3. Sie erhalten Benachrichtigung bei Genehmigung oder Ablehnung

Urlaubsübersicht

Ihr Urlaubskonto:

Verfügbare Urlaubstage:	25 Tage/Jahr
Bereits genommen:	5 Tage
Beantragt (ausstehend):	3 Tage
Verbleibend:	17 Tage

Status-Übersicht:

- ☒ **Ausstehend:** Warten auf Genehmigung
- ☑ **Genehmigt:** Urlaub wurde genehmigt
- ✗ **Abgelehnt:** Urlaub wurde abgelehnt (mit Begründung)

Urlaubsanträge verwalten

Antrag ansehen:

1. Navigieren Sie zu "**Meine Urlaube**"
2. Sehen Sie alle Ihre Anträge mit Status

Antrag stornieren:

- Nur möglich bei Status "Ausstehend"
- Klicken Sie auf den Antrag
- Klicken Sie "**Stornieren**"
- Bestätigen Sie die Stornierung

⚠ **Genehmigte Urlaube** können Sie nicht selbst stornieren. Kontaktieren Sie Ihren Administrator.

Urlaubsplanung - Tipps

💡 Best Practices:

Frühzeitig planen:

- Stellen Sie Anträge mindestens 2 Wochen im Voraus
- Berücksichtigen Sie Projekttermine
- Koordinieren Sie sich mit Ihrem Team

Jahresplanung:

- Nutzen Sie Ihren Jahresurlaub rechtzeitig
- Resturlaub verfällt am Ende des Jahres (je nach Unternehmensrichtlinie)
- Planen Sie um Feiertage herum für maximale Freizeit

Dokumentation:

- Geben Sie immer einen Grund an (hilft bei der Planung)
- Bei längeren Urlauben: Informieren Sie Ihr Team zusätzlich
- Aktualisieren Sie Ihre Abwesenheitsnotiz

Abwesenheiten

Arten von Abwesenheiten

cflux unterstützt verschiedene Abwesenheitstypen:

Typ	Wann verwenden	Beispiel
Urlaub	Regulärer Erholungsurlaub	Sommerurlaub, Weihnachten
Krankheit	Bei Krankheit/Unfall	Grippe, Arzttermin
Home Office	Arbeit von zu Hause	Geplante Home-Office-Tage
Weiterbildung	Schulungen, Kurse	Konferenz, Workshop
Persönlich	Persönliche Gründe	Umzug, Behördentermin
Sonstige	Alle anderen Gründe	-

Krankmeldung

Wenn Sie krank sind:

1. So schnell wie möglich melden:

- Klicken Sie auf "**Abwesenheit melden**"
- Wählen Sie Typ: "**Krankheit**"
- Geben Sie das Von-Datum an (heute)
- Bis-Datum: Schätzung oder leer lassen
- Klicken Sie "**Melden**"

2. Bei Genesung:

- Aktualisieren Sie den Antrag mit dem tatsächlichen Enddatum
- Oder lassen Sie ihn vom Admin aktualisieren

Ärztliche Bescheinigung:

Ab dem 3. Krankheitstag ist in vielen Unternehmen ein Arztzeugnis erforderlich. Klären Sie dies mit Ihrem Administrator.

Home Office

So melden Sie Home Office:

1. Klicken Sie auf "**Abwesenheit melden**"
2. Wählen Sie Typ: "**Home Office**"
3. Geben Sie das Datum an
4. Optional: Grund (z.B. "Konzentrierte Arbeit", "Kinderbetreuung")
5. Klicken Sie "**Melden**"

Tipp: Einige Unternehmen erfordern Vorab-Genehmigung für Home Office. Stellen Sie den Antrag rechtzeitig!

Weiterbildung

Schulungen und Kurse erfassen:

1. Klicken Sie auf "**Abwesenheit melden**"
 2. Wählen Sie Typ: "**Weiterbildung**"
 3. Geben Sie Details an:
 - Zeitraum
 - Name des Kurses/der Konferenz
 - Ort (optional)
 4. Klicken Sie "**Melden**"
-

Mein Profil

Profil bearbeiten

So aktualisieren Sie Ihr Profil:

1. Klicken Sie auf Ihr **Profilbild** oder **Namen** (oben rechts)
2. Wählen Sie "**Profil bearbeiten**"
3. Ändern Sie:
 - Vorname
 - Nachname
 - E-Mail-Adresse
4. Klicken Sie "**Speichern**"

Passwort ändern

So ändern Sie Ihr Passwort:

1. Navigieren Sie zu "**Profil**" → "**Passwort ändern**"
2. Geben Sie ein:
 - Aktuelles Passwort
 - Neues Passwort (mindestens 8 Zeichen)
 - Neues Passwort bestätigen
3. Klicken Sie "**Passwort ändern**"

Ihr Passwort wurde geändert! Sie werden automatisch abgemeldet und müssen sich mit dem neuen Passwort anmelden.

💡 Passwort-Tipps:

- Verwenden Sie ein einzigartiges Passwort
 - Mindestens 8 Zeichen
 - Mischung aus Groß-/Kleinbuchstaben, Zahlen und Sonderzeichen
 - Ändern Sie Ihr Passwort regelmäßig
-

Berichte und Übersichten

Persönliche Statistiken

Übersicht aufrufen:

1. Navigieren Sie zu "**Meine Reports**" oder "**Dashboard**"

2. Wählen Sie den Zeitraum:

- Heute
- Diese Woche
- Dieser Monat
- Benutzerdefiniert

Was Sie sehen:

Zusammenfassung:

Gesamte Arbeitsstunden: 160h

Arbeitstage: 20

Durchschnitt/Tag: 8h

Urlaubstage: 2

Aufschlüsselung nach Projekt:

Website Redesign: 80h (50%)

Mobile App: 80h (50%)

Wochenübersicht:

KW 1: 40h

KW 2: 42h

KW 3: 38h

KW 4: 40h

Reports exportieren

PDF Export:

1. Öffnen Sie Ihren Report

2. Klicken Sie auf "**Als PDF exportieren**"

3. PDF wird automatisch heruntergeladen

Excel Export:

1. Öffnen Sie Ihren Report

2. Klicken Sie auf "**Als Excel exportieren**"

3. Excel-Datei wird heruntergeladen

💡 **Verwendung:**

- Für persönliche Unterlagen
- Für Jahresgespräche
- Zur Nachverfolgung Ihrer Produktivität

Monatsabschluss

Am Ende jedes Monats:

1. Überprüfen Sie Ihre Zeiteinträge
 2. Stellen Sie sicher, dass alle Tage erfasst sind
 3. Exportieren Sie Ihren Monatsbericht
 4. Optional: Unterschreiben und an HR senden (je nach Unternehmensrichtlinie)
-

Häufige Fragen

Allgemein

F: Kann ich meine Zeiteinträge nachträglich bearbeiten?

A: Nein, nur Administratoren können Zeiteinträge korrigieren. Dies dient der Transparenz und Nachvollziehbarkeit. Wenn Sie einen Fehler gemacht haben, wenden Sie sich an Ihren Administrator.

F: Was passiert, wenn ich vergesse auszustempeln?

A: Kontaktieren Sie Ihren Administrator. Er kann den Zeiteintrag manuell korrigieren und eine Ausstempelzeit nachtragen.

F: Kann ich mehrere Projekte gleichzeitig buchen?

A: Nein, Sie können nur auf ein Projekt gleichzeitig eingestempelt sein. Stempeln Sie aus und wieder ein, um das Projekt zu wechseln.

F: Wie werden Pausen erfasst?

A: Stempeln Sie aus, wenn Sie eine längere Pause machen. Kurze Pausen (Kaffee, WC) müssen nicht erfasst werden - diese sind in der Arbeitszeit enthalten.

Urlaub

F: Wie viele Urlaubstage habe ich?

A: Sie sehen Ihr Urlaubskontingent in Ihrem Profil und auf der Dashboard-Seite. Standardmäßig sind dies 25 Tage/Jahr (kann je nach Vertrag variieren).

F: Mein Urlaubsantrag wurde abgelehnt - was nun?

A: Sprechen Sie mit Ihrem Vorgesetzten über die Gründe. Sie können einen alternativen Zeitraum beantragen.

F: Kann ich Urlaubstage ins nächste Jahr mitnehmen?

A: Dies hängt von Ihrer Unternehmensrichtlinie ab. Fragen Sie Ihren Administrator oder HR.

F: Was ist "Halber Urlaubstag"?

A: Sie können einzelne Tage als halben Urlaubstag buchen (4 Stunden). Zum Beispiel: Vormittag frei, Nachmittag arbeiten.

Krankheit

F: Muss ich bei Krankheit trotzdem melden?

A: Ja, melden Sie Ihre Krankheit so schnell wie möglich über cflux. Informieren Sie zusätzlich Ihren Vorgesetzten direkt (Telefon, E-Mail).

F: Brauche ich ein Arztzeugnis?

A: In der Regel ab dem 3. Krankheitstag. Klären Sie dies mit Ihrem Unternehmen.

F: Werden Krankheitstage von meinem Urlaubskonto abgezogen?

A: Nein, Krankheitstage sind getrennt von Urlaubstagen und beeinflussen Ihr Urlaubskonto nicht.

Technische Probleme

F: Ich kann mich nicht anmelden - was tun?

A:

1. Überprüfen Sie Ihre E-Mail-Adresse und Passwort
2. Versuchen Sie einen Passwort-Reset (kontaktieren Sie Admin)
3. Leeren Sie Ihren Browser-Cache
4. Versuchen Sie einen anderen Browser
5. Kontaktieren Sie Ihren IT-Administrator

F: Die Seite lädt nicht richtig

A:

1. Drücken Sie F5 zum Neuladen
2. Leeren Sie den Browser-Cache (Strg+Shift+Entf)
3. Aktualisieren Sie Ihren Browser auf die neueste Version
4. Kontaktieren Sie den IT-Support

F: Ich sehe keine Projekte zum Einstempeln

A: Wenden Sie sich an Ihren Administrator. Er muss Ihnen erst Projekte zuweisen.

Best Practices

F: Was ist die beste Art, Beschreibungen zu formulieren?

A: Seien Sie spezifisch aber knapp:

- Gut: "Bugfix Login-Formular", "Meeting Projekt X", "Code Review Feature Y"
- Schlecht: "Arbeit", "Diverses", "Meeting"

F: Sollte ich jeden Tag einen Report erstellen?

A: Nicht nötig. Nutzen Sie Reports für:

- Wochenrückblicke
- Monatsabschlüsse
- Jahresgespräche
- Projektdokumentation

F: Wie detailliert sollten meine Zeitbuchungen sein?

A: Ein guter Mittelweg:

- Erfassen Sie alle Arbeitsstunden
 - Wechseln Sie das Projekt, wenn Sie an etwas anderem arbeiten
 - Kurze, aussagekräftige Beschreibungen
 - Ehrliche Zeiterfassung
-

Kontakt & Support

Bei Fragen oder Problemen:**Technischer Support:**

- E-Mail: support@cflux.ch (fiktiv)
- Telefon: +41 XX XXX XX XX
- Öffnungszeiten: Mo-Fr, 08:00-17:00 Uhr

Administratoren:

- Für Zeitkorrekturen
- Für Projektzuweisungen
- Für Urlaubsgenehmigungen
- Für Benutzerrechte

HR-Abteilung:

- Für Urlaubsfragen
 - Für Arbeitszeit-Richtlinien
 - Für Vertragsangelegenheiten
-

Glossar

Begriff	Bedeutung
Einstempeln	Start der Arbeitszeiterfassung
Ausstempeln	Ende der Arbeitszeiterfassung
Zeiteintrag	Eine erfasste Arbeitsperiode (von-bis)
Projekt	Arbeitsprojekt, dem Zeiten zugeordnet werden
Urlaubskonto	Ihre verfügbaren Urlaubstage
Abwesenheit	Jede Nicht-Arbeitszeit (Urlaub, Krankheit, etc.)
Status	Zustand eines Antrags (Ausstehend, Genehmigt, Abgelehnt)
Report	Zusammenfassende Auswertung Ihrer Zeiten

Shortcuts & Tipps

Tastatur-Shortcuts:

- **Alt+E** - Einstempeln
- **Alt+A** - Ausstempeln
- **Alt+U** - Urlaub beantragen
- **Alt+R** - Reports öffnen

Mobile Nutzung:

- cflux ist mobilfreundlich gestaltet
- Zeiterfassung funktioniert auf Smartphone/Tablet
- Responsive Design passt sich an Ihr Gerät an

Browser-Empfehlungen:

- Chrome, Firefox, Safari, Edge (neueste Versionen)
- JavaScript muss aktiviert sein
- Cookies müssen erlaubt sein

Ende des Benutzerhandbuchs

Wir wünschen Ihnen produktives Arbeiten mit cflux!

Bei Fragen steht Ihnen unser Support-Team gerne zur Verfügung.

cflux Administrator-Handbuch

Für Administratoren und System-Manager

Version 1.0 - Januar 2025

Inhaltsverzeichnis

1. [Administrator-Rolle](#)
 2. [Benutzerverwaltung](#)
 3. [Projektverwaltung](#)
 4. [Zeitmanagement](#)
 5. [Urlaubsverwaltung](#)
 6. [Rechnungswesen](#)
 7. [Reporting & Analytics](#)
 8. [System-Administration](#)
 9. [Best Practices](#)
-

Administrator-Rolle

Berechtigungen

Als Administrator haben Sie **vollständigen Zugriff** auf alle Funktionen:

Benutzer:

- Benutzer erstellen, bearbeiten, löschen
- Rollen zuweisen (USER ↔ ADMIN)
- Urlaubskontingente festlegen
- Benutzer aktivieren/deaktivieren

Projekte:

- Projekte erstellen, bearbeiten, löschen
- Benutzer zu Projekten zuweisen
- Projekt-Status verwalten

Zeiterfassung:

- Zeiteinträge aller Benutzer einsehen
- Zeiteinträge korrigieren
- Zeiteinträge löschen

Urlaub & Abwesenheiten:

- Alle Anträge einsehen
- Anträge genehmigen/ablehnen
- Anträge stornieren

Reports:

- Team-weite Reports
- Projekt-Auswertungen
- Export-Funktionen

Rechnungen:

- Rechnungen erstellen
- Rechnungen versenden
- Zahlungen tracken

Verantwortlichkeiten

Als Administrator sind Sie verantwortlich für:

1. Tägliche Aufgaben:

- Urlaubsanträge prüfen und bearbeiten
- Zeitkorrekturen durchführen
- Benutzer-Support leisten

2. Wöchentliche Aufgaben:

- Arbeitszeitübersichten prüfen
- Projekt-Auslastung kontrollieren

- System-Health überprüfen

3. Monatliche Aufgaben:

- Monatsberichte erstellen
- Abwesenheitsstatistiken prüfen
- Rechnungen erstellen
- Datensicherung kontrollieren

4. Jährliche Aufgaben:

- Urlaubskontingente aktualisieren
- Jahresberichte erstellen
- System-Review durchführen

Benutzerverwaltung

Benutzer erstellen

Manuelles Anlegen:

1. Navigieren Sie zu "**Verwaltung**" → "**Benutzer**"
2. Klicken Sie auf "**Neuer Benutzer**"
3. Füllen Sie das Formular aus:

E-Mail:	max.mustermann@firma.ch
Vorname:	Max
Nachname:	Mustermann
Rolle:	USER / ADMIN
Urlaubstage:	25 (Standard)
Aktiv:	✓ Ja

4. Klicken Sie "**Erstellen**"

Initiales Passwort:

- System generiert temporäres Passwort
- Benutzer erhält E-Mail mit Login-Daten
- Benutzer muss Passwort bei erster Anmeldung ändern

Bulk-Import (CSV):

Für viele Benutzer auf einmal:

1. Navigieren Sie zu "**Verwaltung**" → "**Benutzer**" → "**Import**"
2. Laden Sie die CSV-Vorlage herunter
3. Füllen Sie die CSV-Datei aus:

```
email,firstName,lastName,role,vacationDays  
max@firma.ch,Max,Mustermann,USER,25  
anna@firma.ch,Anna,Schmidt,USER,25  
admin@firma.ch,Admin,User,ADMIN,30
```

4. Laden Sie die Datei hoch
5. Prüfen Sie die Vorschau
6. Klicken Sie "**Importieren**"

Benutzer bearbeiten

Profildaten ändern:

1. Öffnen Sie "**Verwaltung**" → "**Benutzer**"
2. Klicken Sie auf den Benutzer
3. Klicken Sie "**Bearbeiten**"
4. Ändern Sie die Daten:
 - Name
 - E-Mail (wird als neuer Login verwendet)
 - Urlaubstage
 - Rolle
5. Klicken Sie "**Speichern**"

Rolle ändern:

USER zu ADMIN machen:

```
Benutzer öffnen → Rolle: ADMIN auswählen → Speichern
```

ADMIN zu USER zurückstufen:

```
Benutzer öffnen → Rolle: USER auswählen → Speichern
```

⚠ **Wichtig:** Es sollte immer mindestens ein ADMIN-Benutzer existieren!

Passwort zurücksetzen

Wenn ein Benutzer sein Passwort vergessen hat:

1. Öffnen Sie den Benutzer
2. Klicken Sie "**Passwort zurücksetzen**"
3. Wählen Sie:
 - **Temporäres Passwort generieren** (wird per E-Mail gesendet)
 - **Eigenes Passwort setzen** (geben Sie ein neues Passwort ein)
4. Bestätigen Sie

Der Benutzer kann sich nun mit dem neuen Passwort anmelden.

Benutzer deaktivieren

Soft Delete (empfohlen):

Wenn ein Mitarbeiter das Unternehmen verlässt:

1. Öffnen Sie den Benutzer
2. Setzen Sie "**Aktiv**" auf **Nein**
3. Speichern Sie

Effekt:

- Benutzer kann sich nicht mehr anmelden
- Alle historischen Daten bleiben erhalten
- Zeiteinträge und Urlaube sind weiterhin in Reports sichtbar
- Benutzer kann bei Bedarf reaktiviert werden

Hard Delete (mit Vorsicht):

⚠ Nur in Ausnahmefällen!

1. Öffnen Sie den Benutzer
2. Klicken Sie "**Löschen**"
3. Bestätigen Sie die Sicherheitsabfrage

Effekt:

- Benutzer wird aus System entfernt
- Alle Zeiteinträge gehen verloren
- Unwiderruflich!

Empfehlung: Verwenden Sie immer Soft Delete (Deaktivieren).

Urlaubskontingent verwalten

Initiales Kontingent setzen:

Bei neuem Benutzer oder Jahreswechsel:

1. Öffnen Sie den Benutzer
2. Feld "**Urlaubstage**": Setzen Sie die Anzahl (z.B. 25)
3. Speichern Sie

Anpassungen während des Jahres:

Urlaubstage hinzufügen:

Beispiel: Sonderurlaub für Hochzeit
→ Urlaubstage von 25 auf 28 erhöhen

Urlaubstage reduzieren:

Beispiel: Teilzeit-Wechsel
→ Urlaubstage von 25 auf 20 reduzieren

Übersicht für alle Benutzer:

Report: "Urlaubsübersicht" zeigt:

- Kontingent pro Benutzer
- Genommener Urlaub
- Geplanter Urlaub (genehmigt aber noch nicht genommen)
- Verbleibende Tage

Projektverwaltung

Projekt erstellen

1. Navigieren Sie zu "**Verwaltung**" → "**Projekte**"

2. Klicken Sie "**Neues Projekt**"

3. Füllen Sie das Formular aus:

Name:	Website Redesign 2025
Beschreibung:	Kompletter Relaunch der Firmenwebsite
Aktiv:	✓ Ja
Kunde:	Acme Corp (optional)
Budget:	100 Stunden (optional)

4. Klicken Sie "**Erstellen**"

Benutzer zu Projekten zuweisen

Einzelzuweisung:

1. Öffnen Sie das Projekt
2. Klicken Sie "**Benutzer zuweisen**"
3. Wählen Sie Benutzer aus der Liste
4. Klicken Sie "**Zuweisen**"

Mehrfachzuweisung:

1. Öffnen Sie das Projekt
2. Aktivieren Sie die Checkboxen bei allen gewünschten Benutzern
3. Klicken Sie "**Ausgewählte zuweisen**"

Benutzer entfernen:

1. Öffnen Sie das Projekt
2. Klicken Sie auf das "X" neben dem Benutzer
3. Bestätigen Sie die Entfernung

 **Tipp:** Nur zugewiesene Benutzer können auf das Projekt Zeiten buchen!

Projekt-Lifecycle

Aktives Projekt:

Status: Aktiv ✓
→ Benutzer können Zeiten buchen
→ Erscheint in Projekt-Listen

Projekt pausieren:

Status: Inaktiv X
→ Keine neuen Zeitbuchungen möglich
→ Historische Daten bleiben erhalten
→ Kann jederzeit reaktiviert werden

Projekt abschließen:

1. Setzen auf Inaktiv
2. Final-Report erstellen
3. Optional: Rechnung erstellen
4. Projekt archivieren

Projekt-Übersicht

Dashboard zeigt für jedes Projekt:

Project: Website Redesign
Status: Aktiv

Team: 4 Mitarbeiter
Gesamtstunden: 120h von 150h Budget
Auslastung: 80%
Letzte Buchung: 15.01.2025 17:30
Top Contributor: Max Mustermann (45h)

Alerts:

-  Warnung bei 80% Budget-Auslastung

-  Kritisch bei 100% Budget überschritten
 -  Keine Aktivität seit >7 Tagen
-

Zeitmanagement

Zeiteinträge prüfen

Team-Übersicht:

1. Navigieren Sie zu "**Verwaltung**" → "**Zeiteinträge**"

2. Filter setzen:

- Zeitraum (heute, diese Woche, ...)
- Benutzer
- Projekt
- Status (eingestempelt/ausgestempelt)

Was Sie sehen:

15.01.2025			
<hr/>			
Max Mustermann	08:00 - 12:00	Website Redesign	4h
	13:00 - 17:00	Mobile App	4h
Gesamt: 8h ✓			
Anna Schmidt	08:30 - ???	CRM System	Eingestempelt 
Status: 1 Benutzer noch eingestempelt			

Zeiteinträge korrigieren

Wann korrigieren?

- Vergessenes Ausstempeln
- Falsche Projektbuchung
- Zeitfehler (zu früh/zu spät)

So korrigieren Sie:

1. Suchen Sie den Zeiteintrag

2. Klicken Sie auf "**Bearbeiten**"

3. Ändern Sie:

- Einstempel-Zeit
- Ausstempel-Zeit
- Projekt
- Beschreibung

4. **Wichtig:** Geben Sie einen Korrektur-Grund an!

5. Klicken Sie "**Speichern**"

Best Practice:

Korrektur-Grund Beispiele:

- "Vergessenes Ausstempeln nachtragen"
- "Falsche Projektbuchung korrigiert"
- "Zeitanpassung nach Rücksprache mit MA"

X Nicht einfach leer lassen!

Audit Trail: Alle Korrekturen werden geloggt:

- Wer hat korrigiert
- Wann wurde korrigiert
- Was wurde geändert
- Grund der Korrektur

Zeiteintrag löschen

⚠ Nur in Ausnahmefällen!

Gründe zum Löschen:

- Doppelte Buchung
- Test-Eintrag
- Irrtümliche Buchung

So löschen Sie:

1. Zeiteintrag öffnen
2. Klicken Sie "**Löschen**"
3. Geben Sie Lösch-Grund an
4. Bestätigen Sie

💡 Besser: Korrigieren statt Löschen (für Audit Trail)

Offene Zeiteinträge schließen

Problem: Benutzer hat vergessen auszustempeln

Lösung 1 - Manuell:

1. Finden Sie den offenen Eintrag
2. Bearbeiten Sie ihn
3. Setzen Sie Ausstempel-Zeit
4. Speichern Sie

Lösung 2 - Automatisch: Nächtlicher Cronjob schließt offene Einträge:

Konfiguration in .env:
AUTO_CLOCK_OUT_TIME=18:00
AUTO_CLOCK_OUT_ENABLED=true

→ Alle offenen Einträge werden um 18:00 Uhr geschlossen

Urlaubsverwaltung

Anträge bearbeiten

Übersicht der Anträge:

Navigieren Sie zu "**Verwaltung**" → "**Urlaubsanträge**"

Ausstehende Anträge (3)

Max Mustermann
01.02. - 07.02.2025 (5 Tage)
Typ: Urlaub
Grund: Familienurlaub
[Genehmigen] [Ablehnen]

Anna Schmidt
15.03. - 15.03.2025 (0.5 Tage)
Typ: Urlaub
Grund: Arzttermin
[Genehmigen] [Ablehnen]

Antrag genehmigen

Standard-Genehmigung:

1. Klicken Sie "**Genehmigen**"
2. Optional: Kommentar hinzufügen
3. Bestätigen Sie

Effekt:

- Status → APPROVED
- Urlaubstage werden vom Kontingent abgezogen
- Benutzer erhält Benachrichtigung
- Erscheint in Kalenderübersicht

Prüfungen vor Genehmigung:

1. Verfügbare Urlaubstage:

Kontingent: 25 Tage
Genommen: 10 Tage
Beantragt: 5 Tage
Verbleibend: 10 Tage ✓

2. Team-Auslastung:

- Sind genug Mitarbeiter anwesend?
- Gibt es bereits viele Urlaube im Zeitraum?
- Ist das Projekt ausreichend besetzt?

3. Projekttermine:

- Gibt es kritische Deadlines?
- Ist der Mitarbeiter zwingend erforderlich?

4. Vorlaufzeit:

- Wurde rechtzeitig beantragt?
- Mindestens 2 Wochen im Voraus (Richtlinie)

Antrag ablehnen

Ablehnung mit Begründung:

1. Klicken Sie "**Ablehnen**"
2. **Wichtig:** Geben Sie einen Grund an!

Beispiele:

"Projektdeadline am 05.02., bitte alternativen Zeitraum wählen"
"Team-Meeting am 02.02. - Ihre Anwesenheit erforderlich"
"Bereits 2 Kollegen im Urlaub, minimal Besetzung"

3. Bestätigen Sie

Effekt:

- ✗ Status → REJECTED
- ✗ Keine Urlaubstage abgezogen
- ✗ Benutzer erhält Benachrichtigung mit Begründung

Best Practice:

- Immer konstruktive Begründung geben
- Alternative Zeiträume vorschlagen
- Bei Ablehnung: Persönliches Gespräch empfohlen

Genehmigten Urlaub stornieren

Wann nötig?

- Dringende Projektanforderung
- Mitarbeiter selbst storniert
- Fehlerhafte Genehmigung

So stornieren Sie:

1. Öffnen Sie den genehmigten Antrag
2. Klicken Sie "**Stornieren**"
3. Geben Sie Grund an
4. Bestätigen Sie

Effekt:

- Urlaubstage werden zurückgebucht
- Status → CANCELLED
- Benutzer wird benachrichtigt

⚠ Achtung: Nur in Ausnahmefällen! Stornierung bereits genehmigter Urlaube ist problematisch.

Urlaubskalender

Team-Kalenderübersicht:

Februar 2025						
Mo	Di	Mi	Do	Fr	Sa	So
				1	2	
3	4	5	6	7	8	9
✉	✉	✉	✉	✉		Max M.
10	11	12	13	14	15	16
✉	✉	✉				Anna S.
17	18	19	20	21	22	23
✉	✉					Tom K.
24	25	26	27	28		

Legende:

- ✉ Urlaub
- ⌚ Krankheit
- 🏡 Home Office

Export-Funktionen:

- iCal Export (für Outlook/Google Calendar)
- PDF Export (für Aushang)
- Excel Export (für Planung)

Rechnungswesen

Rechnung erstellen

Schritt 1 - Grunddaten:

1. Navigieren Sie zu "**Verwaltung**" → "**Rechnungen**"
2. Klicken Sie "**Neue Rechnung**"
3. Wählen Sie:

Kunde: Acme Corp
Projekt: Website Redesign
Rechnungsdatum: 31.01.2025
Fälligkeitsdatum: 28.02.2025 (30 Tage)

Schritt 2 - Positionen:

Manuelle Positionen:

Pos	Beschreibung	Menge	Einheit	Preis	MwSt	Total
1	Backend Entwicklung	40	Stunden	120 CHF	8.1%	5'188.80
2	Frontend Entwicklung	30	Stunden	120 CHF	8.1%	3'891.60
3	Projektmanagement	10	Stunden	150 CHF	8.1%	1'621.50
				Subtotal:	9'600.00	
				MwSt 8.1%:	777.60	
				Total:	10'377.60 CHF	

Aus Zeitbuchungen:

1. Klicken Sie "**Aus Zeiteinträgen generieren**"
2. Wählen Sie Zeitraum
3. Wählen Sie Benutzer/Projekte
4. System fasst automatisch zusammen:

Max Mustermann - Website Redesign: 40h × 120 CHF = 4'800 CHF
Anna Schmidt - Website Redesign: 30h × 120 CHF = 3'600 CHF

5. Prüfen und anpassen Sie

Schritt 3 - Details:

Zahlungsbedingungen:

Zahlbar innerhalb 30 Tagen
 2% Skonto bei Zahlung binnen 10 Tagen
 Anzahlung 50%

Notizen:
Vielen Dank für Ihren Auftrag!

Fußzeile:
Bankverbindung: IBAN CH...

Schritt 4 - QR-Rechnung (Schweiz):

Automatische Generierung:

- QR-Code mit allen Zahlungsinformationen
- IBAN
- Betrag
- Referenznummer
- Rechnungsadresse

Rechnung versenden

Als PDF:

1. Öffnen Sie die Rechnung
2. Klicken Sie "**PDF generieren**"
3. Vorschau prüfen
4. "**Herunterladen**" oder "**Per E-Mail senden**"

E-Mail-Versand:

An: buchhaltung@kunde.ch
Betreff: Rechnung RE-2025-001 - Website Redesign
Text: [Standard-Template oder benutzerdefiniert]
Anhang: RE-2025-001.pdf

Status nach Versand:

- Status: DRAFT → SENT
- Versanddatum wird gespeichert
- Tracking: "Wann wurde versendet"

Zahlungsstatus verwalten

Zahlung erfassen:

Wenn Kunde bezahlt hat:

1. Öffnen Sie die Rechnung
2. Klicken Sie "**Zahlung erfassen**"
3. Geben Sie ein:

Zahlungsdatum:	15.02.2025
Betrag:	10'377.60 CHF
Zahlungsart:	Banküberweisung
Referenz:	Buchungsbeleg XYZ

4. Speichern Sie

Status:

- Status: SENT → PAID
- Überfällig-Warnung wird entfernt

Teilzahlungen:

Bei Anzahlungen oder Ratenzahlungen:

Rechnungsbetrag:	10'377.60 CHF
<hr/>	
Zahlung 1:	5'000.00 CHF (15.02.)
Zahlung 2:	5'377.60 CHF (15.03.)
<hr/>	
Bezahlt:	10'377.60 CHF ✓
Offen:	0.00 CHF

Mahnwesen

Überfällige Rechnungen:

Automatische Erkennung:

- Status → OVERDUE wenn Fälligkeitsdatum überschritten
- Dashboard zeigt überfällige Rechnungen

Mahnungen erstellen:

1. Mahnung (nach 7 Tagen):

Betreff:	Zahlungserinnerung RE-2025-001
Ton:	Freundlich
Inhalt:	"Möglicherweise haben Sie unsere Rechnung übersehen..."

2. Mahnung (nach 14 Tagen):

Betreff:	1. Mahnung RE-2025-001
Ton:	Bestimmt aber höflich
Mahngebühr:	50 CHF

3. Mahnung (nach 30 Tagen):

Betreff: 2. Mahnung RE-2025-001
Ton: Ernst
Mahngebühr: 100 CHF
Androhung: Inkasso

Finanz-Übersicht

Dashboard "Rechnungen":

Monat Januar 2025

Rechnungen erstellt: 15
Gesamtvolumen: 250'000 CHF

Status:

Draft: 3 (20'000 CHF)
Versendet: 7 (150'000 CHF)
Bezahlt: 4 (70'000 CHF)
Überfällig: 1 (10'000 CHF) 

Ausstehend: 160'000 CHF
Durchschn. Zahlungsziel: 22 Tage

Reporting & Analytics

Standard-Reports

1. Team-Übersicht:

Zeitraum: Januar 2025

Mitarbeiter	Stunden	Tage	Projekte	Urlaub
Max M.	160h	20	3	0
Anna S.	152h	19	2	1
Tom K.	144h	18	4	2
Gesamt:	1280h	160	-	10
Durchschnitt:	160h	20	3	1.25

2. Projekt-Auslastung:

Projekt	Team	Stunden	Budget	Auslastung
Website Redesign	4	320h	400h	80% ✓
Mobile App	3	240h	200h	120% 
CRM System	2	80h	150h	53% 

3. Urlaubsübersicht:

Mitarbeiter	Kontingent	Genommen	Geplant	Verfügbar
Max M.	25	5	3	17
Anna S.	25	8	0	17
Tom K.	25	3	5	17

Custom Reports erstellen

Report-Builder:

1. Navigieren Sie zu "**Reports**" → "**Neuer Report**"

2. Wählen Sie:

Typ: Zeiterfassung / Urlaub / Projekt
Zeitraum: 01.01.2025 - 31.01.2025
Filter:
 Benutzer: [Alle / Auswahl]
 Projekte: [Alle / Auswahl]
 Status: [Alle / Aktiv / Inaktiv]

Gruppierung: Nach Benutzer / Projekt / Woche / Monat
Sortierung: Name / Stunden / Datum

3. "**Vorschau**" → Prüfen

4. "**Erstellen**" → Report wird generiert

Speichern für Wiederverwendung:

- Reports können als Vorlage gespeichert werden
- Monatliche Reports automatisieren

Export-Funktionen

PDF Export:

- Formatierte Berichte
- Firmen-Logo/Header
- Professionelles Layout
- Geeignet für Präsentationen

Excel Export:

- Alle Rohdaten
- Pivot-Tabellen möglich
- Weitere Analyse in Excel
- Archivierung

CSV Export:

- Einfaches Format
- Import in andere Systeme
- Datenverarbeitung
- Backup

Dashboards

Admin-Dashboard:

Aktuelle Auslastung

 20/25 Mitarbeiter @ Arbeit

Heute eingestempelt: 20
Home Office: 3
Urlaub/Krank: 2

Ausstehende Aktionen

- 🔔 Urlaubsanträge: 5 warten auf Genehmigung
- ⌚ Offene Zeiteinträge: 2 seit gestern offen
- ฿ Überfällige Rechnung: 1 (10'000 CHF)

Monatsstatistik (Januar 2025)

Gesamtstunden:	3'200h
Urlaubstage:	45
Krankheitstage:	12
Rechnungsvolumen:	250'000 CHF

System-Administration

Backup & Restore

Automatisches Backup:

Konfiguration in `.env`:

```
BACKUP_ENABLED=true
BACKUP_SCHEDULE="0 2 * * *" # Täglich um 2 Uhr
BACKUP_RETENTION_DAYS=30
BACKUP_LOCATION=/opt/cflux/backups
```

Manuelles Backup:

```
# Datenbank-Backup
docker exec timetracking-db pg_dump -U timetracking timetracking > \
backup_$(date +%Y%m%d).sql

# Komprimieren
gzip backup_$(date +%Y%m%d).sql

# Auf externes Storage kopieren
scp backup_*.sql.gz backup-server:/backups/cflux/
```

Restore:

```
# Backup entpacken
gunzip backup_20250115.sql.gz

# Datenbank wiederherstellen
docker exec -i timetracking-db psql -U timetracking timetracking <
backup_20250115.sql
```

Benutzer-Aktivität überwachen

Audit Log:

Timestamp	User	Action	Details
2025-01-15 14:30:22	admin@firma.ch	USER_CREATED	max@firma.ch
2025-01-15 14:35:10	admin@firma.ch	PROJECT_ASSIGNED	Max → Projekt ABC
2025-01-15 14:40:55	max@firma.ch	CLOCK_IN	Projekt ABC
2025-01-15 15:20:30 (Korrektur)	admin@firma.ch	TIME_ENTRY_CORRECTED	Entry #123
2025-01-15 16:00:00	admin@firma.ch	ABSENCE_APPROVED	Urlaub Max 5 Tage

Zugriff auf Audit Log:

- Navigieren Sie zu "**System**" → "**Audit Log**"
- Filter nach Benutzer, Datum, Aktion
- Export als CSV möglich

System-Einstellungen

Globale Konfiguration:

Zeiterfassung	
Max. Arbeitszeit/Tag:	10 Stunden
Pflicht-Pause ab:	6 Stunden
Auto-Ausstempeln:	18:00 Uhr
Zeitbuchung in Zukunft:	<input type="checkbox"/> Erlauben
 Urlaub	
Standard Urlaubstage:	25 Tage
Min. Vorlaufzeit:	2 Wochen
Max. Urlaubstage am Stück:	20 Tage
Resturlaub übertragbar:	<input checked="" type="checkbox"/> Ja (bis 31.03.)
 Rechnungen	
Standard Zahlungsziel:	30 Tage
MwSt-Satz:	8.1%
Rechnungsnummer-Format:	RE-{JAHR}-{NUMMER}
Automatische Nummerierung:	<input checked="" type="checkbox"/> Ja
 E-Mail	
SMTP Server:	smtp.firma.ch
Port:	587
Verschlüsselung:	TLS
Absender:	noreply@cflux.firma.ch

Wartungsarbeiten

Regelmäßige Tasks:

Täglich:

- Backup prüfen
- Offene Zeiteinträge schließen (Auto)
- Überfällige Rechnungen markieren (Auto)

Wöchentlich:

- Log-Dateien rotieren
- Performance-Metriken prüfen
- Disk Space monitoren

Monatlich:

- Alte Logs archivieren
- Backup-Rotation
- Security Updates prüfen

Jährlich:

- Urlaubskontingente zurücksetzen
- Jahresarchiv erstellen
- System-Review

Wartungsmodus**System für Wartung sperren:**

```
# Wartungsmodus aktivieren
docker exec timetracking-backend npm run maintenance:on

# Wartungsmodus deaktivieren
docker exec timetracking-backend npm run maintenance:off
```

Effekt:

- Benutzer sehen Wartungshinweis
 - Login nicht möglich
 - Admin-Zugang bleibt bestehen
 - Keine Datenverluste
-

Best Practices**Tägliche Routine****Morgens (08:00-09:00):**

1. Dashboard-Check
2. Urlaubsanträge prüfen (max. 24h Reaktionszeit)
3. Offene Zeiteinträge vom Vortag schließen
4. Krankheitsmeldungen prüfen

Mittags:

1. Aktuelle Team-Auslastung prüfen
2. Zeitkorrekturen durchführen

Abends (17:00-18:00):

1. Tagesübersicht erstellen
2. Kritische Alerts prüfen

3. Backup-Status kontrollieren

Wöchentliche Aufgaben

Montags:

- Wochenplanung: Team-Auslastung
- Projekt-Status-Updates
- Urlaubs-Kalender prüfen

Freitags:

- Wochenreport erstellen
- Zeiteinträge-Qualität prüfen
- Ausstehende Genehmigungen abarbeiten

Monatliche Aufgaben

Monatsanfang:

- Vormonat abschließen
- Monatsberichte erstellen
- Urlaubsübersicht aktualisieren

Monatsende:

- Rechnungen erstellen
- Projekt-Auswertungen
- Budget-Kontrolle

Kommunikations-Richtlinien

Urlaubsanträge:

- Reaktionszeit: Max. 24 Stunden
- Bei Ablehnung: Immer Begründung + Alternative
- Bei Genehmigung: Kurzes Bestätigung-Kommentar

Zeitkorrekturen:

- Immer Korrektur-Grund dokumentieren
- Benutzer informieren über größere Änderungen
- Bei Unstimmigkeiten: Persönliches Gespräch

Probleme:

- Offene Kommunikation
- Lösungsorientiert
- Dokumentation im System

Datenschutz & Compliance

DSGVO-Konformität:

- Mitarbeiter-Daten minimieren
- Zugriffskontrolle (Rollen)
- Audit-Logs führen
- Datenlöschung auf Anfrage

Schweizer OR (Obligationenrecht):

- Korrekte Arbeitszeiterfassung
- Pausenregelungen einhalten
- Überstunden-Tracking
- Aufbewahrungspflicht (10 Jahre)

Tips:

- Regelmäßige Datenschutz-Schulungen
 - Privacy by Design
 - Verschlüsselung nutzen
 - Zugriffsrechte regelmäßig prüfen
-

Troubleshooting

Häufige Admin-Probleme

Problem: Benutzer kann sich nicht anmelden**Checkliste:**

1. Ist Benutzer aktiv? (Status prüfen)
2. Passwort korrekt? (Temporäres PW setzen)
3. E-Mail-Adresse korrekt?
4. Account gesperrt? (Nach zu vielen Fehlversuchen)
5. System-weites Problem? (Andere Benutzer betroffen?)

Lösung:

- Benutzer reaktivieren
- Passwort zurücksetzen
- Account entsperren (nach 30 Min automatisch)

Problem: Zeiteinträge fehlen**Checkliste:**

1. Falscher Zeitraum gewählt?
2. Filter aktiv? (Projekt, Status)
3. Benutzer hat vergessen zu buchen?
4. Daten verloren gegangen? (Backup prüfen)

Lösung:

- Filter zurücksetzen
- Benutzer kontaktieren
- Bei Datenverlust: Backup einspielen

Problem: Reports zeigen falsche Zahlen

Checkliste:

1. Zeitzone korrekt?
2. Filter richtig gesetzt?
3. Inkonsistente Daten? (Offene Einträge)
4. Bug im System?

Lösung:

- Offene Zeiteinträge schließen
- Datenintegrität prüfen
- Bei Fehlern: Support kontaktieren

Anhang

Checklisten

Neuer Mitarbeiter Onboarding:

- Benutzer anlegen
- Urlaubskontingent setzen (Standard: 25)
- Zu Projekten zuweisen
- Willkommens-E-Mail mit Login-Daten
- Einführung in cflux geben
- Erste Zeitbuchung prüfen

Mitarbeiter Offboarding:

- Laufende Projekte abschließen
- Letzte Zeiteinträge prüfen
- Resturlaub auszahlen/dokumentieren
- Final-Report erstellen
- Benutzer deaktivieren (nicht löschen!)
- Zugriffsrechte entziehen

Monatsabschluss:

- Alle Zeiteinträge prüfen
- Urlaubsanträge bearbeiten
- Krankheitstage dokumentieren
- Projekt-Auswertungen erstellen
- Rechnungen erstellen
- Monatsreport an Management
- Backup kontrollieren

Kontakte

Technischer Support:

- E-Mail: support@cflux.ch
- Hotline: +41 XX XXX XX XX
- Öffnungszeiten: Mo-Fr 08:00-17:00

Entwickler:

- GitHub: github.com/mpue/cflux
 - Issues: github.com/mpue/cflux/issues
-

Ende Administrator-Handbuch

Vielen Dank für Ihre Arbeit als cflux Administrator!

Bei Fragen steht Ihnen unser Support-Team zur Verfügung.

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,
    "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",  
  "duration": 32400, // Sekunden (9 Stunden)  
  "durationFormatted": "09:00:00",  
  "description": "Completed feature X implementation"  
}
```

```
"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
}
}
}

```

Benutzer-Zeiteinträge abrufen

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": "Project A",  
      "description": "Project A description",  
      "status": "ACTIVE",  
      "createdBy": "user-123",  
      "lastModified": "2025-01-15T10:00:00.000Z",  
      "clients": [  
        {  
          "id": "client-123",  
          "name": "Client A",  
          "status": "ACTIVE",  
          "lastModified": "2025-01-15T10:00:00.000Z"  
        }  
      ]  
    },  
    {  
      "id": "proj-456",  
      "name": "Project B",  
      "description": "Project B description",  
      "status": "INACTIVE",  
      "createdBy": "user-456",  
      "lastModified": "2025-01-15T10:00:00.000Z",  
      "clients": []  
    }  
  ]  
}
```

```
{  
    "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  
    }  
},  
]  
,"pagination": {  
    "total": 10,  
    "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: Überlappendende 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": "PENDING",  
            "days": 5, // Arbeitstage (ohne Wochenende)  
            "createdAt": "2025-01-15T13:00:00.000Z"  
        }  
    ]  
}
```

```
"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": "2025-02-05T00:00:00.000Z"
        }
    ]
}
```

```
        "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
            }
        ]
    }
}
```

Benutzer-Zusammenfassung

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",
            "name": "John Doe",
            "department": "Sales"
        }
    ]
}
```

```

    "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
    }
]
```

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

Code	Bedeutung	Verwendung
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