

Installation & Start

1. Backend starten

```
# Im Projektverzeichnis  
./stop.sh && ./start.sh
```

2. Frontend starten

```
# Next.js App  
cd rico-ops-hub  
npm install  
echo "NEXT_PUBLIC_API_BASE=http://localhost:8000" > .env.local  
npm run dev
```

3. URLs

- **Backend API:** <http://localhost:8000>
- **Frontend:** <http://localhost:3000>
- **API Docs:** <http://localhost:8000/api/v1/docs>

API Endpoints

v2 Core

- `GET /v2/core/prompts` - Prompt Library
- `POST /v2/core/prompts` - Neuer Prompt
- `GET /v2/core/runs` - Runs & Telemetry
- `GET /v2/core/kb/search` - Knowledge Base Suche

v2 Practice

- `GET /v2/practice/stats` - Praxis-KPIs
- `GET /v2/practice/patients` - Patienten
- `GET /v2/practice/appointments` - Termine
- `GET /v2/practice/invoices` - Rechnungen

v2 Finance

- `GET /v2/finance/kpis` - Finanz-KPIs (MRR, ARR, Cash, Runway)
- `GET /v2/finance/forecast` - 12-Monats Forecast

v2 Cashbot

- `POST /v2/cashbot/scan` - Cashflow-Scan starten
- `GET /v2/cashbot/findings` - Findings anzeigen
- `POST /v2/cashbot/dispatch/{id}` - Finding an n8n senden

Frontend Seiten

- **Home** (`/`) - Dashboard mit Health, KPIs, Quick Actions
- **Agents** (`/agents`) - Multi-Provider Konsole
- **Cashbot** (`/cashbot`) - Cashflow-Radar
- **Finance** (`/finance`) - Finanz-KPIs & Forecast
- **Practice** (`/practice`) - Tierheilpraxis Verwaltung
- **Prompts** (`/prompts`) - Prompt Library
- **Runs** (`/runs`) - Telemetry & Kosten

Features

Health Check 2.0

- Provider-Status (OpenAI, Claude, Perplexity)
- Latenz-Messung

- ENV_SOURCE Anzeige

✅ Auto-Race Mode

- First-success-wins
- Meta-Daten (Provider, Model, Duration)
- Saubere Cancellation

✅ Cashbot

- KI-gestützte Cashflow-Analyse
- Priorisierte Findings
- n8n Integration (Webhook)

✅ Finanz-KPIs

- MRR/ARR Berechnung
- Cash on Hand
- Burn Rate & Runway
- 12-Monats Forecast

✅ Tierheilpraxis

- Patienten-Verwaltung
- Terminplanung
- Rechnungsverwaltung
- KPIs Dashboard

Sicherheit

- ✅ `.env.local` Vorrang vor `.env`
- ✅ Keine Secrets in Logs/UI
- ✅ CORS für Next.js (localhost:3000)
- ✅ Einheitliches Fehlermapping

Tests

```
# Backend Tests
python3 -m pytest tests/test_v2_apis.py -v

# Frontend Tests
python3 -m pytest tests/test_frontend_integration.py -v
```

Sanity Checks

```
# Health Check
curl http://localhost:8000/check-keys

# Practice Stats
curl http://localhost:8000/v2/practice/stats

# Finance KPIs
curl http://localhost:8000/v2/finance/kpis

# Cashbot Scan
curl -X POST http://localhost:8000/v2/cashbot/scan

# Cashbot Findings
curl http://localhost:8000/v2/cashbot/findings
```

Nächste Schritte

1. **Backend starten:** `./start.sh`
2. **Frontend starten:** `cd rico-ops-hub && npm run dev`
3. **Testen:** Sanity-Checks ausführen
4. **Anpassen:** Prompts, KPIs, n8n-Webhooks konfigurieren

Troubleshooting

- **Import Errors:** Backend-Pfade prüfen

- **CORS Issues:** localhost:3000 in main.py
 - **ENV Issues:** .env.local vs .env
 - **Database:** SQLite wird automatisch erstellt
-

Status:  Vollständig implementiert **Letzte Aktualisierung:** 2024-01-15