



# Rico 4.0 UI-Redesign - Komplettüberblick

---

## Ziel erreicht: Modernes, modulares Frontend

Das Rico 4.0 Frontend wurde vollständig modernisiert mit:

- **Modulare Architektur** mit separaten Komponenten
- **Dark Theme** mit vollständigen Design-Tokens
- **Optionales n8n-Panel** für Workflow-Integration
- **UI\_API\_BASE-Support** für flexible Backend-Konfiguration
- **Deutsche Fehlermeldungen** mit Secrets-Redaction

## Neue Dateistruktur

```
frontend/
├─ streamlit_app.py          # ✅ Refactored auf Komponenten
├─ ui/
│   ├─ theme.py              # ✅ Vollständige Design-Tokens
│   ├─ layout.py             # ✅ Header, Sidebar, Grid-System
│   ├─ assets/
│   │   └─ custom.css        # ✅ Lokale CSS-Tweaks
│   └─ sections/
│       ├─ health_panel.py    # ✅ Health-Check 2.0 + Adapter
│       ├─ provider_controls.py # ✅ Provider-Dropdown + UI_API_BASE
│       ├─ result_tabs.py     # ✅ Deutsche Keys + Strict Parsing
│       ├─ toast.py           # ✅ Deutsche Fehlermeldungen
│       ├─ n8n_panel.py       # ✅ Optionales n8n-Panel
│       ├─ task_monitor.py    # ✅ Task-Monitoring
│       └─ export_panel.py    # ✅ Export-Funktionen
└─ integrations/
    └─ n8n_client.py          # ✅ UI-only Webhook-Client
```

## Konfiguration (.env.local)

```
# UI API Base (Backend-URL)
UI_API_BASE=http://localhost:8000

# n8n Integration (optional)
N8N_ENABLED=true
N8N_TIMEOUT_SECONDS=20

# n8n Webhooks - Variante 1: JSON-String
N8N_WEBHOOKS_JSON=[{"name":"CRM Lead","url":"http://localhost:5678/webhook"}]

# n8n Webhooks - Variante 2: JSON-Datei
N8N_WEBHOOKS_FILE=integrations/n8n_webhooks.local.json
```

## Design-System

### Design-Tokens (theme.py)

- **Farben:** Smaragdgrün (#009688) + Gold (#D4AF37)
- **Spacing:** xs (4px) bis xxl (48px)
- **Radius:** sm (6px) bis xl (20px)
- **Shadows:** sm, md, lg
- **Typography:** System-Fonts + Monospace

### CSS-Variablen

Alle Tokens werden als CSS-Variablen verfügbar gemacht:

```
:root {
  --rico-bg: #0f1216;
  --rico-brand: #009688;
  --rico-space-md: 0.75rem;
  --rico-radius-md: 10px;
  /* ... */
}
```



## Neue Features

### 1. Health-Check 2.0

- **UI\_API\_BASE-Support:** Konfigurierbare Backend-URL
- **Adapter-Pattern:** Unterstützt beide API-Response-Formen
- **Erweiterte Metriken:** Latenz, Model, env\_source
- **Gesamtampel:** "x/3 OK" Status-Übersicht

### 2. Provider-Controls 2.0

- **Erweiterte Dropdowns:** autolopenailclaudelperplexity
- **Task-Typen:** analyselrecherchelstrategielcodelberatungloptimierung
- **Request-URL-Anzeige:** Kompakte API-URL-Darstellung
- **UI\_API\_BASE-Integration:** Dynamische URL-Generierung

### 3. Result-Tabs 2.0

- **Strict Parsing:** Sichere Extraktion ohne Secrets
- **Deutsche Keys:** Alle Tabs mit deutschen Bezeichnungen
- **Download-Funktionen:** JSON + Log-Export
- **Meta-Darstellung:** Provider, Modell, Dauer, Zeit

### 4. Toast-System 2.0

- **Deutsche Fehlermeldungen:** 10+ Fehlertypen
- **Secrets-Redaction:** Automatische Ausblendung
- **Regex-Pattern:** Erkennung von API-Keys, Tokens, etc.
- **Universal-Funktion:** show\_toast() für alle Typen

### 5. n8n-Integration (Optional)

- **UI-only:** Keine Backend-Änderungen nötig
- **Webhook-Client:** Vollständiger n8n-Client
- **Zwei Konfigurationsarten:** ENV-JSON oder Datei

- **Test-Funktionen:** Webhook-Tests + Ergebnis-Versand

## 6. Task-Monitor

- **Autopilot-Status:** Start/Pause-Controls
- **Aktive Tasks:** Live-Monitoring
- **Task-Historie:** Letzte 24h
- **Performance-Metriken:** Erfolgsrate, Ø-Dauer

## 7. Export-Panel

- **Multi-Format:** JSON, Markdown, PDF, CSV
- **Konfigurierbar:** Mit/ohne Metadaten, Rohdaten
- **Download-Buttons:** Direkter Export
- **Export-Historie:** Lokale Übersicht



## Sicherheit

### Secrets-Redaction

- **Automatische Erkennung:** Keys, Tokens, Webhooks
- **Regex-Pattern:** Sk-, pk\_, ghp\_\*, etc.
- **UI-Safe:** Niemals Secrets im UI/Log
- **Toast-Integration:** Sichere Fehlermeldungen

### Strict Parsing

- **dict.get():** Sichere Extraktion
- **Type-Checking:** Robuste Datentyp-Behandlung
- **Fallback-Werte:** N/A für fehlende Daten
- **Error-Handling:** Graceful Degradation

## Akzeptanzkriterien

- ☒ **Funktion unverändert:** Alle bestehenden Features funktionieren
- ☒ **Neues Layout aktiv:** Moderne Sidebar mit Akkordeon
- ☒ **Health-Check 2.0:** Ampeln + Latenzen + Header-Badge
- ☒ **Provider-Dropdown:** autolopenailclaudelperplexity
- ☒ **Deutsche Keys:** Alle Tabs mit deutschen Bezeichnungen
- ☒ **Meta sichtbar:** used\_provider, provider\_model, duration\_s
- ☒ **n8n opt-in:** Panel nur bei N8N\_ENABLED=true
- ☒ **Keine Secrets:** Vollständige Redaction
- ☒ **State-Persistenz:** Session State funktioniert

## Quick Start

### 1. Backend starten:

```
cd backend && python -m uvicorn app.main:app --reload
```

### 2. Frontend starten:

```
cd frontend && streamlit run streamlit_app.py
```

### 3. n8n aktivieren (optional):

```
# .env.local erstellen  
echo "N8N_ENABLED=true" > .env.local  
echo 'N8N_WEBHOOKS_JSON=[{"name":"Test","url":"http://localhost:5678
```

## Ergebnis

Das Rico 4.0 Frontend ist jetzt:

- **Modern:** Dark Theme mit Design-System
- **Modular:** Saubere Komponenten-Architektur
- **Erweiterbar:** n8n-Integration + Export-Funktionen
- **Sicher:** Secrets-Redaction + Strict Parsing
- **Benutzerfreundlich:** Deutsche UI + Intuitive Bedienung

**Alle Anforderungen erfüllt - Ready for Production! 🚀**