

📄 KI-Rechnungsverarbeitung v2.1

Installations- & Nutzungsanleitung

Automatische PDF-Extraktion mit ChatGPT API

📄 Inhaltsverzeichnis

1. Überblick
 2. Systemvoraussetzungen
 3. Installation (Schritt-für-Schritt)
 4. Erste Nutzung
 5. 3 Verfügbare Versionen
 6. Troubleshooting
 7. Kosten & Performance
 8. Support
-

📄 Überblick

Dieses Tool extrahiert automatisch strukturierte Daten aus PDF-Rechnungen mittels künstlicher Intelligenz (ChatGPT).

Was es kann:

- 📄 PDF-Rechnungen automatisch auslesen
- 📄 Strukturierte Datenextraktion:
 - Rechnungsnummer
 - Datum & Fälligkeitsdatum
 - Lieferant & Adresse
 - Beträge (Netto/Brutto/MwSt)
 - IBAN & BIC
 - Steuernummer & USt-IdNr
- 📄 Excel-Export (.xlsx)
- 📄 3 verschiedene Benutzeroberflächen

Performance:

- ⚡ < 10 Sekunden pro Rechnung
 - 📄 95%+ Genauigkeit
 - 💰 ~0,0005€ pro Rechnung (API-Kosten)
-

📄 Systemvoraussetzungen

Betriebssystem

- 📄 macOS 10.14+
- 📄 Windows 10+
- 📄 Linux (Ubuntu 20.04+)

Software

- 📄 Python 3.10 oder höher
- 📄 Git (optional, für Klonen)
- 📄 Internetverbindung (für OpenAI API)

OpenAI Account

- 📄 OpenAI Account erstellen: <https://platform.openai.com>
 - 📄 API-Key generieren
 - 📄 Guthaben aufladen (min. 5€ empfohlen)
-

📄 Installation (Schritt-für-Schritt)

Schritt 1: Projekt herunterladen

Option A: Mit Git (empfohlen)

```
git clone https://github.com/Luyzz22/ki-rechnungsverarbeitung.git
cd ki-rechnungsverarbeitung
```

Option B: Als ZIP

1. Öffne: <https://github.com/Luyzz22/ki-rechnungsverarbeitung>
2. Klicke: "Code" → "Download ZIP"
3. Entpacke die ZIP-Datei
4. Öffne Terminal/CMD im entpackten Ordner

Schritt 2: Virtual Environment erstellen

macOS/Linux:

```
python3 -m venv venv
source venv/bin/activate
```

Windows (CMD):

```
python -m venv venv
venv\Scripts\activate
```

Windows (PowerShell):

```
python -m venv venv
venv\Scripts\Activate.ps1
```

Erfolg: Du siehst `(venv)` vor deinem Prompt

Schritt 3: Dependencies installieren

```
pip install -r requirements.txt
```

Dauer: 30-60 Sekunden

Installiert:

- OpenAI API Client
- PyPDF2 (PDF-Parsing)
- pandas & openpyxl (Excel)
- rich (CLI-Formatierung)
- tkinter (GUI - meist vorinstalliert)

Schritt 4: OpenAI API-Key einrichten

4.1 API-Key erstellen

1. Gehe zu: <https://platform.openai.com/api-keys>
2. Login mit deinem OpenAI Account
3. Klicke: "Create new secret key"
4. Name: "Rechnungsverarbeitung"
5. **WICHTIG:** Kopiere den Key sofort! (Wird nur einmal angezeigt)

Format: `sk-proj-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`

4.2 Key in Projekt einfügen

macOS/Linux:

```
echo "OPENAI_API_KEY=sk-proj-DEIN-KEY-HIER" > .env
```

Windows (CMD):

```
echo OPENAI_API_KEY=sk-proj-DEIN-KEY-HIER > .env
```

Oder manuell:

1. Erstelle Datei `.env` im Projektordner
2. Inhalt:

```
OPENAI_API_KEY=sk-proj-DEIN-KEY-HIER
```

3. Speichern

⚠ **WICHTIG:** Der `.env`-File enthält deinen privaten API-Key! Nicht teilen!

Schritt 5: Test-Ordner vorbereiten

```
mkdir -p test_rechnungen
```

Kopiere deine PDF-Rechnungen in diesen Ordner:

macOS/Linux:

```
cp ~/Desktop/Rechnungen/*.pdf test_rechnungen/
```

Windows:

```
copy %USERPROFILE%\Desktop\Rechnungen\*.pdf test_rechnungen\
```

Oder: Ziehe PDFs per Drag & Drop in den `test_rechnungen` Ordner

Schritt 6: Installation testen

```
python invoice_parser.py
```

Erwartetes Ergebnis: ``

📄 KI-RECHNUNGSVERARBEITUNG v1.0

📄 Verarbeite 2 Rechnungen...

[1/2] 📄 Rechnung_2024_001.pdf 📄 Musterfirma GmbH 📄 199.99€

[2/2] 📄 Rechnung_2024_002.pdf 📄 Beispiel AG 📄 1234.56€

=====

📄 Fertig! 2 Rechnungen verarbeitet 📄 Export: rechnungen_export.xlsx 📄
Gesamt: 1434.55€ 📄 Excel wurde geöffnet!

```
**📄 Installation erfolgreich!**
```

```
---
```

```
## 📄 Erste Nutzung
```

```
### Schnellstart (GUI - Empfohlen)
```

```
```bash
```

```
1. Virtual Environment aktivieren
```

```
source venv/bin/activate # macOS/Linux
```

```
venv\Scripts\activate # Windows
```

```
2. GUI starten
```

```
python invoice_parser_gui.py
```

Im GUI-Fenster:

1. 📄 PDFs AUSWÄHLEN klicken
2. Wähle 1-100 PDF-Rechnungen aus
3. 📄 JETZT VERARBEITEN klicken

4. Warte (Live-Log zeigt Fortschritt)
5. Excel öffnet sich automatisch!

**Optional:** Button `EXCEL ÖFFNEN` zum erneuten Öffnen

---

## 3 Verfügbare Versionen

---

### Version 1: GUI (Grafisches Fenster)

```
python invoice_parser_gui.py
```

**Perfekt für:**

- Nicht-technische User
- Live-Demos
- Gelegentliche Nutzung
- Drag & Drop

**Features:**

- Modernes Dark Theme
  - Datei-Browser
  - Live-Fortschrittsbalken
  - Farbcodiertes Logging
  - Excel-Button
  - Auto-Open Excel
- 

### Version 2: CLI Pretty (Schönes Terminal)

```
python invoice_parser_v2.py
```

**Perfekt für:**

- Power-User
- Server ohne GUI
- Präsentationen
- Screenshots

**Features:**

- Farbiges Terminal-Output
- Unicode-Box-Grafiken
- Strukturierte Tabellen
- Live-Progress-Bar
- Summary-Panel
- Auto-Open Excel

PDFs müssen im `test_rechnungen/` Ordner liegen!

---

### Version 3: CLI Basic (Simples Terminal)

```
python invoice_parser.py
```

**Perfekt für:**

- Automation (Cron-Jobs)
- Scripts
- CI/CD-Pipelines
- Headless-Server

**Features:**

- Minimalistisches Output
- Schnelle Verarbeitung
- Scriptable
- Auto-Open Excel

PDFs müssen im `test_rechnungen/` Ordner liegen!

---

## Troubleshooting

---

Problem: "OPENAI\_API\_KEY nicht gefunden"

#### Lösung:

```
Prüfe ob .env existiert
ls -la .env

Prüfe Inhalt
cat .env

Sollte zeigen:
OPENAI_API_KEY=sk-proj-...

Falls leer oder falsch:
echo "OPENAI_API_KEY=sk-proj-DEIN-KEY" > .env
```

---

### Problem: "Module not found"

#### Lösung:

```
Virtual Environment aktiviert?
Du solltest (venv) im Prompt sehen!

Falls nicht:
source venv/bin/activate # macOS/Linux
venv\Scripts\activate # Windows

Dependencies nochmal installieren:
pip install -r requirements.txt
```

---

### Problem: "Insufficient quota" oder "Rate limit"

**Ursache:** Kein Guthaben auf OpenAI Account

#### Lösung:

1. Gehe zu: <https://platform.openai.com/account/billing>
2. "Add payment method"
3. Lade min. 5€ auf
4. Warte 5 Minuten
5. Versuche nochmal

---

### Problem: "PDF kann nicht gelesen werden"

#### Mögliche Ursachen:

- ❌ PDF ist passwortgeschützt
- ❌ PDF ist gescannt (Bild, kein Text)
- ❌ PDF ist korrupt

#### Lösung:

- ❌ Passwort-Schutz entfernen
- ❌ Gescannte PDFs: OCR-Software nutzen
- ❌ PDF neu herunterladen

---

### Problem: "Extrahierte Daten sind falsch"

**Ursache:** KI hat Rechnung falsch interpretiert

#### Lösung:

- ❌ PDF-Qualität prüfen (ist Text lesbar?)
- ❌ Rechnung nochmal verarbeiten
- ❌ Bei 5+ falschen Ergebnissen: Feedback senden

**Durchschnittliche Accuracy:** 95%+

---

### Problem: GUI startet nicht

macOS/Linux:

```
tkinter installiert?
python3 -m tkinter

Sollte Fenster öffnen
```

Windows:

- tkinter ist meist vorinstalliert
- Falls nicht: Python neu installieren mit tkinter-Option

## ☒ Kosten & Performance

### OpenAI API-Kosten

Model: gpt-4o-mini

Kosten pro Rechnung:

- Input: ~3000 Tokens = 0,00045€
- Output: ~200 Tokens = 0,00006€
- **Total: ~0,0005€ pro Rechnung**

### Beispiel-Rechnungen

Rechnungen/Monat	Kosten/Monat	Kosten/Jahr
10	0,05€	0,60€
50	0,25€	3,00€
100	0,50€	6,00€
500	2,50€	30,00€
1000	5,00€	60,00€

Zum Vergleich:

- Manuelle Eingabe: 40€/Stunde
- 100 Rechnungen manuell: 66€
- 100 Rechnungen mit Tool: 0,50€
- **Ersparnis: 65,50€ (99,2%)**

### Performance

Geschwindigkeit:

- 1 Rechnung: ~10 Sekunden
- 10 Rechnungen: ~2 Minuten
- 100 Rechnungen: ~17 Minuten
- 1000 Rechnungen: ~3 Stunden

Genauigkeit:

- Standardrechnungen: 95-98%
- Komplexe Rechnungen: 90-95%
- Gescannte PDFs: 70-85% (nicht empfohlen)

## ☒ Best Practices

Für beste Ergebnisse:

1. **Verwende digitale PDFs**
  - ☒ Von Software generiert
  - ☒ Nicht gescannt/fotografiert
2. **Batch-Verarbeitung**
  - ☒ Verarbeite mehrere auf einmal
  - ☒ Gruppiere nach Monat/Lieferant

3. Qualitätskontrolle

- Stichproben prüfen
- Excel nach Verarbeitung checken
- Auffälligkeiten notieren

4. Backup

- Original-PDFs behalten
- Excel-Exports sichern
- Regelmäßige Backups

📄 Excel-Output

Spalten im Export:

Spalte	Beispiel	Typ
dateiname	Rechnung_001.pdf	Text
rechnungsnummer	RE-2024-001	Text
datum	2024-10-15	Datum
lieferant	Musterfirma GmbH	Text
betrag_brutto	199.99	Zahl
betrag_netto	168.06	Zahl
mwst_betrag	31.93	Zahl
iban	DE89370400440532013000	Text
verarbeitet_am	2024-10-15 15:30:00	Datum/Zeit

**Dateiname:** rechnungen\_export.xlsx  
**Format:** Excel 2007+ (.xlsx)  
**Kompatibel mit:** Excel, LibreOffice, Google Sheets, DATEV

🔒 Datenschutz & Sicherheit

Lokale Verarbeitung

- Nur PDF-Text wird an OpenAI gesendet
- Keine Bilder/Metadaten
- Verarbeitung über verschlüsselte API

API-Key Sicherheit

- ⚠️ .env Datei NIE committen/teilen!
- ⚠️ Key regelmäßig rotieren
- ⚠️ Spending Limits in OpenAI setzen

DSGVO-Konformität

- OpenAI ist GDPR-compliant
- Daten werden nicht trainiert (API)
- 30 Tage Retention (dann gelöscht)

**Empfehlung:** Für sensible Daten Self-Hosted-LLM nutzen (v3.0+)

🛠 Support

GitHub Issues

<https://github.com/Luyzz22/ki-rechnungsverarbeitung/issues>

Email Support

---

## 📄 Changelog

---

### v2.1 (2024-10-15)

- 📄 Auto-Open Excel in allen 3 Versionen
- 📄 Excel-Button in GUI
- 📄 Verbesserte Error-Messages

### v2.0 (2024-10-15)

- 📄 GUI mit Tkinter
- 📄 CLI v2 mit Rich-Formatting
- 📄 Screenshots & Dokumentation

### v1.0 (2024-10-15)

- 📄 Initial Release
- 📄 Basic CLI
- 📄 OpenAI Integration
- 📄 Excel Export

---

## 📄 Lizenz

---

### Proprietary Software

Für kommerzielle Nutzung kontaktieren Sie:

- 📄 [Luis@schenk.com](mailto:Luis@schenk.com)
- 📄 Pricing: Ab 12.000€ (Beta: 5.000€)

---

## 📄 Credits

---

Entwickelt von: Luis Schenk  
GitHub: <https://github.com/Luyzz22>  
Built with: Python, OpenAI, Rich, Tkinter

### Powered by:

- OpenAI API (ChatGPT)
- PyPDF2 (PDF Parsing)
- pandas (Data Processing)
- Rich (Terminal Formatting)

---

Made with ♥ in Weinheim, Germany 🇩🇪

© 2024 Luis Schenk - All Rights Reserved

---