

Amazon Steuer Automatisierung v7

Automatisiere die Verwaltung von Amazon-Rechnungen mit Betragsextraktion und Ordnerstrukturierung.

Features

- 📁 Automatischer Download von Amazon-Rechnungen
- 📄 PDF-Betragsextraktion (Tabellen, Position, Keywords)
- 📁 Automatische Ordnerstrukturierung (Steuer-YYYY/KWXX/)
- 💬 Telegram-Benachrichtigungen
- 📊 Detailliertes Logging
- 🔍 Modulare Architektur

Installation

Voraussetzungen

- Python 3.8+
- pip

Schnellstart

```
# 1. In das files/ Verzeichnis gehen  
cd ~/Dokumente/vsCode/API\ -\ Amazon\ Selenium/v7/files  
  
# 2. Setup ausführen  
bash setup.sh  
  
# 3. Konfiguration erstellen  
cp config.yaml.example config.yaml  
nano config.yaml  
  
# 4. Starten  
python3 main.py --help
```

Verwendung

Kompletter Workflow

```
python3 main.py
```

Nur Amazon Download

```
python3 main.py --only-download  
python3 main.py --year 2024 # Nur 2024
```

Nur PDF Verarbeitung

```
python3 main.py --only-process
```

Dateien ohne Betrag anzeigen

```
python3 main.py --check
```

Konfiguration

Bearbeite `config.yaml`:

```
PROF  
paths:  
  download_dir: ./amazon_downloads      # Amazon Downloads  
  temp_dir: ./temp                      # Temporäre Dateien  
  steuer_dir: ./steuer                  # Steuer-Ordner  
  log_dir: ./logs                       # Log-Dateien  
  
amazon:  
  script_path: ./amazon_invoice_downloader.py  
  config_path: ./amazon_config.yaml  
  
notifications:  
  telegram:  
    enabled: false  
    token: ""          # oder TELEGRAM_BOT_TOKEN env var  
    chat_id: ""        # oder TELEGRAM_CHAT_ID env var  
  
logging:  
  level: INFO      # DEBUG, INFO, WARNING, ERROR  
  to_file: true
```

Telegram Benachrichtigungen

Aktivieren

1. Erstelle einen Bot bei [@BotFather](#)
2. Erhalte deine Chat-ID bei [@userinfobot](#)
3. Setze in `config.yaml`:

```
notifications:
  telegram:
    enabled: true
    token: "dein_bot_token"
    chat_id: "deine_chat_id"
```

Oder nutze Environment-Variablen:

```
export TELEGRAM_BOT_TOKEN="dein_bot_token"
export TELEGRAM_CHAT_ID="deine_chat_id"
python3 main.py
```

Ordnerstruktur

```
steuer/
├── Steuer-2024/
│   ├── KW01/
│   │   ├── 2024_KW01_0042,50_EUR.pdf
│   │   └── 2024_KW01_0125,99_EUR.pdf
│   ├── KW02/
│   │   └── 2024_KW02_0089,00_EUR.pdf
│   ...
└── Steuer-2025/
    └── ...
```

Dateiname-Format: YYYY_KWXX_BETRAG_EUR.pdf

- **YYYY:** Jahr
- **XX:** Kalenderwoche (01-53)
- **BETRAG:** Betrag mit Komma (z.B. 0042,50)

Troubleshooting

"Config-Datei nicht gefunden"

```
cp config.yaml.example config.yaml
nano config.yaml
```

"Python3 nicht gefunden"

```
# Ubuntu/Debian
sudo apt-get install python3
```

```
# macOS  
brew install python3
```

"Module fehlen"

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

"Amazon Downloader Skript nicht gefunden"

Stelle sicher, dass `amazon_invoice_downloader.py` unter dem konfigurierten Pfad existiert.

"Keine Beträge erkannt"

- Prüfe mit `python3 main.py --check`
- Manche PDF-Formate werden möglicherweise nicht erkannt
- Manuelle Nachbearbeitung erforderlich

Logs

Logs werden in `./logs/automation_YYYYMMDD.log` gespeichert.

```
# Letzte Logs anzeigen  
tail -f logs/automation_*.log  
  
# Nach Fehlern suchen  
grep ERROR logs/automation_*.log
```

Module

PROF

Modul	Funktion
<code>config.py</code>	Konfigurationsverwaltung
<code>logger.py</code>	Logging-Setup
<code>amazon_downloader.py</code>	Amazon-Integration
<code>pdf_processor.py</code>	PDF-Betragsextraktion
<code>file_manager.py</code>	Datei- und Ordnerverwaltung
<code>notification.py</code>	Telegram-Benachrichtigungen

Performance

- PDF-Verarbeitung: ~1-2s pro Datei

- Amazon Download: Abhängig von Anzahl der Rechnungen
- Gesamlauf: ~30-60s für 50 Rechnungen

Sicherheit

-  Speichere Telegram-Credentials nicht in `config.yaml`
-  Nutze Environment-Variablen: `TELEGRAM_BOT_TOKEN`, `TELEGRAM_CHAT_ID`
-  Relative Pfade verwenden (portabel)

Lizenz

Privat

Support

Für Probleme:

1. Prüfe die Logs: `tail -f logs/automation_*.log`
2. Führe Setup erneut aus: `bash setup.sh`
3. Prüfe die Konfiguration: `nano config.yaml`

^

Version: 2.0 (Repariert und optimiert)

Letztes Update: Januar 2026