Pflichtenheft

Projekt: Personalisierter Idealo Crawler für Folbt Hillrichs Gewerbe
Datum: 01.06.2025
Bearbeiter: Jakob

---

1. Zielsetzung

Das bestehende Open-Source-Programm (Idealo Crawler) wird im Rahmen dieses Projekts personalisiert und erweitert, um es an die spezifischen Bedürfnisse des Kunden Folbt Hillrichs Gewerbe anzupassen. Ziel ist es, eine performante, stabile und konfigurierbare Lösung bereitzustellen, die auf einem V-Server (z. B. Hostinger) oder einem privaten Server (z. B. Raspberry Pi) betrieben werden kann.
Die gescrapten Daten werden in Google Sheets ausgegeben, und es sollen Benachrichtigungen per E-Mail eingerichtet werden.

---

2. Ausgangssituation

Der Kunde nutzt derzeit einen Idealo Crawler mit folgenden Basisfunktionen:
- Scrapen bestimmter Idealo-Kategorien
- Speicherung der Daten in Google Sheets
- Filterung nach: - Preis - FBA-Marge - Anzahl Bewertungen - BSR (Best Seller Rank) - Blacklists - Keepa 30-Tage Durchschnittspreis (ungenau)

Bereits durchgeführte frühere Erweiterungen:
- Verbesserter Multithread-Support für schnelleres Crawling
- Scannen des gesamten Idealo-Katalogs (nicht nur einzelner Kategorien)
- Ermittlung der BuyBox-Verkäufer und BuyBox-Ja/Nein
- Nutzung des Keepa 90-Tage-Durchschnittspreises (funktioniert nicht zuverlässig)
- Ermittlung der geschätzten Verkaufszahlen im letzten Monat (funktioniert nicht zuverlässig)

---

3. Anforderungen

3.1 Funktionale Anforderungen

- **Crawler-Logik:**
  - Komplettes Durchlaufen aller Idealo-Kategorien
  - Nach vollständigem Durchlauf automatischer Neustart

  - Zusätzlich dauerhafter Scrape des „Schnäppchen“-Tabs

  - Umgang mit Duplikaten:
    * Zeitbasierte Filterung, z. B. „ignoriere identisches Ergebnis für X Tage“ (alternativ: intelligentes Update-Handling)
- **Konfigurierbarkeit (über Konfigurationsdatei):**
  - Mindestmarge (einstellbar)

  - Minimaler Profitbetrag

  - Unterste Preisgrenze

  - Einstellbare Blacklists und Whitelists (optional)
- **Performance:**
  - Optimierung des Codes für schwächere Hardware (z. B. Raspberry Pi)

  - Verbesserte Ressourcennutzung, z. B. effizienteres Multithreading, geringere Speicherlast
- **Datenspeicherung und Ausgabe:**
  - Speicherung aller Ergebnisse in einem Google Sheets Dokument

  - Automatisches Update existierender Sheets, keine Duplikate
- **Benachrichtigungen:**
  - Push-Benachrichtigungen oder E-Mail-Benachrichtigungen (z. B. bei Fehlern, bei erfolgreichen Crawlvorgängen)
- **Amazon SP-API Integration:**
  - Abrufen des 30-Tage-Durchschnittspreises direkt von Amazon

  - Abrufen von FBA- und Amazon-Gebühren

  - Nutzung bestehender API-Schlüssel des Kunden
- **Zusätzliche Kennzahlen:**
  - Durchschnittlicher BSR (Best Seller Rank)
- **Protokollierung:**
  - Server-Logs über laufende Vorgänge und Fehler

  - Optional: Logfile-Rotation oder Archivierung

3.2 Nicht-funktionale Anforderungen

- **Betriebsumgebung:**
  - Lauffähig auf einem gemieteten V-Server (z. B. Hostinger)

  - Alternativ lauffähig auf einem privaten Server (z. B. Raspberry Pi)

  - Setup erfolgt vorerst ohne Docker (manuelles Deployment, Kunde übernimmt Setup)
- **Benutzerverwaltung:**
  - Kein Multi-User-Zugang innerhalb des Programms erforderlich
  - Mehrere Nutzer dürfen auf das Google Sheets Ergebnis zugreifen
- **Sicherheit:**
  - Sicherer Umgang mit API-Keys und Zugangsdaten
  - Kein öffentlich zugängliches Web-Interface

---

4. Technische Rahmenbedingungen

- **Programmiersprache:**
  Python (bestehender Code)
- **APIs:**
  - Idealo-Website (Scraping, unter Berücksichtigung der aktuellen Bot Detection Mechanismen)
  - Keepa API (für historische Preisdaten)
  - Amazon SP-API (für aktuelle Preisdaten, FBA-Gebühren etc.)
- **Konfiguration:**
  - Konfigurationsdateien (z. B. JSON, YAML, INI) zur Einstellung der Filter und Parameter
  - Keine grafische Benutzeroberfläche notwendig
- **Ausgabeformate:**
  - Google Sheets (direkte API-Anbindung, automatisches Schreiben)
- **Logging:**
  - Konsolen-Logs und Log-Dateien auf dem Server

---

5. Offene Punkte

Thema	Status
API-Schlüssel für SP-API	Vom Kunden bereitgestellt
Authorisierungscode SP-API	Noch zu klären

Thema	Status
Bot Detection bei Idealo umgehen	Erfordert neuen Ansatz, prüfen
Konfigurationsdatei Format	Wird noch im Detail abgestimmt
E-Mail-Benachrichtigungssystem	SMTP-Setup wird festgelegt

---

6. Zeitplan

Meilenstein	Termin
Pflichtenheft finalisieren	05.06.2025
Setup Entwicklungsumgebung	10.06.2025
Codeanpassungen und Optimierungen starten	15.06.2025
Integration der Amazon SP-API + Tests	15.07.2025
Integration der E-Mail-Benachrichtigung	22.07.2025
Gesamtfunktionstest und Debugging	25.07.2025
Deployment auf V-Server + Abnahme	30.07.2025

---

7. Sonstiges

- Testtabellen/Beispieltabellen aus der bisherigen Arbeit sind vorhanden
- Der Kunde übernimmt das Deployment auf dem Hostinger-Server
- Eventuelle spätere Dockerisierung oder zusätzliche Features (z. B. weitere Filter, Ländervergleiche) können als Folgeprojekt geplant werden
- Kommunikation:
  - Wichtige Fragen per E-Mail
  - Updates regelmäßig per Chat
