Himalaya Email-Management

Automatisierte Email-Filterung und -Sortierung

Dr. Jo

28. Januar 2026

Inhaltsverzeichnis

1 Einführung **4**
1.1 Systemanforderungen . . . 4

2 Installation und Konfiguration **4**
2.1 Himalaya Installation . . . 4
2.2 Basiskonfiguration . . . 4
2.2.1 Account-Setup . . . 4
2.2.2 Passwort-Speicherung (macOS Keychain) . . . 5
2.3 Verbindungstest . . . 5

3 Email-Filter-System **5**
3.1 Architektur . . . 5
3.2 Filter-Konfiguration . . . 6
3.2.1 Struktur . . . 6
3.2.2 Match-Typen . . . 6
3.2.3 Action-Typen . . . 6
3.3 Aktuelle Filterregeln (27 Regeln) . . . 6
3.4 Filter-Script . . . 7
3.4.1 Funktionsweise . . . 7
3.4.2 Wichtige Features . . . 7
3.4.3 Parsing-Besonderheit . . . 8

4 Ausführung **8**
4.1 Manuelle Ausführung . . . 8
4.1.1 Direkter Aufruf . . . 8
4.1.2 Via Alias (empfohlen) . . . 8
4.2 Automatische Ausführung . . . 8
4.2.1 Täglich um 8:00 Uhr . . . 8
4.2.2 Andere Zeitpunkte . . . 9

5 Verwaltung **9**
5.1 Filter hinzufügen . . . 9
5.2 Filter deaktivieren . . . 10
5.3 Dry-Run (Test-Modus) . . . 10
5.4 Logs anzeigen . . . 10

6 Himalaya Grundbefehle **10**
6.1 Email-Verwaltung . . . 10
6.2 Ordner-Verwaltung . . . 10
6.3 Suche . . . 11

7 Troubleshooting **11**
7.1 Häufige Probleme . . . 11
7.1.1 Authentifizierungs-Fehler . . . 11
7.1.2 Filter funktionieren nicht . . . 11
7.1.3 LaunchAgent startet nicht . . . 11
7.1.4 Node.js nicht gefunden . . . 12
7.2 Performance-Optimierung . . . 12
7.2.1 Viele Emails . . . 12
7.2.2 Logging reduzieren . . . 12

8 Best Practices **12**
8.1 Sicherheit . . . 12
8.2 Filter-Design . . . 13
8.3 Wartung . . . 13

9 Erweiterte Konfiguration **13**
9.1 Mehrere Accounts . . . 13
9.2 Custom Folder-Mapping . . . 13
9.3 Email-Vorlagen . . . 13

10 Zusammenfassung **14**
10.1 Systemarchitektur . . . 14
10.2 Wichtige Dateien . . . 14
10.3 Statistik . . . 14

11 Schnellreferenz **14**
11.1 Wichtigste Befehle . . . 14
11.2 Nützliche Aliases . . . 15

12 Changelog **15**
12.1 Version 1.0 (Januar 2026) . . . 15

13 Support und Ressourcen **15**
13.1 Offizielle Dokumentation . . . 15
13.2 Weitere Hilfe . . . 16

1 Einführung

Dieses Handbuch beschreibt das vollständige Setup eines automatisierten Email-Management-Systems basierend auf **Himalaya**, einem CLI-basierten Email-Client für macOS. Das System ermöglicht:

- Automatische Filterung und Sortierung eingehender Emails
- Spam-Erkennung und automatisches Löschen unerwünschter Nachrichten
- Kategorisierung wichtiger Emails in dedizierte Ordner
- Flexible Ausführung (manuell oder zeitgesteuert)
- Sichere Passwortverwaltung über macOS Keychain

1.1 Systemanforderungen

- macOS (getestet auf macOS 14+)
- Homebrew Package Manager
- Node.js (für Filter-Automatisierung)
- IMAP/SMTP Email-Account (z.B. United Domains)

2 Installation und Konfiguration

2.1 Himalaya Installation

Installation via Homebrew:

```
brew install himalaya
```

Versionsprüfung:

```
himalaya --version
# Ausgabe: himalaya 1.1.0
```

2.2 Basiskonfiguration

Die Konfigurationsdatei befindet sich unter `~/.config/himalaya/config.toml`.

2.2.1 Account-Setup

```
[accounts.roar]
default = true
email = "resa@roar.com"

# Backend-Konfiguration
backend.type = "imap"
backend.host = "imaps.udag.de"
backend.port = 993
backend.encryption = "tls"
backend.login = "resa@roar.com"
backend.auth.type = "password"
backend.auth.command = "security find-generic-password -s roar-imap -w"

```

```
# SMTP fuer Versand
message.send.backend.type = "smtp"
message.send.backend.host = "smtps.udag.de"
message.send.backend.port = 465
message.send.backend.encryption = "tls"
message.send.backend.login = "resa@roar.com"
message.send.backend.auth.type = "password"
message.send.backend.auth.command = "security find-generic-password -s
    roar-smtp -w"

# Ordner-Mapping
folder.alias.inbox = "INBOX"
folder.alias.sent = "INBOX.Sent"
folder.alias.drafts = "INBOX.Drafts"
folder.alias.trash = "INBOX.Trash"
```

2.2.2 Passwort-Speicherung (macOS Keychain)

Sicheres Speichern der Passwörter:

```
# IMAP Passwort
security add-generic-password -a resa@roar.com \
    -s roar-imap -w "IHR_PASSWORT"

# SMTP Passwort
security add-generic-password -a resa@roar.com \
    -s roar-smtp -w "IHR_PASSWORT"
```

Passwort abrufen (Test):

```
security find-generic-password -s roar-imap -w
```

2.3 Verbindungstest

```
# Ordner auflisten
himalaya folder list

# Emails abrufen
himalaya envelope list --page-size 10

# Email lesen
himalaya message read <ID>
```

3 Email-Filter-System

3.1 Architektur

Das Filter-System besteht aus drei Komponenten:

1. **Filter-Konfiguration** (`email-filters.json`)
   JSON-Datei mit allen Filterregeln
2. **Filter-Script** (`apply-email-filters.js`)
   Node.js-Script zur Ausführung der Filter
3. **LaunchAgent** (optional)
   macOS-Daemon für zeitgesteuerte Ausführung

3.2 Filter-Konfiguration

Speicherort: `~/.config/himalaya/email-filters.json`

3.2.1 Struktur

```
{
  "filters": [
    {
      "name": "Beschreibung",
      "enabled": true,
      "match": {
        "type": "from",           // oder "subject"
        "pattern": "regex-pattern",
        "ignoreCase": true
      },
      "action": {
        "type": "move",
        "folder": "INBOX.Zielordner"
      }
    }
  ],
  "settings": {
    "applyToExisting": false,
    "markAsRead": false,
    "logActions": true,
    "dryRun": false
  }
}
```

3.2.2 Match-Typen

- `from`: Absender-Email-Adresse
- `subject`: Email-Betreff

3.2.3 Action-Typen

- `move`: Email in Ordner verschieben
- `delete`: Email löschen (via `INBOX.Trash`)

3.3 Aktuelle Filterregeln (27 Regeln)

Absender	Aktion	Zielordner
Lösch-Filter (10)		
Temu	Löschen	INBOX.Trash
Otodom	Löschen	INBOX.Trash
Statista	Löschen	INBOX.Trash
Chainstack	Löschen	INBOX.Trash
Rallies	Löschen	INBOX.Trash
Mermaid	Löschen	INBOX.Trash
Pipedream	Löschen	INBOX.Trash

Fortsetzung auf nächster Seite

Absender	Aktion	Zielordner
Xmind	Löschen	INBOX.Trash
Yutori	Löschen	INBOX.Trash
Monday.com	Löschen	INBOX.Trash
Junk-Filter (1)		
LinkedIn	Spam	INBOX.Junk
Sortier-Filter (16)		
AliExpress	Sortieren	INBOX.alibaba
Amazon	Sortieren	INBOX.amazon
Anthropic	Sortieren	INBOX.anthropic
Google	Sortieren	INBOX.google
Adobe	Sortieren	INBOX.adobe
eBay	Sortieren	INBOX.ebay
Gelato/Shopify	Sortieren	INBOX.gelato shopify
Hugging Face	Sortieren	INBOX.huggingface
JetBrains	Sortieren	INBOX.jetbrains
Replit	Sortieren	INBOX.replit
ING Bank	Sortieren	INBOX.ing
Sotheby's	Sortieren	INBOX.sothebys
Lindenmüller	Sortieren	INBOX.walles
Kimi	Sortieren	INBOX.kimi
Polymarket	Sortieren	INBOX.polymarket
Cremieux	Sortieren	INBOX.cremieux

3.4 Filter-Script

Das Node.js-Script befindet sich unter:
`~/.config/himalaya/apply-email-filters.js`

3.4.1 Funktionsweise

1. Lädt Filterkonfiguration aus JSON
2. Ruft alle Emails aus INBOX ab
3. Wendet jeden Filter der Reihe nach an
4. Verschiebt passende Emails in Zielordner
5. Protokolliert alle Aktionen in Log-Datei

3.4.2 Wichtige Features

- **Regex-Pattern-Matching**: Flexible Absender-Erkennung
- **Case-Insensitive**: Groß-/Kleinschreibung ignoriert
- **OR-Verknüpfung**: Multiple Patterns via `|`
- **Dry-Run-Modus**: Test ohne tatsächliche Änderungen
- **Detailliertes Logging**: Alle Aktionen nachvollziehbar

3.4.3 Parsing-Besonderheit

Das Script parsed die tabellarische Ausgabe von Himalaya:

```
| ID | FLAGS | SUBJECT | FROM | DATE |
```

Wichtig: Die FLAGS-Spalte kann leer sein! Das Script nutzt daher feste Spalten-Indizes statt Filter.

4 Ausführung

4.1 Manuelle Ausführung

4.1.1 Direkter Aufruf

```
node ~/.config/himalaya/apply-email-filters.js
```

4.1.2 Via Alias (empfohlen)

Alias in `~/.zshrc`:

```
alias emailfilter='node ~/.config/himalaya/apply-email-filters.js'
```

Anwendung:

```
emailfilter
```

4.2 Automatische Ausführung

4.2.1 Täglich um 8:00 Uhr

LaunchAgent erstellen:
`~/Library/LaunchAgents/com.himalaya.email-filters.plist`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
    "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.himalaya.email-filters</string>

    <key>ProgramArguments</key>
    <array>
        <string>/usr/local/bin/node</string>
        <string>/Users/mac/.config/himalaya/apply-email-filters.js</
           string>
    </array>

    <key>StartCalendarInterval</key>
    <dict>
        <key>Hour</key>
        <integer>8</integer>
        <key>Minute</key>
        <integer>0</integer>
    </dict>

    <key>StandardOutPath</key>
```

```
    <string>/Users/mac/.config/himalaya/filter-output.log</string>

    <key>StandardErrorPath</key>
    <string>/Users/mac/.config/himalaya/filter-error.log</string>
</dict>
</plist>
```

Aktivierung:

```
launchctl load ~/Library/LaunchAgents/com.himalaya.email-filters.plist
```

Deaktivierung:

```
launchctl unload ~/Library/LaunchAgents/com.himalaya.email-filters.plist
```

4.2.2 Andere Zeitpunkte

Ändern Sie `Hour` (0-23) und `Minute` (0-59) in der plist-Datei.

Mehrere Zeitpunkte pro Tag:

```
<key>StartCalendarInterval</key>
<array>
    <dict>
        <key>Hour</key>
        <integer>8</integer>
        <key>Minute</key>
        <integer>0</integer>
    </dict>
    <dict>
        <key>Hour</key>
        <integer>18</integer>
        <key>Minute</key>
        <integer>0</integer>
    </dict>
</array>
```

5 Verwaltung

5.1 Filter hinzufügen

1. Ordner erstellen (falls nötig):

```
himalaya folder create INBOX.neuer-ordner
```

2. Filter in `email-filters.json` hinzufügen:

```
{
  "name": "Neuer Filter",
  "enabled": true,
  "match": {
    "type": "from",
    "pattern": "absender-pattern",
    "ignoreCase": true
  },
  "action": {
    "type": "move",
    "folder": "INBOX.neuer-ordner"
  }
}
```

3. Testen:

```
emailfilter
```

5.2 Filter deaktivieren

Setzen Sie `ënabled": false` in der JSON-Datei.

5.3 Dry-Run (Test-Modus)

In `email-filters.json`:

```
"settings": {
  "dryRun": true
}
```

Das Script zeigt an, was es tun würde, ohne Änderungen vorzunehmen.

5.4 Logs anzeigen

```
# Hauptlog
tail -f ~/.config/himalaya/filter.log

# Fehler
tail -f ~/.config/himalaya/filter-error.log

# Standardausgabe
tail -f ~/.config/himalaya/filter-output.log
```

6 Himalaya Grundbefehle

6.1 Email-Verwaltung

```
# Emails auflisten
himalaya envelope list --page-size 20

# Email lesen
himalaya message read <ID>

# Email verschieben
himalaya message move INBOX.Zielordner <ID>

# Email loeschen
himalaya message move INBOX.Trash <ID>

# Email senden
himalaya message write
```

6.2 Ordner-Verwaltung

```
# Ordner auflisten
himalaya folder list

# Ordner erstellen
```

```
himalaya folder create INBOX.neuer-ordner

# Ordner loeschen
himalaya folder delete INBOX.ordner
```

6.3 Suche

```
# Nach Absender suchen
himalaya envelope list | grep "absender@example.com"

# Nach Betreff suchen
himalaya envelope list | grep -i "wichtig"
```

7 Troubleshooting

7.1 Häufige Probleme

7.1.1 Authentifizierungs-Fehler

Problem: `Authentication failed`
Lösung:

1. Passwort in Keychain prüfen:

```
security find-generic-password -s roar-imap -w
```

2. Bei Bedarf neu setzen:

```
security delete-generic-password -s roar-imap
security add-generic-password -a resa@roar.com \
    -s roar-imap -w "NEUES_PASSWORT"
```

7.1.2 Filter funktionieren nicht

Problem: Emails werden nicht verschoben
Lösung:

1. Dry-Run aktivieren und testen
2. Log-Datei prüfen:

```
tail -50 ~/.config/himalaya/filter.log
```

3. Pattern testen:

```
node -e "console.log(/pattern/i.test('test-string'))"
```

7.1.3 LaunchAgent startet nicht

Problem: Filter laufen nicht automatisch
Lösung:

1. Status prüfen:

```
launchctl list | grep himalaya
```

2. Neu laden:

```
launchctl unload ~/Library/LaunchAgents/com.himalaya.email-filters.
    plist
launchctl load ~/Library/LaunchAgents/com.himalaya.email-filters.
    plist
```

3. Berechtigung prüfen:

```
ls -l ~/Library/LaunchAgents/com.himalaya.email-filters.plist
chmod 644 ~/Library/LaunchAgents/com.himalaya.email-filters.plist
```

7.1.4 Node.js nicht gefunden

Problem: `node: command not found`
Lösung:

1. Node.js-Pfad finden:

```
which node
# z.B. /usr/local/bin/node oder /opt/homebrew/bin/node
```

2. In plist-Datei korrigieren:

```
<string>/opt/homebrew/bin/node</string>
```

7.2 Performance-Optimierung

7.2.1 Viele Emails

Bei großen Mailboxen (>1000 Emails):

- `applyToExisting`: `false` setzen (nur neue Emails)
- Filter nach Priorität sortieren (häufigste zuerst)
- Alte Emails manuell archivieren

7.2.2 Logging reduzieren

In `email-filters.json`:

```
"settings": {
  "logActions": false
}
```

8 Best Practices

8.1 Sicherheit

- **Niemals** Passwörter in Config-Dateien speichern
- Immer macOS Keychain verwenden
- Config-Dateien mit `chmod 600` schützen:

```
chmod 600 ~/.config/himalaya/config.toml
```

- Regelmäßige Backups der Filterregeln

8.2 Filter-Design

- **Spezifisch vor generisch**: Spezielle Filter zuerst
- **Testen**: Neue Filter im Dry-Run testen
- **Dokumentieren**: Aussagekräftige Namen verwenden
- **Regex-Vorsicht**: Einfache Patterns bevorzugen

8.3 Wartung

- Monatlich: Log-Dateien rotieren/löschen
- Quartalsweise: Filterregeln überprüfen und aufräumen
- Bei Problemen: Logs konsultieren vor Änderungen

9 Erweiterte Konfiguration

9.1 Mehrere Accounts

Himalaya unterstützt mehrere Email-Accounts:

```
[accounts.roar]
default = true
email = "resa@roar.com"
# ... config ...

[accounts.privat]
email = "privat@example.com"
# ... config ...
```

Account wechseln:

```
himalaya -a privat envelope list
```

9.2 Custom Folder-Mapping

Weitere Ordner in `config.toml` mappen:

```
folder.alias.archive = "INBOX.Archive"
folder.alias.important = "INBOX.Important"
```

9.3 Email-Vorlagen

Vorlagen in `~/.config/himalaya/templates/`:

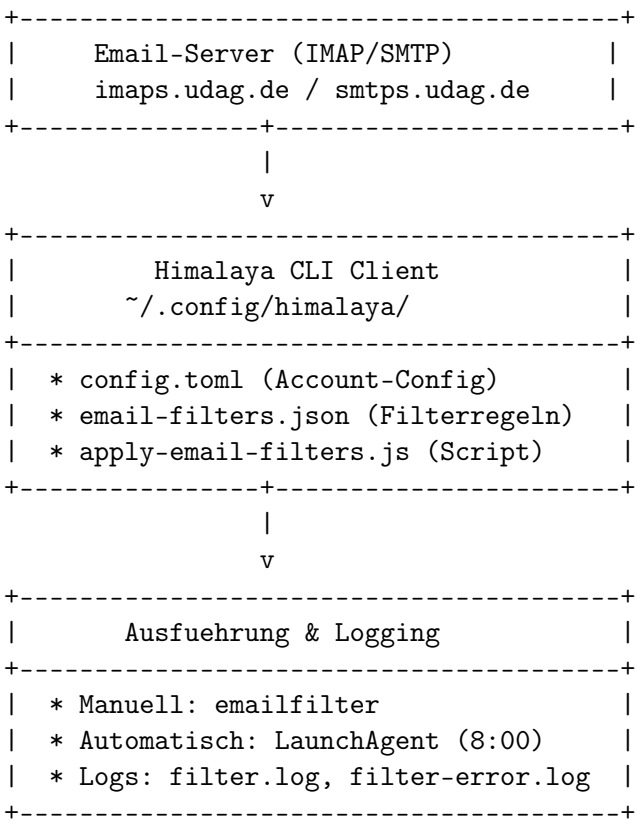
```
# template.eml
From: resa@roar.com
To: empfaenger@example.com
Subject: Betreff

Nachrichtentext
```

Verwenden:

```
himalaya message send < ~/.config/himalaya/templates/template.eml
```

10 Zusammenfassung

10.1 Systemarchitektur



10.2 Wichtige Dateien

Datei	Zweck
`~/.config/himalaya/config.toml`	Himalaya-Konfiguration
`~/.config/himalaya/email-filters.json`	Filterregeln (27)
`~/.config/himalaya/apply-email-filters.js`	Filter-Script
`~/Library/LaunchAgents/*.plist`	Automatisierung
`~/.config/himalaya/filter.log`	Haupt-Log
`~/.zshrc`	Shell-Alias

10.3 Statistik

- **27 Filterregeln**: 10 Lösch-, 1 Junk-, 16 Sortier-Filter
- **17 Email-Ordner**: Automatische Kategorisierung
- **Ausführung**: Manuell via `emailfilter` oder täglich um 8:00 Uhr
- **Sicherheit**: Passwörter in macOS Keychain

11 Schnellreferenz

11.1 Wichtigste Befehle

```
# Filter ausfuehren
emailfilter

# Emails anzeigen
himalaya envelope list

# Ordner anzeigen
himalaya folder list

# Logs anzeigen
tail -f ~/.config/himalaya/filter.log

# LaunchAgent Status
launchctl list | grep himalaya

# Filter bearbeiten
vim ~/.config/himalaya/email-filters.json
```

11.2 Nützliche Aliases

```
# In ~/.zshrc
alias emailfilter='node ~/.config/himalaya/apply-email-filters.js'
alias emaillist='himalaya envelope list --page-size 20'
alias emaillog='tail -f ~/.config/himalaya/filter.log'
alias emailconf='vim ~/.config/himalaya/email-filters.json'
```

12 Changelog

12.1 Version 1.0 (Januar 2026)

- Initiales Setup mit Himalaya 1.1.0
- 27 Filterregeln implementiert
- Node.js-basiertes Filter-Script
- LaunchAgent für Automatisierung
- macOS Keychain Integration
- Vollständige Dokumentation

13 Support und Ressourcen

13.1 Offizielle Dokumentation

- Himalaya: https://pimalaya.org/himalaya/
- GitHub: https://github.com/soywod/himalaya
- LaunchAgent: `man launchd.plist`

13.2 Weitere Hilfe

Bei Problemen oder Fragen:

1. Log-Dateien konsultieren
2. Dry-Run-Modus zum Testen nutzen
3. Konfiguration gegen Beispiele prüfen
4. GitHub Issues durchsuchen