

☐☐ Kompletter Einsteiger-Guide für Hermes Agent (hermes-ai.net)

Was ist Hermes Agent?

Hermes Agent ist ein Open-Source-KI-Agent von **Nous Research** (MIT-Lizenz). Er ist weder ein reiner Coding-Copilot noch ein simpler Chatbot, sondern ein **autonomer, selbst-verbessernder Agent**, der mit der Zeit klüger wird.

“ i Wichtig zu wissen: hermes-ai.net ist eine **inoffizielle, unabhängige Community-Seite** mit lokalisierten Docs (u.a. auf Deutsch). Die offizielle Software und Dokumentation kommen von Nous Research (hermes-agent.nousresearch.com bzw. GitHub).

Kernmerkmale auf einen Blick

Feature	Beschreibung
☐☐ Selbstlernender Loop	Erstellt Skills aus Erfahrung, verbessert sie im Betrieb, baut ein Modell von dir über Sessions hinweg
☐☐ 15+ Messaging-Plattformen	Telegram, Discord, Slack, WhatsApp, Signal, Matrix, E-Mail, SMS u.v.m.
☐☐ 47 eingebaute Tools	Websuche, Dateioperationen, Terminal-Befehle
☐☐ MCP-Integration	Bindet beliebige Model-Context-Protocol-Server ein
☐ Cron-Automationen	Geplante Aufgaben, Ergebnis an jeden Kanal
☐☐ 6 Runtime-Backends	local, Docker, SSH, Daytona, Singularity, Modal

⚡ Schnellstart – in ~60 Sekunden startklar

Voraussetzungen

- **Betriebssystem:** Linux, macOS oder **WSL2** unter Windows
- **Modell:** Ein LLM mit **großem Kontextfenster** empfohlen
- **API-Key:** Bereithalten für deinen Anbieter (oder ein OpenAI-kompatibler Endpoint)

Schritt 1 – Installation (Linux / macOS / WSL2)

```
# Installer herunterladen
curl -fsSL0 https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh

# Skript ANSCHAUEN, bevor du es ausführst (Sicherheits-Best-Practice!)
less install.sh

# Ausführen
bash install.sh
```

Danach Shell neu laden:

```
source ~/.bashrc    # oder: source ~/.zshrc
```

🗨️ **Windows-Nutzer:** Der empfohlene Weg ist **WSL2**. Installiere WSL2, öffne ein Ubuntu/WSL-Terminal und führe dort den Linux-Befehl aus. Das erspart dir viele Abhängigkeits- und Automatisierungsprobleme.

Schritt 2 – Modell konfigurieren

```
hermes model    # LLM-Anbieter und Modell wählen
hermes tools    # aktivierte Tools einstellen
hermes setup    # kompletter Setup-Assistent
```

Bei `hermes model` wählst du die Verbindungsart:

- **OAuth-Login**
- **API-Key**
- **Custom OpenAI-kompatibler Endpoint** (Base-URL, API-Key, Modellname, Kontextfenster eingeben)

☐ Du kannst `hermes model` jederzeit erneut ausführen, um das Modell zu wechseln.

Schritt 3 – Loslegen und Chatten

```
hermes
```

Es erscheint ein Willkommens-Banner mit gewähltem Modell, Tools und Skills. Dann einfach tippen:

```
> Was kannst du gerade für mich tun?
```

☐ Die wichtigsten Workflows

Hermes das Terminal nutzen lassen

```
> Prüfe die Festplattennutzung und liste die fünf größten Verzeichnisse.
```

Hermes führt die Befehle aus, erklärt die Ergebnisse und untersucht bei Bedarf weiter.

Slash-Befehle (im Chat `/` tippen)

Befehl	Funktion
<code>/help</code>	Alle Befehle anzeigen
<code>/tools</code>	Verfügbare Tools auflisten
<code>/model</code>	Modell interaktiv wechseln

Befehl	Funktion
<code>/skills</code>	Skills anzeigen/verwalten
<code>/save</code>	Aktuelle Konversation speichern

Nützliche Tastenkürzel

- **Mehrzeilige Eingabe:** `Alt+Enter` oder `Ctrl+J` (für Code, Logs, lange Prompts)
- **Aufgabe unterbrechen:** Einfach neue Nachricht senden oder `Ctrl+C`
- **Session fortsetzen:**

```
hermes --continue    # letzte Session fortsetzen
hermes -c            # Kurzform
```

Erweiterte Nutzung

Sandboxed Terminal (mehr Isolation)

```
hermes config set terminal.backend docker
hermes config set terminal.backend ssh
```

Messaging-Plattformen anbinden

```
hermes gateway setup
```

Danach erreichst du Hermes über Telegram, Discord, WhatsApp, Signal, E-Mail, Home Assistant u.v.m.

Sprachmodus (Voice)

```
pip install "hermes-agent[voice]"
pip install faster-whisper    # optional: lokale Spracherkennung
```

Im CLI aktivieren:

```
/voice on
```

- `Ctrl+B` zum Aufnehmen
- `/voice tts` lässt Hermes Antworten vorlesen

Skills durchsuchen & installieren

```
hermes skills search kubernetes
hermes skills search react --source skills-sh
hermes skills install openai/skills/k8s
```

⚠ **Sicherheit:** Prüfe Skills von Dritten immer vor der Installation – du siehst so, welche Tools und Daten sie nutzen. Community-Skills findest du auf **agentskills.io**.

MCP-Server verbinden

Bearbeite `~/hermes/config.yaml`:

```
mcp_servers:
  github:
    command: npx
    args: ["-y", "@modelcontextprotocol/server-github"]
    env:
      GITHUB_PERSONAL_ACCESS_TOKEN: "ghp_xxx"
```

In Editoren nutzen (ACP-Server)

```
pip install -e '[acp]'
hermes acp
```

☐ Befehls-Referenz (Cheat Sheet)

Befehl	Beschreibung
--------	--------------

<code>hermes</code>	Chat starten
<code>hermes model</code>	LLM-Anbieter/Modell wählen
<code>hermes tools</code>	Tools konfigurieren
<code>hermes setup</code>	Kompletter Setup-Assistent
<code>hermes doctor</code>	Installation/Konfiguration diagnostizieren
<code>hermes update</code>	Auf neueste Version aktualisieren
<code>hermes gateway</code>	Messaging-Gateway starten/konfigurieren
<code>hermes --continue</code>	Vorherige Session fortsetzen

Empfohlene Reihenfolge zum Einstieg

1. **Installieren** → `install.sh` ausführen
2. **Modell einrichten** → `hermes setup` durchlaufen
3. **Ersten Chat starten** → `hermes` und eine echte Aufgabe stellen
4. **Slash-Befehle erkunden** → mit `/help` beginnen
5. **Bei Problemen** → `hermes doctor` ausführen
6. **Nächste Ebene** → Gateway (Messaging), Skills und MCP-Server ergänzen

Nützliche Links

- **Community-Guide (Deutsch):** `hermes-ai.net/de/docs/quickstart/`
- **Offizielle Docs:** `hermes-agent.nousresearch.com/docs`
- **GitHub (101K+ Stars):** `github.com/NousResearch/hermes-agent`
- **Community-Skills:** `agentskills.io`
- **Reddit:** `r/hermesagent`