Ollama auf Windows 11 installieren

Lokale KI für die Lagebild-Narration · Standardmodell mistral-nemo · ~15 Minuten plus Modell-Download · keine Admin-Rechte nötig

1 Warum lokal?

Ollama betreibt Sprachmodelle vollständig auf dem eigenen Rechner. Für die KI-Narration des Lagebilds heißt das: Das Delta-Briefing verlässt nie das System, es braucht kein Konto, keinen API-Schlüssel und keine Konzern-Freigabe für externe Dienste. In Kennzeichnung und Betreiber-Nachweis erscheint dieser Weg als deployment_class „local“ — solange die Adresse auf diesen Rechner zeigt (localhost, 127.0.0.1, ::1). Zeigt sie auf einen anderen Rechner, heißt die Klasse „remote“: Die Daten verlassen diesen Rechner, und ob das eigene Netz oder das Internet dahinter liegt, kann das Werkzeug nicht wissen. Der Nachweis hält die Adresse als SHA-256-Hash fest.

2 Voraussetzungen

• Windows 11, 64-bit; keine Administratorrechte nötig (Ollama installiert sich ins Benutzerprofil).

• Plattenplatz: ~4 GB für Ollama selbst plus Modell — mistral-nemo lädt rund 7 GB, mistral rund 4 GB.

• Arbeitsspeicher: 16 GB empfohlen für mistral-nemo (12B); mit 8 GB das kleinere mistral (7B) wählen.

• Ohne Grafikkarte läuft alles auf der CPU — eine Narration dauert dann bis zu einer Minute; eine NVIDIA- oder AMD-GPU beschleunigt spürbar und wird automatisch genutzt.

3 Installieren

• Im Browser ollama.com/download/windows öffnen und OllamaSetup.exe herunterladen.

• Die Datei per Doppelklick starten und dem Assistenten folgen — mehr Auswahl gibt es nicht.

• Danach läuft Ollama im Hintergrund; im Infobereich der Taskleiste erscheint das Lama-Symbol. Nach einem Neustart startet es automatisch mit.

Prüfen, dass die Installation angekommen ist — PowerShell oder Eingabeaufforderung öffnen (Windows-Taste, „powershell“ tippen):

```
ollama --version
```

4 Modell laden

Das Standardmodell der Narration ist mistral-nemo (12B, gutes Deutsch, Management-Ton). Der Download läuft einmalig:

```
ollama pull mistral-nemo

# Alternative fuer Rechner mit 8 GB RAM / for 8 GB machines:
ollama pull mistral
```

Kurztest direkt in Ollama — die erste Antwort dauert am längsten, weil das Modell in den Speicher geladen wird:

```
ollama run mistral-nemo "Antworte mit einem Satz: Was ist ein Lagebild?"
# Chat beenden / leave the chat: /bye
```

5 Verkabelungs-Check mit SituationReport

Im Repository- bzw. Release-Ordner zeigt die providers-Liste die entdeckten KI-Anbieter, und der test-Befehl macht eine erste gekennzeichnete Probe-Narration über Ollama:

```
python -m llm providers
python -m llm models
python -m llm test
python -m llm test --lang en --model mistral
```

6 Modell und Adresse waehlen

Liegen mehrere Modelle in Ollama, entscheidet der Name, welches benutzt wird. Der Befehl models listet, was tatsaechlich installiert ist, und markiert die Voreinstellung. Der Name muss genau so geschrieben werden, wie er dort steht — mit Tag, also qwen3.8:latest statt qwen3.8. Kennt Ollama den Namen nicht, sagt SituationReport das mit dem passenden ollama-pull-Befehl.

```
# Installierte Modelle auflisten / list installed models:
python -m llm models

# Ein bestimmtes Modell benutzen / use a specific model:
python -m llm test --model qwen3.8:latest

# Ollama auf einem anderen Rechner / Ollama on another host:
python -m llm models --base-url http://gpu-box:11434
```

In der Oberflaeche stehen neben dem Anbieter die Felder Modell und Adresse. Das Modellfeld schlaegt die installierten Modelle vor, laesst sich aber auch frei beschriften — ein Modell, das erst noch geladen wird, kann so schon eingetragen werden. Leer bedeutet: der Anbieter entscheidet. Beide Felder werden gemerkt und stehen beim naechsten Start wieder da; sie gehoeren zu diesem Rechner und wandern deshalb NICHT in die gespeicherte Solution-Konfiguration, die man an Kolleginnen und Kollegen weitergibt.

Laeuft Ollama nicht auf diesem Rechner, traegt man die Adresse ein — im Feld Adresse, per --base-url bzw. --llm-base-url, oder einmalig ueber die Umgebungsvariable OLLAMA_HOST, die Ollama selbst benutzt. Ohne Angabe gilt http://localhost:11434.

7 Benutzen

CLI: --narrate ergänzt das Delta-Briefing um den Abschnitt „Narration (Entwurf)“ — inklusive KI-Kennzeichnung, Zahlen-Wächter und Betreiber-Nachweis (llm_audit.jsonl neben der Ausgabe). GUI: im Solutions-&-Portfolios-Fenster die Checkbox „KI-Narration (Entwurf)“ anhaken und daneben den Provider wählen, dann wie gewohnt „Delta-Briefing …“:

```
python -m portfolio --delta prev.json now.json --narrate --output delta.html --browser

# Anderes Modell / different model:
python -m portfolio --delta prev.json now.json --narrate --llm-model mistral --output delta.html
```

8 Wenn etwas hakt

• „Python wurde nicht gefunden“ (Hinweis auf den Microsoft Store): Windows leitet den Befehl python auf einen Store-Platzhalter um. Die Befehle aus Abschnitt 5 bis 7 im Repo-Ordner ausführen und dort .venv\Scripts\python.exe statt python verwenden (oder einmalig .venv\Scripts\Activate.ps1) — alternativ mit installiertem Python-Launcher py -m llm test, oder die App-Ausführungsaliase python.exe/python3.exe deaktivieren (Einstellungen > Apps > Erweiterte App-Einstellungen).

• „Could not reach Ollama“ / Verbindung abgelehnt: Ollama ist nicht gestartet — über das Startmenü „Ollama“ öffnen und auf das Lama-Symbol im Infobereich warten.

• „Ollama does not know model …“: das Modell fehlt noch — ollama pull mistral-nemo ausführen (Abschnitt 4).

• Antwort dauert lange: auf reiner CPU ist bis zu einer Minute je Narration normal; mistral statt mistral-nemo halbiert die Wartezeit ungefähr.

• Platz auf C: knapp — Modellordner verlegen:

```
# Modelle auf D: statt C: ablegen / store models on D:
setx OLLAMA_MODELS "D:\ollama-models"
# danach Ollama beenden und neu starten / then restart Ollama
```

• Beenden: Rechtsklick auf das Lama-Symbol im Infobereich → „Quit Ollama“.

9 Datenschutz in einem Satz

Ollama lauscht nur auf dem eigenen Rechner (localhost:11434); weder Briefing noch Narration verlassen das System — der Unterschied zum externen Weg (Claude-API) bleibt über die deployment_class in Banner und Audit jederzeit sichtbar.