Spickzettel: die Werkzeuge deines Brains

Deine KI ruft diese Werkzeuge selbst auf. Du kannst sie aber auch mit Namen verlangen, zum Beispiel: „Nutze smart_recall und suche nach dem Deploy."

Werkzeug	Wofür	Beispielaufruf
session_start	Briefing zu Beginn einer Sitzung: letzter Stand, neue Lektionen.	`session_start(focus="Login-Fehler")`
smart_recall	Wissen zu einer Aufgabe suchen (nach Bedeutung, nicht nach Wortlaut).	`smart_recall(query="Login-Schleife beheben")`
remember_context	Eine Tatsache unter einem Namen merken.	`remember_context(key="auth_architecture", content="Sessions laufen über Keycloak")`
recall_context	Eine gemerkte Tatsache per Name zurückholen.	`recall_context(key="auth_architecture")`
learn_from_attempts	Nach einem Fix festhalten, was klappte und was nicht.	`learn_from_attempts(topic="deploy:api", outcome="success", what_worked="...")`
causal_trace	Bei einem Fehler: bekannte Ursachen und Lösungen suchen.	`causal_trace(problem="Login bricht nach Neustart ab")`
recall_best_solution	Die beste bekannte Lösung zu einem Thema holen.	`recall_best_solution(topic="deploy:api")`
brain_predict	Vor einem riskanten Schritt: was früher dabei schiefging.	`brain_predict(context="Keycloak von 21 auf 24 heben")`
brain_from_git	Das Brain aus der Git-Historie füllen.	`brain_from_git(repo_path=".", limit=100)`
brain_doctor	Prüft Schlüssel, Brain und Verbindung.	`brain_doctor()`
forget_context	Gemerkte Einträge löschen (Namen, auch mit *).	`forget_context(keys=["wip:*"])`
session_end	Am Ende der Sitzung zusammenfassen, was getan wurde.	`session_end(summary="Login-Fehler behoben", files_changed=["src/auth.ts"])`

Einrichten in einer Zeile

```
Claude Code:  /plugin marketplace add cachly-dev/cachly-mcp
              /plugin install cachly-brain@cachly
Andere:       npx @cachly-dev/mcp-server@latest autopilot
Prüfen:      npx @cachly-dev/mcp-server@latest doctor
```