

CLAUDE.md-Vorlage und Auftragsmuster

Zwei Vorlagen zum Übernehmen: die Projektdatei und zehn Aufträge.

Teil 1 — CLAUDE.md zum Ausfüllen

Ersetz die Platzhalter in eckigen Klammern. Streich, was auf dich nicht zutrifft.

[Projektname]

[Zwei bis drei Sätze: Was tut dieses Projekt, für wen?]

Technik

- Sprache/Framework: [z. B. PHP 8.3 / Laravel 11]
- Datenbank: [z. B. MySQL]
- Frontend: [z. B. Blade + Tailwind]

Befehle

- Starten: `[Befehl]`
- Tests: `[Befehl]`
- Abhängigkeiten installieren: `[Befehl]`

Regeln

- Antworte auf Deutsch. Kommentare auf Deutsch.
- Führe nach jeder Änderung die Tests aus und melde, wenn etwas rot wird.
- Zeig bei allem, was mehr als eine Datei berührt, zuerst einen Plan.
- Keine neuen Abhängigkeiten ohne Rückfrage.
- [eigene Regel]

Verbote

- `[Datei/Ordner]` wird nie geändert.
- Keine Änderungen an der Datenbankstruktur ohne ausdrücklichen Auftrag.
- [eigenes Verbot]

Fallen in diesem Projekt

<!-- Hier kommt jedes Mal eine Zeile rein, wenn etwas schiefgegangen ist. -->
<!-- Beispiel: Preise stehen in Cent, nicht in Euro. -->

Richtwert: unter 200 Zeilen. Wenn du etwas ergänzt, streich etwas anderes.

Teil 2 — Zehn Auftragsmuster

Verstehen

1 — Projektüberblick > Verschaff dir einen Überblick über dieses Projekt und erklär mir in einfachen Worten, > was es tut, wie es aufgebaut ist und wo die wichtigsten Dateien liegen.

2 — Ablauf nachvollziehen > Was passiert, wenn jemand [Aktion] auslöst? Geh den Weg Schritt für Schritt durch > und nenn mir die beteiligten Dateien.

3 — Einstiegspunkt finden > Wo würde ich anfangen, wenn ich [Funktion] hinzufügen wollte? Noch nichts ändern.

Planen

4 — Plan anfordern > Ich möchte [Ziel]. Zeig mir zuerst einen Plan: welche Dateien, welche Schritte, > in welcher Reihenfolge. Ändere noch nichts.

5 — Plan korrigieren > Schritt [Nummer] brauchen wir nicht, [Begründung]. Und bevor du Schritt [Nummer] > machst, schreib bitte einen Test dafür.

Bauen

6 — Tests zuerst > Schreib zuerst Tests für Folgendes: [Fall 1]. [Fall 2]. [Fall 3]. > Zeig mir die Tests, bevor du implementierst.

7 — Umsetzen mit Grenzen > Setz [Funktion] um. Nicht anfassen: [Dateien/Bereiche]. > Nach der Änderung Tests laufen lassen und mir das Ergebnis zeigen.

Aufräumen

8 — Zurück > Verwirf alle Änderungen seit dem letzten Commit.

9 — Sichern > Ist der Arbeitsstand sauber? Wenn nicht, commit alles Offene mit einer passenden Nachricht.

Festhalten

10 — Skill erzeugen > Hier sind meine letzten Erklärungen zum Thema [Thema]: [Text]. > Fass das zu einer wiederverwendbaren Anleitung zusammen, die du künftig befolgst.

Drei Formulierungen, die Aufträge schwächen

Vermeiden	Besser
„Mach das besser.“	„Ändere nur [Stelle], weil [Grund]. Der Rest bleibt.“
„Wie füge ich X ein?“ (an den Agenten)	„Füg X ein, so dass Y, ohne Z kaputtzumachen.“
„Räum mal auf.“	„Benenne in allen Dateien [alt] in [neu] um. Sonst nichts.“