deer-workflow

Ein Open-Source-Runtime für dynamische Agent-Workflows in TypeScript, bei dem die Orchestrierung im Code und die semantische Arbeit durch austauschbare Coding Agents erledigt wird.

STERNE	LIZENZ	SPRACHE	VERSION	ANGELEGT	ZULETZT GEÄNDERT
517	MIT	TypeScript	v0.2.0	2026-07-26	2026-08-09

Was es ist

deer-workflow ist eine code-first Implementierung von Graph Engineering: TypeScript definiert die gültigen Ausführungspfade, während Coding Agents die inhaltliche Arbeit innerhalb jedes Knotens übernehmen. Das Projekt ist ein Pilot für DeerFlow 3.0 von ByteDance und steht unter der MIT-Lizenz. Es richtet sich an Entwickler, die Agent-Workflows als reviewbaren Code statt als undurchsichtige Konversation abbilden wollen.

Was es kann

- Workflows aus natürlichsprachlichen Beschreibungen generieren lassen, die als TypeScript-Module ausgegeben werden.
- Verschiedene Coding Agents als Runtime nutzen: OpenAI Codex CLI als Standard, Claude Code und Pi als eingebaute Alternativen.
- Workflows interaktiv mit phasenbewusster TUI oder für Server und CI mit JSONL-Event-Stream über stdout ausführen.
- Parallele Ausführung von Knoten ermöglichen, wie bei der gleichzeitigen Recherche mehrerer Themen.
- Als npm-Paket global installiert werden oder direkt aus dem Quellcode mit Bun gestartet werden.
- Qualitätsprüfungen mit einem einzigen Befehl durchführen.

So fängst du an

Voraussetzungen sind Bun als Runtime und ein angemeldeter Codex CLI-Account; dann lässt sich das Projekt auf zwei Wegen nutzen.

```
bun install --global @deerwork-ai/deer-workflow
deer-workflow create "Beschreibung des gewünschten Workflows" > workflow.ts
deer-workflow run ./workflow.ts --input '{"topics":["Beispiel"]}'
deer-workflow run ./workflow.ts --input '{...}' --print
git clone https://github.com/deerwork-ai/deer-workflow.git
cd deer-workflow && bun install
bun run dev -- --help
bun run check
```

Grenzen und Bedingungen

- Die Runtime erfordert Bun; Node.js wird nicht explizit als unterstützt genannt.
- Für die Standardfunktion ist ein Account bei OpenAI für Codex CLI nötig; alternative Agents müssen separat installiert sein.
- Das Projekt befindet sich in einem frühen Stadium mit Version v0.2.0 und 517 GitHub-Sternen; die API kann sich noch ändern.
- Die semantische Arbeit wird an externe Agent-Runtimes delegiert; ohne diese funktioniert die Workflow-Generierung nicht.
- Die Beispiele (Deep Research, Blog Writer) müssen aus dem Repository geklont werden, bevor sie laufen.

Für wen sich das lohnt

Das Projekt lohnt sich für technische Selbständige und kleine Firmen, die wiederkehrende Agent-Aufgaben als versionierbaren TypeScript-Code abbilden wollen und bereit sind, mit einer jungen Open-Source-Software zu arbeiten. Wer auf No-Code-Oberflächen angewiesen ist, keine TypeScript-Kenntnisse hat oder stabile, langfristig garantierte APIs benötigt, ist hier nicht richtig.



Zusammengestellt von **localdev.ai**
Quelle: **github.com/deerwork-ai/deer-workflow** · Stand 2026-08-09

Grundlage: README.md: Abschnitt 'Why Deer Workflow' — Code-first-Philosophie und austauschbare Agents; README.md: Abschnitt 'Quick Start' — Installationsbefehle, deer-workflow create, deer-workflow run, --print/-p Flag; README.md: Abschnitt 'Set up' — git clone, bun install, bun run dev -- --help; README.md: Abschnitt 'Validate changes' — bun run check; README.md: Abschnitt 'Contribute' — Codex CLI als Stand