Code als Community-Mitglied beitragen

Der Ablauf eines Code-Beitrags zum Exponential-Kernel aus Sicht eines Community-Mitglieds - Fork, Branch, Coding-Standards und Namenspräfixe, Commit-Nachrichten mit Präfix, das Dateimanifest, Tests, das CLA, der Pull…

Source file: `doc/ger-de/hilfe/gemeinschaft/mitwirken/code-beitragen.md` Last changed 2026-11-07
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/community/contributing/contributing-code.md

Code-Beiträge gehen als GitHub-Pull-Requests an das Kernel-Repository se7enxweb/exponential. Die verbindlichen Regeln stehen in `CONTRIBUTING.md`; diese Seite ordnet sie in der Reihenfolge, in der Sie ihnen begegnen.

Inhalt

1. Bevor Sie Code schreiben
2. Während Sie Code schreiben
3. Bevor Sie den Pull Request eröffnen
4. Der Pull Request
5. Code, der kein Kernel-Code ist

Bevor Sie Code schreiben

1. **Suchen Sie nach einem Issue.** Durchsuchen Sie den Tracker; behebt Ihre Änderung ein bekanntes Issue, nennen Sie es. Für eine größere Funktion eröffnen Sie zuerst ein Issue und beschreiben die Idee, damit Sie nichts bauen, das nicht gemergt werden kann. Tun Sie dasselbe, wenn das Verhalten, das Sie ändern möchten, gewollt oder auslegungsfähig sein könnte: Eine solche negative Funktion ist etwas, worauf sich andere Websites verlassen; die Änderung braucht daher eine klare Begründung, funktioniert am besten als Option statt als neuer Standard und kann mehrere Review-Runden brauchen, bevor sie angenommen wird.
2. **Forken und klonen**: `git clone https://github.com/your-username/exponential.git`, dann `composer install` (`CONTRIBUTING.md`, „Getting Started“).
3. **Installieren Sie die Hooks** mit `make hooks`: Der pre-commit-Hook prüft das Manifest, die PHP-Syntax und den Code-Stil; der commit-msg-Hook prüft die Nachricht.
4. **Branch anlegen**: `git checkout -b feature/your-feature`.

Während Sie Code schreiben

Regel	Quelle
Folgen Sie den PSR-Standards und den Coding-Standards des Projekts	`CONTRIBUTING.md`, „Making Changes“
Benennen Sie neuen Code nach der Präfixregel: Neuer Kernel-Code verwendet exp; vorhandene Bezeichner wie `eZContentObject` oder `eZINI` werden nie umbenannt	ebenda

Regel	Quelle
Templates und Stylesheets der Administrationsdesigns folgen Die Administration gestalten: kein `<style>`-Block und kein `style=""`-Attribut, gemeinsame Komponenten und Tokens	ebenda
Fügen Sie Tests hinzu, die ohne Ihre Änderung fehlschlagen und mit ihr bestehen	`CONTRIBUTING.md`, „Code Reviews“
Eine Änderung pro Pull Request: „Eine Korrektur und ein Refactoring sind zwei Pull Requests“	ebenda

Bevor Sie den Pull Request eröffnen

- **Commit-Nachrichten** beginnen mit `Added:` , `Updated:` , `Fixed:` , `Removed:` oder `Renamed:` , zum Beispiel `Fixed: The redirect keeps the query string on the own host` (`CONTRIBUTING.md`).
- **Das Manifest**: „Jeder Commit, der eine Datei hinzufügt, ändert oder entfernt, aktualisiert ihre Zeile in `share/filelist.md5` (`make manifest-fix`)“.
- **Die Tests bestehen**, und die Dokumentation ist aktualisiert, wenn sich Verhalten geändert hat.
- **Ihr CLA liegt** bei 7x **vor** (Das CLA erklärt).

Die Prüfungen, die die CI ausführt, beschreibt Qualitätsprüfungen.

Der Pull Request

1. Pushen Sie Ihren Branch in Ihren Fork und eröffnen Sie einen Pull Request gegen `main`.
2. Füllen Sie die Vorlage aus und kreuzen Sie das CLA-Kästchen an.
3. Rechnen Sie mit Review-Runden: „7x führt deutlich mehr Review-Runden als üblich durch, bevor es einen Commit oder Pull Request annimmt, manchmal Hunderte von Überarbeitungen“ (`CONTRIBUTING.md`, „Code Reviews“).
4. **Squashen und rebasen** Sie Ihren Pull Request auf `main`, solange er offen ist; force-pushen Sie nur in Ihren eigenen Branch.
5. Nach dem Mergen wird die Historie von `main` nie umgeschrieben; eine spätere Korrektur ist ein neuer Pull Request.

Reviews geben und erhalten enthält Ratschläge für die Review-Runden.

Code, der kein Kernel-Code ist

Ihr Code	Wohin er geht	CLA?
Eine Korrektur an einer Erweiterung mit eigenem Repository	Die Pull Requests dieses Repositorys	Fragen Sie den Maintainer; das CLA deckt Code ab, „den 7x“ in das Projekt „aufnehmen soll“
Eine Korrektur an einer Bibliothek des Kernels (`ezc*` -Klassen)	Das Repository des Forks, `se7enxweb/expcomponents-<name>`	Code unter Apache 2.0; siehe Zu expcomponents beitragen

Ihr Code	Wohin er geht	CLA?
Ihre eigene Erweiterung, Ihr Theme oder Seitenpaket	Ihr eigenes Repository, Projects, Packagist	Nein (Erweiterungen und Themes teilen)
Eine Korrektur an einer Bibliothek, von der Exponential abhängt	Das eigene Projekt der Bibliothek	Dessen Regeln

Verwandte Seiten

- Als Community-Mitglied mitwirken
- Code beitragen (Unterstützer)
- Mit Git entwickeln
- Mit Pull Requests zurückgeben
- Umgangsformen bei Pull Requests
- Freie Dokumentation: Diese Seite darf unter der GNU FDL 1.3 frei kopiert, übersetzt und verbessert werden.

> **Werden Sie Exponential-Partner**
>
> Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.
>
> Partner: hier beginnen | Wie Sie sich bewerben

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.