Claude Code で 600 テストケースを書いて得た知見

自己紹介

白石 祐大

- 株式会社よりそう テックリード
- 株式会社BOXIV AI エバンジェリスト
- 元ユニークビジョン株式会社

はじめに

- Claude Code をフル活用してテストが無かったバックエンドのプロジェクトにテストを導入
- 最終的に全自動で全コントローラのテストケースを書けるようになった
- つまづいたポイントと解決方法を紹介

前提

- 技術スタック
  - Cloud Functions for Firebase
  - Cloud SQL
  - TypeScript
  - Prisma
- テストがないプロジェクトに初めてテストを導入する
- 全コントローラに対してテストケースを作成する
- CC は自分でテストを実行して確認できる状態

承認が面倒

公式の DevContainer を導入して承認をスキップ

作業の重複や抜け漏れ

TODO.md で全コントローラを管理
1 回で複数のことを支持しない

- TODO.md の残り実装を進めるコマンド
- 型エラーと linter エラーを修正させるコマンド

CC が途中で作業を勝手に止める

ターミナルで claude コマンドを for 文で 10 回繰り返す

余計な実装が増える

CLAUDE.md に実装方針を書く → 守ってくれないことも多くあんまり意味なかった
linter ルールで CC がやりがちな実装をエラーにする → 効果抜群

- prisma.create の禁止
- Factory 系クラスに追加のメソッド定義を禁止
- 記法の統一
  - user.id! → user.id as string

エラーメッセージに、どのように修正するのかを書く

初手で生成されるコードの品質が低い

型エラーや linter エラーが多い状態で生成され、修正に時間がかかる

- hooks で即時フォーマット
- hooks で即時型エラー通知

まとめ

- 何度も実行すれば作業が終わるようにコマンドを設計する
- ルール < 既存のコード << プロンプト <<<< 静的解析
- 静的解析で自分の代わりに即時フィードバックしてくれる状態を作る

現在のプログラミングは、

- 自動テストループ
- 品質を担保するガード
  が書ければ終わる。

参考: