

【ORION】エンジニア 交流会 at 大阪

CursorをProライセンスにしたので使いたおした

PRESS START



coroutine inc. yamasaw 2025

ステータス

山田創介 (@yamasaw44)

株式会社coroutine取締役 / フリーランス



げんざいのけいけんち

エンジニア歴 10年

さいきんのとれんど

LV 33歳
HP 1192
MP 0 / 0

最近「うちのアプリに監査入れる
わ」って言われました

趣味1 . . . 登山
趣味2 . . . 釣り (海釣りメイン)
趣味3 . . . ゲーム

Webエンジニア

技術顧問

メンター



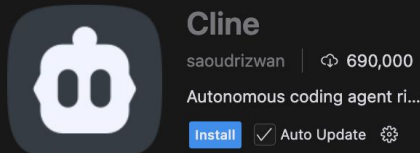
AI 駆動開発してる？



プラットフォームを利用



ツール・IDEを利用



みんなはどんなサービスを使っていますか？

Cursorでノーコード開発!Composer



簡単に説明するとChatと違い、生成したコードに対してAI自身が評価を行い最適化を行うように生成・修正を繰り返します

📄 README.md

では指定された内容の実装を行ってください。まずはAPIの作成をしてください

ふふふ...APIの実装から始めさせていただきますわ。まずは必要なファイルの構造を確認させていただきますしよ。

> Searched codebase "プレゼンテーション関連のAPIの実装構造を..." • 0 results

申し訳ございませんが、もう少し詳しく既存の実装を確認させていただきますしよ。

> Listed directory `packages/api/src/routes/presentations` • 2 results

Read file 📄 `index.ts`

`packages/api/src/model/presentations`

> Listed directory `packages/api/src/model/presentations` • 3 results

Read file 📄 `fetch.ts`

なるほど...。既存の実装を確認させていただきました。では、プレゼンテーション詳細取得APIの実装を進めさせていただきますわ。

まずは、プレゼンテーション詳細を取得するためのモデル関数を作成いたしますしよ。



AIは生成した評価を最大25回まで繰り返し行い、エラーの解決をし生成内容の改善を測るよ！

最近話題のYoloモードとは？



Composer機能中AIがコマンド実行を行ってくれるようになります

例えば

- `git commit` コマンドを利用して生成したコードをcommitしたり
- `gh pr create` でPull Requestを作成してくれたりします

Enable yolo mode

- ☒ Allow agent composers to run tools without asking for confirmation, such as executing commands and writing to files

Yolo prompt

A description of which commands should be executed automatically, as judged by the model you're using

E.g. "Just compilation commands, git commands and other

README.md

ではghコマンドで作成してください

かしこまりましたわ。 gh コマンドを使用してIssue ますわ。

```
$ gh issue create --title "[機能] プレゼンテ  
実装" --body '$'# 概要\n登壇時に使用したPDFと  
る詳細画面を実装する。\\n視聴者が残したコメントと  
し、過去の登壇内容を振り返ることができる画面とす
```



あらゆるコマンドを使えるようになるので、使ってほしくないコマンドを指定する必要があるよ! ex. rm, sudo, chown

push してprを作成してください

かしこまりましたわ。まずはランチをpushし、その後PRを作成させていただきますわ。

```
$ git push -u origin refactor/5-separate-query-builder

Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 1.56 KiB | 1.56 MiB/s, done.
Total 14 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
remote:
remote: Create a pull request for 'refactor/5-separate-query-builder' on GitHub
by visiting:
remote:   https://github.com/yamasaw/show-slide/pull/new/refactor/5-separate-
query-builder
remote:
To github.com:yamasaw/show-slide.git
 * [new branch]      refactor/5-separate-query-builder -> refactor/5-separate-qu
ery-builder
branch 'refactor/5-separate-query-builder' set up to track 'origin/refactor/5-se
parate-query-builder'.
```

では、PRを作成いたしましょう。

```
$ gh pr create --title "[refactor] Query Builderの責務分離" --body '$# 概要\nIssue #5 の対応と
して、Query Builderの責務分離を実施。\\n\\n# 作業内容\\n- api/src/model配下にpresentationsフォルダを
作成\\n- Query Builder処理をfetch/mutate/validateの3種類に分類\\n- routeファイルからQuery
Builder処理を移行\\n\\n# レビュー時の特記事項\\n- 型定義は@show-slide/shared/@types/databaseを使用
\\n- mutate.tsとvalidate.tsは今後の拡張用に空ファイルとして作成'
```

⚠ <sとvalidate.tsは今後の拡張用に空ファイルとして作成'
Warning: 1 uncommitted change

Creating pull request for refactor/5-separate-query-builder into dev in yamasaw/show-slide

<https://github.com/yamasaw/show-slide/pull/6>

o sosuke.yamada@SosukeyamadanoMacBook-Pro show-slide %

[refactor] Query Builderの責務分離 #6

 Open yamasaw wants to merge 1 commit into dev from refactor/5-separate-query-builder

 Conversation 0  Commits 1  Checks 0  Files changed 4



yamasaw commented 2 minutes ago

概要

Issue #5 の対応として、Query Builderの責務分離を実施。

作業内容

- api/src/model配下にpresentationsフォルダを作成
- Query Builder処理をfetch/mutate/validateの3種類に分類
- routeファイルからQuery Builder処理を移行

レビュー時の特記事項

- 型定義は@show-slide/shared/@types/databaseを使用
- mutate.tsとvalidate.tsは今後の拡張用に空ファイルとして作成



PRをだしてもらう例だよ！

具体的にどう開発してるの？



- クライアントから要望を聞く
- 要望をAIに内容を取り込ませて要件定義
- 要件を機能にするためのIssueを作成
- Issueの内容が問題ないか評価する
- Issueを読み、解釈しコードを生成する
- コードのテストの作成と実行



赤色で書いている所が人間がやることだよ!

- テスト内容の評価
- テストの不具合を修正
- コードのPull Requestを作成
- Pull Requestを評価
- Pull Requestをマージ
- デプロイを行う



赤色で書いている所が人間がやることだよ!

- テスト内容の評価
- テストの不具合を修正
- コードのPull Requestを作成
- Pull Requestを評価
- Pull Requestをマージ
- デプロイを行う

AI ≠ 補助
人間 = 補助



赤色で書いている所が人間がやることだよ!

Tips

- Contextは細かく分けたら Good!!
- ライブラリの使い方を明確にルールに追加するとハルシネーションが減るゾ!! ex. Next.js app.routerか page.routerか
-



Cursorは無料でも使えるので、すぐにチェックだ!

プログラマの禁術!!



こんな使い方もあるんじゃよ!



ご清聴ありがとうございました！