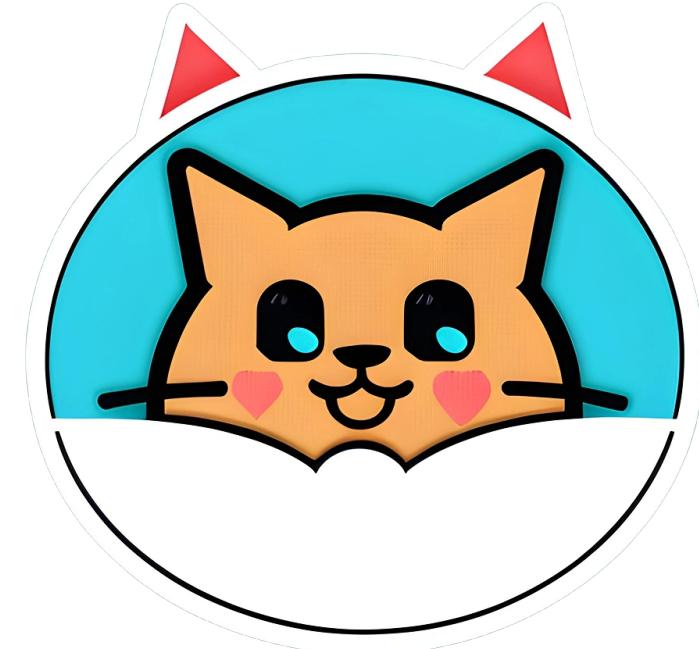


「フロントエンドのテストコード不要じゃない?」に対する答え

自己紹介

- @www_yto(watanabe yuito)
- フロントエンドエンジニア
- SaaS の開発
 - React, Next.js, TypeScript



話したいこと

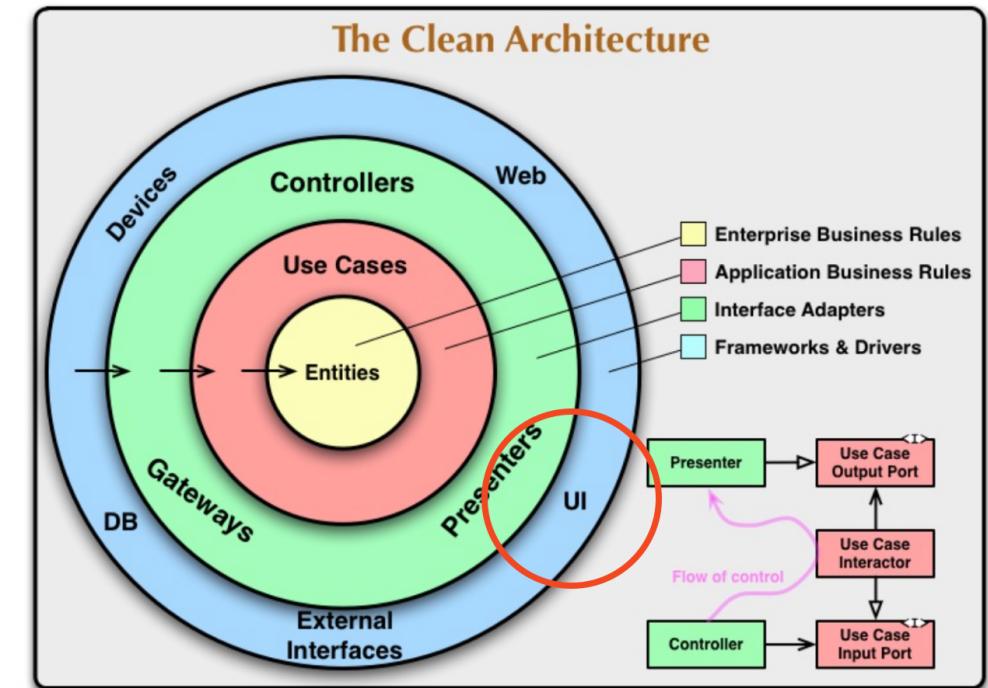
- フロントエンドのテスト不要説
- テストの対象/レベル/種類
- どこに張るか

「フロントエンドのテストコード不要じゃない」に対する答え

フロントエンドのテスト不要説

テスト不要の論理

- フロントエンドはクリーンアーキテクチャで言う所の一番外側
- ドメインがない
- テスト対象(≒ドメイン)が存在しない/ROIが低い



○ テストはドメインに対して書きたい

 フロントエンドにはドメインがない

ドメインとは

知識、影響力、または活動の領域。

ユーザーがプログラムを適用する対象エリアは、ソフトウェアのドメインである。

A sphere of knowledge, influence, or activity. The subject area to which the user applies a program is the domain of the software.

—Eric Evans



> ユーザーがプログラムを適用する対象エリア

- ユーザーが操作するUI
- ユーザーが操作することで発生するデータの変更
- ユーザーが操作することで発生するAPIのリクエスト

- ✖ フロントエンドにはドメインがない
- (エンタープライズ)ビジネスロジックがない
- ドメインはある

フロントエンドのドメイン

ユーザーの操作と状態管理

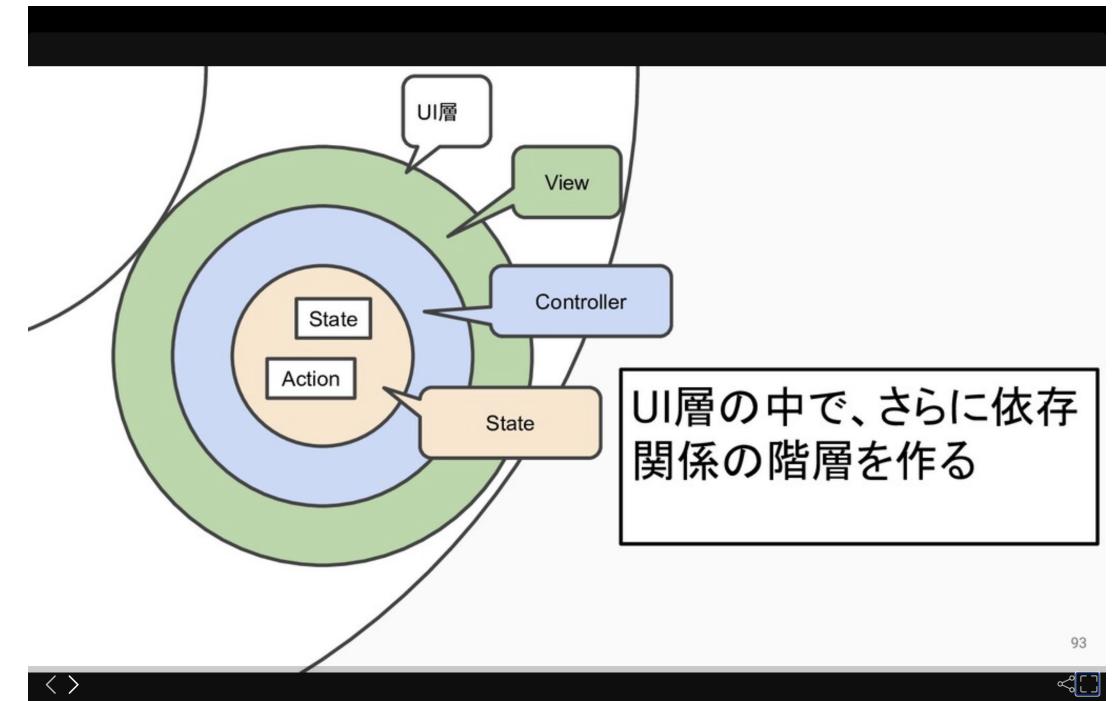
$ui = f(state)$

 dan 2
@dan_abramov2

$ui = f(data)(state)$

午前9:55 · 2024年1月4日 · 4.9万 件の表示

WEBフロントエンドにおけるソフトウェア設計の考察



テストの対象/レベル/種類

テストの対象

1. ライブラリが提供する関数
2. ロジックを担う関数
3. UIを表現する関数
4. Web API クライアント
5. API サーバー
6. DB サーバー

テスト種類

1. 機能テスト
2. 非機能テスト
3. ホワイトボックステスト
4. リグレッションテスト

テストレベル

1. 静的テスト(Static)
2. ユニットテスト(Unit)
3. 結合テスト(Integration)
4. E2E テスト(End to End)

トレードオフ

- テストの信頼度
- 速さ, 壊れやすさ

どこに張るか

フロントエンドのドメイン(再掲)

ユーザーの操作と状態管理

統合テスト

ユーザーの操作によるモジュールの相互作用をテストする

戦略: テスティングトロフィー

<https://www.testingjavascript.com/>

THE FOUR TYPES OF TESTS

End to End

A helper robot that behaves like a user to click around the app and verify that it functions correctly.

Sometimes called “functional testing” or e2e.

Integration

Verify that several units work together in harmony.

Unit

Verify that individual, isolated parts work as expected.

Static

Catch typos and type errors as you write the code.



統合テスト例

1. セレクトボックスの操作する
2. URLの検索クエリが変化する
3. 検索クエリの変化により、データ取得APIが呼ばれる
4. 一覧表示内容が更新される

まとめ

「フロントエンドのテストコード不要じゃない?」

- フロントエンドはドメインがないという説は間違い
- ドメインがあるので、統合テストが必要
- テストの対象/レベル/種類によってテストの張り方が変わる

参考

- WEBフロントエンドにおけるソフトウェア設計の考察
- [The Testing Trophy](#)
- [Clean Architecture](#)
- フロントエンド開発のためのテスト入門 今からでも知りたい自動テスト戦略の必須知識