

TypeScript + Express + Clean ArchitectureでREST APIの開発 環境をつくる

<https://github.com/akihiro-okada/ExpressTypescriptSample>

Express

Node.jsは言わずとしたサーバーサイドJavascriptの実行環境。
ExpressはそのNode.jsでの王道的な開発をスピードアップするためのフレームワークです。

<https://qiita.com/nkjm/items/723990c518acfee6e473>

TypeScript

TypeScript はマイクロソフトによって開発され、メンテナンスされているフリーでオープンソースのプログラミング言語である。

TypeScriptはJavaScriptに対して、省略も可能な静的型付けとクラスベースオブジェクト指向を加えた厳密なスーパーセットとなっている。

<https://ja.wikipedia.org/wiki/TypeScript>

Clean Architecture

- 👉 レイヤー毎に責務を分担するレイヤードアーキテクチャ
- 👉 レイヤー間を疎結合にすることで、メンテナンスしやすくなり、テストも書きやすくなる

<https://qiita.com/nrslib/items/a5f902c4defc83bd46b8>

プロジェクトの作成、設定

👉 Express

👉 TypeScript

👉 Clean Architecture

<https://github.com/sadnessOjisan/ts-clean>

→ 写経させてもらう、MySQLのDBにアクセスする箇所は簡略化

プロジェクト構造

```
app
├─ dist
├─ node_modules
└─ src
    ├─ application
    │   └─ usecase      ・ ・ ユースケース
    ├─ constant         ・ ・ 固定値
    ├─ domain           ・ ・ モデルの管理
    ├─ infrastructure
    │   └─ repository   ・ ・ データを管理
    └─ interface
        ├─ controller   ・ ・ 振り分ける処理
        ├─ request       ・ ・ リクエストデータの定義
        ├─ routes        ・ ・ ルーティング
        └─ serializer    ・ ・ レスポンスを整える
```

デバッグ

👉 以下の記事を参考に

<https://zenn.dev/potepopo/articles/ca93a6bfcca663>

実行コマンド

Get All

```
curl localhost:3000/v1/users
```

Get

```
curl localhost:3000/v1/users/1
```

Create

```
curl -X POST -H "Content-Type: application/json" -d '{"name":"hanako"}' localhost:3000/v1/users
```

Update

```
curl -X PATCH -H "Content-Type: application/json" -d '{"name":"hanao"}' localhost:3000/v1/users/3
```

Delete

```
curl -X DELETE localhost:3000/v1/users/1
```


テストコード

👉 jestを使ってテストを書く

<https://github.com/sadnessOjisan/ts-clean>

Lint + Formatter

👉 eslint ・ ・ JavaScriptコードのエラーチェックを行うLinterと呼ばれるツールの一つ

👉 Prettier ・ ・ ソースコードの整形ツール(コードフォーマッター)の一つ

<https://qiita.com/notakaos/items/85fd2f5c549f247585b1>

まとめ

- 👉 Express + TypeScript + Clean Architectureのコードを動かすところまで作れた
- 👉 テストコード、デバッグ、Lint、Formatterなど、開発環境の構築を作れた
- 👉 初歩的なところで、yarn、npmなどパッケージのバージョン管理にハマった

今後やりたいこと

- ☞ 本番環境設定で動かす

- ☞ PostgreSQLのDBつなぎこみ

- ☞ 環境構築自動化

 - ☞ Dockerなど

- ☞ AWSに載せていく

 - ☞ EC2、RDS