

Web領域のモダンな技術を活用して、
アプリケーションを企画・開発・運用する。

programming project dlopp (dlopp)
発表：前川 大樹

目次

1. 背景と目的
2. 活動内容
3. 学びと今後の展開

1

背景と目的

背景

- ・IT分野の技術は常に速いスピードで変化し続けている
- ・dlopp内で作成したwebアプリケーションにも、それに類する問題が発生していた

目的

- ・古くなったシステムを新しい技術で置き換え、運用コストを削減し、最新技術の恩恵を受けられるようにすること

2

活動内容

1. 作成したアプリケーション
2. プロジェクトの概要と問題点
3. 移行方法
4. 得られた恩恵

1. 作成したアプリケーション



2. プロジェクトの概要と問題点

プロジェクトの概要と問題点

- Nuxt2
- JavaScript
- Firebase
- Heroku

Nuxt 2 に関する問題点

- ・ Nuxt3がパブリックベータになったので、その恩恵を受けたい
- ・ Composition API
- ・ Nitroエンジン
- ・ TypeScriptとの相性の良さ
- ・ useStateで状態管理

TypeScriptとは

- ・ JavaScriptの完全上位互換
- ・ 型定義ができるJavaScript
- ・ null/undefined safe にできる
- ・ エディタによる入力補完が強力

Herokuに関する問題点

- ・ 無料範囲だと、30分アクセスがないとスリープ状態になり、アクセス時の初期描画の時間が長くなる
- ・ もともと検討していたVercelが、orgリポジトリでも無料で利用可能になっていた

Nuxt 2 から 3 に
JavaScriptからTypeScriptに
HerokuからVercelに
移行することにした

3. 移行方法

Nuxt 2 から 3 に移行する方法

- ・ 公式から、「Nuxt Bridge」が提供されている
- ・ 大幅な書き直しやリスクのある変更をせずに、少しずつリプレースできるため、こちらを採用

JavaScriptをTypeScriptに移行する方法

- ・ tsconfigでarrowJs: trueと設定すると、JavaScriptとTypeScriptを両立させることができる
- ・ 共通部分から少しずつ変更していくことが可能

HerokuからVercelに移行する方法

- ・ Herokuで動作しているアプリを消してしまうと、ツイート済みのOGPが消えてしまうので、残す共存させる
- ・ Firebaseとの連携周りで設定ファイルを書く必要がある

3

学びと今後の展開

1. 得られた恩恵
2. 今後の展開

得られた恩恵

- ・ 起動時間の短縮、ホットリロードの高速化
- ・ 入力補完が強力でスペルミスがなくなる
- ・ マウスオーバーで変数の型が確認できる
- ・ 関数の引数や返り値の型が確認できる
- ・ 状態管理が簡単
- ・ 開発効率、保守性の向上

今後の展開

- ・ 完全にNuxt3, TypeScriptに移行させたい
- ・ アプリケーションの規模を拡大させたい
- ・ 今回の活動を外部に発信していき、団体内外での成長サイクルを確立させたい