

# SPAの歴史と Remix SPAモードという到達点

Niigata 5分 Tech #4

Yukiya Nakagawa a.k.a Nkzn

# 自己紹介

- 中川幸哉 a.k.a Nkzn (なかざん)
- '86世代 🐰 の36歳 (来月37になります)
- 新潟生まれ新潟育ち
- 8歳と4歳の父で妻の夫
- 2011～2021年は農業IT 🌾 の人
- 2021年から株式会社モニクル
  - 資産運用お手伝いサービス「マネイロ」のIT裏方



# SPAとは

- **Single Page Application**というWebアプリケーションの一形態
- [狭義] Webサイトに訪問したときにHTML + CSS + JavaScriptを一度だけダウンロードして、その後そのWebサイト内にはいる間は、バックグラウンドで簡単な通信 (Ajax) は行うものの、画面をリロードせずに動き続けるやつ
- システム上の特性としては、Webサイトの訪問によってメモリ上にインストールを行い、Webサイトから出ていくとアンインストールされるアプリ、みたいな抽象度で捉えてもいい

# SPAで大抵必要になるもの

- インフラ
  - 静的ファイルの配信サーバー
- ビルド
  - 記述したJavaScript文法を動作するJavaScript文法に変換する**トランスパイラー**
  - JavaScriptファイルをモジュールとしてモジュール間やライブラリ間の依存性解決を行い、適切な粒度のファイルにまとめる**バンドラー**
- 特殊な役割のライブラリ
  - バックグラウンド通信
    - だいたいfetchでいい
  - 画面の動的書き換え
    - 今回の文脈ではReact
  - 画面の動的書き換えとHistory（アドレスバー&履歴）を同期させる**画面遷移ライブラリ**
- サーバーサイドレンダリング(SSR)
  - [狭義]外部からのアクセスに対し適切なHTMLを返せる仕組み

# React向けSPAツールチェーンの歴史

# ビルド職人の朝は早い時代

トランスパイラー	バンドラー	画面遷移	SSR
Babel	webpack	React Router等	Express + JSDOM

- 一通り自分で組んでいました
- .babelrcとかwebpack.config.jsとかを頑張って書いてた
- Babel筋やwebpack筋という言葉が生まれた
  - 英語圏ではbabel-fuとかwebpack-fu (クンフーに由来)
- SSRは気合いで小さなサーバーサイドアプリケーションを書く


# Create React App時代

トランスパイラー	バンドラー	画面遷移	SSR
(Babel) 隠蔽	(webpack) 隠蔽	React Router等	Express + JSDOM

- Create React Appという小さなフレームワークが登場
- Reactが動いてビルドしてくれる
- 画面遷移は自分で追加する必要あり
- 設定ファイルを隠蔽しすぎてて使いづらかった
- SSRは気合いで小さなサーバーサイドアプリケーションを書く

# Next.js Pages Router時代前期

トランスパイラー	バンドラー	画面遷移	SSR
Babel	webpack	next/router	標準サポート

- フォルダとファイルで画面遷移が組めるようになった
- babelやwebpackの設定に介入するのもそれなりに容易で拡張性 
- SSRを前提としたフレームワークで厳密にはSPAではない
- React Routerがビッグバンリリースでヘイトを貯めていたり、CRAの開発が遅かったりしたことから、SSRの要否とは別の要因で導入が進んだ



# Vite登場

トランスパイラー	バンドラー	画面遷移	SSR
esbuild + Rollup	Rollup	React Router等	Vike (vite-plugin-ssr)

- 粒度としてはCreate React Appに近い
- Vue向けのツールとして誕生したが、他プラットフォームにも対応を進めてくれたので足を向けて眠れない
- 後発だけあってSSR向けのプラグインも出てきて手厚い
- 相変わらず画面遷移は自前で入れる
- Go製のesbuildでの開発時ビルドがくっそ早い

# Next.js Pages Router時代後期

トランスパイラー	バンドラー	画面遷移	SSR
SWC	Turbopack	next/router	標準サポート

- ビルドの速度面の不満をBabelやwebpackでは解決できなかったのでお引越し
- TurbopackはRust製webpack後継なので考え方は割と近いっぽい
- Rust製のSWCでのビルドがくっそ早い
- もちろんSPAではないけど歴史的経緯で代わりに使う人が多い
- next exportでSPAとしての出力もできんこともない

# Remix

トランスパイラー	バンドラー	画面遷移	SSR
(Vite) 隠蔽	(Vite) 隠蔽	(React Router) 隠蔽	標準サポート

- React Routerチームが作ったNext.jsみたいなやつ
- 学習コストを減らすためのアイデアがたくさん入っており、これにインスパイアされる形でApp Routerが生まれている
- 通信と画面遷移を高度に連携していて使い勝手がいい
- Viteを使っているが、remix.config.jsで隠蔽されてて弄りづらい
- これもサーバーが手厚いのでSPAではない

# Next.js App Router

トランスパイラー	バンドラー	画面遷移	SSR
SWC	Turbopack	next/navigation	標準サポート

- Next.jsチームが作ったRemixみたいなやつ
- Meta社Reactチームと組んで、React Server Componentsというサーバーだけで実行するPHPみたいなReact（語弊）の実験場初めての導入事例となる
- 画面遷移や通信周りの考え方がそこそこ変わった
- もうSPAでもなんでもないんだけど惰性で初手で選ぶ人も多い

いまここ

Ima koko

← App Router

Remix SPA →

もう小さくSPAやる方法が  
Viteしかおらん

# SPAやるならVite一択？

- SSRがいない社内システムやBtoB向けのSaaSを作ってるときに、SSRを軸に設計されたフレームワークを使うのはtoo muchなのでSPAで済ませたい
- とはいえFile-system based routingが意外と気持ちいいので、そのためだけに必要以上に重いNext.jsやRemixでプロジェクトを立ち上げがち
- 画面遷移ライブラリが上手いこと統合されたViteがあればなあ

**Remix SPA mode**



# Remix SPAモード (experimental)

トランスパイラー	バンドラー	画面遷移	SSR
Vite	Vite	(React Router) 隠蔽	不要

- 別件でvite.config.jsを露出するモードができた (experimental)
  - このモードではremix.config.jsは使用禁止
- サーバーに頼らずにブラウザだけの範囲で通信と画面遷移を統合させる方法も実装された (v2.5.0 clientLoader/clientAction)
- Vite製SPAの手厚いプラグインとして動作するRemixが生まれた
- SSRしたくなったらSPAモードを外して普通のRemixになればいい

# 通信と画面遷移が統合された Vite管理のSPA

長らく空席だった  
業務用SaaSと相性のいい  
SPAフレームワーク

Remixは流行らなくてもいいけど  
Remix SPAモードは流行ってほしい

完