

3rd party scriptでもReactを使いたい！ Preact + Reactのハイブリッド開発

@2024/11/06 【Qiita Bash】推しフロントエンド技術について語ろう！

齋藤 成之

X: @nakaakist



2020 東京大学理学系研究科博士課程修了

2020-2021 日立製作所

2021-2024 株式会社プレイド

2021-現在 株式会社RightTouch

レーザー物理で博士号取得後、日立製作所でTプラットフォームの設計・開発に従事。

その後、プレイド/RightTouchで、テックリード/フルスタックエンジニアとしてアプリケーション開発に従事。

好きなもの: 旬

RightTouch

あらゆる人を負の体験から解放し、
可能性を引き出す

設立:2021年12月

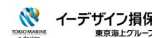
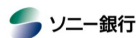
従業員:40名、うちエンジニア15名(2024年8月1日現在)

資本金:10,000,000円(資本準備金含む)

沿革

- | | |
|----------|--|
| 2021年12月 | 株式会社 RightTouch 設立 |
| 2022年3月 | 次世代のコンタクトセンターを創ることに賛同いただいたお客様との実証実験を経て、KARTE RightSupport(β版)をリリース |
| 2023年10月 | Webサイトとコールセンターの分断をなくし、問い合わせ体験を抜本から変革する新プロダクト「RightConnect by KARTE」β版を提供開始 |
| 2023年10月 | RightSupport by KARTEの正式版をリリース |

主な導入企業様



34

18636

Qiita記事数



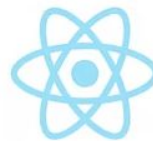
preact

34
記事

19
フォロワー

フォロー

☆: 36k
npm DL: 4M



React

18636
記事

43228
フォロワー

フォロー

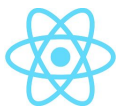
☆: 228k
npm DL: 25M

Preactとは？

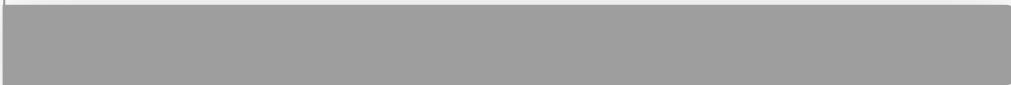
Preact = 超軽量なReact



preact
4.6 kB



react + react-dom
44.4 kB



※gzip後の値。後述のpreact/compatを入れるとpreactのサイズは2 kb程度増える

Reactと似たAPI。preact/compatという補助ライブラリでほぼ Reactのように使える

Preact

```
import { useState } from "preact/compat";

export const Counter = () => {
  const [count, setCount] = useState(0);

  return (
    <button onClick={() => setCount(count + 1)}>
      {count}
    </button>
  );
};
```

React

```
import { useState } from "react";

export const Counter = () => {
  const [count, setCount] = useState(0);

  return (
    <button onClick={() => setCount(count + 1)}>
      {count}
    </button>
  );
};
```

ただし完全互換ではないので、React用のライブラリをそのまま入れるのは避けた方が無難

何に使える？

我々の利用例：カスタマーサポート向けのツール (=サポートアクション) をWebサイトに埋め込み

既存のwebサイト(お客様環境)

Preactで追加したボタン

- すでに存在するお客様のウェブサイト、コーチマークやヘルプアイコンを追加表示できる
- サイトに<script>タグで我々のスクリプト埋めて配布する形 (=3rd party script)
- サイト本体のパフォーマンスへの影響を最小限にするため、**スクリプトの軽量化**が必須
- 2023年末から約1年間本番運用

軽量FWにはSvelte等もあるが、すでにReactが開発のメインだった我々にはPreactが適していた

1

開発生産性の向上

full-Reactにすることで、技術キャッチアップの効率化、コードの共通化により開発生産性が向上

2

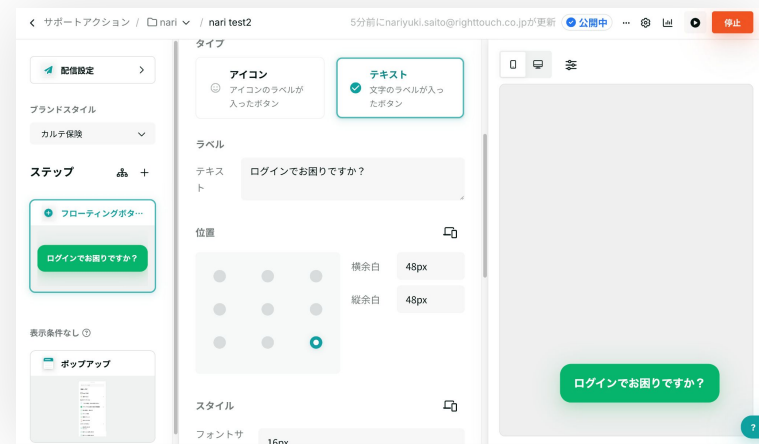
滑らかな編集プレビューの実現

サポートアクションの編集画面上で、編集内容を即時反映するプレビューが簡単に作れる

開発生産性の向上例：コードの共通化

編集画面のプレビューとサイトに配布する script のコードを共通化可能

編集画面 お客様のツール運用担当がコンテンツ編集



編集内容を
サイト側に反映

サイト



編集フォーム

プレビュー



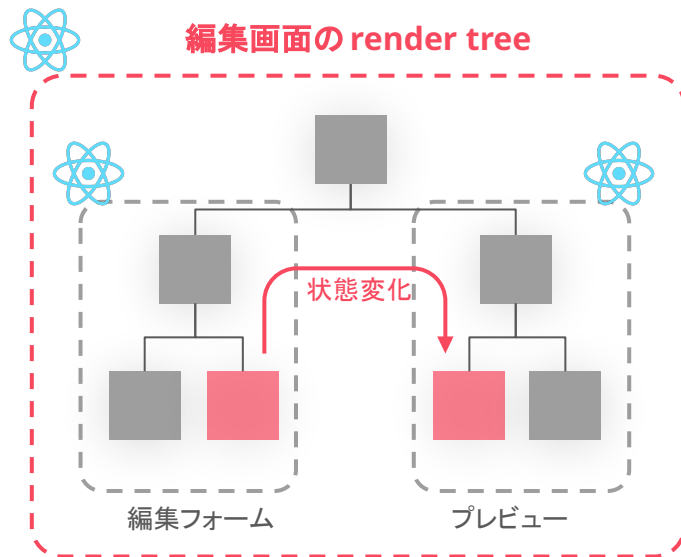
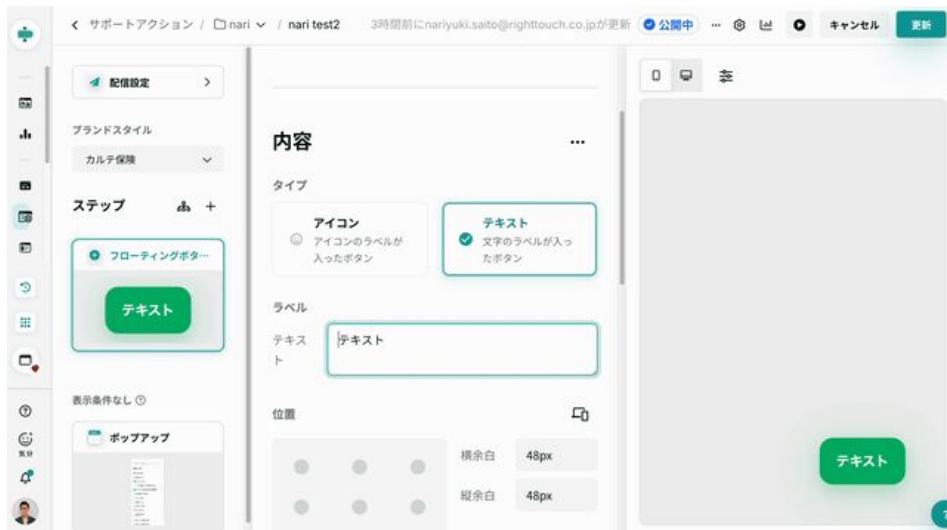
ビルド

ビルド

サポートアクションのコード

滑らかな編集プレビューの実現

編集画面全体を同じReactのrender treeに収めることで、編集内容を即時でプレビュー反映



運用上の工夫

- **React/Preact向けに2つビルドするにあたって互換性の考慮**
 - サポートアクションのコード自体はpreact/compatを前提に書いている。
 - このとき、preact特有の構文が混入すると、react用にビルドしたときにバグる可能性がある。
 - 下記の構文はeslintで禁止
 - “class”、“onChange”属性
 - preact/compatからの直接インポート (“react”に”preact/compat”へのエイリアスをはる)
- **ランタイムでの外部ライブラリ依存はゼロに**
 - バンドルサイズ節約とバグ防止
 - スタイリングにはゼロランタイムCSS in JS(vanilla-extract)を利用
- その他、ビルド時のバンドルサイズのチェックなど

今のところ、Preact起因での障害は起きていない

**サイズ制限の厳しいプロジェクトに
Preactをぜひ使ってみてください！**