弁護士ドットコム

Vueで Webコンポーネントを作って Reactで使う

2024.10.30

クラウドサイン事業本部　辻 佳佑

自己紹介

辻 佳佑 Keisuke Tsuji

フロントエンドエンジニア

弁護士ドットコム株式会社
クラウドサイン事業本部　Product Engineering 部

＜ 趣味 ＞
ボイトレ・生成 AI・服作り（超初心者）・
アンチエイジング

＜ X（Twitter）アカウント ＞
@t0daaay

会社紹介

弁護士ドットコム

VISION

まだないやり方で、世界を前へ。

Drive a paradigm shift for the better world.

MISSION

「プロフェッショナル・テック 」で、次の常識をつくる。

Be the Professional-Tech Company.

プロフェッショナルだからできること。
専門知とテクノロジーで、社会に貢献する。

OUR SERVICE

運営サービス

弁護士ドットコム

日本最大級の無料法律相談ポータルサイト
月間サイト訪問者数約1,600万人
登録弁護士数は27,782人、国内弁護士におけるシェアが61%
売上は、弁護士からの広告掲載・各種サービス利用費用／
有料会員ユーザーからの会員費用

弁護士ドットコム ニュース

時事問題の弁護士解説を中心としたメディア
身近な出来事を高い専門性を保ちつつ、弁護士が法的観点からわかりやすく・やわらかく解説

弁護士ドットコム CAREER

弁護士事務所、企業法務職を中心とした人材紹介事業
弁護士ドットコムの運営によって築いたネットワークによる
独自求人

BUSINESS LAWYERS

最新の法改正や実務について分かりやすく解説する
日本最大級の企業法務ポータルサイト
会員数は約100,000人を突破
「ビジネスロイヤーズライブラリー」「ビジネスロイヤーズコンプライアンス」を中心に収益化

税理士ドットコム

税理士に無料で相談・検索できる日本最大級の
税務相談ポータルサイト
売上は、税理士とユーザーのマッチングによる成功報酬

CLOUDSIGN

契約の締結から管理までデジタルで完結させる
契約マネジメントプラットフォーム
東京海上日動、野村證券、三菱地所、トヨタ等で導入
地方自治体は238自治体のうち157自治体が導入（シェア約70%）

CLOUDSIGN

契約締結から契約書管理まで可能な クラウド型の電子契約 サービス

契約交渉が済んだ完成済の契約書をアップロードし、相手方が承認するだけで契約を締結することが出来ます
書類の受信者はクラウドサインに登録する必要がありません

まえがき

Vue Fes お疲れ様でした！

控えめに言って
最高でした！

メドピアトラック 11:20~11:35

VueとViteで作るUIコンポーネントライブラリ ~デザインシステムとプロダクトの理想的な分離を目指して~

弁護士ドットコム株式会社が提供するサービス、クラウドサインにUIコンポーネントライブラリを導入した事例を紹介します。クラウドサインではプロダクトの拡大に伴い、デザインシステムとプロダクトの関心の分離を行うため、VueとViteを活用したUIコンポーネントライブラリを導入しました。本セッションでは、その背景とVueとViteを活用したライブラリの開発方法、導入プロセスについてお話しします。

弁護士ドットコム株式会社 クラウドサイン事業本部
フロントエンドエンジニア
辻 佳佑

弁護士ドットコム株式会社のクラウドサインでフロントエンドエンジニアをやってます。 ボイトレ, 服作り, アンチエイジング, 生成AIにハマってます。

ここでは

“ Vue コンポーネント ”

ライブラリの話をしました

セッション後に受けた相談 

コンポーネントライブラリ
導入してみたいんですけど ...

弊社のプロダクト
Vue と React が
共存してるんですよねー 🤔

それ 

“ Webコンポーネント ”
で解決できるかも！

目次

目次

- Web コンポーネントとは
- Vue で Web コンポーネントを作る
- まとめ・感想

Webコンポーネントとは

Web コンポーネントとは

Web コンポーネントは、**再利用可能な**
カプセル化されたオリジナルの
HTML タグを作成できる技術。

Web 標準の技術でモダンブラウザでは
サポート済み。

特定のフレームワークに依存しない
標準化されたコンポーネントを作成することができる。

Web コンポーネントの 主要な3技術

カスタム要素	シャドウ DOM	HTML テンプレート

カスタム要素

独自の HTML タグを定義し、

再利用可能なコンポーネント を作成するための仕組み。

これにより、**シンプルで直感的なタグ** として扱うことができる。

カスタム要素の名前には、ハイフンを1つ以上含む必要がある。

```
<hello-world></hello-world>
```

シャドウ DOM

カスタム要素内の要素やスタイルを

カプセル化 するための仕組み。

これにより、HTML や CSS が他の要素に影響を与えたり、

逆に他の要素から影響を受けたりすることがなくなる。

HTML テンプレート

template タグを使用して、非表示のHTML構造を定義し、

後で動的に再利用できる仕組み。

また、**slot タグ**を組み合わせることで、

テンプレート内に柔軟なコンテンツの差し込みが可能になる。

実装例

シンプルな

`hello-world` コンポーネントの

実装例。

この js を読み込むと

`hello-world` タグが有効になる。

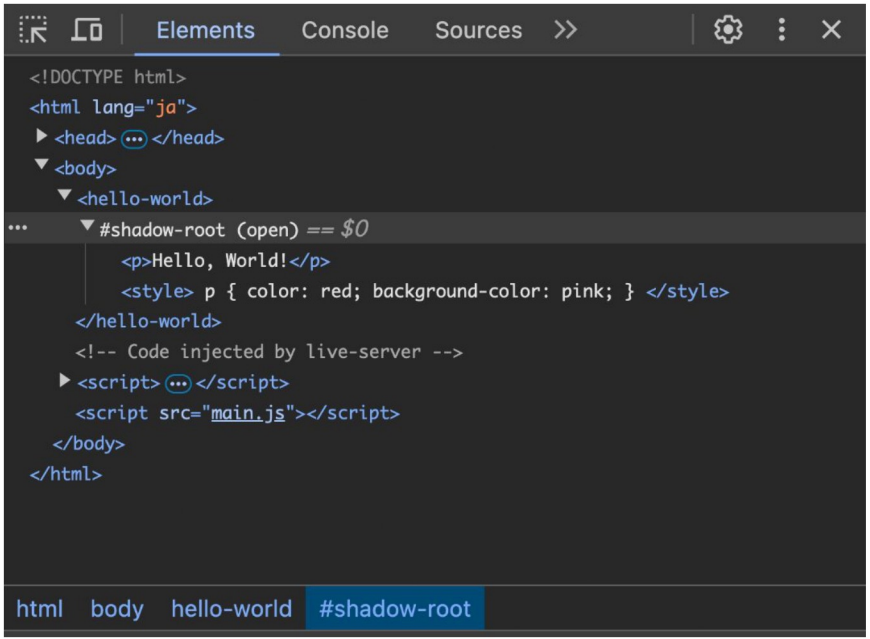
```
class HelloWorld extends HTMLElement {
 constructor() {
   super();
   // シャドウ DOM を作成
   const shadow = this.attachShadow({ mode: "open" });
   // シャドウ DOM 内に HTML と CSS を追加
   shadow.innerHTML = `
   <p>Hello, World!</p>
   <style>
     p {
       color: red;
       background-color: pink;
     }
   </style>
   `;
 }
}

// カスタム要素として 'hello-world' を登録
customElements.define("hello-world", HelloWorld);
```

ブラウザでの表示

表示

Hello, World!

DOM



Vueで
Webコンポーネントを作る

実は Vue の SFC を

“ Web コンポーネント化 ”

できる API がある

defineCustomElement

`defineCustomElement` はWeb コンポーネントを

作成できるAPI（Vue 3.2 で追加）

`defineComponent` と同じ引数を受け取って、

ネイティブのカスタム要素クラスのコンストラクタを返す。

props, emit, slot の扱い

- **props** : 対応するプロパティに反映される
- **emit** : カスタムイベントとして dispatch される
- **slot** : ネイティブなスロット構文と同じ使い方ができる

基本的な使い方

1. 拡張子が `.ce.vue` の SFC を作成し、インポートする
2. `defineCustomElement` でVue の SFC をカスタム要素のクラスのコンストラクタに変換する
3. カスタム要素を登録する

```
import { defineCustomElement } from "vue";
import VueHelloWorld from "./HelloWorld.ce.vue";

// カスタム要素のコンストラクタに変換
const HelloWorld = defineCustomElement(VueHelloWorld);

// カスタム要素として 'hello-world' を登録
customElements.define("hello-world", HelloWorld);
```

フレームワークに依存しない
コンポーネントとして
ライブラリ
で配信するケースが多そう

エントリーポイントの設定例

ライブラリ側の設定。

カスタム要素を登録する関数を用意してライブラリ使用側で呼んでもらうようにする。

また、必要に応じてカスタム要素の型推論ができるように設定する。

```
import { defineCustomElement } from "vue";
import VueHelloWorld from "./components/HelloWorld.ce.vue";

// カスタム要素のコンストラクタに変換
const HelloWorld = defineCustomElement(VueHelloWorld);

// ライブラリ使用側でこれを呼び出して登録 する
export const register = () => {
 customElements.define("hello-world", HelloWorld);
}

// Vue テンプレート で型推論できるように設定
declare module "vue" {
 export interface GlobalComponents {
   HelloWorld: typeof VueHelloWorld;
 }
}

// 必要に応じて JSXなどでも型推論できるように設定
…ここでは省略
```

ライブラリ使用側の実装例

Vue 上で Web コンポーネントを使う例。

カスタム要素登録の関数を呼び出すことで、カスタム要素が設置できるようになる。

```
<script setup lang="ts">
import { register } from "my-library";

// カスタム要素を登録
register();
</script>

<template>
 <hello-world></hello-world>
</template>
```

本題です

Reactで使ってみたい 🙋

実は課題がある ...😢

😢 React の Web コンポーネントの課題

React で Web コンポーネントを直接使う上では以下の制限がある。

- 複雑なデータ（オブジェクト、配列、関数など）をHTML属性で
  Web コンポーネントに渡そうとすると、
  [object Object] のように文字列化された値が渡されてしまう

- Web コンポーネントの DOM イベントを宣言的にリッスンできないため、
  `ref` と `addEventListener` を使った命令的な実装が必要になる

解決策

👉 ラッパーコンポーネント

ラッパーコンポーネントとは

前述の課題を解決するために、React と Web コンポーネント の間に挟む
橋渡し用の React コンポーネント のこと。

どうやってラップする？

Lit で簡単にラップできる

Lit は **Web コンポーネント** のための
ライブラリ。

`@lit/react` のパッケージに
Webコンポーネント をラップする API
`createComponent` が用意されている。

試してみる 🔥

ラップするSFC

シンプルな `input` のコンポーネントを

React で使えるようにすることを考える。

```
// MyInput.ce.vue
<script setup lang="ts">
const model = defineModel();
</script>

<template>
 <input v-model="model" />
</template>
```

ラップする

`defineCustomElement` で生成された
カスタム要素のクラスのコンストラクタを
Lit の `createComponent` に設定し、
Web コンポーネントを
ラッパーコンポーネントでラップする。

`events` プロパティにカスタムイベントを
登録することで React の**イベントプロパティ** で
使用可能にする。

```
import * as React from "react";
import { createComponent, type EventName } from "@lit/react";

const CEMyInput = defineCustomElement(VueMyInput);

const register = () => {…省略}

// React用にWebコンポーネントをラップ
const MyInput = createComponent({
 tagName: "my-input",
 // カスタム要素のクラスのコンストラクタを設定
 elementClass: CEMyInput,
 react: React,
 // DOMイベントをイベントプロパティで使えるように設定
 events: {
   onUpdateModelValue: "update:modelValue" as
EventName<CustomEvent>,
 },
});

export { register, MyInput };
```

React を外部依存関係に設定

React を外部依存関係として設定する。

右記はビルドツールが Vite の場合の設定例。

```
// vite.config.ts
export default defineConfig({
    …省略
    rollupOptions: {
      // vue, react を外部依存関係として指定し、バンドルしないように設定（ライブラリ使用側の vue, react を使う）
      external: ["vue", "react"],
      output: {
        // UMDビルド時、外部の vue, reactモジュールをグローバル変数 Vue, React として参照するようにする
        globals: {
          vue: "Vue",
          react: "React",
        },
      },
    },
  },
});
```

React で使う

`register` を実行し、カスタム要素を登録する。

`MyInput` コンポーネントには以下を設定する。

- **props** された `modelValue` 属性を設定
- **emit** された `update:modelValue` に紐づけたカスタムイベント `onUpdateModelValue` をイベントプロパティで呼び出すよう設定し、`text` を更新するように実装

```tsx
import { useState, useEffect } from "react";
import { register, MyInput } from "my-library";

function App() {
 const [text, setText] = useState("");

 useEffect(() => {
   register();
 }, []);

 const updateText = (e: CustomEvent) => {
   setText(e.detail[0]);
 };

 return (
   <>
     <p>入力内容： {text}</p>
     <MyInput modelValue={text} onUpdateModelValue={updateText} />
   </>
 );
}

export default App;
```

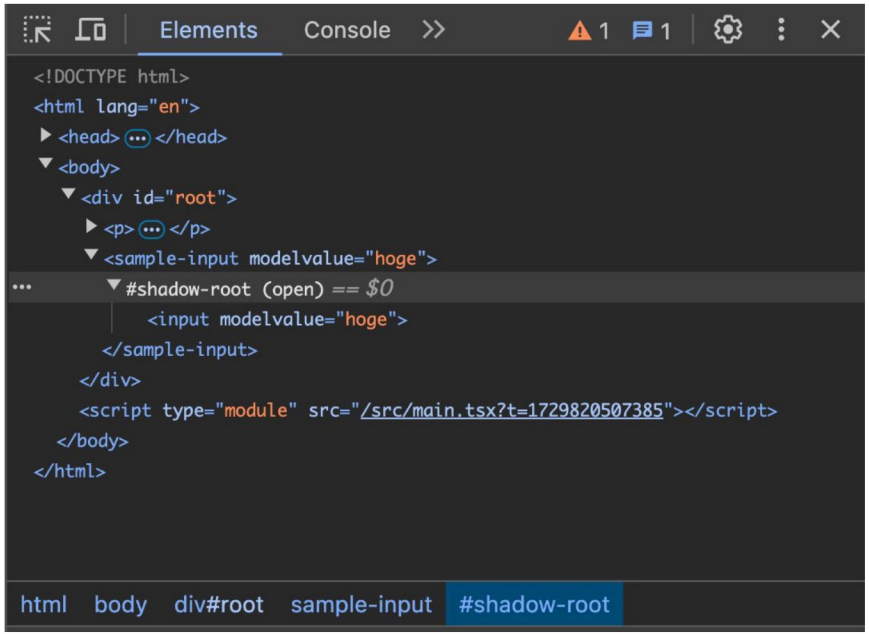
ブラウザでの表示

表示

入力内容: hoge

hoge

DOM



React で使えた 🎉

まとめ・感想

まとめ

- Web コンポーネントを使うと
  フレームワークに依存しない**汎用的なコンポーネント** を作成できる
- Vue には **defineCustomElement** という Web コンポーネントを
  作成するAPIが用意されている
- React で Webコンポーネントを上手く活用するには、
  ラッパーコンポーネント で Web コンポーネントをラップする必要がある

感想

- 複雑なコンポーネントも Vue を使えば Web コンポーネント化しやすそう
- Vue で v-model を使う Webコンポーネントを作ると
  modelValue が Vue 寄りのプロパティ名になってしまうのが気になる
- SSR対応やスロットコンテンツの表示に関してなど
  いくつか未解決の課題はありそうなのが気になる
  （Vueの公式ドキュメントに記載あり）

PROFESSIONAL TECH “LOVE”な
エンジニアを募集中です。

LOVE

次の常識をつくるために。
「みんなの役に立ちたい」「技術力で貢献したい」という想い、
つまり、あなたの LOVE を求めています。
専門知 × テクノロジーの力で、ともに社会に貢献しましょう。

WANTED
TECH JOBS

弁護士ドットコム

最新の取り組みは弁護士ドットコム　クリエイターズブログで発信中です。

ぜひご覧ください。

弁護士ドットコム株式会社

CREATORS' BLOG

弁護士ドットコム　クリエイターズブログ

https://creators.bengo4.com/

弁護士ドットコム

ありがとうございました