Combining React and Svelte components in a single app (interop)

Bob Fanger · Follow
3 min read · Aug 21, 2022

* when using svelte-preproces-react

React and Svelte are frontend frameworks that allow writing apps using components, but they do this in a different and incompatible way.
This article describes how to combine the two.

Why would you combine React with Svelte?

The React ecosystem is older, bigger and still growing. An interoperability solution allows Svelte apps to tap into that ecosystem. Which allows utilizing code written by third parties that don't provide Svelte support.

Another reason is that the existing codebase is already written in React and your team wants to migrate the project to Svelte.

How can you use React inside a Svelte file?

By installing and configuring the **svelte-preprocessor-react** package:

> ***Note:*** *I'm a maintainer of the svelte-preprocessor-react project*

Other lighter, less powerful solutions exists. If you don't need context, SSR or slots/children, one of the following projects might be a better fit: react-svelte, svelte-react, vite-plugin-svelte-bridge or svelte-adapter.

```
npm install --save-dev svelte-preprocessor-react
```

```
// svelte.config.js
import preprocess from "svelte-preprocess";
import preprocessReact from "svelte-preprocess-react/preprocessReact";

export default {
  preprocess: preprocessReact({
    preprocess: preprocess({ sourceMap: true }),
  }),
};
```

The svelte-preprocess is optional, so omit if you don't need Typescript, Sass, PostCSS, etc.

Using React components inside Svelte (sveltify)

Let's say you want to use react-video, you'll use it as if it was a regular svelte component, but prefix the the tag with "react:" so <YouTube /> becomes <**react:**YouTube />

```
<script>
  import YouTube from "react-youtube"
</script>

<react:YouTube videoId="AdNJ3fydeao" />
```

How does that work? The preprocessor uses a **sveltify** utility that wraps any React component into Svelte component. That svelte component renders a DOM element and via a react-portal the React component is rendered into that DOM element.

```
import { sveltify } from "svelte-preprocess-react";
```

sveltify is available for complex use-cases. Due to changes introduced in React 18 and how bundlers work, the **sveltify** utility requires more arguments than the component alone, but the preprocessor abstracts that away as shown in the Readme.

Note: To use React components inside a Svelte project your bundler needs JSX support. Vite supports JSX directly, but for Rollup or Webpack you'll need to setup @babel/preset-react.

Using Svelte components in React (reactify)

The **reactify** utility takes a Svelte component as argument and creates a React component.

```
import { reactify } from "svelte-preprocess-react";
import { YouTube } from 'svelte-yt';

const SvelteYouTube = reactify(YouTube);

export default function App() {
  return <SvelteYouTube videoId="AdNJ3fydeao" />
}
```

To use *.svelte files your bundler needs to integrate the Svelte compiler:
Vite needs @sveltejs/vite-plugin-svelte (included in SvelteKit)
Webpack needs svelte-loader
Rollup needs rollup-plugin-svelte

Caveats and considerations

Typescript & Linting are not aware of the preprocessor

```
<script lang="ts">
  import YouTube from "react-youtube";
  import { used } from "svelte-preprocess-react";
```

```
    used(YouTube); // used() prevents 'importsNotUsedAsValues' errors
</script>
```

There are subtle differences between Slots and JSX children, if a React component wants to inspect and transform it's child nodes you can't use the slot notation.

Adding compatibility layers adds overhead and increases complexity.
If you can prevent using one, please do so.
Intermix frameworks sparingly and use what works best for your product.

These examples are contrived as YouTube wrappers are available in both Svelte and React flavors.

If your goal is write a multi-framework component library https://mitosis.builder.io/ might be a better fit.

Conclusion

Although using a single framework is preferable, using React in Svelte works surprisingly well.

Svelte React Interoperability

Follow

Written by Bob Fanger

11 Followers

Building software in style

More from Bob Fanger

Bob Fanger

The problem with Error handling in Go

The problem I have with errors in golang is two-fold, it's unpredictable and it's very verbose

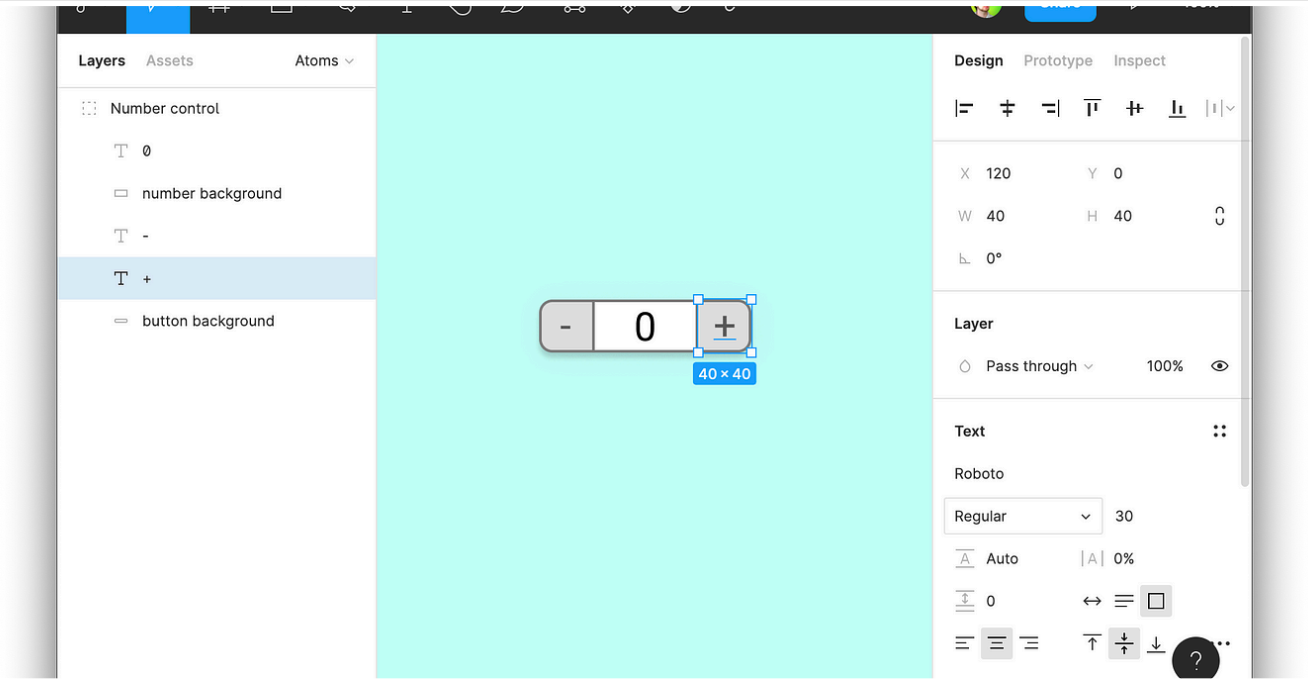
Feb 23, 2018 14

Bob Fanger

styled-jsx with scss in next.js

You probably installed @zeit/next-sass expecting sass to also work inside styled-jsx tags, sadly not. In this article I'll describe how to...

Dec 9, 2019 56 2



Bob Fanger

Storytelling for Frontend Engineers

Storytelling is a useful technique for building interfaces and in this article I will show you how to make components speak for themselves.

Sep 13, 2021 14

See all from Bob Fanger

Recommended from Medium

Dana Prata

Getting Started with Svelte and Vite: A Practical Guide

1. Outline

Feb 9 4

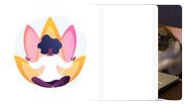
Afan Khan in JavaScript in Plain English

Microsoft is ditching React

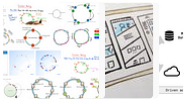
Here's why Microsoft considers React a mistake for Edge.

Jun 6 2.9K 66

Lists



Stories to Help You Grow as a Software Developer
19 stories · 1232 saves



General Coding Knowledge
20 stories · 1430 saves

Medium's Huge List of Publications Accepting Submissions
334 stories · 3171 saves

Natural Language Processing
1604 stories · 1165 saves

Amazon.com Seattle, WA
Software Development Engineer Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by $25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)
- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)
- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Alexander Nguyen in Level Up Coding

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

Jun 1 15.1K 231

Aviv Segal in Melio's R&D blog

Leave useEffect Alone!

Avoiding unnecessary re-renders in React. Every frontend framework has its inner workings and challenges, React is no different.

5d ago 3

Akshay S

Building Web Apps with React, WebAssembly, and Go

Introduction

Feb 5 7

```
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
▶ Warning: <http://[::1]:5173/resources/js/Components/Icons/hamburger.svg /> is using incorrect casing. Use PascalCase for React components, or lowercase for HTML elements.
    at http://[::1]:5173/resources/js/Components/Icons/hamburger.svg
    at button
    at div
    at div
    at nav
    at header
    at NavBar (http://[::1]:5173/resources/js/Components/NavBar/index.jsx:28:34)
    at div
    at AuthenticatedLayout (http://[::1]:5173/resources/js/Layouts/AuthenticatedLayout.jsx:26:47)
    at w2 (http://[::1]:5173/node_modules/.vite/deps/@inertiajs_react.js?v=2a838990:3711:25)
    at ErrorBoundary (http://[::1]:5173/node_modules/.vite/deps/react-error-b
```

react-dom.development.js:86

Josh Praise

How to import SVG files as React Components in Vite App

SVGR Vite plugin basically transforms SVGs into React components but I applied some custom modification to the plugin options to work fine

Apr 3 14

See more recommendations