

[Open in app](#)[Sign up](#)[Sign in](#)

Search



Svelte the Bad Parts (vs React)

Eric Kim · [Follow](#)

4 min read · Sep 29, 2023

[Listen](#)[Share](#)

I have been using React (and React Native) since 2015, and at the same time was actively involved with the React Community.

Recently I started using SvelteKit on my latest projects. Although I love Svelte, there are times I find it extremely frustrating due to its limitations. Here are things that I want to share with you what I find frustrating about Svelte and SvelteKit coming from a React developer background.

DISCLAIMER: I do not want people to think Svelte is a bad library/framework, in fact I think it is awesome and I love it. I am simply stating the bad parts in this blog post because these information are not easily shared and I hope they are resolved in the future. This blog was originally written on 2023-09-29

TypeScript types can only be used within the script tag

You can bypass this by first declaring/initializing inside the script tag and then referencing the variable in the markup, however this causes unnecessary code and fragmentation. This might be because JS inside the script tag and JS in the markup is compiled differently.

```

1  <script lang="ts">
2    const sumAlert = (arg1: number, arg2: number) => {
3      alert(arg1 + arg2)
4    }
5  </script>
6
7  <button
8    on:click={(e) => {
9      const sumAlert = (arg1: number, arg2: number) => {
10        alert(arg1 + arg2)
11      }
12      sumAlert(1, 2)
13    }}>
14    add
15  </button>
16

```

Unexpected token svelte(parse-error)
Unexpected token ts(-1)
View Problem (F8) No quick fixes available

Cannot declare types in .svelte outside of the script tag

Only one Component is allowed per (.svelte) file

A lot of times it is recommended to have one component per file even in React. However, there are times when creating multiple components in a single file is actually beneficial/recommended. For example, if you want to reuse a component only within one component and/or want to keep it private from other components. Unfortunately this is currently not possible in Svelte.

Only one script tag is allowed per component (and file)

I often put style related JS at the bottom of the file. Because `.svelte` only accepts one script tag, I am forced to put all my JS at the top. Note that you can have two script tags if you declare one as `context="module"` but this is for a totally different use case.

Svelte components cannot (easily) be created via pure JS

In my opinion this is probably the biggest downside of Svelte. Unlike React, where component can easily be created via a simple function, in Svelte you must create a `.svelte` file per component. This makes it very hard to seamlessly integrate with JS. For example Svelte components cannot be passed as `props`, instead they must be passed as `slot` (special syntax). Because of this, components cannot (easily) be composed or extended to a new Component using JS only. In Svelte extending/composing components requires not only a new `.svelte` file, but also the component you are composing to be known statically. ([reference1](#), [reference2](#))

Props do not contain element directives (events) and children (slots)

In Svelte, component props (`$$props` or `$$restProps`) do not contain element directives. If you want to create a `Button` component that passes all the on directives

(for example `on:click` `on:change` `on:drag` etc...) it must be done so separately from spreading props. Also, because children is not part of props, `<slot />` must be placed to allow children to be rendered.

+layout does not allow props to be passed

When you create your own Svelte component (for example `Button.svelte`), you can pass props through slot, which then can be received from the children using let directives. Because of this, you may think you can do the same thing for `+layout.svelte` as it is parent (wrapper) of each `+page.svelte` components.

However, this doesn't work, and some people say it is because `+page.svelte` (child component) does not know which `+layout.svelte` the props are coming from statically. This is probably still possible if you can provide typing yourself, but as I mentioned before TypeScript types are not supported inside the markup.

Incorrect TypeScript errors

A lot of SvelteKit related types are generated inside `.svelte` folder when running `svelte-kit sync` or when dev server is running through `vite dev`. This means that people may see incorrect TypeScript errors if they forget to run the above commands and start changing their code.

Also, I honestly don't know why this happens, but sometimes you get TypeScript errors when importing components even though there shouldn't be. Most of the time the error goes away if you open the component you are importing and force-saving by adding whitespace then saving (no code changes). This happens quite often when renaming files or changing (`.svelte`) file paths.

CSS-in-JS is not supported out of the box

It is quite common that different styles depends on JS, and therefore storing entire CSS as JS object and styling html elements as directly would be nice.

If you have any Svelte/SvelteKit frustrations, please share in the comments and I will add them to the list.

Also, if any of my frustrations can be solved, please let me know so I can remove them from the list :)

Svelte

Sveltekit

TypeScript

HTML

React

[Follow](#)

Written by Eric Kim

416 Followers

More from Eric Kim



Mahisa Dyan Diptya

Enter your password

 Eric Kim

Tackling HOC in React Native

Higher Order Component or HOC is a very simple concept with an incredible power. I will start out with some high level overview, and follow...

Mar 25, 2017  1.5K  3



```

3   "version": "0.0.1",
4   "private": true,
5   "scripts": {
6     "start": "node node_modules/react-native/local-cli/cli.js start"
7   },
8   "dependencies": {
9     "react": "15.2.1",
10    "react-native": "0.29.0",
11    "react-native-router-flux": "^3.31.0",
12    "react-redux": "^4.4.5",
13    "redux": "^3.5.2",
14    "redux-logger": "^2.6.1",
15    "redux-promise": "^0.5.3",
16    "redux-thunk": "^2.1.0"
17  },
18  "devDependencies": {

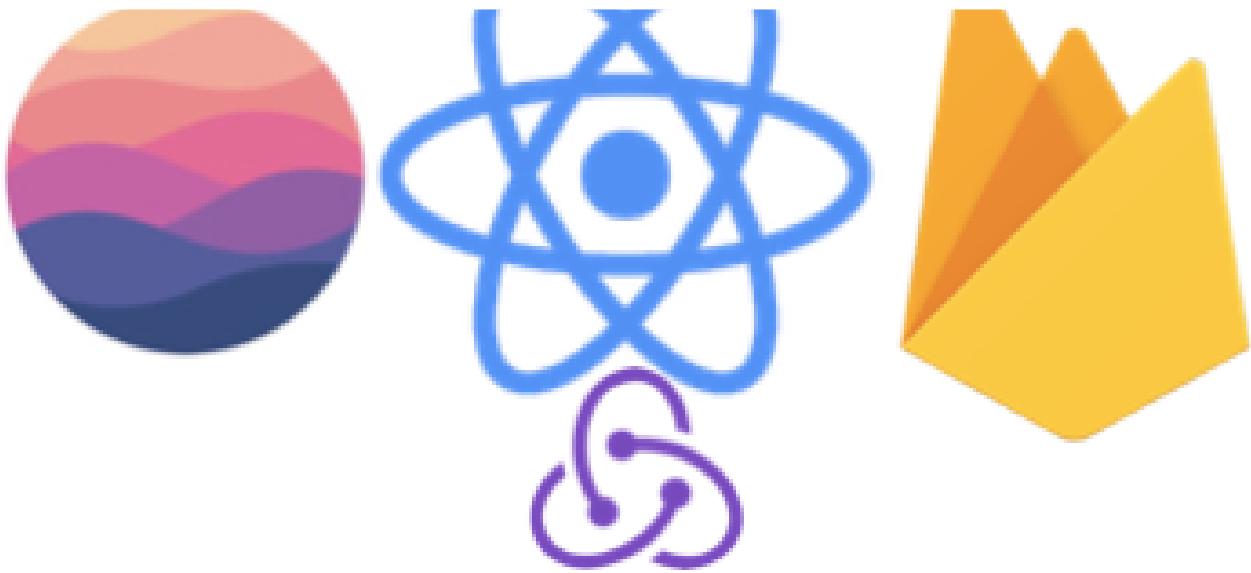
```

 Eric Kim

How to structure your React Native application using Redux?

Introduction

Jul 14, 2016  540  6



D
R
E
W
N
A
T
U
R
E

 Eric Kim in codeburst

Tackling React Native Storage — Part 1

Making data persist in React Native can be hard. I have worked with AsyncStorage, RealmDB, and Firebase using React Native. Today I am...

Dec 12, 2016

1.4K

12



Eric Kim

We are living in an information era, and the most prosperous time in human history.

Little over 150 years ago, most people spent an average of 60 percent of their time just to survive. Those days, people had to spend their...

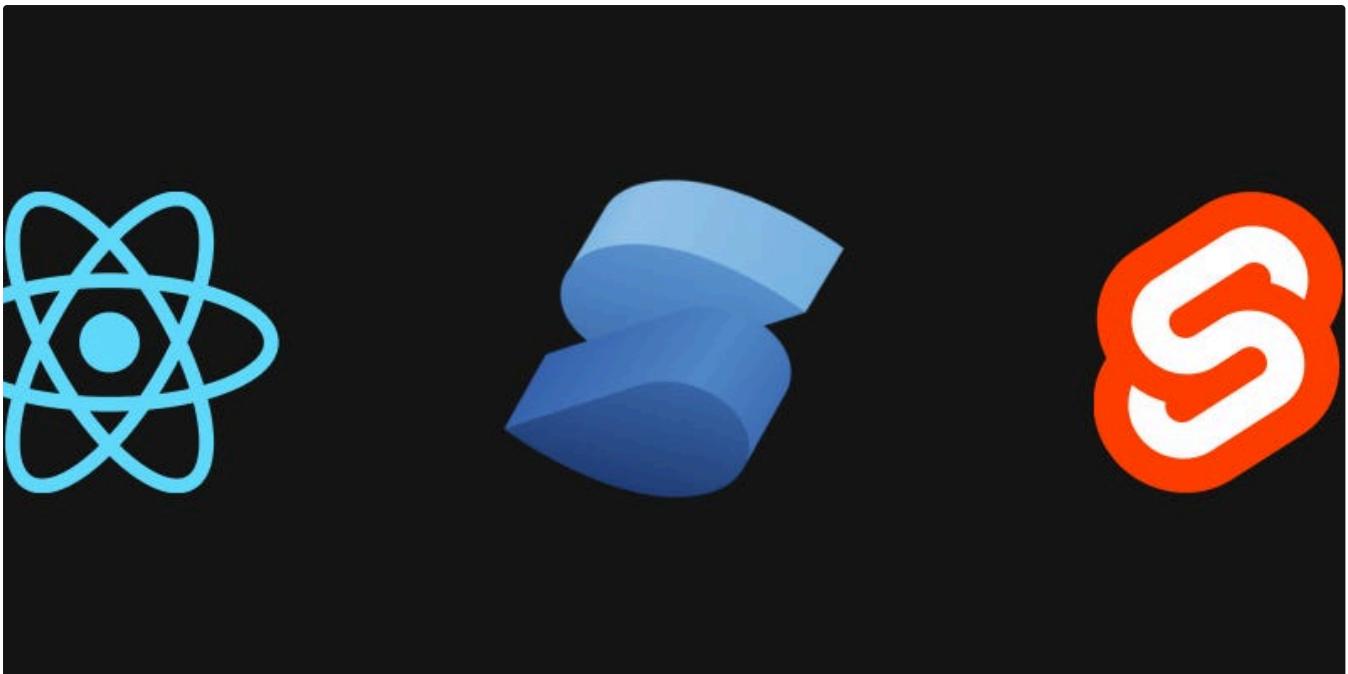
Nov 15, 2016

54



[See all from Eric Kim](#)

Recommended from Medium



Ricardo Cerqueira in MCTW_TDW

React, Svelte and Solid: The differences and advantages

What is the best JavaScript framework for each project?

Feb 23

7





Afan Khan in JavaScript in Plain English

Microsoft is ditching React

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

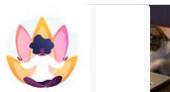
Jun 6

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

- 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



New JavaScript features

JS

```
const fruits = [
  { name: 'pineapple🍍', color: 'yellow' },
  { name: 'apple🍎', color: 'red' },
  { name: 'banana🍌', color: 'yellow' },
  { name: 'strawberry🍓', color: 'red' },
];

// ✅ native group by
const groupedByColor = Object.groupBy(
  fruits,
  fruit => fruit.color
);
```

```
// data-fetcher.js
// ...
const { promise, resolve, reject } =
  Promise.withResolvers();
```

 Tari Ibaba in Coding Beauty

5 amazing new JavaScript features in ES15 (2024)

5 juicy ES15 features with new functionality for cleaner and shorter JavaScript code in 2024.

Jun 2 1.93K 10



 Sviat Kuzhelev

Mastering NextJS Architecture with TypeScript in Mind | Design Abstractions 2024

You'll never know how everything works, but you should understand the system.—by Sviat Kuzhelev.

Jun 17 6.8K



 Andrew Zuo

Async Await Is The Worst Thing To Happen To Programming

I recently saw this meme about async and await.

Jun 22 2.2K 133



See more recommendations