Research Svelte

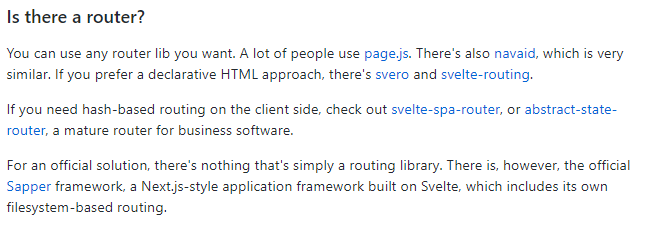
# Features comparing other frameworks?

## What is svelte?

* 1. Js-compiler 🡪 not a framework

1. What does a framework have as features out-of-the-box
   1. Code-splitting (separating code 🡪 for example api calls)
   2. Components (being able to separate components so you don’t have on giant file)
   3. Event handling (click events without needing to use javascript-native eventhandling)
   4. Conditional rendering (show items depending on a Boolean for example)
   5. Testing (test pieces of code)
   6. Routing (define the different pages and navigate between them
2. Which of these things does Svelte come with out of the box
   1. Code-splitting
   2. Components
   3. Event handling
   4. Conditional rendering
3. Why did they not put routing and testing in Svelte out-of-the box?

### Routing

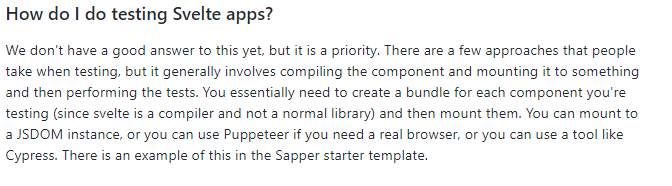


They do say you can use “any” routing library you want but these often lack the in-depth routing you get from for example vue.js their routing, where you actually declare each page instead of just using href on an <a>-tag.

I personally used page.js because a co-worker of Svelte sent me an in-depth pdf file about this library on the official discord server of Svelte, after using it for a while and testing it on my project I found that there really isn’t a difference between just switching between components and using page.js, so i ended up just ditching the idea of using a router since it doesn’t belong to Svelte and doesn’t prove anything else but the fact that this is a feature that almost every website needs.

After asking on of the head workers at Svelte on their discord about routing their argument mostly was that routing is only necessary on a framework and not on js-compilers, which Svelte actually is. They did recommend me an actual framework which uses Svelte but adds extra features like routing.

### Testing



As you can see by this message on the FAQ tab on the official website of Svelte people have already asked the same question about testing as I have, on this topic they generally say you can’t test anything yet use Svelte, even on the Svelte discord server.

You can if you REALLY want to test your code you can create a bundle for each component and use Cypress to test the code, but this is a tedious procedure, they just need to add testing.

C. NO proxying



When you are doing api calls to a public api you mostly (you can in Vue and React AND sapper) you need to let the api know you are authorized and not just a non-authorized person trying to reach the api.

Normally in a framework you can set a proxy name that let’s the api know you are (in my case localhost:5000) a authorized person, Svelte does not provide this, but sapper supports the rollup library which has this feature.

This could also be fixed if reddit allowed my url in the backend to disable the cors issue

# What make Svelte jump out comparing to other Frameworks?

## Less code, geen template tag (vue) of div tag (react)

## 

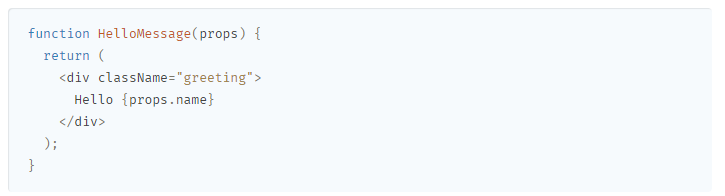
### 

### NO VIRTUAL DOM

A big point Svelte like to make is that they have eliminated the virtual DOM, what they mean by that I’m not sure but what I understand from their explanation is that other frameworks who use the virtual DOM use it to change rendered items, but this takes allot of time.

React has for example stated that it’s a positive thing you can re-render your whole application using the virtual DOM but looking back at that statement now it would make everything much slower.

What Svelte does according to my understanding is that they just completely eliminated this in order to create an instant change rather then checking between the already existing object and the changed object. For example:

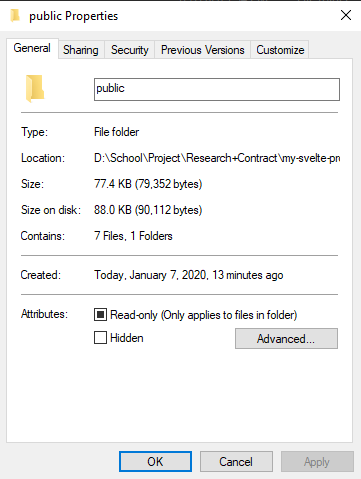
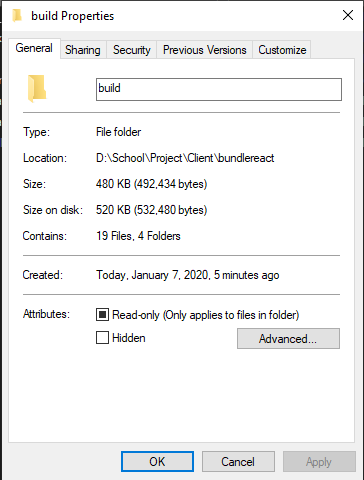


This is an example of someone saying how the page should now look using a render() function, this is the virtual DOM, never in the 3 weeks of my project I’ve had to write any code like this, you can just immediately change an object that’s already declared.

### Build Size

Mostly when building a project using frameworks you would build everything along with it from the framework, which would lead to a large build file which leads to a slow launching website.

The biggest thing for me personally which makes Svelte stand out comparing with Vue and React is that the build file is drastically smaller as seen on the images below:



# The pros and cons of Svelte

## The pros:

* Small build size leads to fast launching and running apps
* Simple for anyone with knowledge of Js, HTML and CSS
* Easy installation

## The cons

* No proxying unless you set up a proxy server yourself
* Separation of code issues (login.js can’t return values for some reason)
* No router
* No testing
* Brand new

## Conclusion

If you are planning on trying out Svelte, use the Sapper framework since it actually is a framework and has allot of the features that were missing.

I can’t recommend Svelte **right now** because it is so new, but once it has all the necessary features that a framework needs to create a web app, I think it could become one of the biggest frameworks out there.

# Sources:

Researching js-frameworks

<https://skillcrush.com/2018/07/23/what-is-a-javascript-framework/>

<https://jsreport.io/the-ultimate-guide-to-javascript-frameworks/>

how does rouing work in vue

<https://vuejs.org/v2/guide/routing.html>

figuring what features are asked after and if they have an explanation for it

<https://github.com/sveltejs/svelte/wiki/FAQ>

what makes svelte unique?

<https://svelte.dev/blog/write-less-code>

<https://svelte.dev/blog/virtual-dom-is-pure-overhead>

<https://svelte.dev/blog/svelte-3-rethinking-reactivity>

proxying issue in svelte

<https://github.com/sveltejs/svelte/issues/3717>