

GitLab Commit Virtual is here. [Register Now](#) for our 24 hour immersive DevOps experience.

Oct 20, 2016 - Jacob Schatz [!\[\]\(c8d96c8885d3000a912c2582004aed63_img.jpg\)](#)

Why we chose Vue.js

Why GitLab went with Vue.js

← Back to engineering

I had a great conversation with an interviewee a few weeks ago about how one should go about choosing a JavaScript framework.

He pointed out that when a major software company releases their secret sauce, there is going to be hype. Devs think to themselves, "That company writes JS differently than me, and they are prominent and successful. Is their

Try [GitLab](#) risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)



Their secret sauce may be awesome, but don't assume awesomeness just because everyone else gets excited. You wouldn't copy and paste an answer from StackOverflow, without understanding it, so why copy and paste an entire framework?

Which brings me to our decision to use [Vue.js](#) at GitLab.

Simplicity and ease of use

Primarily what drew us to Vue.js is that it allows our team to *easily* write *simple* JavaScript. Getting started with Vue.js is extremely easy. Its source code is very readable, and the documentation is the only tutorial you'll ever need. You don't need external libraries. You can use it with or without jQuery. You won't need to install any plugins, though many are available. I like vanilla Vue.js personally, although I can reach for vue-resource when I need it. Hooking Vue.js up to existing code is very straightforward. There's no magic to Vue.js – it's `Object`s all the way down.

I talk to a lot of JavaScript devs and I find it really interesting that the ones who spend the most time in Angular tend to not know JavaScript nearly as well. I don't want that to be me or our devs. Why should we write "not JavaScript?"

Try [GitLab](#) risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)

data will change.

But Vue.js comes with the perfect balance of what it will do for you and what you need to do yourself. If Backbone was anarchy (no one in charge) and Angular is a dictatorship (the Angular team is in charge), I'd say Vue.js is like socialism: you are definitely in charge, but Vue.js is always within reach, a sturdy, but flexible safety net ready to help you keep your programming efficient and your DOM-inflicted suffering to a minimum.

To give you an idea of what I mean, here's a simple [Codepen](#):

```
<div id="journal">
  <input type="text" v-model="message">
  <div>{ message }</div>
</div>
```

```
var journal = new Vue({
  el: '#journal',
  data: {
    message: 'Your first entry'
  }
});
```

Try [GitLab](#) risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)

examples when you're "Getting started", but in reality things get complicated as soon as you try to get your money's worth out of the framework. Not with Vue.js though – real-life usage seems to stay as simple as the docs.

And that is what we love about Vue.js: it's an elegant combination of structure and simplicity. The data for the view goes in an object called `data`, but the data can get there and look however you want. Any functions you'll write as callbacks for events go into a `methods` object, but they can do or return whatever you want. Vue.js just knows when things change and updates your views. And you write less code.

Vue.js + GitLab === Less code

So what problem does this solve for GitLab? When I joined, all the JavaScript was written with JQuery. There is nothing *wrong* with that, except that it takes a lot more code to solve every problem. We knew we could do better. Once we started with Vue.js, we could immediately and consistently solve complex problems in much less time.

A simple, but practical example we're using in production: on a GitLab Issue, the issue's state is displayed as either `closed` or `open`. That simple value can change often and needs to be represented in several views. With JQuery, we had about 30 or so lines of code to propagate those changes, and those lines involved multiple classes and querying the DOM by hand.

Try [GitLab](#) risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)

What [Evan You](#) knows is that creating a kick ass framework isn't just about writing great code. You need excellent documentation, a strong community helping each other learn, a supporting cast of libraries and plugins to help users solve the hard problems, and short feedback loops based on user feedback to keep the framework relevant. Vue.js is all of that, *plus* great code. That's why we're using it. What about you?

Watch the Why We Chose Vue.js webcast

Having trouble [viewing this video](#)? You may need to update your [cookie settings to allow personalization \(personal information\) cookies](#). You might also need to allow us on your adblocker, firewall, or browser privacy settings.

Why we chose Vue.js via @jakecodes

Click to tweet! 



◀ [GitLab UX update](#)

[My first weeks at GitLab: How we ship ...](#) ➤

Try [GitLab](#) risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)



Want a more effective CI/CD pipeline? Try our pro tips

by Valerie Silverthorne

Here's how to take your CI/CD pipeline to the next level with hands on advice about faster builds, better security and more.

[Learn more →](#)



GitLab's guide to safe deployment practices

by Orit Golowinski

It's important to safeguard your deployment process. Here's our best advice to protect your environments.

[Learn more →](#)



Try **GitLab** risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)

How to integrate Azure DevOps repositories with GitLab

by Itzik Gan-Baruch

How to keep your code in an Azure DevOps repository and run CI/CD with GitLab pipelines.

[Learn more →](#)

Follow us: [!\[\]\(9dfdaff1d86ba3c1f8353b4d1b61b8c5_img.jpg\)](#) [!\[\]\(bcef2083a617d3f771f1bcdf2f97158d_img.jpg\)](#) [!\[\]\(2c64db98cee6d30f87a54305b47fe92d_img.jpg\)](#) [!\[\]\(983d60898330cfb9aed2717ca7dcd4a1_img.jpg\)](#)

Sign up for GitLab's twice-monthly newsletter to explore upcoming webcasts, how-to blogs, and stay up-to-date on exciting new features released every month:

Try [GitLab](#) risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)

GitLab is more than just source code management or CI/CD. It is a full software development lifecycle & DevOps tool in a single application.

Try GitLab for Free



Why GitLab?

Product

Resources

All resources

Community

Customers

Solutions	All-Remote	Contribute
Services	Blog	Partners
DevOps lifecycle	Newsletter	Channel Partners
DevOps tools	Events	Explore repositories
Is it any good?	Webcasts	Source code
Releases	Topics	Shop
Pricing	Training	Direction
Get started	Docs	Contributors
	Install	Core Team
		Hall of fame

Support

- [Get help](#)
- [Contact Sales](#)
- [Contact Support](#)
- [Support options](#)
- [Status](#)
- [Customers portal](#)

Company

- [About](#)
- [What is GitLab?](#)
- [Jobs](#)
- [Culture](#)
- [Team](#)
- [Press](#)

[Analysts](#)

[Handbook](#)

[Security](#)

[Contact](#)

[Terms](#)

[Privacy](#)

[Trademark](#)

GIT is a trademark of Software Freedom Conservancy and our use of 'GitLab' is under license

Edit [this page](#) — open [Web IDE](#) — please [contribute.](#) 

Try **GitLab** risk-free for 30 days.

No credit card required. Have questions? [Contact us.](#)

[Get Your Free Trial Today](#)