

# Drizzle Vue - A Truffle Story

Truffle is an organization that maintains a balance of transparency, mentorship, encouragement, inclusivity and humor that I'm thankful for everyday. While much could be written about Truffle's management culture, I want to share my experience of working on a side project here.

Our remote work setup and the openness of our team contributes to a healthy and productive environment. The ability to schedule zoom sessions to brainstorm and discuss ideas fits well with our schedule. Every member contributes to this culture being generous with their time and knowledge. I'm happy and inspired to see team members grow as developers through the subtle mentorship as a result of our culture.

For most of my career, I dreaded mandatory soul draining review type meetings, but at Truffle, I look forward to the incredibly productive and creative weekly one-on-ones I have with Joshua Quintal, our Product Lead. Josh originally developed Drizzle and we often discuss its role in the dapp ecosystem, issues and roadmap during our meetings.

## About Drizzle

Drizzle is a Smart Contract Store with a subscription and mutation API that abstracts away the boilerplate for reading and writing smart contract state as well as listening for contract events.

Drizzle originated as a helper library for a React project Josh was developing. However he came to realize that the Smart Contract Store was an important piece needed in dapp development as applications start to scale. In Web 2.0 the need to manage and reason about state at scale has led to libraries like [1][Redux](#). Similarly, we want to give web 3.0 dapp developers these best practices to leverage as their applications grow in complexity.

## Drizzle's Unintended Front End Frameworks Exclusion

Unfortunately, this is not how Drizzle is understood by developers we interviewed formally and informally. The most common misunderstanding is that Drizzle is coupled with React, the most popular JavaScript front end framework. We can't blame our community as most of our documentation focuses on implementing Dapps using React.

Drizzle was initially released with bindings only for React which created this misperception that excluded developers from utilizing other front end frameworks such as Vue and Angular.

The problem is that we didn't offer an easy solution for Vue or Angular and our documentation doesn't offer guidance. We want to change this! We want to support all the frameworks!

## A Vue Plugin Side Project

I felt this was an important problem to solve in order to improve the dapp developer experience. It was important to get Drizzle working for Vue. Unfortunately, I knew nothing about Vue. In previous organizations I was a part of, I would have shied away from pursuing this project. However, at Truffle I felt comfortable and empowered to champion the idea. Josh encouraged me to spend the time needed to study Vue and experiment with the concepts.

The basic plan was to: - Use Vuex, Vue's popular State management plugin - Transform Drizzle's redux store to be consumed by Vuex - A redux store as an [2][Observable](#) which would allow us to transform/map drizzle state to a more state pieces that Vuex could consume. - This would allow us to have more efficient UI renders by using RxJS techniques like `distinctUntilChanged` - Create basic Vue components mimicking drizzle-react-components - Vue's architecture allows us to use the adaptor as a datasource as well as provide basic support for Components that can render the data. This gives Vue developers to use our basic Components to make quick proof of concepts, and the flexibility to create richer Components when they need to.

A [3][basic proof of concept](#) eventually surfaced and I was happy with the outcome as it showed a path forward to Angular using this approach with RxJS and can't wait to test it out!

## Collaboration And Improvements

Another aspect of Truffle's culture is to promote collaboration and exchange ideas with our community, and meet like minded developers and advocates working on similar problems. We're encouraged to go to meetups and be advocates for this incredible space.

I reached out to Julien Klepatch on reddit after seeing his impressive [4][Drizzle overview video](#) and asked for his feedback on the Vue Plugin. His participation, code contribution, code review and feedback clarified the project's direction and generally improved the codebase. Furthermore, he created 3 videos to explain the new plugin. - [5][Drizzle Vue Plugin: overview](#) - [6][Drizzle Vue Plugin: Custom Components](#) - [7][Drizzle Vue Plugin: Events](#)

We also run a range of educational programs as part of Truffle University, that amongst other things, teaches developers

how to use our full suite of tools. As part of the experience, students are mentored by the team and are encouraged to contribute to open-source projects, such as the Truffle Suite itself. James Kehoe, a developer in our first cohort, uses Vue on a daily basis and took an interest in the plugin. He took the initial design of the Contract Event Logic and turned it into idiomatic Vue. James also improved the documentation and introduced testing to the project. I learned a lot about Vue pairing with James, and look forward to future sessions with him.

## Take A Preview¶

You can preview the [3][Plugin](#) before we officially release it. Be sure to check out Julien's videos for an excellent walkthrough: [5][Drizzle Vue Plugin: overview](#) , [6][Drizzle Vue Plugin: Custom Components](#) and [7][Drizzle Vue Plugin: Events](#)

This project happened because of Truffle's enabling and supportive culture. Truffle is cool like that! Also, co-MVPs James and Julien's improved the Vue Plugin and made it a viable tool. In fact, I consider them to be original contributors and am happy to see their continued involvement in the project.

Come meet the team and learn about our tools at TruffleCon. We're looking forward to meeting you all!

## References¶

1. <https://redux.js.org/>
2. <https://redux.js.org/api/store#subscribe>
3. <https://github.com/trufflesuite/drizzle-vue-plugin>
4. [https://www.youtube.com/watch?v=QH\\_yLPYQEs4](https://www.youtube.com/watch?v=QH_yLPYQEs4)
5. <https://www.youtube.com/watch?v=XaVEZ1ucxac>
6. <https://www.youtube.com/watch?v=ApJwXfWKI7Q>
7. <https://www.youtube.com/watch?v=xyoztqeYd6U>