

Drizzle 1.5.0 - A new beginning

The dawn of a new era for regular releases and a better contributor experience Since the beginning, our goal here at Truffle has been to make dapp development as easy as it can be. To this end Josh Quintal created [Drizzle](#), a managed Redux store to make interfacing with the blockchain just a little bit easier.

From many to one

Today, we are announcing that we are moving to a monorepo. From now on, a single PR can be used to update code and documentation that used to span multiple repos.

Over the last two years, we've developed libraries on top of Drizzle such as `drizzle-react` and `drizzle-react-components`. Since these libraries depend so heavily on each other, it only makes sense for us to move all of these packages to a monorepo to streamline any updates.

In addition, we now also have an experimental Vue.js plugin that fits naturally into the monorepo architecture. It is very likely that going forward we will develop new packages to aid the Drizzle ecosystem in this manner.

Scoped package names

Another big change we are making is towards scoped package names on NPM. This means that instead of:

`drizzle` `drizzle-react` `drizzle-react-components` `drizzle-vue-plugin` We will now have:

`@drizzle/store` `@drizzle/react-plugin` `@drizzle/react-components` `@drizzle/vue-plugin` From an organizational standpoint, this makes it a lot more intuitive to manage your dapp development.

Testing

Prior to the monorepo, testing our libraries together was a slow, error-prone and painful experience. Our maintainers had to build custom dapps to isolate changes and manually link packages together as well as coordinate with ganache for each front end.

The monorepo allows us to have dedicated front end apps that automatically build against the latest libraries and we can conveniently orchestrate them with a single ganache instance in just a couple commands. In other words, you can easily spin up three apps to test various aspects of the monorepo [in seconds](#).

Easily spin up real front ends to test that dapps can work with the current code

Better release strategy

Perhaps the biggest thing that the changes above allow us to do is more frequent releases.

In the past year, we often had to deal with delaying the publishing of a new change because it required testing and updating other repos. Our publishing process was essentially being blocked by having to synchronize changes amongst multiple repos.

With a monorepo, that all changes, which means that you can look forward to more regular releases and a faster turnaround time for getting issues fixed. Better changelogs are also coming as we move toward adopting [conventional commits](#) for each PR that gets merged.

Going forward

It's important to note that Drizzle wouldn't be what it is without the open source community. A big part of why we are making all these changes is because we need you to help us make Web3 a reality together.

Here at Truffle, every member on the team is passionate about improving the developer experience of building on Web3. In the upcoming weeks, we will be doing more regular blog posts targeting the front end space so keep an eye out!

I would also like to specifically thank Amal Sudama for his work on making this transition. It certainly couldn't have happened without him.