

Expectations

Remember: Keep your codebases *simple, yet robust*.

If a new developer to your framework comes along and takes longer than 10 minutes to grasp the high-level architecture, it's likely that you went a little overboard in the engineering department.

Alternatively, you should *never* forgo following fundamental best practices for the sake of simplicity, lest we teach that same newbie dev the *wrong* way of doing things.

The quality & architecture of Conduit implementations should reflect something similar to an early-stage startup's MVP: functionally complete & stable, but not unnecessarily over-engineered.

To write tests, or to not write tests?

TL;DR — we require a minimum of **one** unit test with every repo, but we'd definitely prefer all of them to include excellent testing coverage if the maintainers are willing to add it (or if someone in the community is kind enough to make a pull request :)

We believe that tests are a good concept, and we are big supporters of TDD in general. Building Conduit implementations without complete testing coverage, on the other hand, is a significant time commitment in and of itself, therefore we didn't include it in the spec at first since we believed that if people wanted it, it would be a fantastic "extra credit" aim for the repo. For example, a request for unit tests was made in our Angular 2 repo, and several fantastic community members are presently working on a PR to address it.

Another reason we didn't include them in the spec is from the "Golden Rule" above:

The quality & architecture of Conduit implementations should reflect something similar to an early-stage startup's MVP: functionally complete & stable, but not unnecessarily over-engineered.

Most startups we know that work in consumer-facing apps (like Conduit) don't apply TDD/testing until they have a solid product-market fit, which is smart because they then spend most of their time iterating on product & UI and thus are far more likely to find PMF.

This doesn't mean that TDD/testing === over-engineering, but in certain circumstances that statement does evaluate true (ex: consumer product finding PMF, side-projects, robust prototypes, etc).

That said, we do *prefer* that every repo includes excellent tests that are exemplary of TDD/testing with that framework 👍

Other Expectations

- All the required features (see specs) should be implemented.
- You should publish your implementation on a dedicated GitHub repository with the "Issues" section open.
- You should provide a README that presents an overview of your implementation and explains how to run it locally.
- The library/framework you are using should have at least 300 GitHub stars.
- You should do your best to keep your implementation up to date.

Project Overview

"Conduit" is a social blogging site (i.e. a Medium.com clone). It uses a custom API for all requests, including authentication. You can view a live demo over at <https://demo.realworld.io>