

# Why are there so many packages?

This article is derived from the documentation at <https://www.npmjs.com/package/@validatem/core>.

Dependencies often introduce a lot of unnecessary complexity into a project. To avoid that problem, I've designed Validatem to consist of a lot of small, separately-usable pieces - even much of the core plumbing has been split up that way, specifically the bits that may be used by validator and combinator functions.

This may sound counterintuitive; doesn't more dependencies mean *more* complexity? But in practice, "a dependency" in and of itself doesn't have a complexity cost at all; it's the code that is *in* the dependency where the complexity lies. The bigger a dependency is, the more complexity there is in that dependency, and the bigger the chance that some part of that complexity isn't even being used in your project!

By packaging things as granularly as possible, you end up only importing code into your project *that you are actually using*. Any bit of logic that's never used, is somewhere in a package that is never even installed. As an example: using 10 modules with 1 function each, will add less complexity to your project than using 1 module with 100 functions.

This has a lot of benefits, for both you and me:

- **Easier to audit/review:** When only the code you're actually using is added to your project, there will be less code for you to review. And because each piece is designed to be [loosely coupled](#) and extensively documented, you can review each (tiny) piece in isolation; without having to trawl through mountains of source code to figure out how it's being called and what assumptions are being made there.
- **Easier to version and maintain:** Most of the modules for Validatem will be completely done and feature-complete the moment they are written, never needing any updates at all. When occasionally a module *does* need an update, it will almost certainly not be one that breaks the API, because the API for each module is so simple that there isn't much to break.
- **Easier to upgrade:** Because of almost nothing ever breaking the API, it also means that you'll rarely need to manually upgrade anything, if ever! The vast majority of module updates can be automatically applied, even many years into the future, *even* if a new (breaking) version of `validatem/@core` is ever released down the line.

- **Easier to fork:** If for any reason you want to fork any part of Validatem, you can do so easily - *without* also having to maintain a big pile of validators, combinators, internals, and so on. You only need to fork and maintain the bit where you want to deviate from the official releases.

Of course, there being so many packages means it can be more difficult to find the specific package you want. That is why [the Validatem website has an extensive, categorized list](#) of all the validators, combinators and utilities available for Validatem. Both officially-maintained ones, and third-party modules!

---

Revision #1

Created 10 December 2024 23:38:33 by joepie91

Updated 10 December 2024 23:39:15 by joepie91