Defeating Complexity

- Functional, Data-driven systems facilitate simplicity
- Certain unfamiliarity with the patterns must be overcome to achieve this
- Introducing https://github.com/markbastian/partsbin

https://github.com/markbastian/partsbin

partsbin

You see a tray full of interchangeable parts. You reach in and grab some and start putting things together. Something starts to form. You mostly like what you see but you need to swap out a few things. No problem, it's all super easy. You tinker a bit more and the next thing you know you have a working system.

This is partsbin - a set of reusable components and a philosophy for building simple systems.

The goal of this project is to:

1. Provide some best practices for designing simple, composable systems.
2. Provide a few useful functions for working with Integrant or similar libraries. As such, when the namespace alias `ig` is used I am referring to `[integrant.core :as ig]`.
3. Provide reference implementations for some common parts.

As such, it can be used in a few ways:

1. Read it for good advice on how to build composable, data-driven systems (no need to use any of the code here, or just vendor what you will).
2. Use the top-level utility methods found in `partsbin.system`, `partsbin.middleware`, etc. but define your own `ig/init-key` methods.
3. Use the provided implementations of `ig/init-key` as well.

Installation

Add the following dependency to your project.clj:

clojars [markbastian/partsbin "0.1.2"]