MVVM and you

A pattern for cleaner more testable front end logic

What’s this MVVM thing anyway?

- **M**odel – Your business logic
- **V**iew – The presentation of your data
  - This is *not* limited to presentation for your eyeballs.
- **V**iew **M**odel – An intermediary logic layer, generated by a “Binder” that contains the data and logic to be exposed to the View
- **C**ontroller – Yes, this still exists

Binder!?

- Blatantly stolen from Wikipedia:

"The binder frees the developer from being obliged to write boiler-plate logic to synchronise the view model and view."

- This can be as simple as passing a parameters into a constructor or as complicated as you can possibly make it.
- Moo/Moose Type Coercions fit here well!

Why?

- Separation of concerns!
  - The View knows about the ViewModel
  - The ViewModel knows about the Model
  - The View knows nothing of the Model
  - The Model knows nothing of the View **or** the ViewModel
- The Controller needs to know about at least two of these at any given time, but Controller logic is ideally thin and easily disposable.

View Model Responsibilities

- Interact with the Model
- Embellish the data
  - The doesn’t mean you’re storing your styling information in the View Model
  - This does mean you should be writing your conditional presentation logic here
- Provide hooks for testing
- Be general enough that it can be used on both ends of a Request/Response cycle in a web scenario, or during an event cycle in desktop/mobile.

Caveats

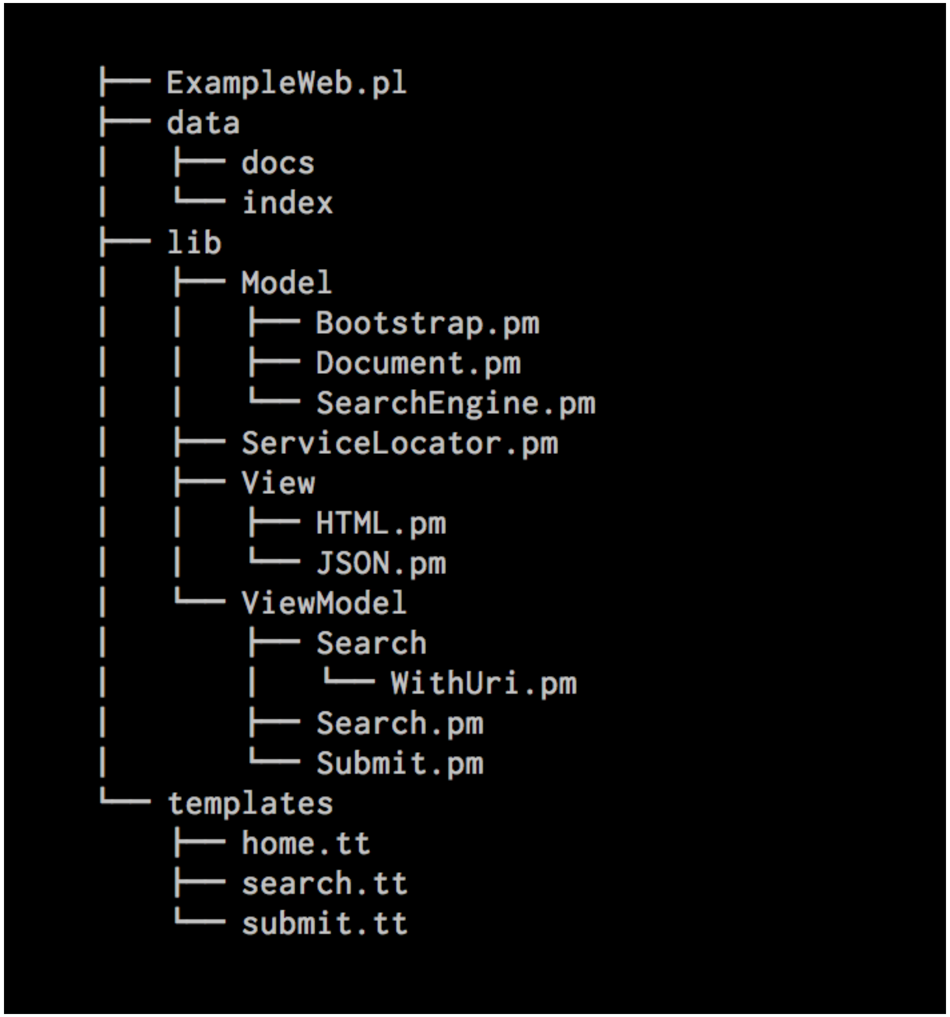
- Can be complicated to implement
  - *Cough* Angular.js *Cough*
- Can be computationally expensive depending on how you do it
  - Eg Microsoft’s XAML implementation
- Some implementations tie View Model creation to the View, which can make it difficult to use your View Models with multiple View types

On to the code!

View Models and Views
Are roughly equivalently named.

To run the example, type:

plackup ExampleWeb.pl



```
├── ExampleWeb.pl
├── data
│   ├── docs
│   └── index
├── lib
│   ├── Model
│   │   ├── Bootstrap.pm
│   │   ├── Document.pm
│   │   └── SearchEngine.pm
│   ├── ServiceLocator.pm
│   ├── View
│   │   ├── HTML.pm
│   │   └── JSON.pm
│   └── ViewModel
│       ├── Search
│       │   └── WithUri.pm
│       ├── Search.pm
│       └── Submit.pm
└── templates
    ├── home.tt
    ├── search.tt
    └── submit.tt
```