
Belgrade Senior Center

Group 1

Spencer Cornish, Taylor Koth, Josh Stephenson-Losey
04.22.2019

System Design and Architecture

Some Design Patterns Used

Built_Redux

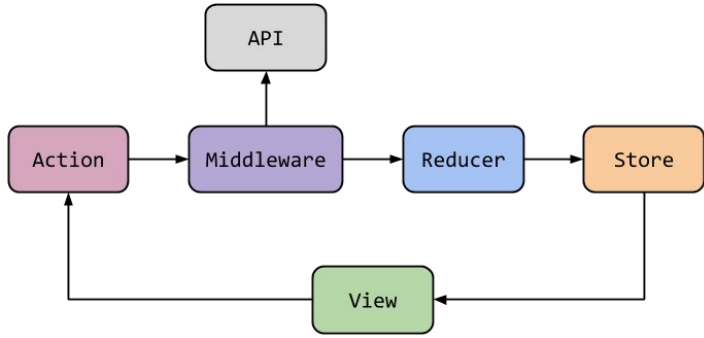
A state management solution, which also enforces immutability.

WUI_Builder

A virtual DOM implementation, designed for performance.

Fractal folder structure

A technique to organize many classes using deep folder structures to organize.



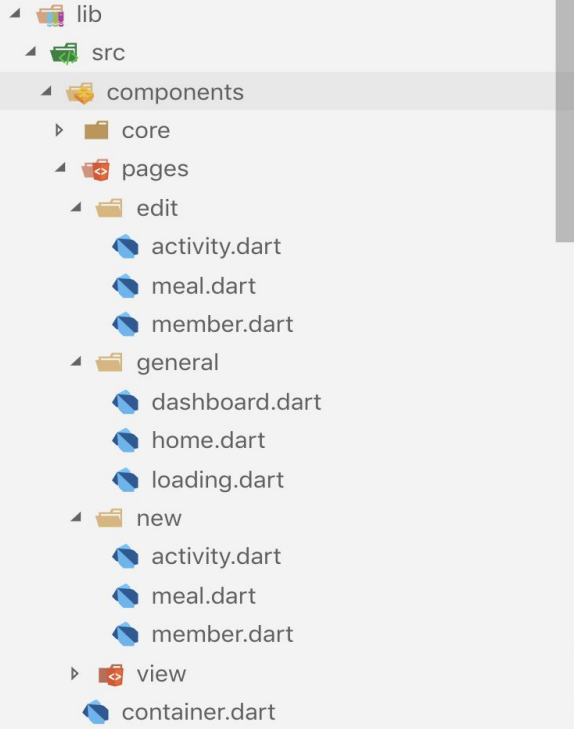
Built Redux

- Enforces Immutability
 - UI thread is never blocked
 - Store is the “source of truth”
 - All objects comparable right away
 - “Middlewares” handle logging, pub/sub, etc.
-

WUI_Builder (and Dart in general)

- Virtual DOM, for performant updates
 - Component reuse, conditional rendering
 - Smart routing and Single Page App control
-

Fractal Folder Structure



```
lib
├── src
│   ├── components
│   │   ├── core
│   │   ├── pages
│   │   │   ├── edit
│   │   │   │   ├── activity.dart
│   │   │   │   ├── meal.dart
│   │   │   │   └── member.dart
│   │   │   └── general
│   │   │       ├── dashboard.dart
│   │   │       ├── home.dart
│   │   │       └── loading.dart
│   │   └── new
│   │       ├── activity.dart
│   │       ├── meal.dart
│   │       └── member.dart
│   └── view
│       └── container.dart
```

A screenshot of a file explorer showing a fractal folder structure. The root is 'lib', which contains 'src'. 'src' contains 'components', 'pages', and 'view'. 'components' contains 'core', 'pages', and 'new'. 'pages' contains 'edit' and 'general'. 'edit' contains 'activity.dart', 'meal.dart', and 'member.dart'. 'general' contains 'dashboard.dart', 'home.dart', and 'loading.dart'. 'new' contains 'activity.dart', 'meal.dart', and 'member.dart'. 'view' contains 'container.dart'.

- Structures components in a smart pattern
 - Represents the HTML tree in files
-


Constant Integration & Constant Delivery


Travis CI Process

- Each commit gets checked for:
 - Formatting
 - Static Analysis
 - Transpilation Errors
 - Failing Unit Tests
 - Failing Functional Tests
 - Every commit to the Master branch gets:
 - All of the above
 - Deployed to the staging environment automatically
-

Code Coverage

- Dart code coverage is messed up - big time!
- But, there is a workaround, kinda..

 bugs chromium ▾ New issue Open issues ▾ 🔍 Search chromium issues...

 Starred by 4 users

Owner: caseq@chromium.org

Status: Assigned (*Open*)

Components: [Platform>DevTools](#)

Modified: 3 days ago

Issue 818806: Support mapping JS coverage to source file via source maps
Reported by kevmoo@google.com on Mon, Mar 5, 2018, 1:50 PM MST

UserAgent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like C

Steps to reproduce the problem:
1. Run coverage over code with source-maps enabled

Code Coverage Continued

- Total number of Dart lines: ~8,601 (Incl. generated files)
- Total covered Dart lines: ~1,779

Raw Coverage: 20.6%

- Unobservable Lines due to Dart 2 coverage bug: ~5,713

Adjusted Coverage: 60.8%

Methodology for Coverage

1. Counted all *.dart lines using bash scripting
2. Counted all lines in files with known coverage
3. Subtracted uncovered lines, *manually*
4. Counted all files which contained an import to 'dart:html'

```
lineCount="$(find lib -name '*.dart' -exec cat {} \; | sed '/^\s*$/d' | wc -l`)"
coveredFileList="lib/src/firebase/dBRefs.dart lib/src/model/*.dart lib/src/state/*.dart lib/src
coveredFileLineCount="$(find ${coveredFileList} -exec cat {} \; | sed '/^\s*$/d' | wc -l`)"
```

Demo Time!
