BDD with Redux

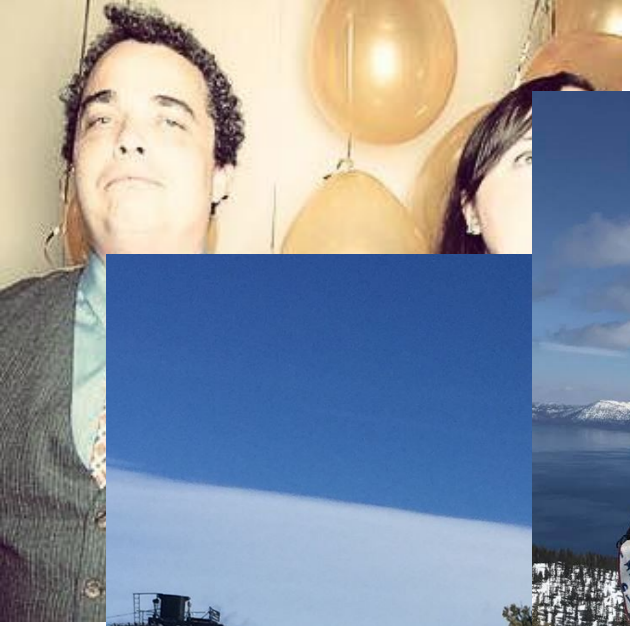
Alex Banks

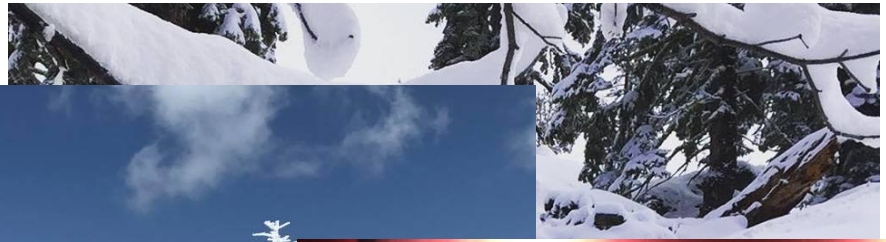
alex@moonhighway.com

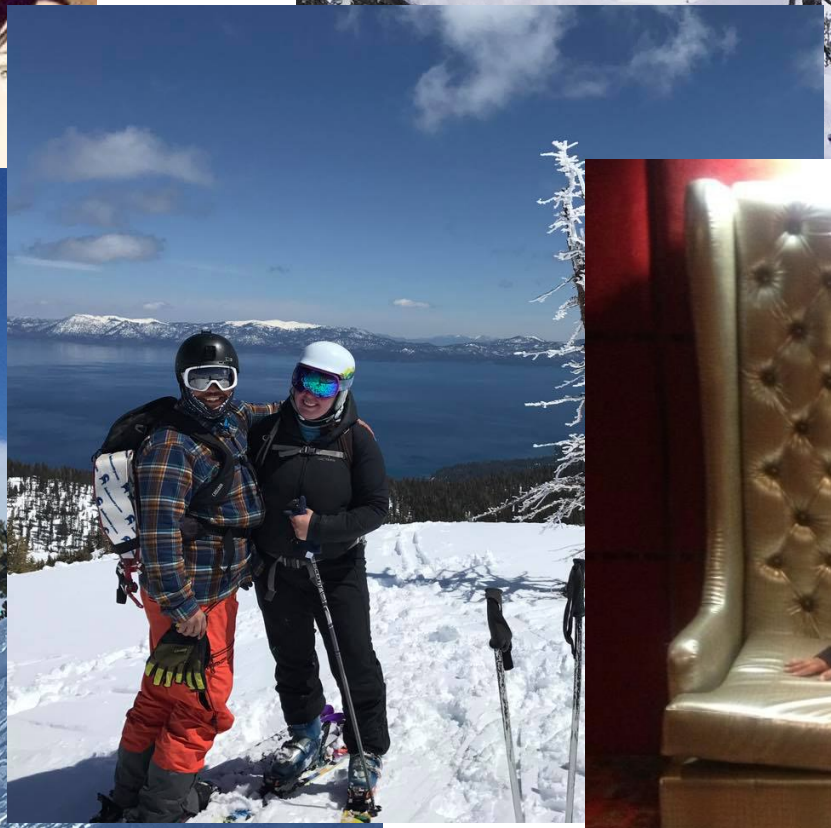
linkedin.com/in/banksalex

@moontahoe

Moon Highway



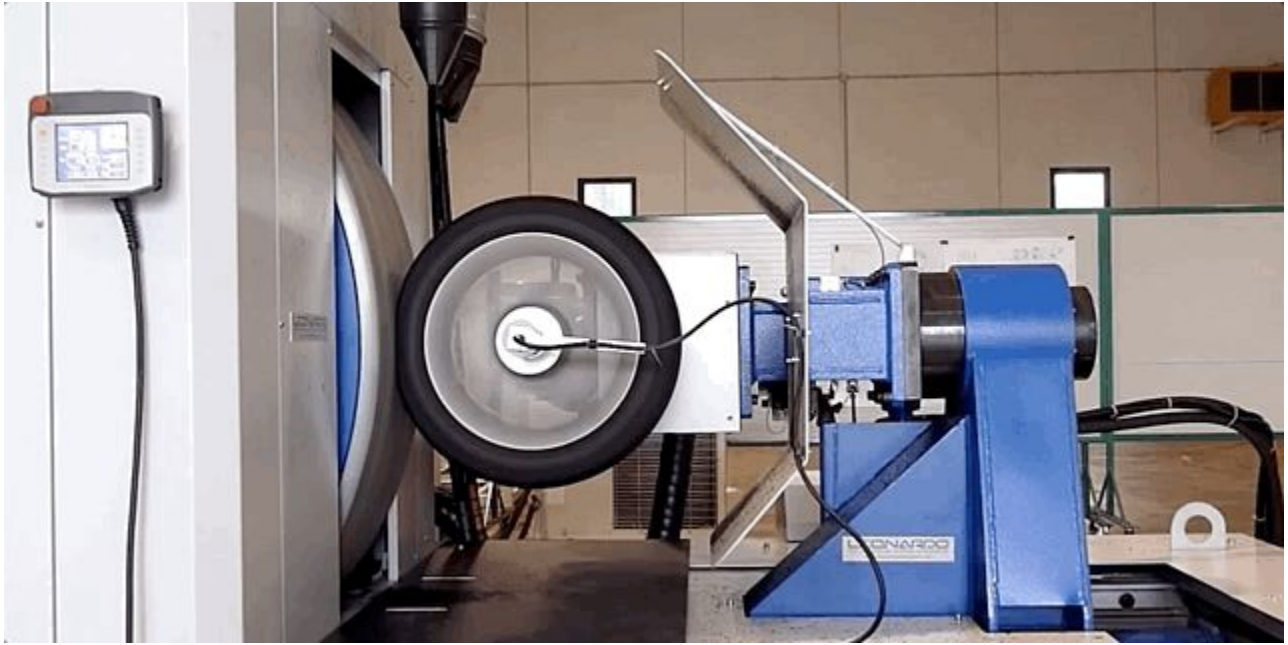





Moon
Highway

TDD Overview

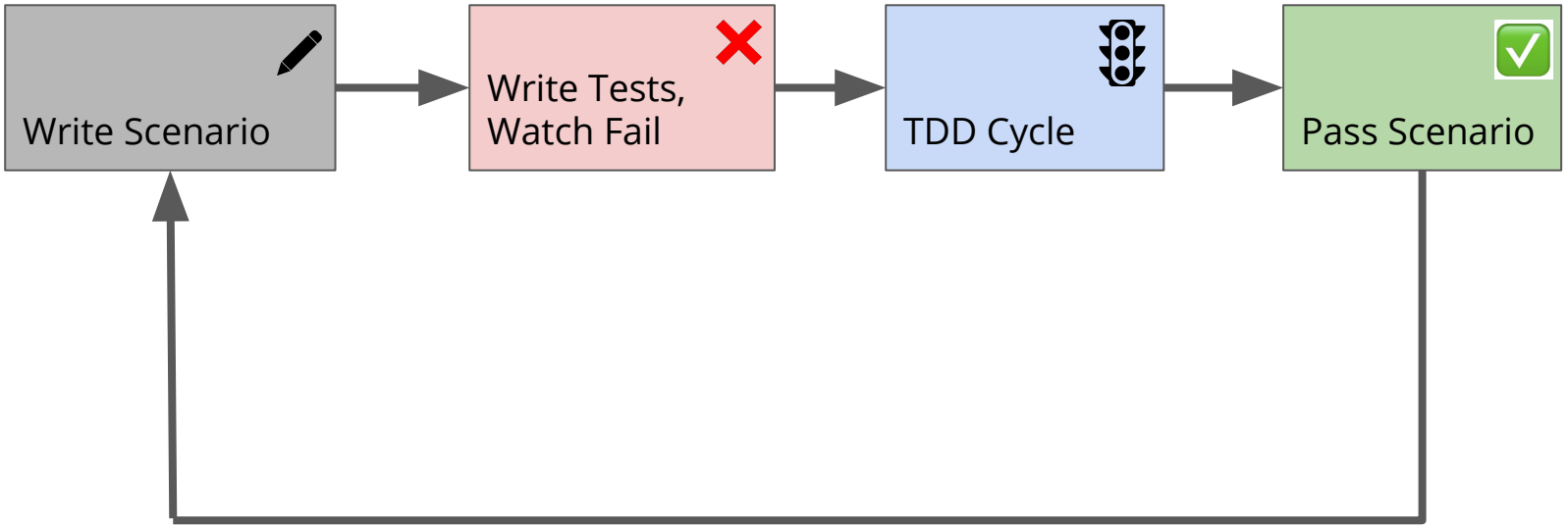
1. Write tests first
2. **Red:** Run tests, watch them fail
3. **Green:** Write ***minimal*** amount of code to make tests pass
4. **Gold:** Refactor code and tests

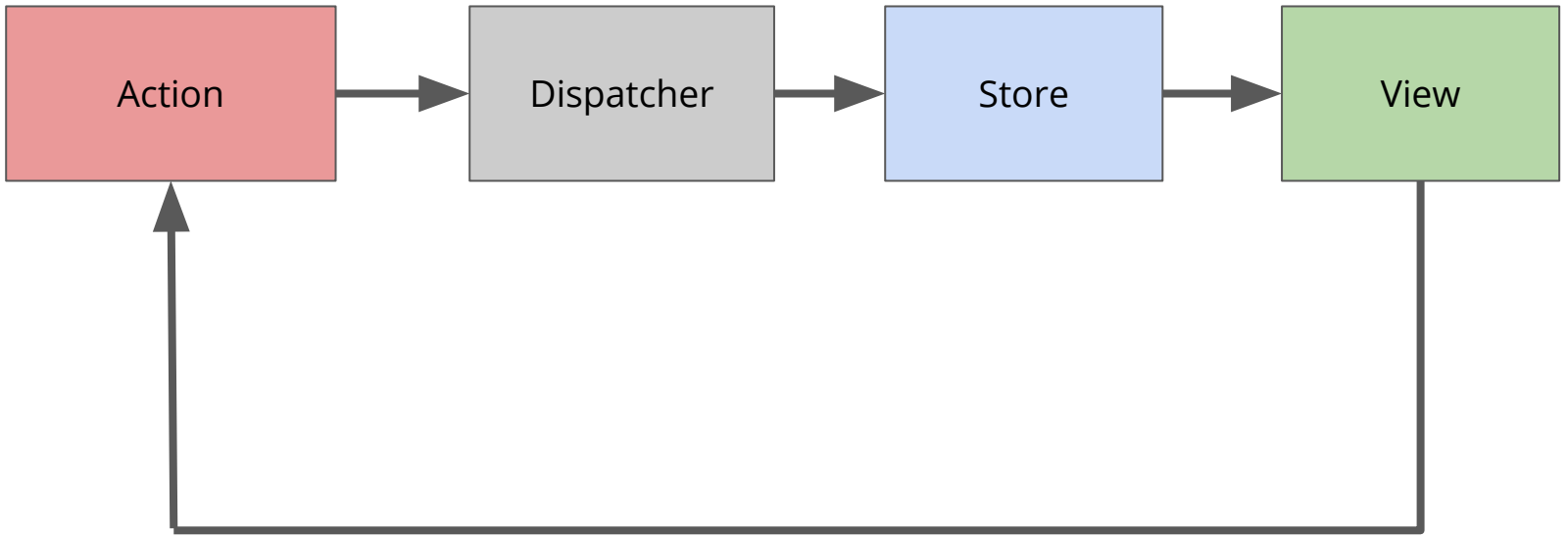

LEONARDO


2014 MAZDA 5
INSURANCE INSTITUTE
FOR HIGHWAY SAFETY
CEN1424

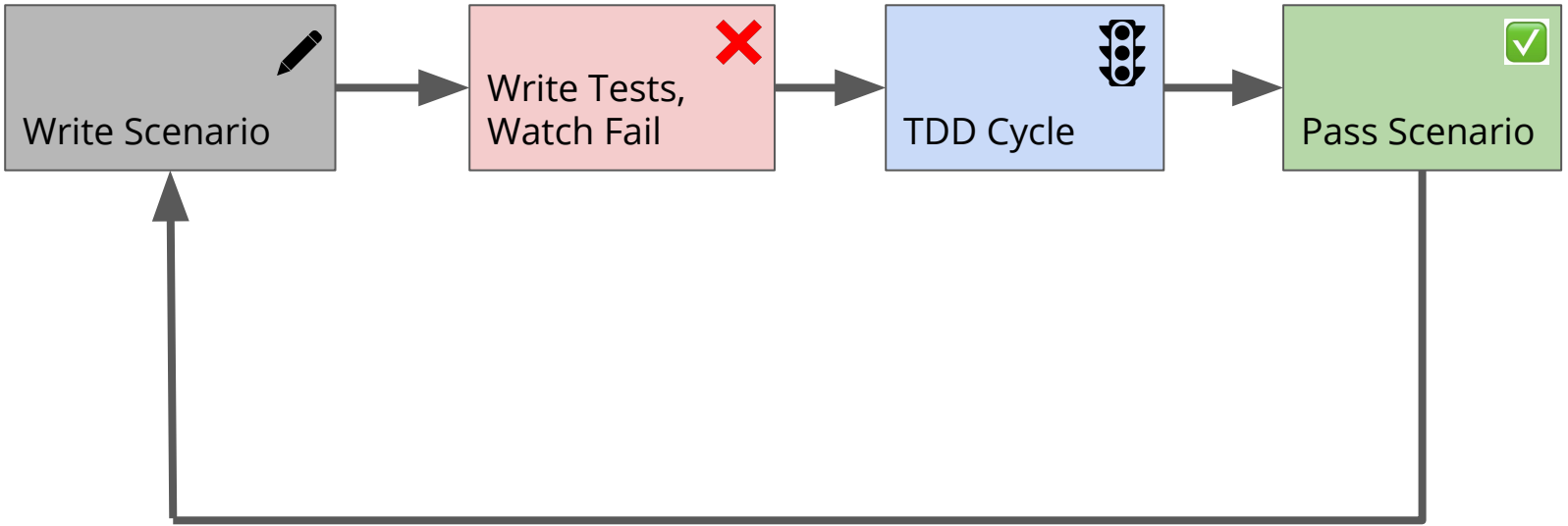
BDD Overview

1. Define the feature or scenario
2. Write the tests
3. **Red:** Run tests, watch them fail
4. **Green:** Write ***minimal*** amount of code/unit tests to make test pass
5. **Gold:** Refactor scenario and code


Write Scenario
Write Tests,
Watch Fail
TDD Cycle
Pass Scenario


Action
Dispatcher
Store
View


Do Not Enter


Write Scenario
Write Tests,
Watch Fail
TDD Cycle
Pass Scenario


Write Feature

Gherkin

```
Feature: Managing Colors
   As a color aficionado
   I would like to manage a list of colors
   So that I can demonstrate expertise
```

```
Scenario: Adding Colors
    Given I already have the following colors:
        | color     | hex     |
        | big blue  | #0000FF |
        | lawn      | #00FF00 |
        | crimson   | #FF0000 |
    When I create a store
    And I add the color:
        | party pink | #FFC0CB |
    Then I should have the following colors:
        | color      | hex     |
        | big blue   | #0000FF |
        | lawn       | #00FF00 |
        | crimson    | #FF0000 |
        | party pink | #FFC0CB |
```

```
Feature: Using the store

  As a web developer I want the ability

  to customize store instances

  So that I can add platform features
```

```
Scenario: Injecting Middleware
    Given I have a middleware
    When I create a store
    Then the store instance should execute the middleware
```

Gherkin Best Practices

1. Keep it simple
2. Keep it non-technical
3. Don’t think about how it should work
4. Think about what should happen
5. Make it readable


Write the Test


```
Scenario: Adding Colors
      Given I already have the following colors:
            | color     | hex     |
            | big blue  | #0000FF |
            | lawn      | #00FF00 |
            | crimson   | #FF0000 |
      When I create a store
      And I add the color:
            | party pink | #FFC0CB |
      Then I should have the following colors:
            | color      | hex     |
            | big blue   | #0000FF |
            | lawn       | #00FF00 |
            | crimson    | #FF0000 |
            | party pink | #FFC0CB |
```

```
Feature: Managing Colors

  Scenario: Adding Colors
    Given I have the following colors in state:
    When I create a store
    When I add the color:
    Then I should see the following colors:

1 scenario (1 passed)
4 steps (4 passed)
0m00.009s
✧ Done in 1.07s.
```

```
Scenario: Adding Colors
    Given I already have the following colors:
          | color     | hex     |
          | big blue  | #0000FF |
          | lawn      | #00FF00 |
          | crimson   | #FF0000 |
    When I create a store
    And I add the color:
          | party pink | #FFC0CB |
    Then I should have the following colors:
          | color      | hex     |
          | big blue   | #0000FF |
          | lawn       | #00FF00 |
          | crimson    | #FF0000 |
          | party pink | #FFC0CB |
```

```
Given('I already have the following colors:',
    function (dataTable) {
        const [, ...rows] = dataTable.rawTable
        const colors = rows.map(([name, hex]) => ({name, hex}))
        this.initState = {colors}
    }
)
```

```
Scenario: Adding Colors
    Given I already have the following colors:
          | color      | hex     |
          | big blue   | #0000FF |
          | lawn       | #00FF00 |
          | crimson    | #FF0000 |
    When I create a store
    And I add the color:
          | party pink | #FFC0CB |
    Then I should have the following colors:
          | color      | hex     |
          | big blue   | #0000FF |
          | lawn       | #00FF00 |
          | crimson    | #FF0000 |
          | party pink | #FFC0CB |
```

```
When('I create a store',
    function () {
        this.store = storeFactory(this.initState)
    }
)
```

```
Scenario: Adding Colors
    Given I already have the following colors:
          | color     | hex     |
          | big blue  | #0000FF |
          | lawn      | #00FF00 |
          | crimson   | #FF0000 |
    When I create a store
    And I add the color:
          | party pink | #FFC0CB |
    Then I should have the following colors:
          | color      | hex     |
          | big blue   | #0000FF |
          | lawn       | #00FF00 |
          | crimson    | #FF0000 |
          | party pink | #FFC0CB |
```

```
When('I add the color:',
   function (dataTable) {
      const [ name, hex ] = dataTable.rawTable[0]
      const { addColor } = actions
      this.store.dispatch(addColor(name, hex))
   }
)
```

```
Scenario: Adding Colors
    Given I already have the following colors:
          | color     | hex     |
          | big blue  | #0000FF |
          | lawn      | #00FF00 |
          | crimson   | #FF0000 |
    When I create a store
    And I add the color:
          | party pink | #FFC0CB |
    Then I should have the following colors:
          | color      | hex     |
          | big blue   | #0000FF |
          | lawn       | #00FF00 |
          | crimson    | #FF0000 |
          | party pink | #FFC0CB |
```

```
Then('I should have the following colors:',
   function (dataTable) {
         const [, ...rows] = dataTable.rawTable
         const expectedColors = rows.map(
            ([name, hex]) => ({name, hex})
         )
         const { colors } = this.store.getState()
         expect(colors).to.deep.equal(expectedColors)
   }
)
```

```
Scenario: Removing Colors
   Given I already have the following colors:
     | color     | hex     |
     | big blue  | #0000FF |
     | lawn      | #00FF00 |
     | crimson   | #FF0000 |
   When I create a store
   When I remove the color "lawn"
   Then I should have the following colors:
     | color       | hex     |
     | big blue    | #0000FF |
     | crimson     | #FF0000 |
```

```
Scenario: Removing Colors
  Given I already have the following colors:
    | color     | hex     |
    | big blue  | #0000FF |
    | lawn      | #00FF00 |
    | crimson   | #FF0000 |
  When I create a store
  When I remove the color "lawn"
  Then I should have the following colors:
    | color       | hex     |
    | big blue    | #0000FF |
    | crimson     | #FF0000 |
```

```
Scenario: Injecting Middleware
    Given I have a middleware
    When I create a store
    Then the store instance should execute the middleware
```

```
Given('I have a middleware',
    function () {
        this.testMiddlware = store => next => action => {
            this.middlwareSpy(action.type)
            return next(action)
        }
    }
)
```

```
When('I create a store',
    function () {
        const middleware = this.testMiddlware ?
            [this.testMiddlware] : null
        this.store = storeFactory(
            this.initState,
            middleware
        )
 })
```

```
Then('the store instance should execute my middleware',
    function () {
        this.store.dispatch({ type: 'TEST_ACTION' })
        assert.calledWith(this.middlwareSpy, 'TEST_ACTION')
    }
)
```

Cucumber Best Practices

1. Keep it simple
2. Mock components
3. Don’t worry about the details
4. Define how to interface with the component


TDD Cycle
CITY SPEED LIMIT 25
7 Ave SLOW ZONE
BUSES ONLY
W 41ST
ONE WAY

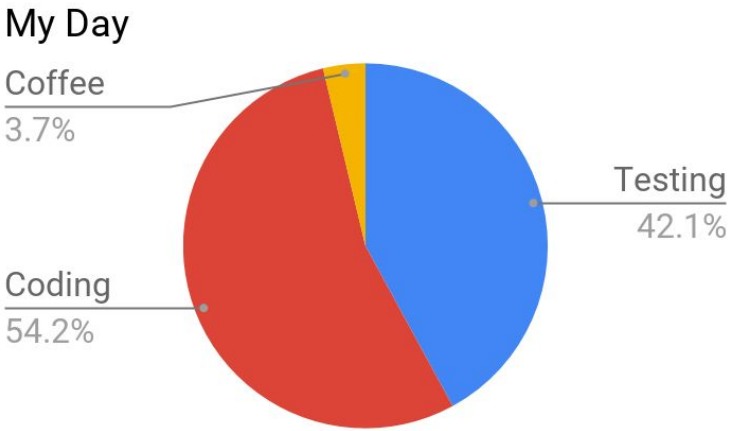
TDD Best Practices

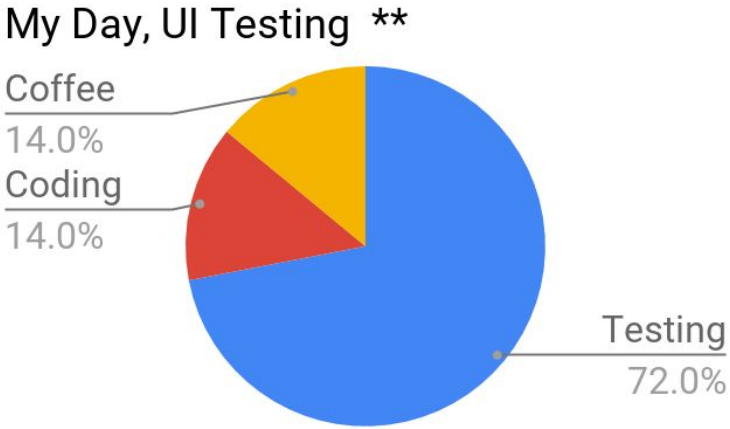
1. Now worry about specific details
2. Write only the tests that you need
3. Focus tests, mock units
4. Use the tests to guide you
5. Write only the code to make the scenario pass


Refactoring


UI Testing

How My Time Is Spent





*** Data based on detailed study of how I felt doing this.*

Jest Snapshots

2. node

```
FAIL __tests__/Link.react-test.js
  ● renders correctly

    expect(value).toMatchSnapshot()

    Received value does not match stored snapshot 1.

    - Snapshot
    + Received

     <a
       className="normal"
     -  href="http://www.facebook.com"
     +  href="http://www.instagram.com"
        onMouseEnter={[Function]}
        onMouseLeave={[Function]}>
     -  Facebook
     +  Instagram
      </a>

       at Object.<anonymous> (__tests__/Link.react-test.js:14:16)

  × renders correctly (10ms)

Snapshot Summary
 › 1 snapshot test failed in 1 test suite. Inspect your code changes or press `u` to update
```

App Components

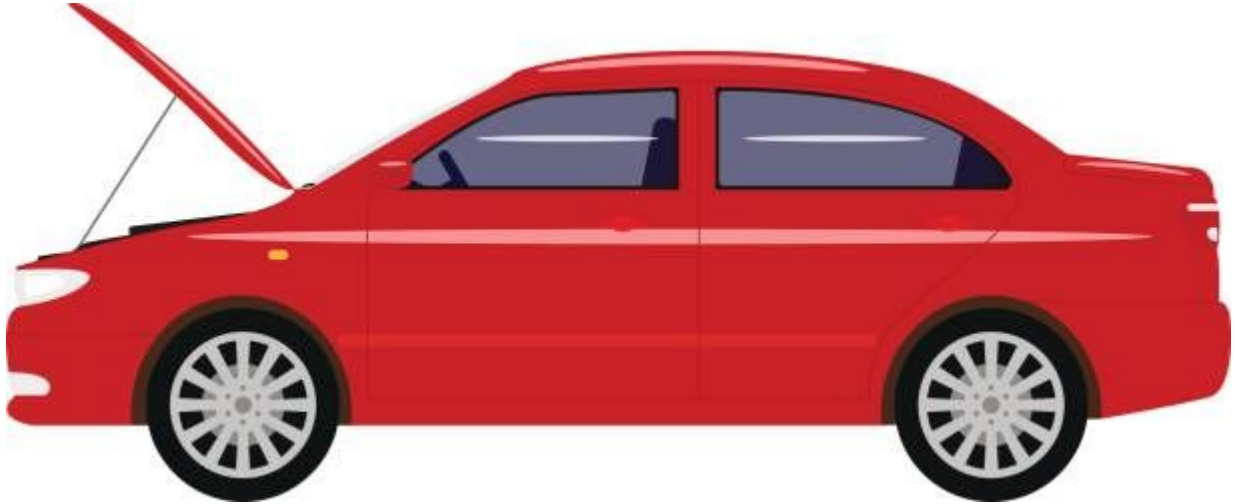
React
UI

Redux
Client

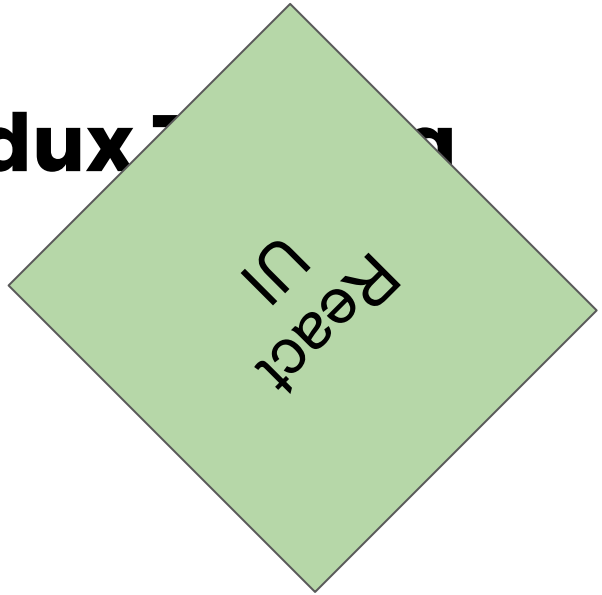
Backend
API

Database


RACING



Redux

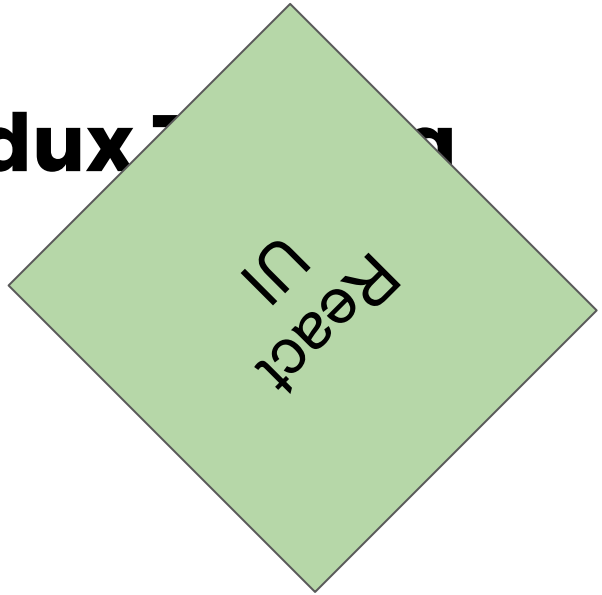


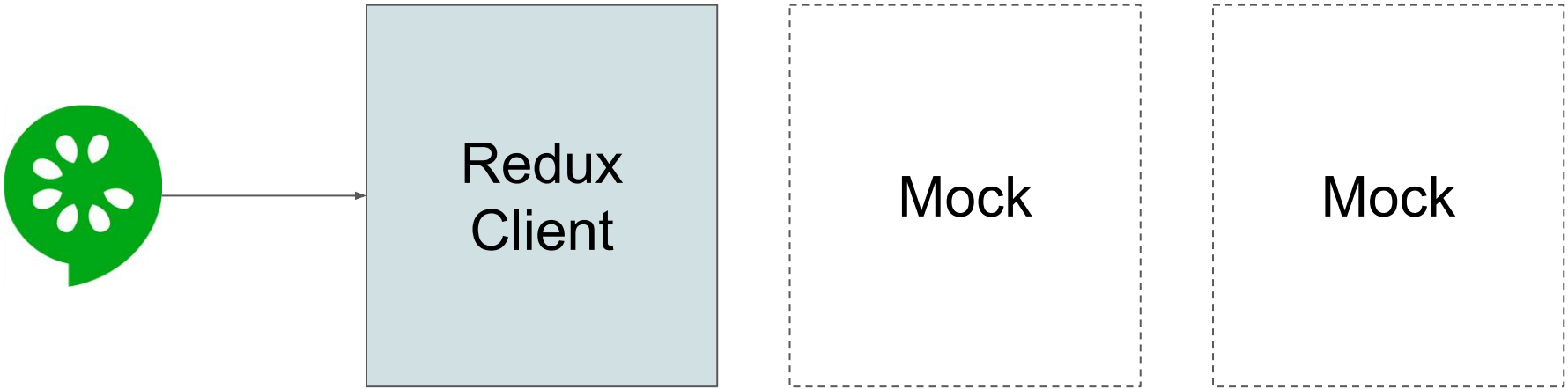
Redux Client

Backend API

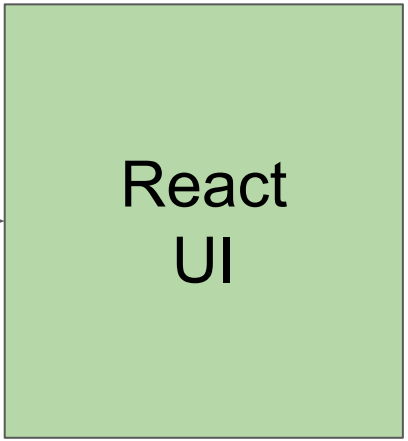
Database

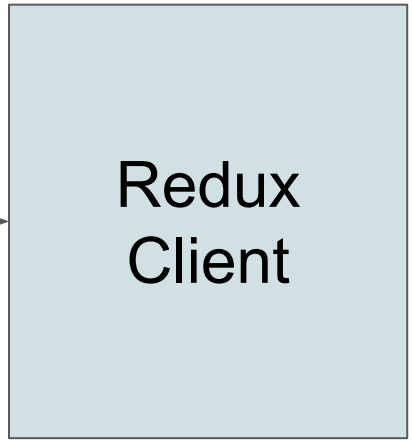
Redux T g

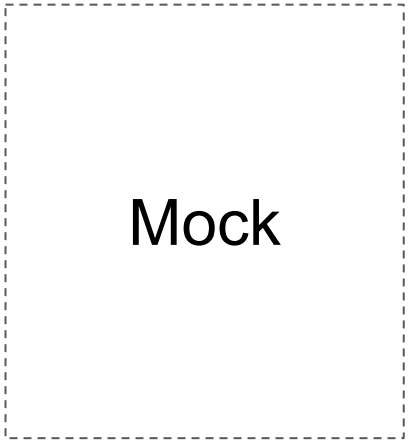


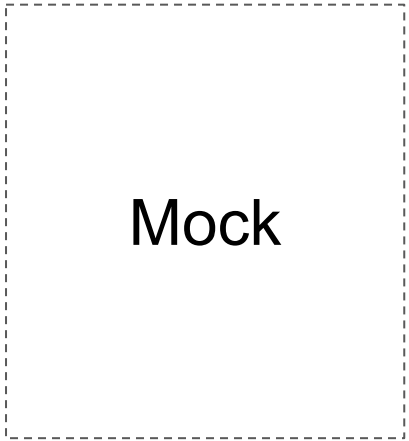


Redux Testing

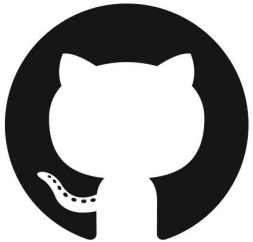








Thanks!



www.github.com/moontahoe/color-manager-redux

www.github.com/moontahoe/color-manager-react

Alex Banks: @moontahoe