AN EVENING WITH GUSTO

SCALING FRONTEND ARCHITECTURE
Hello, World!

- Full Stack Software Engineer
- Typing #WithGusto
- Musician, Ocean Enthusiast, Cyclist
BUILDING EXTENSIBLE UI IS HARD.
HOW DO WE CHOOSE?
MAINTAINING LARGE FRONTEEND APPLICATIONS

- Production application, large team
  - Very selective rewrites (we’ve got a business to run, yo!)
  - *Maintain quality
- Gusto’s journey to modern JavaScript
  - Choosing a framework
  - Formulating a build chain
APP DEMO
LET'S TAKE A WALK DOWN MEMORY LANE...
WE WANT TO TAKE ADVANTAGE OF THE BEST TOOLS AVAILABLE TO US.
IF IT AIN'T BROKE, DON'T FIX IT.
AN EVENING WITH GUSTO

SYMPTOMS OF PAIN

- No centralized application state
SYMPTOMS OF PAIN

- No centralized application state
- Inefficient DOM manipulation

```javascript
class AccountantHomeView extends Backbone.View
  template: HandlebarsTemplates["home/accountant_home"]
  className: 'home-accountant-home-view'

  initialize: (options) ->
    @user = options.user
    @companies = options.companies
    @companies.on('add', @render, @)
```
AN EVENING WITH GUSTO

SYMPTOMS OF PAIN

- No centralized application state
- Inefficient DOM manipulation
- Non-trivial composability
AN EVENING WITH GUSTO

SYMPTOMS OF PAIN

- No centralized application state
- Inefficient DOM manipulation
- Non-trivial composability
- Is this the tool to use for the next several years?
INSPIRING CHANGE

- Pick your battles
- Research & Experiment
- Create an execution/migration plan
- *Get buy-in from the team
AN EVENING WITH GUSTO

GOALS

- Incremental migration
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design
AN EVENING WITH GUSTO

GOALS

- *Incremental migration*
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design

```javascript
class BenefitsRouter extends Backbone.Router
  routes:
    'benefits': 'benefits'

benefits: (highlight) =>
  view = React.createElement(BenefitsPage)
  AppView.showView(view, 'company-benefits')
```
AN EVENING WITH GUSTO

GOALS

- **Incremental migration**
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design

```javascript
class BenefitsView extends Backbone.View {
    className: 'employee-benefits-view',
    template: HandlebarsTemplates['employees/benefits'],
    mixins: [ReactSubComponent]

    render: () => {
        template = @template(@context())
        @el.html(template)

        // Append React Section
        view = React.createElement(
            EmployeeBenefits,
            employeeId: @model.id
        )
        $el = $('<div />').appendTo(@$('#employee-benefits'))
        @renderSubComponent(view, $el.get(0))
    }

    tearDownSubviews: () => {
        super
        @unmountSubComponents()
    }
```
AN EVENING WITH GUSTO

GOALS

- Incremental migration
- **Declarative**
- Encapsulated/Composable
- Testable
- Interface well with Design
AN EVENING WITH GUSTO

GOALS

- Incremental migration
- *Declarative*
- Encapsulated/Composable
- Testable
- Interface well with Design

```javascript
const StateFilingRequirementsView = React.createClass({
  _minReasonableSUIPercentage: 0.1,

  getInitialState() { return {}; },

  render() {
    return (
      <div>
        <PercentInput valueLink={{this.linkState('sui_rate')}} />
        { this.state.sui_rate && this.renderSuiWarning() }
      </div>
    );
  },

  renderSuiWarning() {
    if (this.state.sui_rate < this._minReasonableSUIPercentage) {
      return (
        <div className="sui-rate-low alert alert-warning hide">
          <em>Please double check:</em>
          The amount you entered appears a bit low. Please take another moment to verify that you really intended to input '{this.state.sui_rate}% { this.state.sui_rate !== 0 && 'and not ${this.state.sui_rate * 100}%'.
        </div>
      );
    }
  }
});
```
AN EVENING WITH GUSTO

GOALS

▸ Incremental migration
▸ Declarative
▸ Encapsulated/Composable
▸ Testable
▸ Interface well with Design

```
const Ssn = React.createClass({
  propTypes: {
    value: React.PropTypes.string.isRequired,
    masked: React.PropTypes.bool
  },

  getDefaultProps() {
    return { masked: true };  
  },

  _normalizedValue() {
    return this.props.value.replace(/-/g, '');
  },

  _formattedValue() {
    const ssn = this._normalizedValue();
    return `${ssn.slice(0, 3)}-${ssn.slice(3, 5)}-${ssn.slice(5, 9)} `;
  },

  _maskedValue() {
    return `XXX-XX-`${this._normalizedValue().slice(5, 9)} `;
  },

  render() {
    return (
      <span>
        { this.props.masked ? this._maskedValue() : this._formattedValue() }
      </span>
    );
  }
});
```
AN EVENING WITH GUSTO

GOALS

▸ Incremental migration
▸ Declarative
▸ Encapsulated/Composable
▸ Testable
▸ Interface well with Design

const Profile = React.createClass({
  render() {
    return (  
      <div>
        <FullName
          firstName={this.props.firstName}
          lastName={this.props.lastName}
          />
        <Email value={this.props.email} />
        <Ssn value={this.props.ssn} />
      </div>
    );
  
});
AN EVENING WITH GUSTO

GOALS

- Incremental migration
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design
AN EVENING WITH GUSTO

GOALS

- Incremental migration
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design

```javascript
const Blocked = React.createClass({
  render() {
    return (
      <PageContainer>
        <PageContainer.Header>
          Add Health Benefits
        </PageContainer.Header>
        
        <PageContainer.Body>
          <LayoutSimple>
            <LayoutSimple.Header icon='icons/color-line/icon-color-store.svg'>
              {'Tell Us Where You\'re Located'}
            </LayoutSimple.Header>

            <LayoutSimple.Paragraph>
              We'd love to help you set up health benefits for your team! In order to provide you with the best experience, please finish adding company addresses to Gusto (step 1 of onboarding) and let us know where your company is located.
            </LayoutSimple.Paragraph>

            <LayoutSimple.Link
              icon='plus'
              label='Add Addresses'
              href='/company/locations'
            />
          </LayoutSimple>
        </PageContainer.Body>
      </PageContainer>
    );
  }
});
```
YAY!
CONTINUOUS IMPROVEMENT
HOW DO WE LOAD JAVASCRIPT ON A PAGE?

- Write individual files, concatenate them together for use by the browser
  - Compile and Polyfill
- In development:
  - Compile on every file change
- In production:
  - Digest Assets
  - Minify
WHAT CAN YOUR BUNDLE MANAGER DO FOR YOU?

- Dependency management
- Bundle optimization (lazy loading, common package caching, tree shaking)
- Hot loading
- Fast testing
Development had ground to halt, refreshes could take minutes

Subject to changes on the window

Little bundle optimization we could do

Slow testing

JavaScript was effectively a second class citizen
AN EVENING WITH GUSTO

GOALS

▸ Incremental migration
▸ Speed, speed, speed
▸ Better development tooling
▸ Move towards modular JavaScript
▸ Bundle optimization
▸ Move all assets off of Rails
DEMO
WHAT DID WE ACCOMPLISH?

- Implemented React with Flux over Backbone
- Huge speed/productivity boosts with move to Webpack
- Rallied engineering team behind these upgrades
- What’s next?
  - Bundle optimization
  - Break out the assets to be standalone
  - Optimize build speed with Webpack tools
THANK YOU!
Q&A