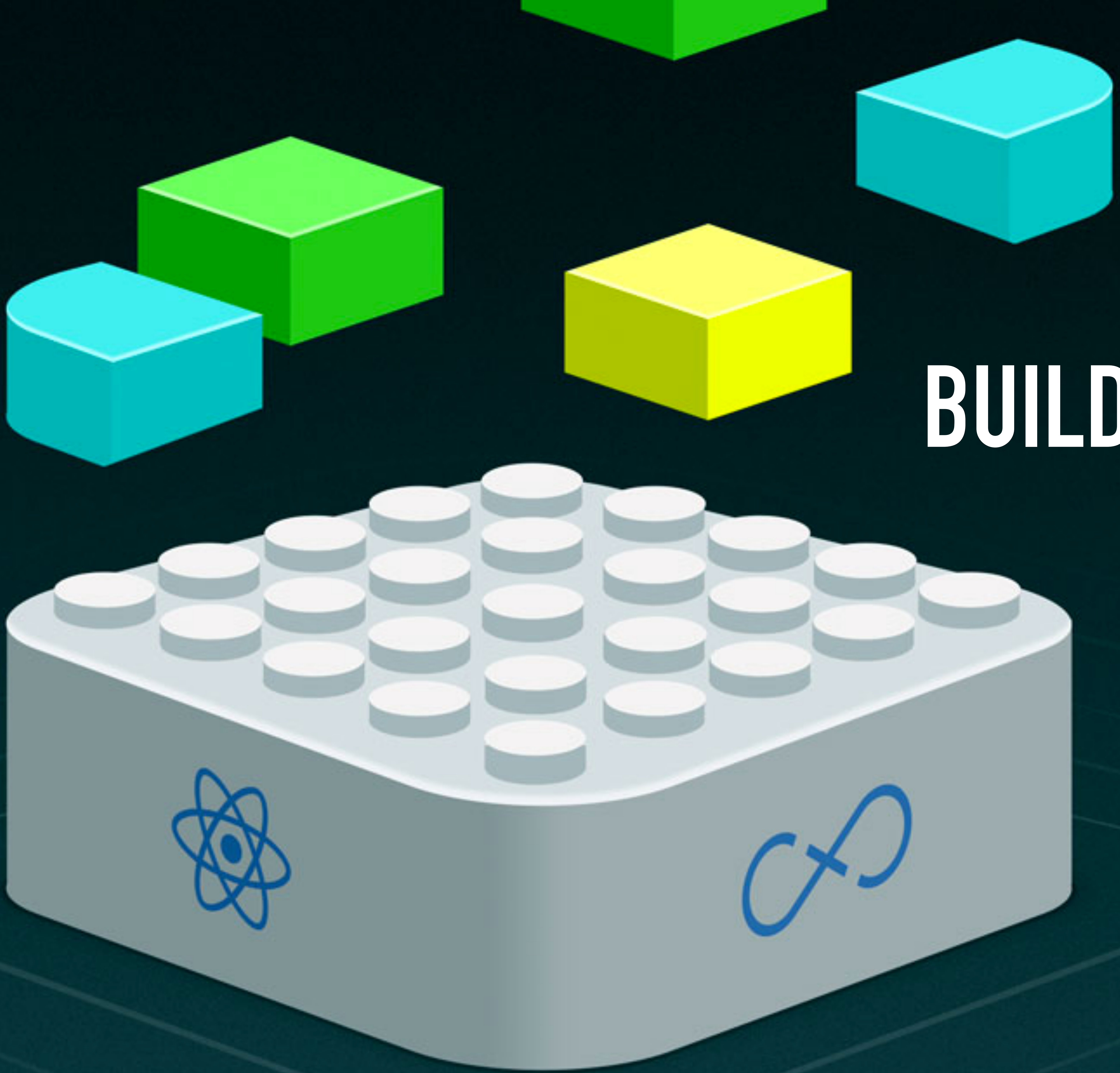
AN EVENING WITH GUSTO

SCALING FRONTEND ARCHITECTURE

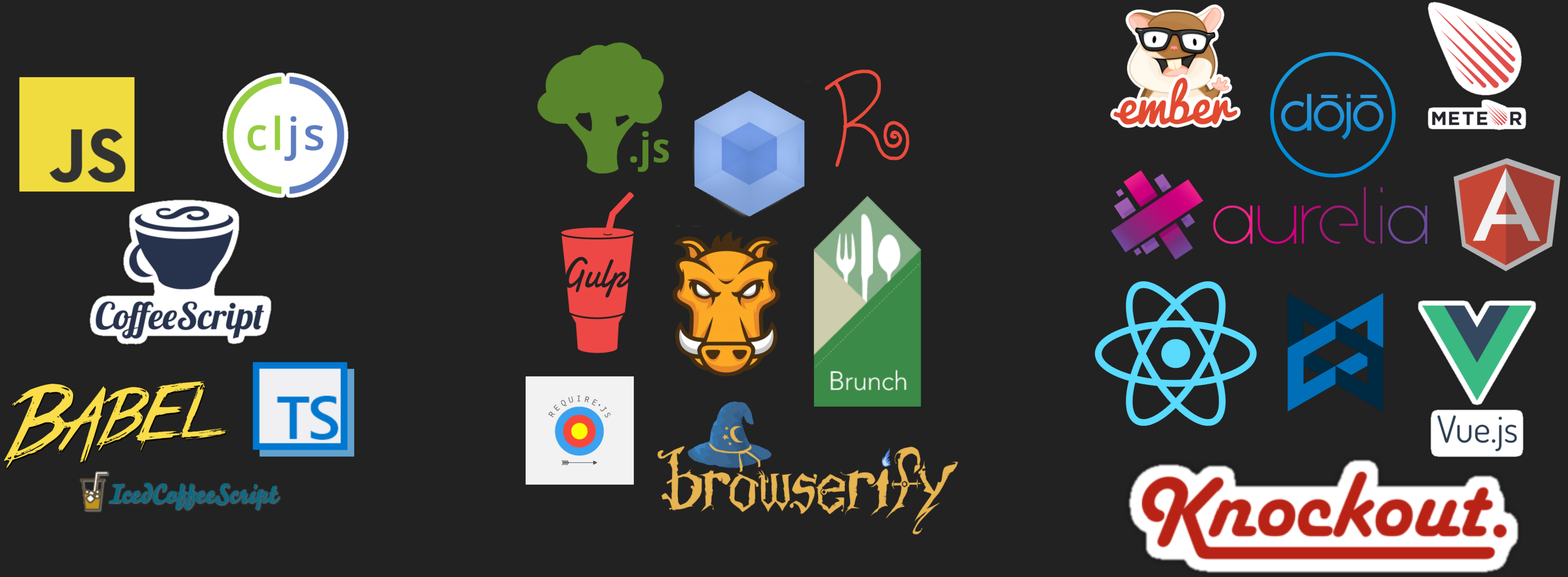
HELLO, WORLD!

- Full Stack Software Engineer
- Typing #WithGusto
- Musician, Ocean Enthusiast, Cyclist



BUILDING EXTENSIBLE UI IS HARD.

STATE OF JAVASCRIPT



HOW DO WE CHOOSE?

MAINTAINING LARGE FRONTEND 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...

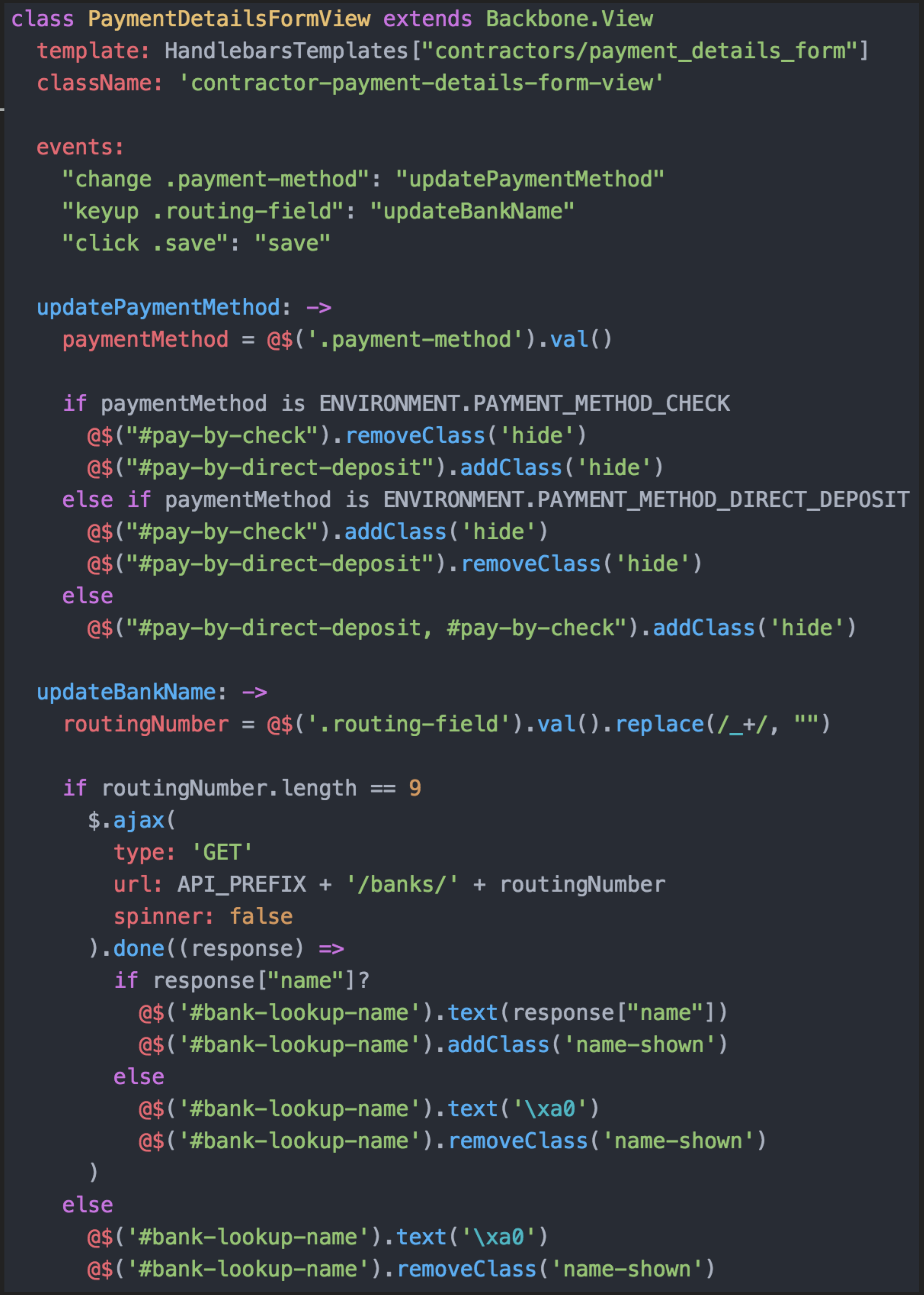
CoffeeScript
handlebars
jQuery

WE WANT TO TAKE ADVANTAGE OF THE BEST TOOLS AVAILABLE TO US.

IF IT AIN'T BROKE, DON'T FIX IT.

SYMPTOMS OF PAIN

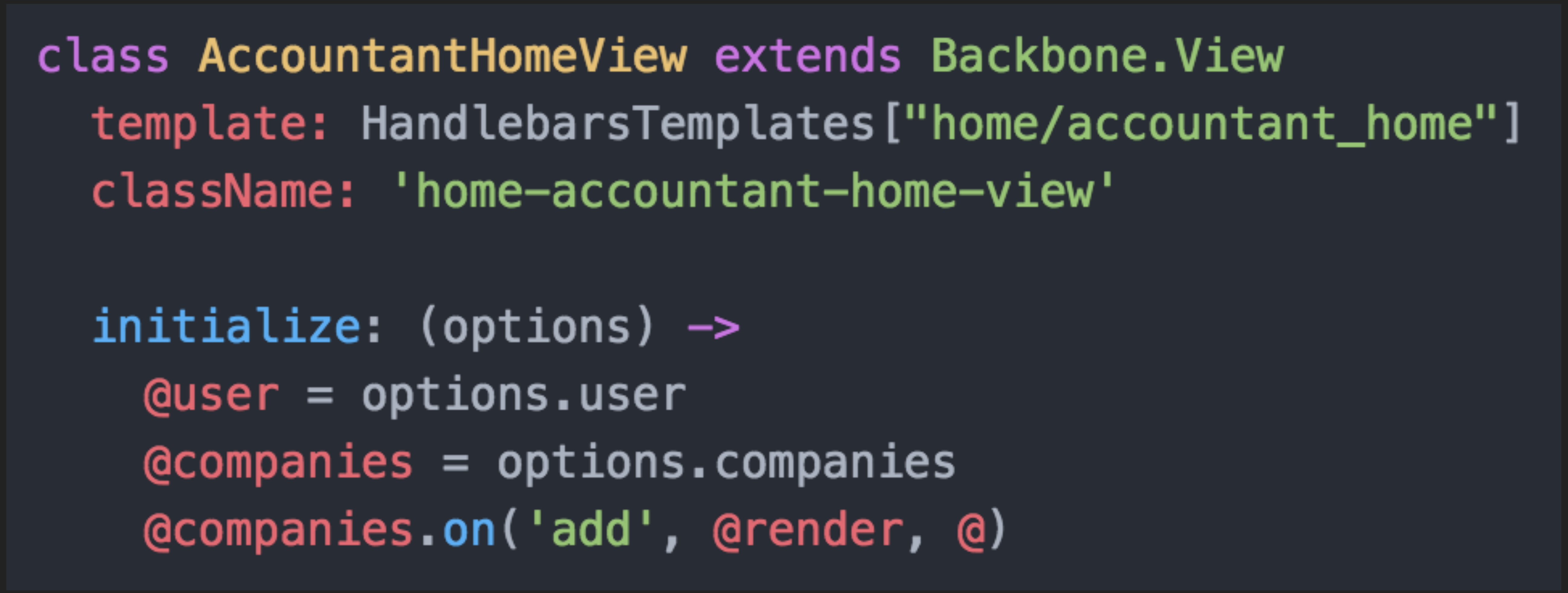
- No centralized application state



```coffeescript
class PaymentDetailsFormView extends Backbone.View
  template: HandlebarsTemplates["contractors/payment_details_form"]
  className: 'contractor-payment-details-form-view'

  events:
    "change .payment-method": "updatePaymentMethod"
    "keyup .routing-field": "updateBankName"
    "click .save": "save"

  updatePaymentMethod: ->
    paymentMethod = @$('.payment-method').val()

    if paymentMethod is ENVIRONMENT.PAYMENT_METHOD_CHECK
      @$("#pay-by-check").removeClass('hide')
      @$("#pay-by-direct-deposit").addClass('hide')
    else if paymentMethod is ENVIRONMENT.PAYMENT_METHOD_DIRECT_DEPOSIT
      @$("#pay-by-check").addClass('hide')
      @$("#pay-by-direct-deposit").removeClass('hide')
    else
      @$("#pay-by-direct-deposit, #pay-by-check").addClass('hide')

  updateBankName: ->
    routingNumber = @$('.routing-field').val().replace(/_+/, "")

    if routingNumber.length == 9
      $.ajax(
        type: 'GET'
        url: API_PREFIX + '/banks/' + routingNumber
        spinner: false
      ).done((response) =>
        if response["name"]?
          @$('#bank-lookup-name').text(response["name"])
          @$('#bank-lookup-name').addClass('name-shown')
        else
          @$('#bank-lookup-name').text('\xa0')
          @$('#bank-lookup-name').removeClass('name-shown')
      )
    else
      @$('#bank-lookup-name').text('\xa0')
      @$('#bank-lookup-name').removeClass('name-shown')
```

SYMPTOMS OF PAIN

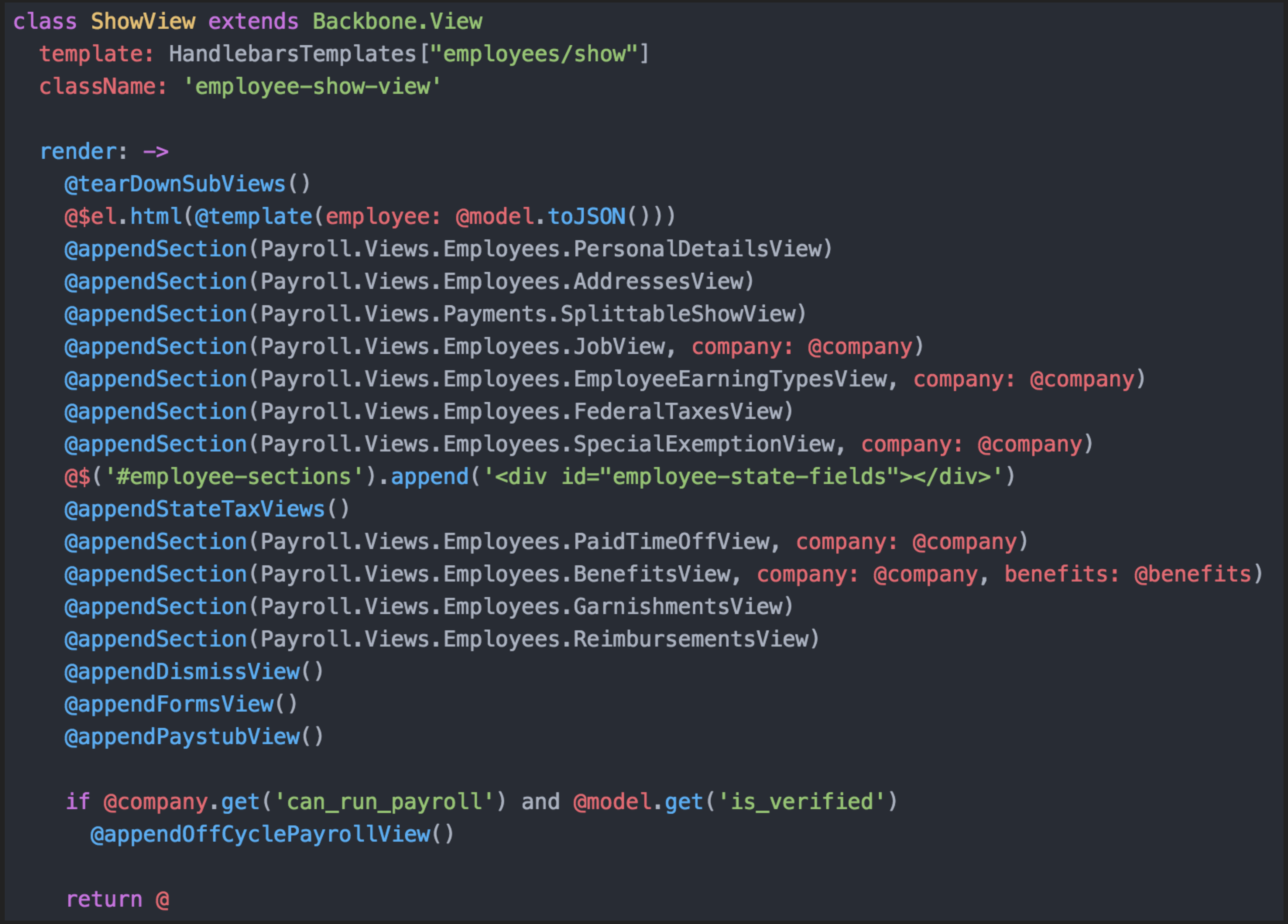
- No centralized application state
- Inefficient DOM manipulation



```
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, @)
```

SYMPTOMS OF PAIN

- No centralized application state
- Inefficient DOM manipulation
- Non-trivial composability



```
class ShowView extends Backbone.View
  template: HandlebarsTemplates["employees/show"]
  className: 'employee-show-view'

  render: ->
    @tearDownSubViews()
    @$el.html(@template(employee: @model.toJSON()))
    @appendSection(Payroll.Views.Employees.PersonalDetailsView)
    @appendSection(Payroll.Views.Employees.AddressesView)
    @appendSection(Payroll.Views.Payments.SplittableShowView)
    @appendSection(Payroll.Views.Employees.JobView, company: @company)
    @appendSection(Payroll.Views.Employees.EmployeeEarningTypesView, company: @company)
    @appendSection(Payroll.Views.Employees.FederalTaxesView)
    @appendSection(Payroll.Views.Employees.SpecialExemptionView, company: @company)
    @$('#employee-sections').append('<div id="employee-state-fields"></div>')
    @appendStateTaxViews()
    @appendSection(Payroll.Views.Employees.PaidTimeOffView, company: @company)
    @appendSection(Payroll.Views.Employees.BenefitsView, company: @company, benefits: @benefits)
    @appendSection(Payroll.Views.Employees.GarnishmentsView)
    @appendSection(Payroll.Views.Employees.ReimbursementsView)
    @appendDismissView()
    @appendFormsView()
    @appendPaystubView()

    if @company.get('can_run_payroll') and @model.get('is_verified')
      @appendOffCyclePayrollView()

    return @
```

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

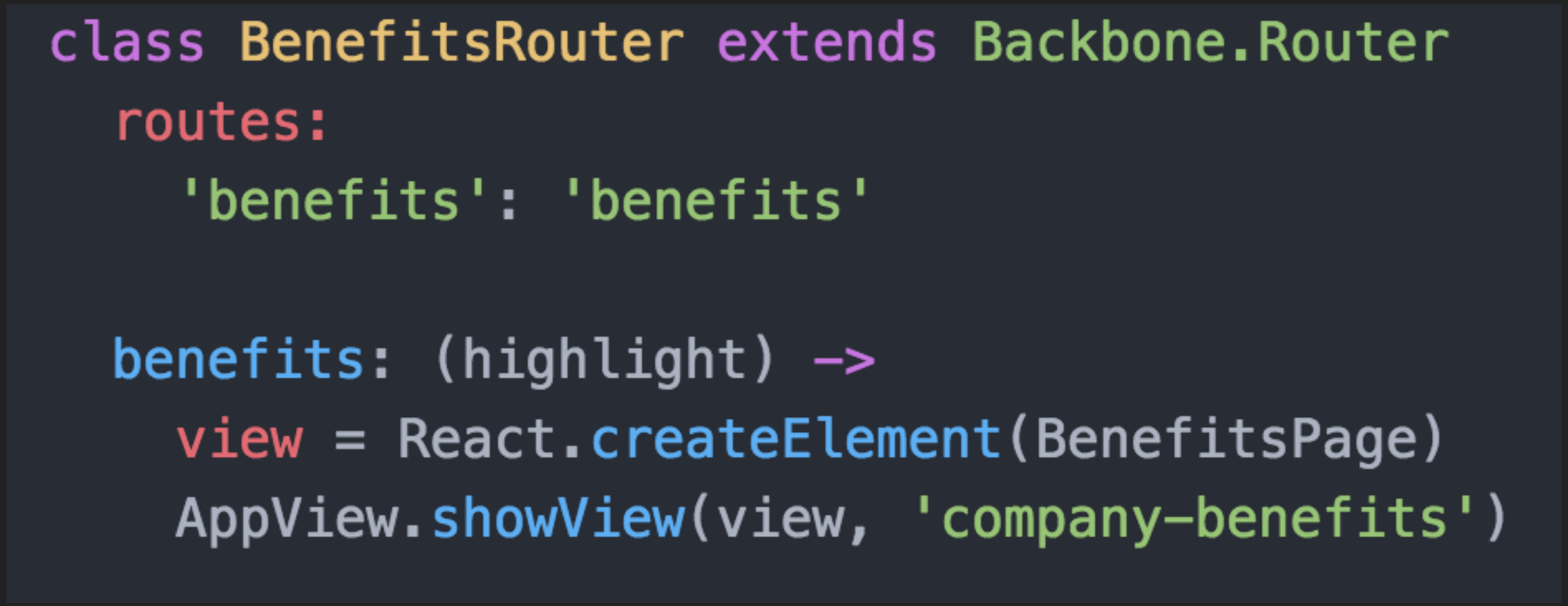
GOALS

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

React

GOALS

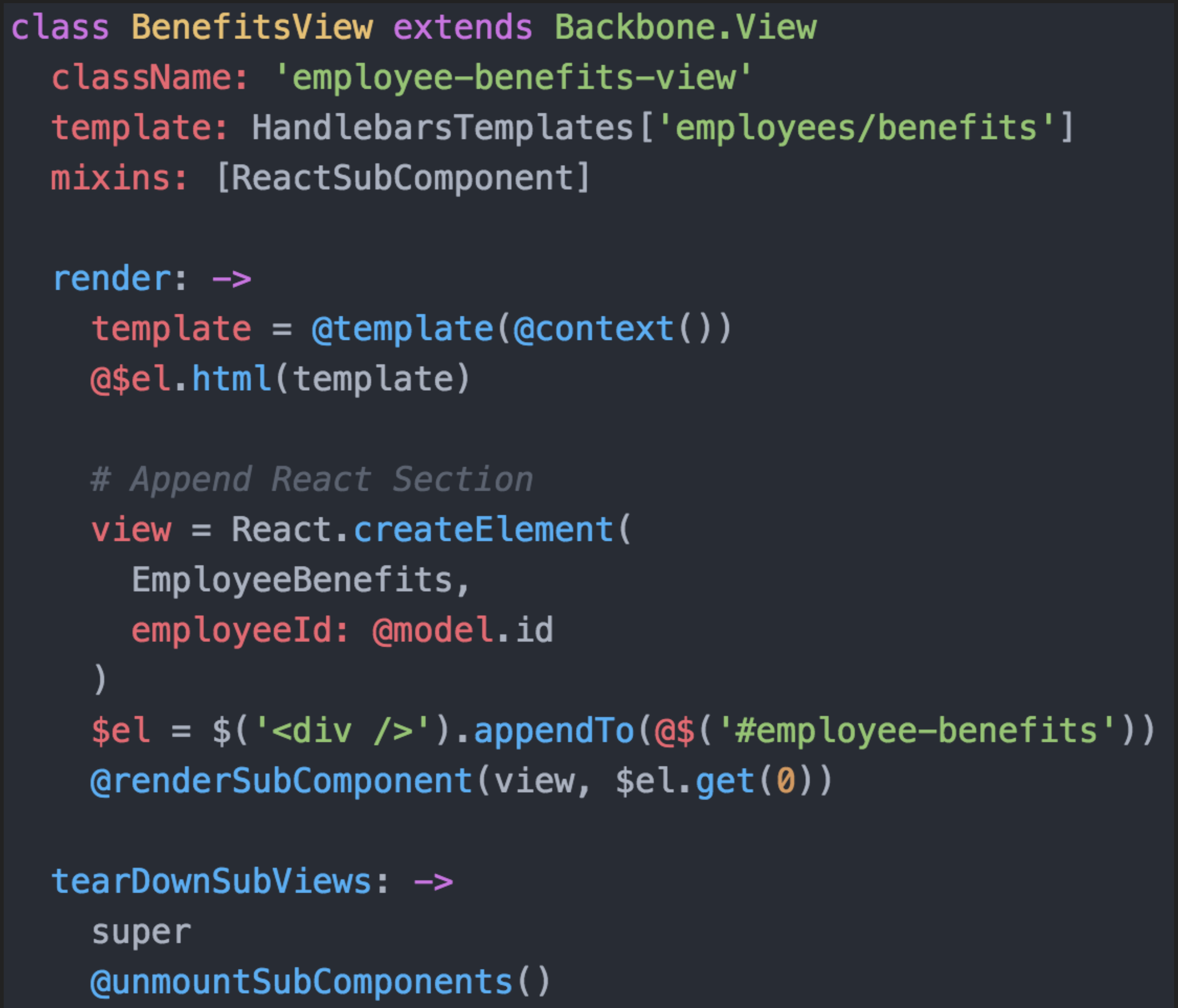
- *Incremental migration*
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design



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

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

GOALS

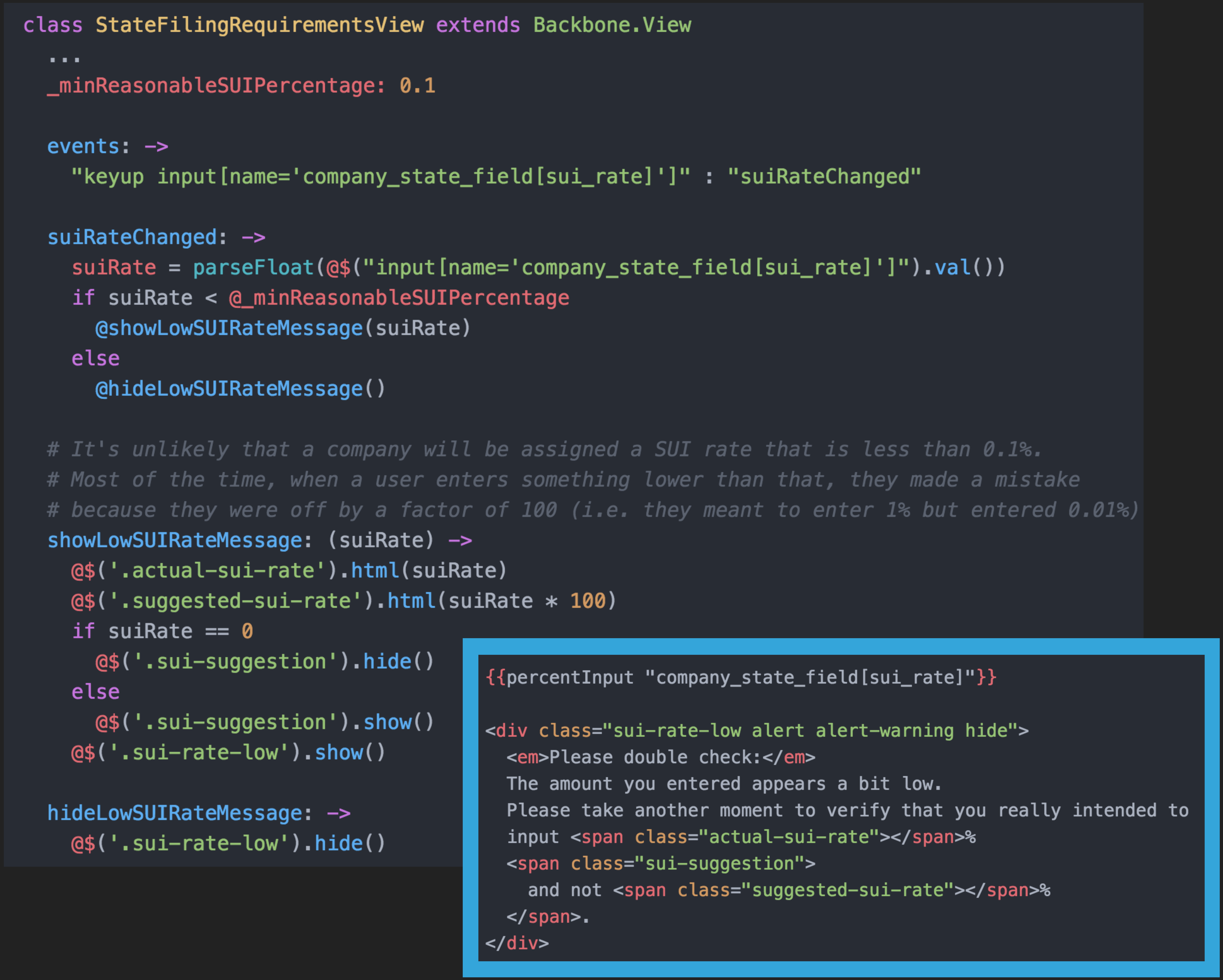
- *Incremental migration*
- Declarative
- Encapsulated/Composable
- Testable
- Interface well with Design



```
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()
```

GOALS

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

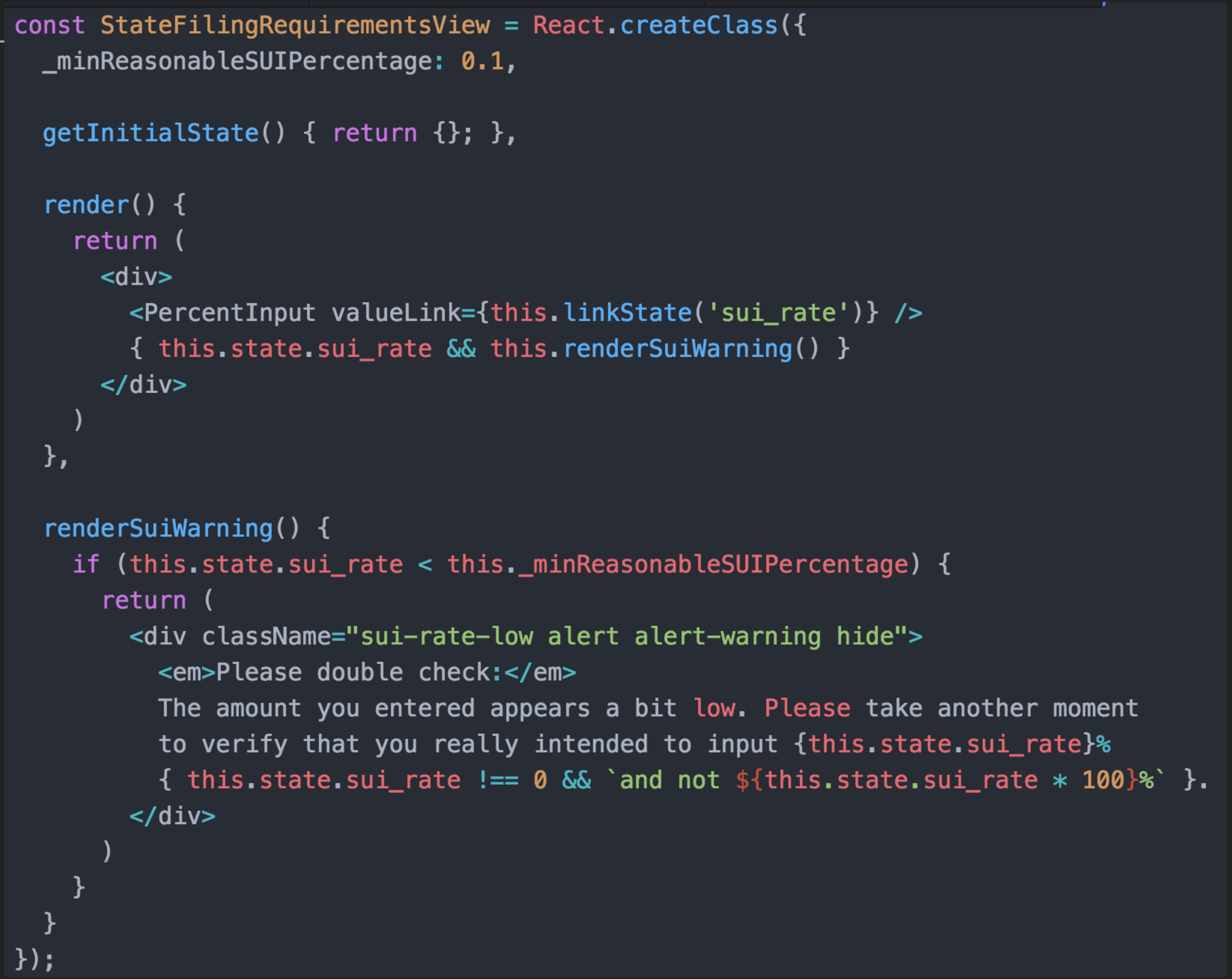


```
class StateFilingRequirementsView extends Backbone.View
  ...
  _minReasonableSUIPercentage: 0.1

  events: ->
    "keyup input[name='company_state_field[sui_rate]']" : "suiRateChanged"

  suiRateChanged: ->
    suiRate = parseFloat(@$("input[name='company_state_field[sui_rate]']").val())
    if suiRate < @_minReasonableSUIPercentage
      @showLowSUIRateMessage(suiRate)
    else
      @hideLowSUIRateMessage()

  # It's unlikely that a company will be assigned a SUI rate that is less than 0.1%.
  # Most of the time, when a user enters something lower than that, they made a mistake
  # because they were off by a factor of 100 (i.e. they meant to enter 1% but entered 0.01%)
  showLowSUIRateMessage: (suiRate) ->
    @$('.actual-sui-rate').html(suiRate)
    @$('.suggested-sui-rate').html(suiRate * 100)
    if suiRate == 0
      @$('.sui-suggestion').hide()
    else
      @$('.sui-suggestion').show()
    @$('.sui-rate-low').show()

  hideLowSUIRateMessage: ->
    @$('.sui-rate-low').hide()
```

```
{{percentInput "company_state_field[sui_rate]"}}

<div class="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 <span class="actual-sui-rate"></span>%
  <span class="sui-suggestion">
    and not <span class="suggested-sui-rate"></span>%
  </span>.
</div>
```

GOALS

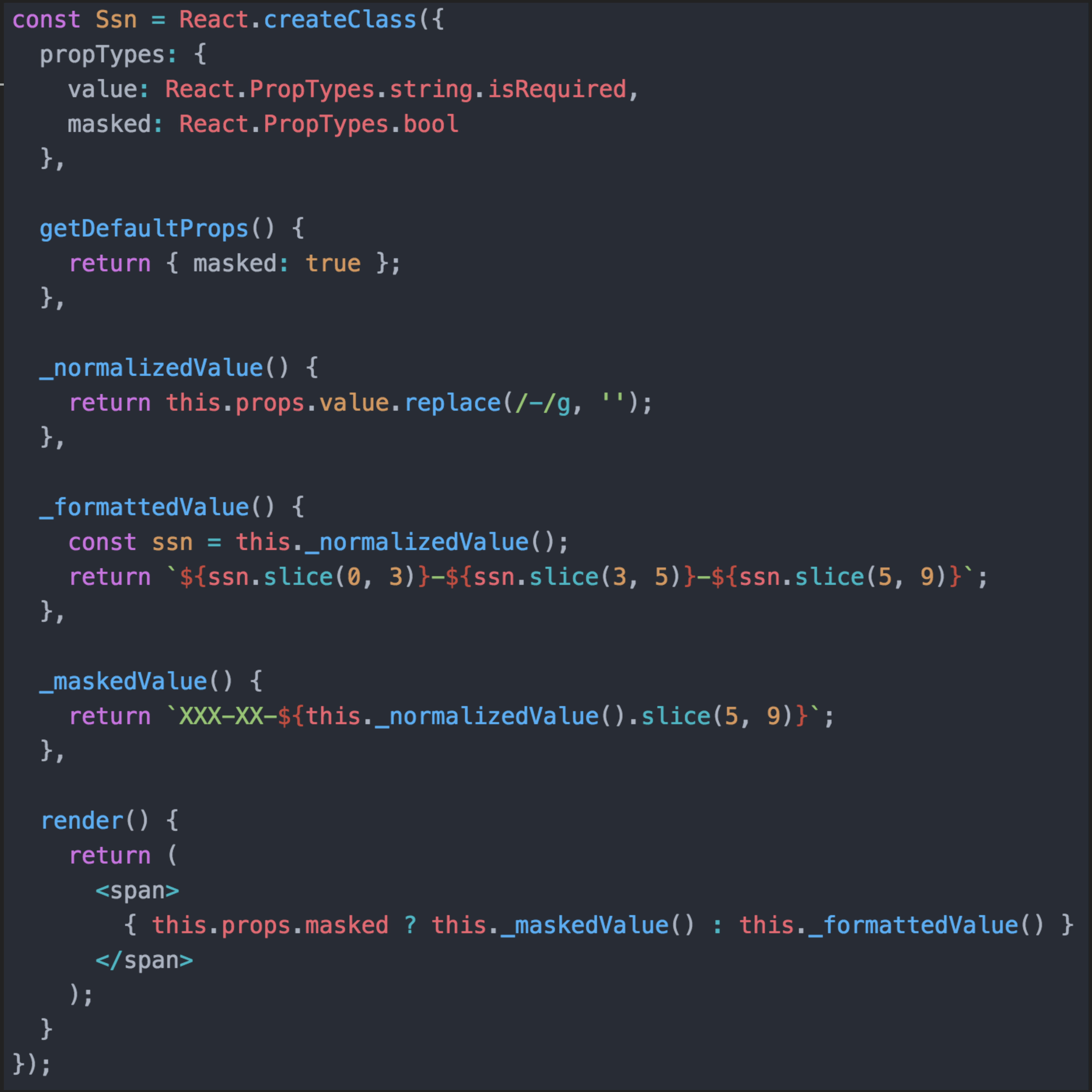
- Incremental migration
- *Declarative*
- Encapsulated/Composable
- Testable
- Interface well with Design



```
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>
      )
    }
  }
});
```

GOALS

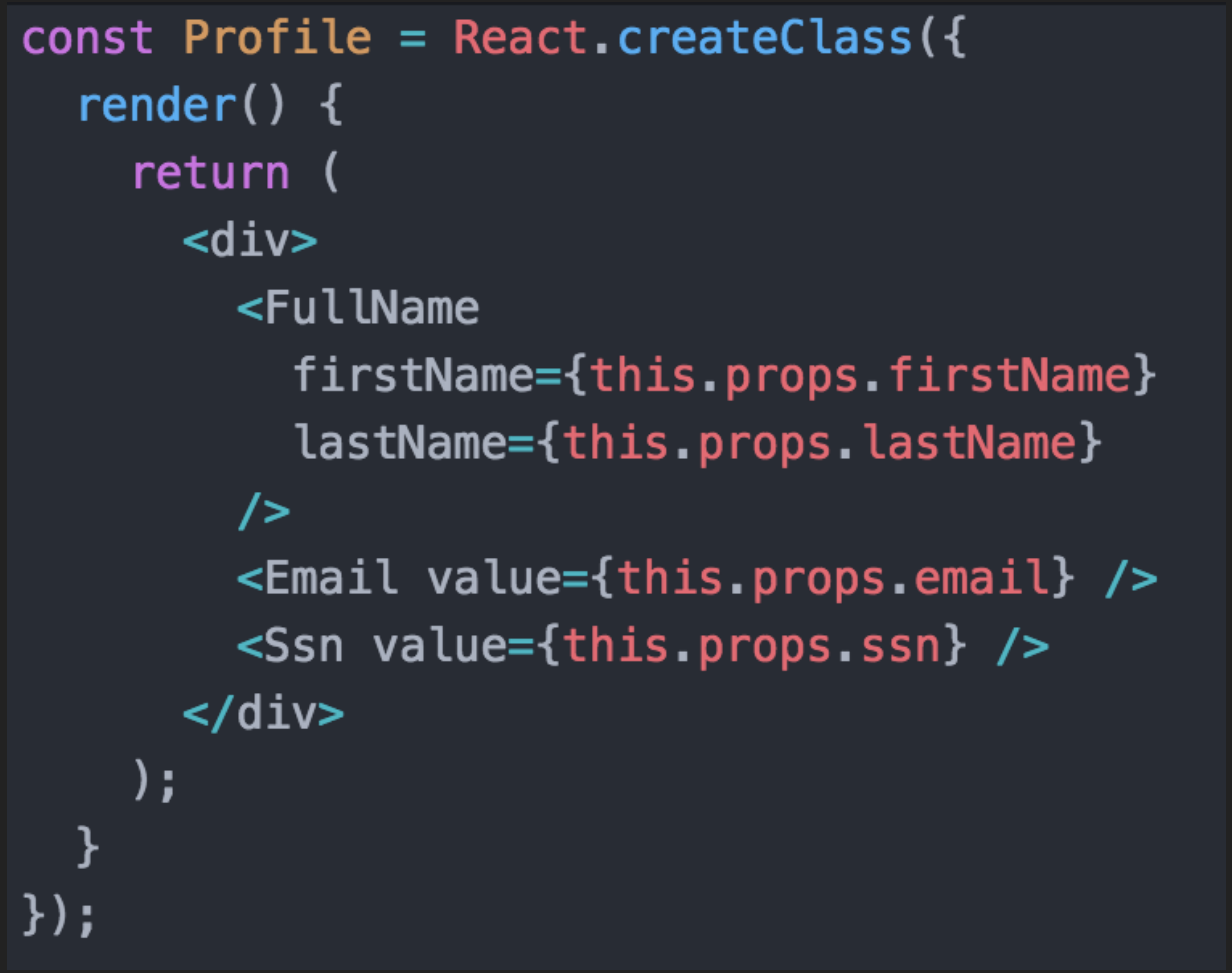
- Incremental migration
- Declarative
- *Encapsulated/Composable*
- Testable
- Interface well with Design



```
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>
    );
  }
});
```

GOALS

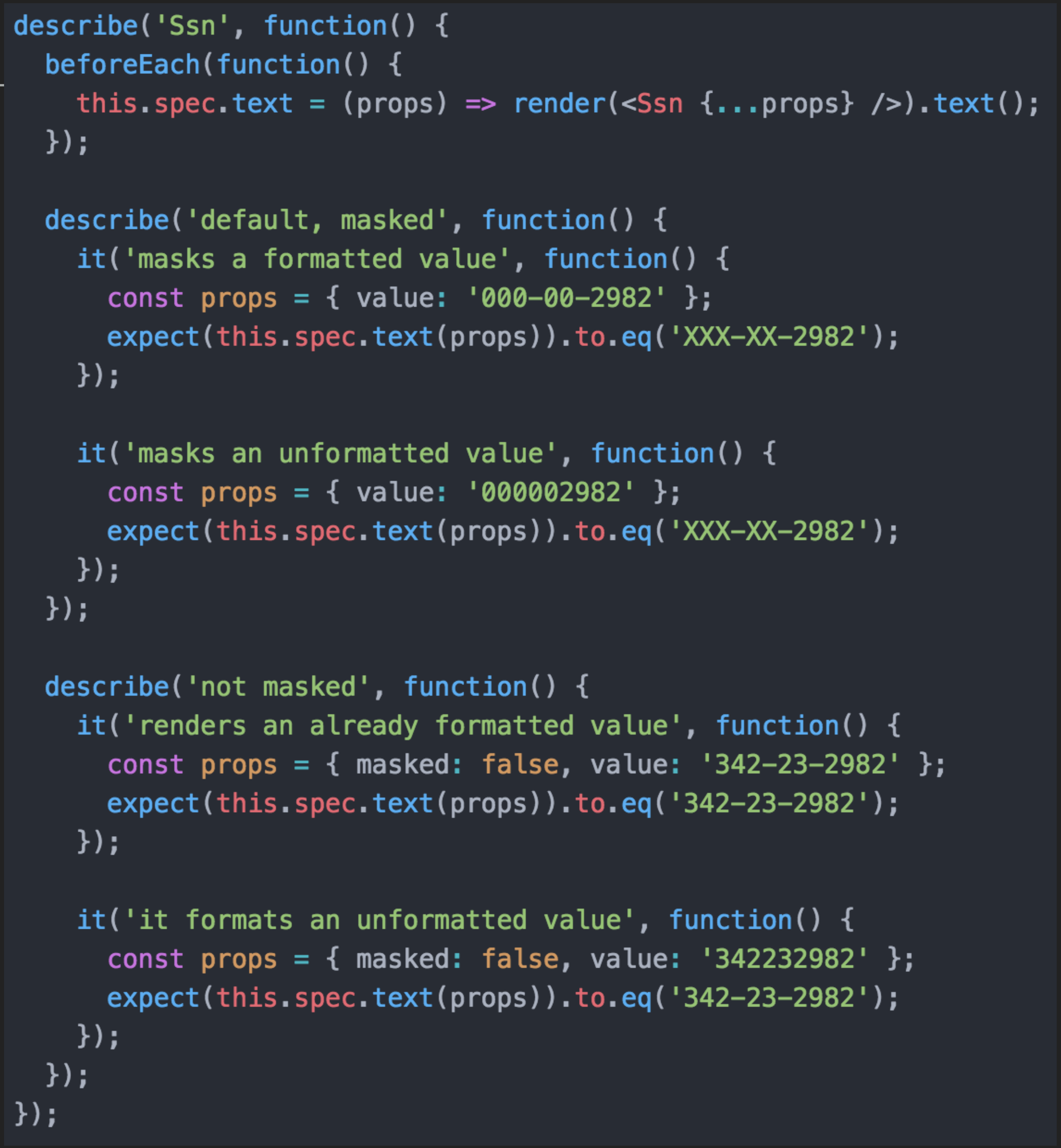
- Incremental migration
- Declarative
- *Encapsulated/Composable*
- Testable
- Interface well with Design



```
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>
    );
  }
});
```

GOALS

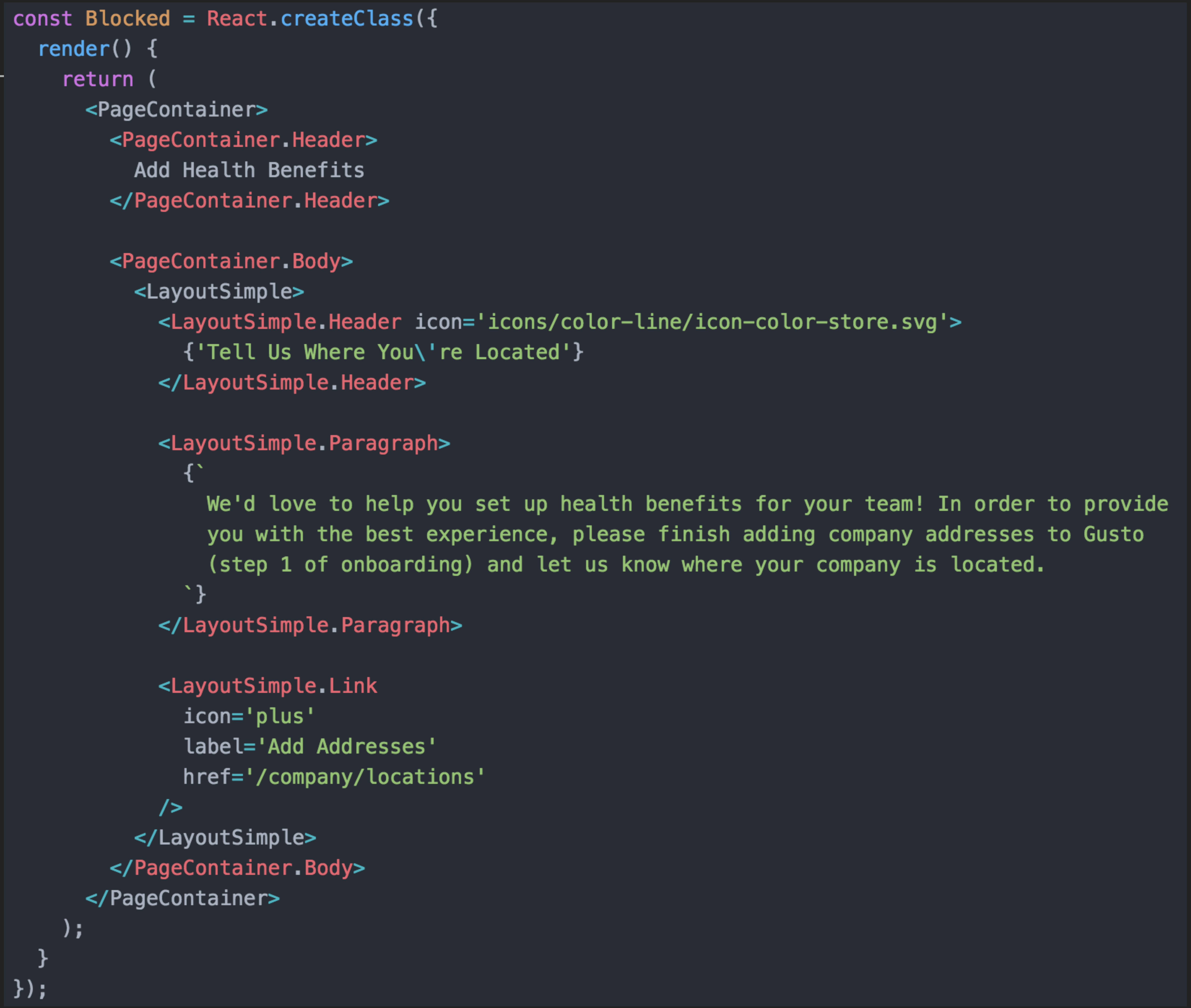
- Incremental migration
- Declarative
- Encapsulated/Composable
- *Testable*
- Interface well with Design



```
describe('Ssn', function() {
  beforeEach(function() {
    this.spec.text = (props) => render(<Ssn {...props} />).text();
  });

  describe('default, masked', function() {
    it('masks a formatted value', function() {
      const props = { value: '000-00-2982' };
      expect(this.spec.text(props)).to.eq('XXX-XX-2982');
    });

    it('masks an unformatted value', function() {
      const props = { value: '000002982' };
      expect(this.spec.text(props)).to.eq('XXX-XX-2982');
    });
  });

  describe('not masked', function() {
    it('renders an already formatted value', function() {
      const props = { masked: false, value: '342-23-2982' };
      expect(this.spec.text(props)).to.eq('342-23-2982');
    });

    it('it formats an unformatted value', function() {
      const props = { masked: false, value: '342232982' };
      expect(this.spec.text(props)).to.eq('342-23-2982');
    });
  });
});
```

GOALS

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



```
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?

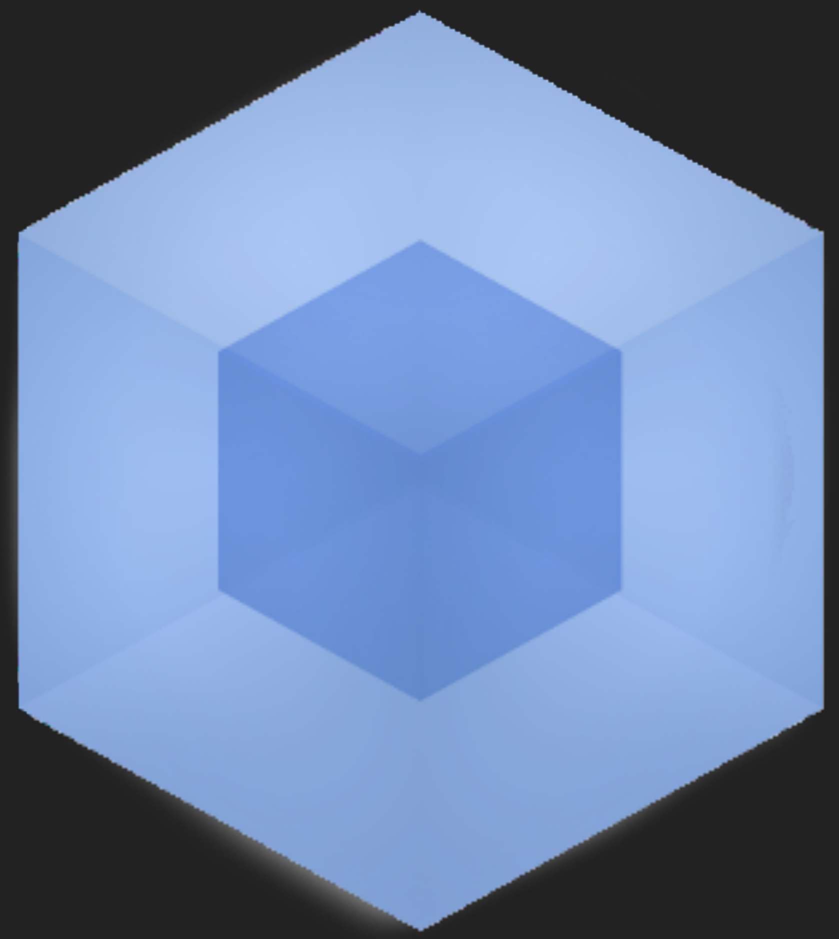
- Dependency management
- Bundle optimization (lazy loading, common package caching, tree shaking)
- Hot loading
- Fast testing

RAILS ASSET PIPELINE CHALLENGES

- 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

GOALS

- Incremental migration
- Speed, speed, speed
- Better development tooling
- Move towards modular JavaScript
- Bundle optimization
- Move all assets off of Rails



Webpack

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