Escape from the server side rendering with ReactJS

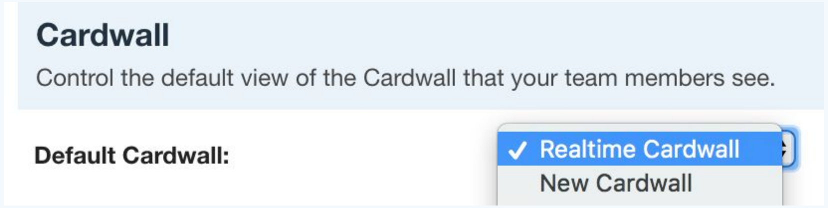
Artur Kremens

Assembla - Who we are?

- Tasks management
- Milestones
- GIT, SVN, P4
- Code reviews

Update existing page elements

- Use feature flags
- Allow team or beta users to see new changes
- Redirect to old views when user does not have enable feature flag
- Allow beta users to return to use old version



Cardwall

Control the default view of the Cardwall that your team members see.

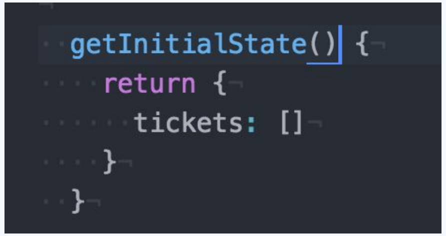
Default Cardwall: ✓ Realtime Cardwall

Use React in Rails application

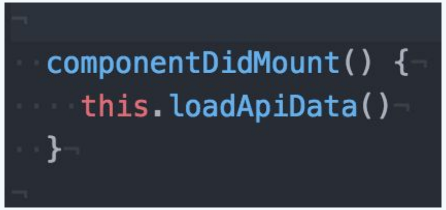
- `gem 'react-rails'`
- Select React.js build `config.react.variant = :development`
- Generate or create component `rails generate react:component ExampleComponent someProps:string`
- Render component in Rails view `react_component('ExampleComponent', someProps: 'TestMe')`
- Profit!

How to work with data

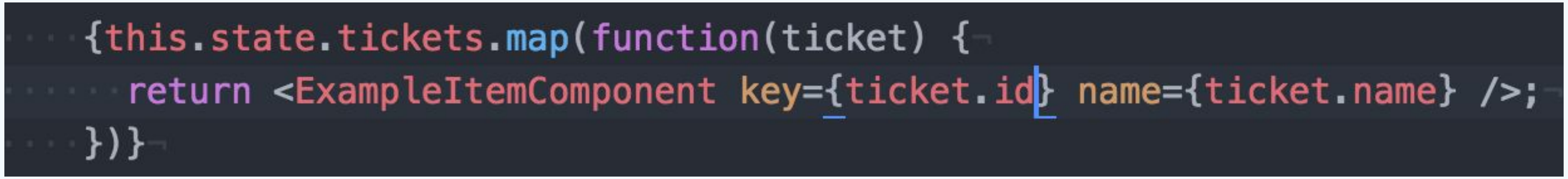
- Use data managers
- Define initial state
- Call Api on `componentDidMount`
- Update collection
- Render collection



```
getInitialState() {
  return {
    tickets: []
  }
}
```



```
componentDidMount() {
  this.loadApiData()
}
```



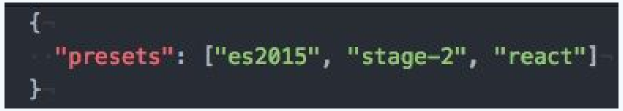
```
{this.state.tickets.map(function(ticket) {
  return <ExampleItemComponent key={ticket.id} name={ticket.name} />;
})}
```

React + Redux + Rails - how to?

- `gem 'browserify-rails'`
- `npm adduser|whoami`
- `npm init --scope=USERNAME`
- `npm install react --save`
- `npm install enzyme --save-dev`
- `npm install`

Base configuration

- Rails.application.config.assets.paths << "node_modules" in assets.rb
- config.browserify_rails.commandline_options = '-t babelify' in application.rb
- .babelrc



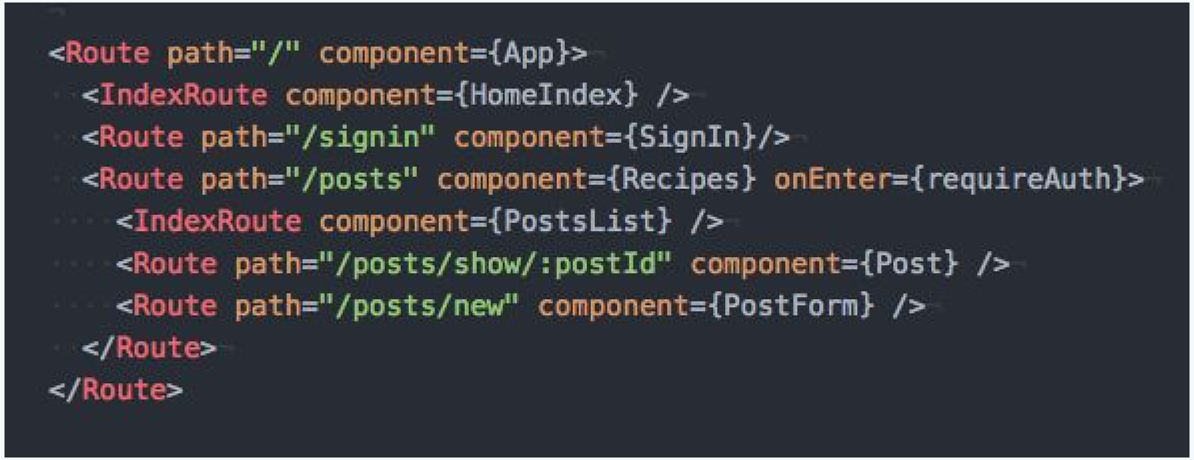
```
{
  "presets": ["es2015", "stage-2", "react"]
}
```

- app/assets/application.js

Assets structure

- routes
- store
- reducers
- pages
- containers
- components
- actions

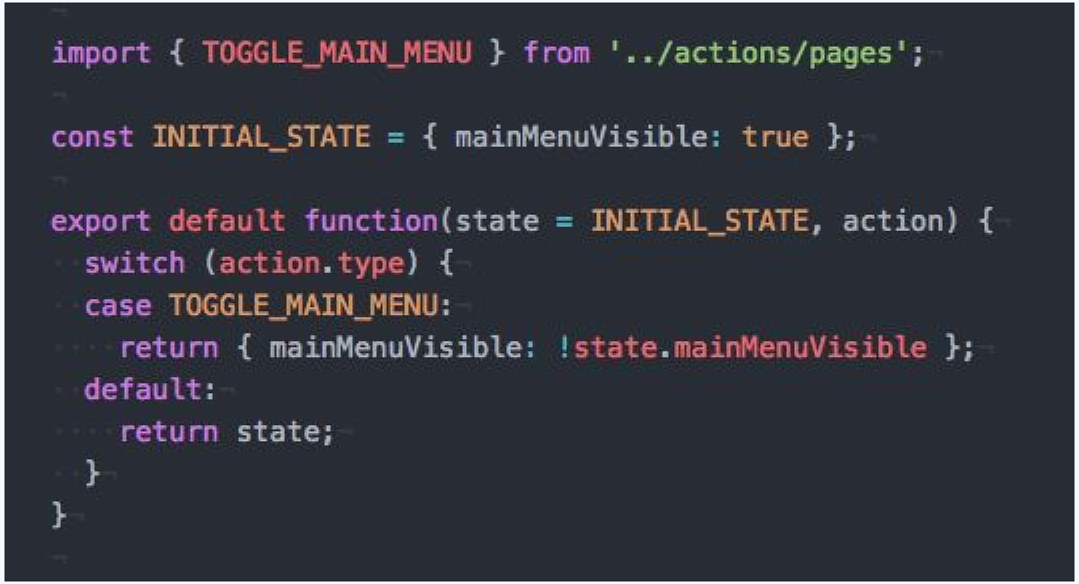
Route



```
<Route path="/" component={App}>
  <IndexRoute component={HomeIndex} />
  <Route path="/signin" component={SignIn}/>
  <Route path="/posts" component={Recipes} onEnter={requireAuth}>
    <IndexRoute component={PostsList} />
    <Route path="/posts/show/:postId" component={Post} />
    <Route path="/posts/new" component={PostForm} />
  </Route>
</Route>
```

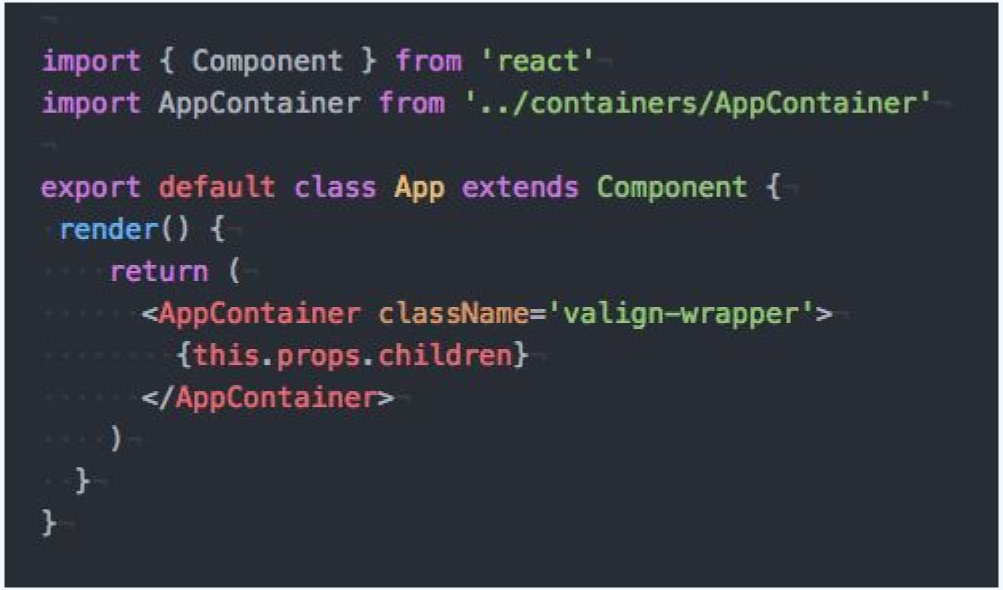
Reacttraining react-router

Reducers



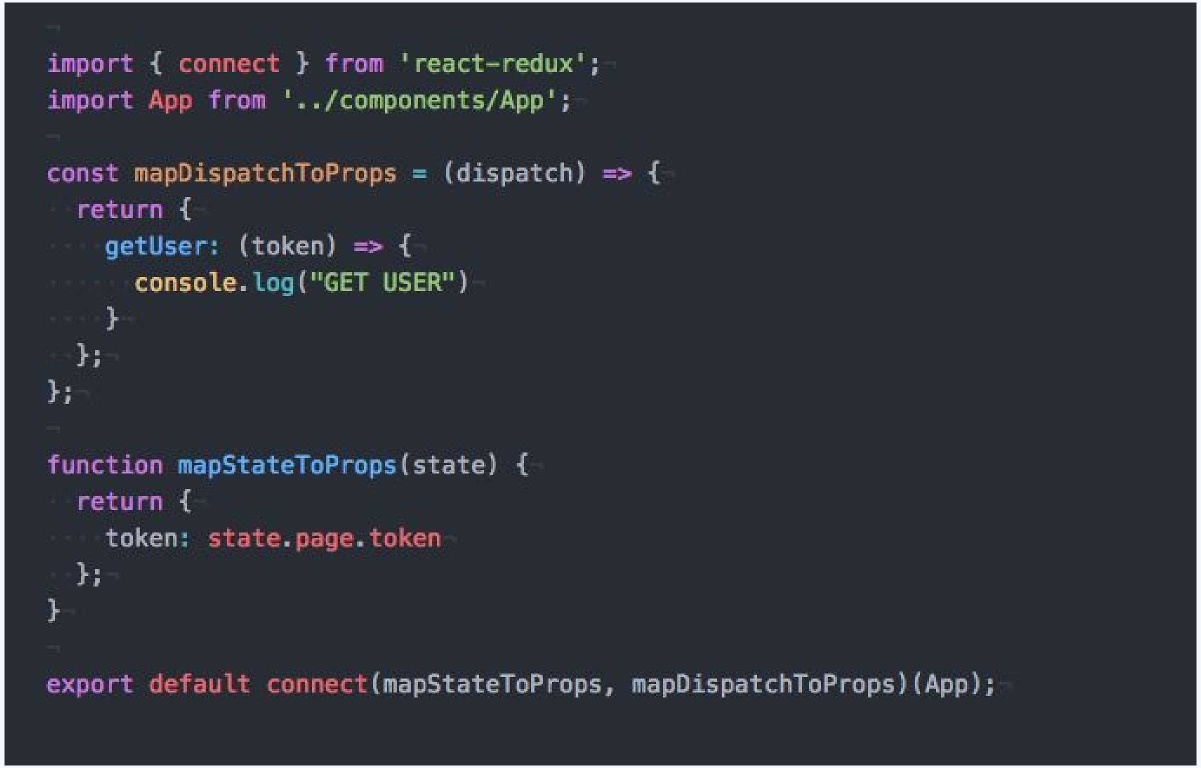
```
import { TOGGLE_MAIN_MENU } from '../actions/pages';

const INITIAL_STATE = { mainMenuVisible: true };

export default function(state = INITIAL_STATE, action) {
  switch (action.type) {
  case TOGGLE_MAIN_MENU:
    return { mainMenuVisible: !state.mainMenuVisible };
  default:
    return state;
  }
}
```

Pages



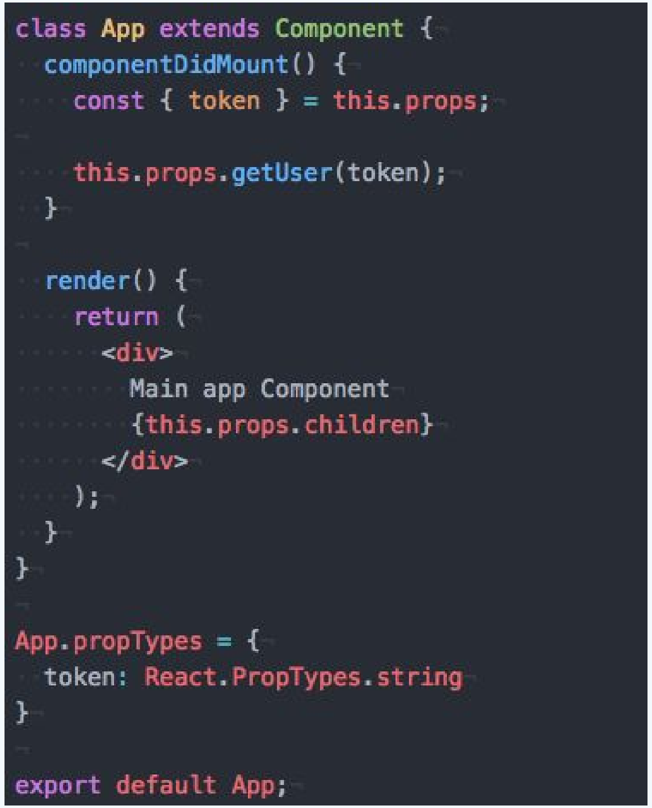
```
import { Component } from 'react'
import AppContainer from '../containers/AppContainer'

export default class App extends Component {
 render() {
    return (
      <AppContainer className='valign-wrapper'>
        {this.props.children}
      </AppContainer>
    )
  }
}
```

Containers



```
import { connect } from 'react-redux';
import App from '../components/App';

const mapDispatchToProps = (dispatch) => {
  return {
    getUser: (token) => {
      console.log("GET USER")
    }
  };
};

function mapStateToProps(state) {
  return {
    token: state.page.token
  };
}

export default connect(mapStateToProps, mapDispatchToProps)(App);
```

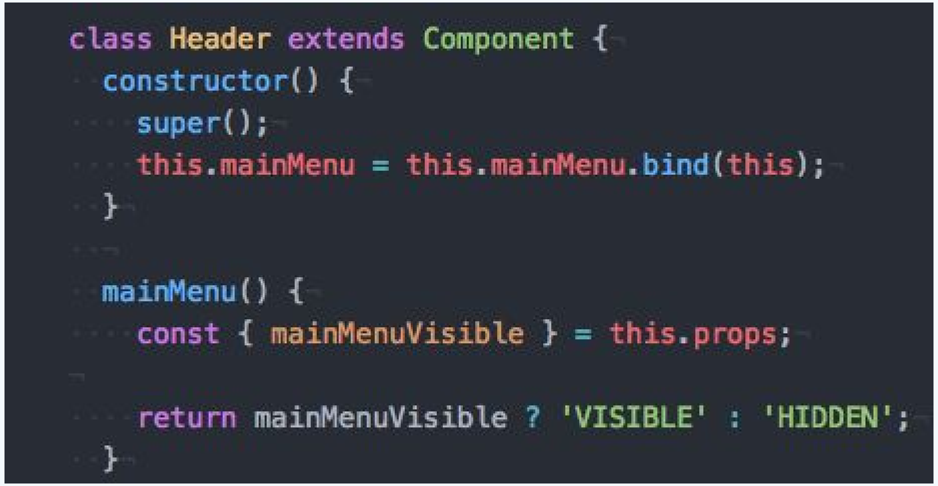
Components



```
class App extends Component {
  componentDidMount() {
    const { token } = this.props;

    this.props.getUser(token);
  }

  render() {
    return (
      <div>
        Main app Component
        {this.props.children}
      </div>
    );
  }
}

App.propTypes = {
  token: React.PropTypes.string
}

export default App;
```

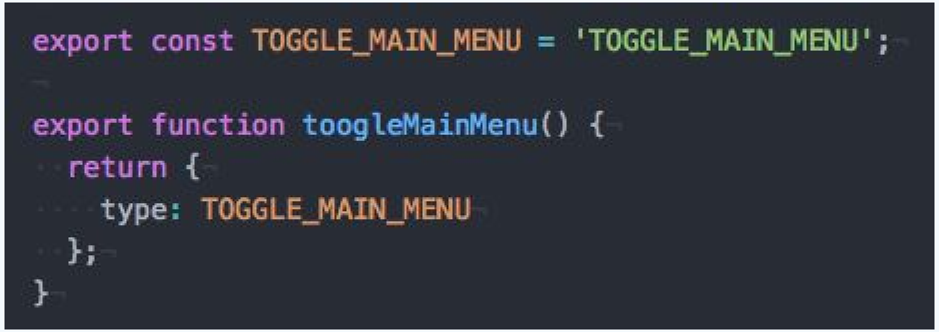
Components

- ES6 classes no longer autobind this to non React methods
- this.myOwnMethod = this.myOwnMethod.bind(this);



```
class Header extends Component {
  constructor() {
    super();
    this.mainMenu = this.mainMenu.bind(this);
  }

  mainMenu() {
    const { mainMenuVisible } = this.props;

    return mainMenuVisible ? 'VISIBLE' : 'HIDDEN';
  }
```

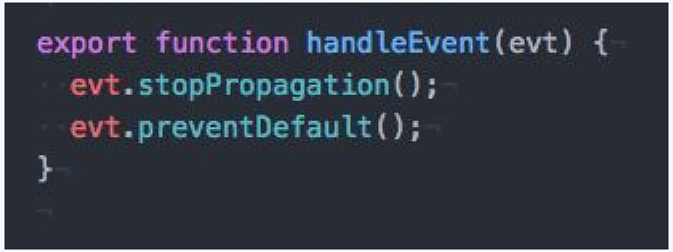
Actions



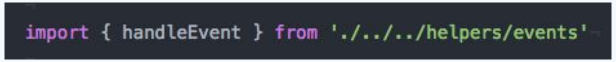
```
export const TOGGLE_MAIN_MENU = 'TOGGLE_MAIN_MENU';

export function toogleMainMenu() {
  return {
    type: TOGGLE_MAIN_MENU
  };
}
```

Don’t repeat yourself

- helpers



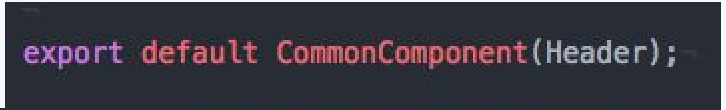
```
export function handleEvent(evt) {
  evt.stopPropagation();
  evt.preventDefault();
}
```



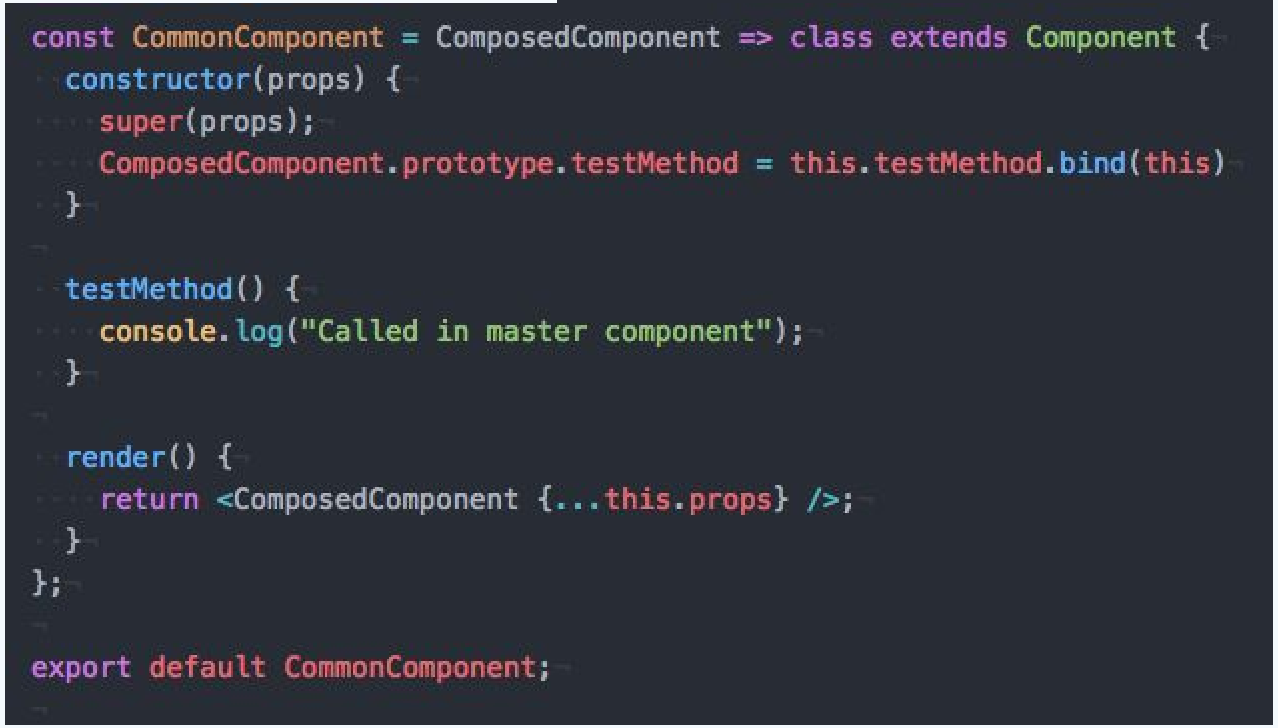
```
import { handleEvent } from './../../helpers/events'
```

Don’t repeat yourself

- Composed components



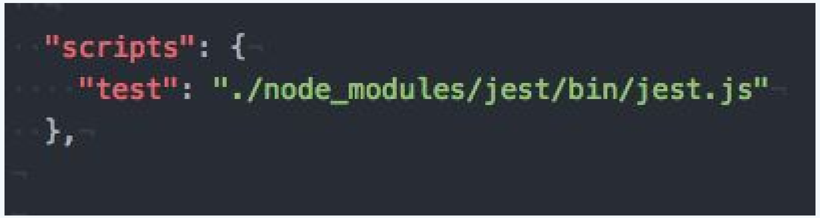
```
export default CommonComponent(Header);
```



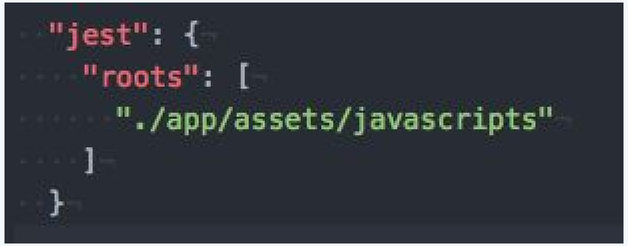
```
const CommonComponent = ComposedComponent => class extends Component {
  constructor(props) {
    super(props);
    ComposedComponent.prototype.testMethod = this.testMethod.bind(this)
  }

  testMethod() {
    console.log("Called in master component");
  }

  render() {
    return <ComposedComponent {...this.props} />;
  }
};

export default CommonComponent;
```

Testing

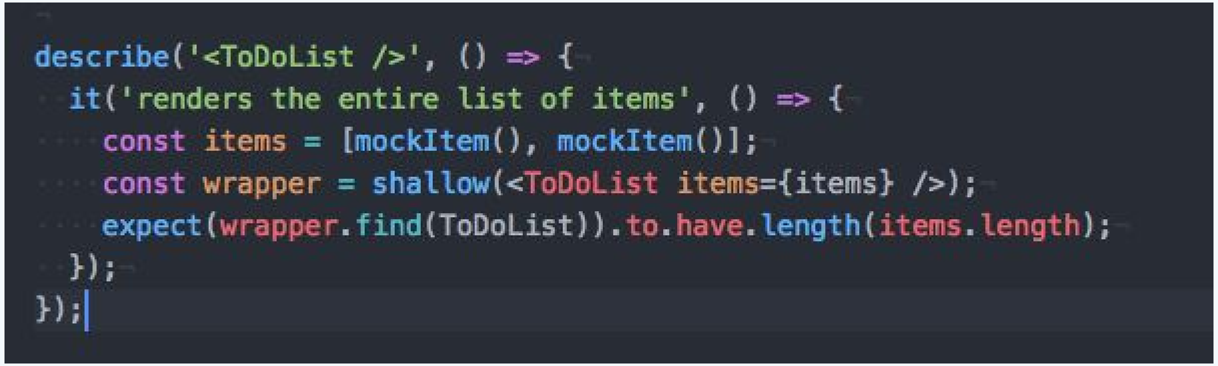
- Jest
- Enzyme
- npm test



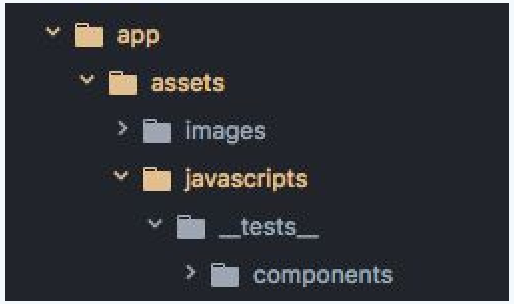
```
"scripts": {
  "test": "./node_modules/jest/bin/jest.js"
},
```



```
"jest": {
  "roots": [
    "./app/assets/javascripts"
  ]
}
```



```
describe('<ToDoList />', () => {
  it('renders the entire list of items', () => {
    const items = [mockItem(), mockItem()];
    const wrapper = shallow(<ToDoList items={items} />);
    expect(wrapper.find(ToDoList)).to.have.length(items.length);
  });
});
```



```
app
  assets
    images
    javascripts
      __tests__
        components
```