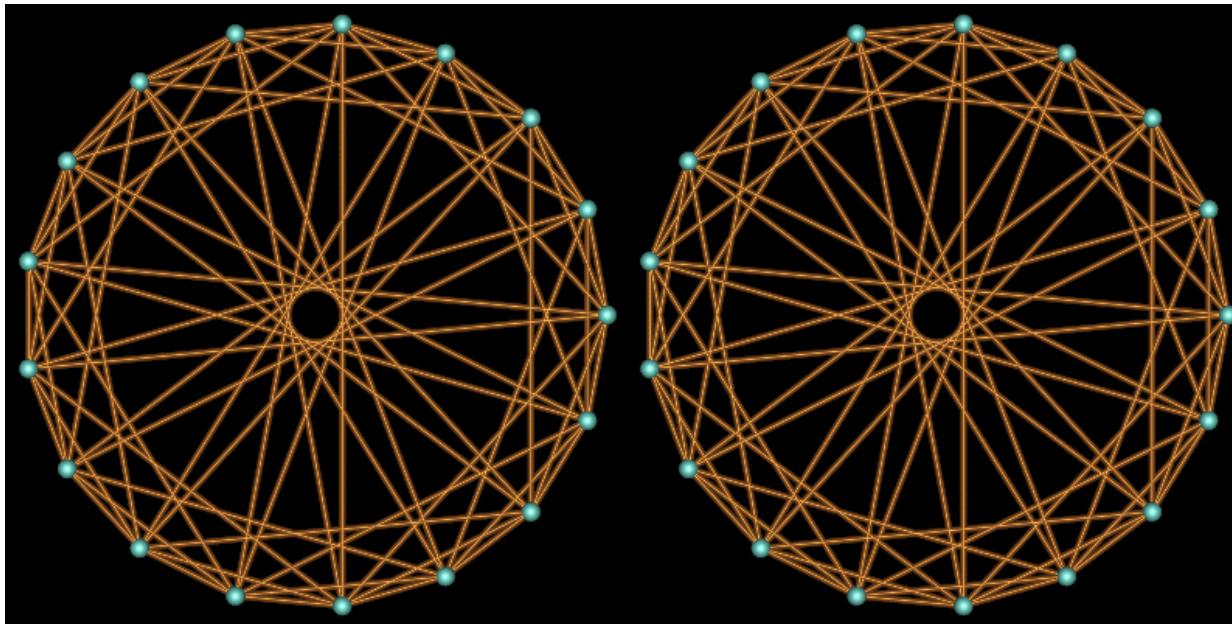


Building Isomorphic Web Apps with React

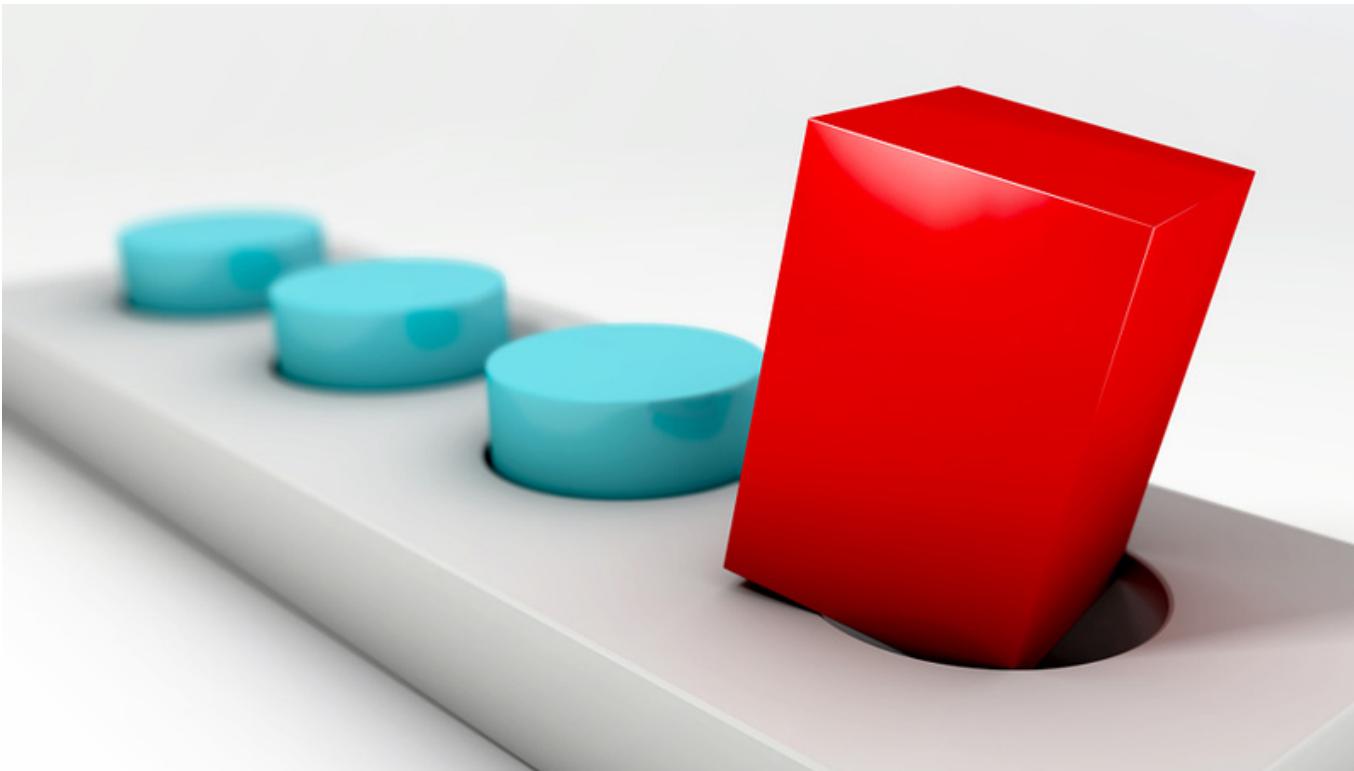
github.com/elyseko/iso-react-demo

@sfdrummerjs
drummerjs.com

Of equal forms



Practical Isomorphic



Why Isomorphic?

- SEO Crawlers
- Performance
- Dev Benefits

Syntax

Overview of Isomorphic Concepts

Code walkthrough

Pros/Cons of Isomorphic Apps

Syntax

```
let foo = "hello";  
const items = [];
```

```
<div>  
  <h1 className="title">{this.props.title}</h1>  
  <Feed></Feed>  
  <Footer></Footer>  
</div>
```

Apps and Libraries

With React



Non React



React Isomorphic frameworks

- Pellet (Veo)
- React Nexus
- Relay (Facebook)

React makes isomorphic possible

- Virtual DOM
- Functional View
- `renderToString`

In the Browser

- Single Page Application (SPA)
- React (vs .14)
- React Router (vs 1.0)
- Browser Sync to serve/update the file

Isomorphic React Example



Mage Knight Board Game
Published in 2011



Android: Netrunner
Published in 2012



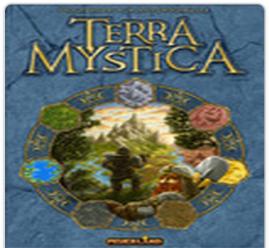
Legendary: Secret Wars -
Volume 1
Published in 2015



La Granja
Published in 2014



Eldritch Horror
Published in 2013



Terra Mystica
Published in 2012



Folder structure

- server
 - app.js (server entry)
 - routes.js
- client
 - main.js (client entry)
 - routes.js
- shared
 - routes.js
 - components
 - app.jsx
 - home.jsx

Build tools



Browsersync

Webpack lets you:

- Build for each environment
- Include environment specific code
- Chunk code into smaller downloadable files

```
entry: {
  app: ['./src/client/main.js']
},
output: {
  path: buildPath,
  filename: 'browser.js'
},
module: {
  loaders: [
    {
      test: /\.jsx?$/,
      exclude: /nodeModulesPath/,
      loader: 'babel-loader'
    }
  ]
}
```

Abstract server/browser differences

```
resolve: {  
  extensions: ['', '.js', '.json', '.jsx'],  
  alias: {  
    '$history': 'history/lib/createBrowserHistory'  
  }  
}
```

Defining environments

```
var definePlugin = new webpack.DefinePlugin({
  __DEV__: JSON.stringify(
    JSON.parse(process.env.BUILD_DEV || 'true')
  ),
  __PROD__: JSON.stringify(
    JSON.parse(process.env.BUILD_PROD || 'false')
  )
});
```

Gulp + Browser Sync

- Replaced webpack dev server
- Gulp makes it easy to run browser sync

```
gulp.task('browser-sync', function () {
  browserSync.init({
    server: path.join(__dirname, "/")
  });
});
```

Shared routes

```
<Route component={ App } path="/">
  <IndexRoute component={ Home } />
  <Route component={ Detail } path="game:id" />
</Route>
```

Isomorphic React Example

Client Router

```
<Router history={createHistory()}  
  routes={sharedRoutes}  
  createElement={createElement}>  
</Router>
```

Server Router

```
match({ routes, location }, (error, redirectLocation, renderProps) => {
  if (redirectLocation)
  else if (error)
  else if (renderProps == null)
  else // handle pre fetching data
})
```

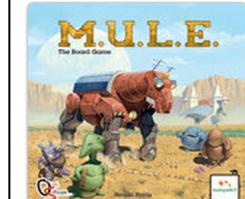
Render the view

```
ReactDOM.render(routes,  
  document.getElementById('react')  
)
```

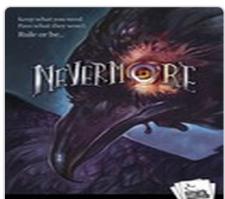
Building the view

```
render() {  
  return (  
    <div className="app">  
      <h3>Isomorphic React Example</h3>  
      <div className="main-content">  
        {this.props.children}  
      </div>  
    </div>  
  );  
}
```

Isomorphic React Example



M.U.L.E. The Board Game
Published in 2015



Nevermore
Published in 2015



Star Wars: Imperial
Assault
Published in 2014



Treasure Hunter
Published in 2015



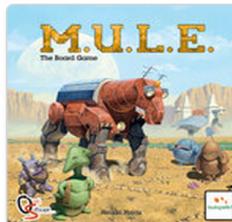
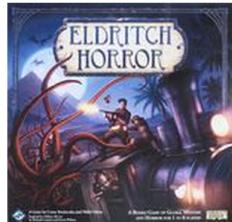
Child Views

```
<div className="ui six doubling cards">  
  {this.renderCards()}  
</div>
```

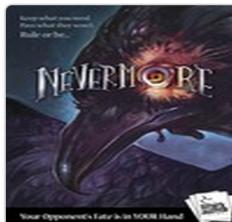
Render the cards

```
cards.push(  
  <Card key={"card-item" + index} {...currentCard}>  
  </Card>  
)
```

Eldritch Horror



M.U.L.E. The Board Game
Published in 2015



Nevermore
Published in 2015



Star Wars: Imperial
Assault



Treasure Hunter
Published in 2015

Detail page code

```
<h1>{item.name}</h1>
<img className="ui image" src={item.thumbnail} />
<div className="ui hidden divider"></div>

<div className="ui six doubling cards">
  {viewUtils.renderCards(related)}
</div>
```

Declaring our data needs

```
static requestData() {  
    return [{request: "getCards"}];  
}
```

Getting the data

```
// runs once for each component that the router  
// knows about  
function createElement(Component, props) {  
  return <DataWrapper component={Component} {...props}>  
}
```

Higher Order Component

```
render() {  
  return (  
    <this.props.component {...this.props}  
      data={this.state.data} />  
  );  
}
```

On the Server

Server render is for bots and crawlers



Serving the page

```
<head>
  <title><%= title%></title>
</head>
<body>
  <div id="react"><%- html%></div>
  <script>$data = function(){return<%- data%>};</script>
  <script type="text/javascript" src="/build/browser.js"></script>
</body>
```

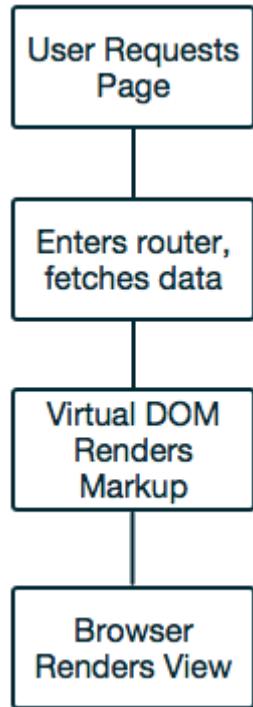
Render on the server

```
let html = ReactDOM.renderToString(<RoutingContext {...renderProps}>)
res.render('pages/index', {
    "title": "Test",
    "html": html,
    data: JSON.stringify(viewData)});
```

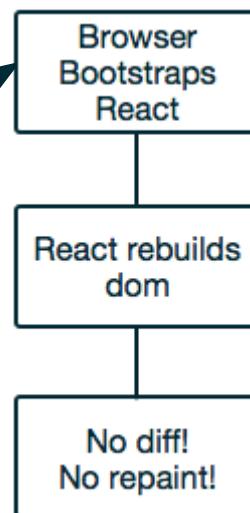
Getting the data

```
let callback = (err, data)=> {
  viewData[data.id] = data.result;
}
routeUtils.batchRequests(requests, callback, renderProps.params);
```

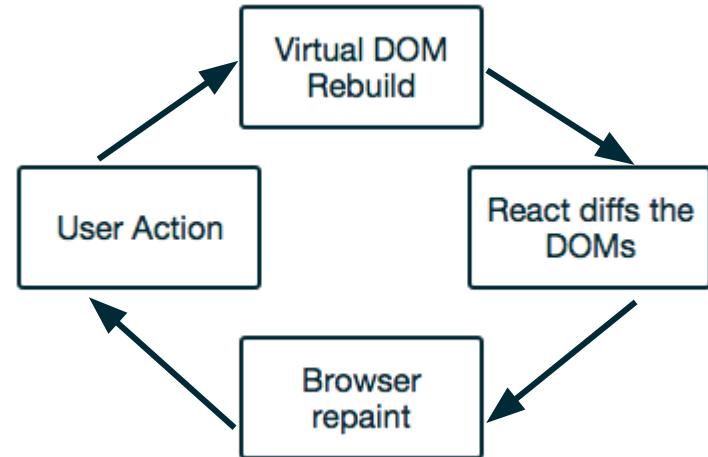
Server



Initial Browser Load



SPA



Getting the data to the client

```
res.render('pages/index', {  
    "title": "Home",  
    "html": html,  
    data: JSON.stringify({viewData})  
});
```

Reading the data on the client

```
let cache = new Cache(window.$data());
```

Handling node vs browser



Pros of building an isomorphic App

- Serve crawlers and bots
- Initial load times are very fast
- Code reuse between server and client

Cons of building an isomorphic app

- Server render performance
- Multiple environments = complications

The future of isomorphic

- Relay
- Virtual dom starting to appear more places - Ember
- Virtual DOM optimizations
- Meteor works with react

Recommendations

- React
- React Router
- Webpack
- Gulp & Browser Sync
- Entry points not abstraction
- Relay

Questions?

github.com/elyseko/iso-react-demo

elyse1213@gmail.com

@sfdrummerjs

drummerjs.com

References

- [React](<https://facebook.github.io/react/>)
- [React Router](<https://github.com/rackt/react-router>)
- [JSX](<https://facebook.github.io/react/docs/jsx-in-depth.html>)
- [Webpack](<https://webpack.github.io/>)
- [Browser Sync](<http://www.browsersync.io/>)
- [List of Isomorphic Apps](<http://isomorphic.net/>)