

React

Relay

GraphQL

on Rails

Jason Waldrip

CTO @  Brandfolder

github.com/jwaldrip

Liz Henderson

Developer @  Brandfolder

github.com/liz303

Why React?

Surprisingly Simple

Simply express how your app should look at any given point in time, and React will automatically manage all UI updates when your underlying data changes.

Declarative

When the data changes, React conceptually hits the "refresh" button, and knows to only update the changed parts. It also does this in a way that is efficient by using a "Virtual DOM"

DRY Components

React is all about building reusable components. In fact, with React the only thing you do is build components. Since they're so encapsulated, components make code reuse, testing, and separation of concerns easy.



WOODCHUCK
HARD CIDER
REAL CIDER
FROM A REAL CIDERY.™

01 - Woodchuck

STEP RIGHT UP!



Logos

Product Shots

Sell Sheets

Media

People

New Products

POS

FAQ/UPC Sheets

Videos

Q section: "Logos"

Search results for *section: "Logos"*

Share Search



Logos 19 Assets



WC Logo Color 2014



WC Logo black 2014



WC Logo Black 2014 weathered



WC Logo Winter 2014



WC Logo Summer 2014



JSX

It's Javascript... except it's also HTML.

Why Relay?

Declarative

Never again communicate with your data store using an imperative API. Simply declare your data requirements using GraphQL and let Relay figure out how and when to fetch your data.

Co-location

Data queries live next to the views that rely on them, so you can easily reason about your app.

Smart

Relay aggregates queries into efficient network requests to fetch only what you need.

Mutations

Relay lets you mutate data on the client and server using GraphQL mutations, and offers automatic data consistency, optimistic updates, and error handling.

But "Relay"...

While React allows you do DRY things up, having to implement flux everywhere can be expensive in http calls and makes React **not** so easy.

Why GraphQL?

WYNIWYG

What you need is what you get. Don't get the fields or resources you don't need. Keep the request and response as light as it possibly can be.

Easy To Implement

GraphQL is pleasantly easy to implement. You don't have to worry about routing unlike traditional api's as every request hits a simple endpoint and the query language does the ***REST***.

Why Rails?

Because DHH Loves "Integrated Applications"

Rails is great for building API's but not so great for front ends. So why not have your Rails app just serve your JS based React app?

Resources

- <https://github.com/rmosolgo/graphql-ruby>
- <https://facebook.github.io/{graphql,relay,react}>
- GraphQL/React/Relay/Redux on Discourse
- GraphQL on Slack