# Declarative data fetching in React

Handle your requests like it ain't nothin'

# Hi, I'm Charles! 👋

Ping me if you're about that React Native life too!
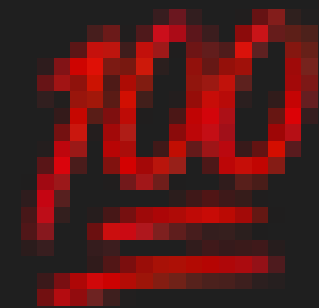
# 2 areas to improve

⛵

🎣

Navigation

Data fetching

# Navigation 💯

`yarn add react-router-navigation`

ImperativeDeclarative

# Axios

```
async function getUser(userID) {
  try {
    const response = await axios.get(`/api/users/${userID}`)
    console.log("👌", response)
  } catch (error) {
    console.log("😭", error)
  }
}
```

# react-refetch

```
export default connect((props) => ({
  userFetch: `/api/users/${props.params.userId}`
}))(Profile)
```

OK, this hoc looks great but what if I'm more of a render props type of person, hun 😤?!

# React Apollo v2.1

```js
// queries.js
const GET_USER = gql`
{
  user {
    firstName
  }
}
`;
```

```js
// app.js
<Query query={GET_USER}>
  {({ data, error, loading }) => {
    if (loading) return <p>Loading...</p>
    if (error) return <p>Error :(</p>

    return (
      <p>Welcome back {data.user.firstName}!</p>
    )
  }}
</Query>
```

# Data fetching through a component?

## React Apollo  🤔 ❓

### GraphQL          REST

🎊 🎊 🎊

# REACT DATA FETCHING

*yet another* library for declarative data fetching in React.

```
import { Fetch, ConnnectedFetch, requestToApi }
             from "react-data-fetching"
```

# Quick comparison

```jsx
// react-apollo
<Query query={GET_USER}>
  {({ data, error, loading }) => {
    if (loading) return <p>Loading...</p>
    if (error) return <p>Error :O</p>

    return (
      <p>
        Welcome back {data.user.firstName}!
      </p>
    )
}}
</Query>
```

```jsx
//react-data-fetching
<Fetch
  path="/users/octocat"
  loader={<p>Loading...</p>}
  timeout={5000}
>
  {({ data, error }) => {
    if (error) return <p>Error :O</p>

    return (
      <p>
        Welcome back {data.firstName}!
      </p>
    )
}}
</Fetch>
```

# requestToApi

```
_onSignUp = async () => {
  const apiResponse = await requestToApi({
    url: 'https://my-app.com/api/v1/users',
    body: {{ email: 'midoriya@shonen.com', password: XXX }},
    headers: {{ Cache-Control: 'no-cache' }},
    method: 'POST',
    onProgress:
      progression => ('♻️ Progressing...', progression),
    timeout: 2500,
    onTimeout: () => console.log('⏱️ Timeout!'),
  })

  this.setState(() => ({ accountCreated: apiResponse.isOK }))
}
```

# &lt;ConnectedFetch /&gt;

```
<ConnectedFetch
  api="https://api.github.com"
  headers={{ Cache-Control: 'public' }}
  loader={<p>Loading…</p>}
  timeout={5000}
  onIntercept={this._onErrorIntercepted} // 😉
  //...
>

  <AppRoot />
</ConnectedFetch>
```

# React Data Fetching

GitHub:
https://git.io/rdf

In-depth article:
https://medium.com/p/2140a1d36cc8

💖 Awesome contributors 💖

Aakansha Doshi
@ad1992

Mathieu Dumont
@Madumo

Luong Dang Hai
@jarvisluong

# Thank you! ✨

@Charles_Mangwa on Twitter, come say hi!...
I'll be waiting for y'all!!!