

GraphQL State Management

**Ankita Kulkarni
Educator**

@kulkarniankita9

<https://kulkarniankita.com/>

Hi, I'm Ankita! 🙌



@kulkarniankita9

Quick Note

- All snippets and my slides are available at:
bit.ly/graphql-state-management-slides



@kulkarniankita9

What is a State Manager?



@kulkarniankit9

**What is a good
State Manager? 🤔**

@kulkarniankita9

Data is Normalized



@kulkarniankita9

**Data transforms
consistently**

@kulkarniankita9

**You can Visualize Global
State**

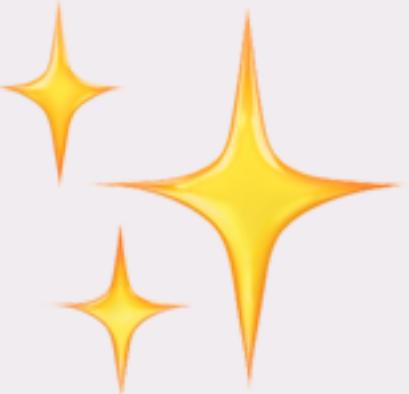


@kulkarniankita9

**Debugging is
easier**

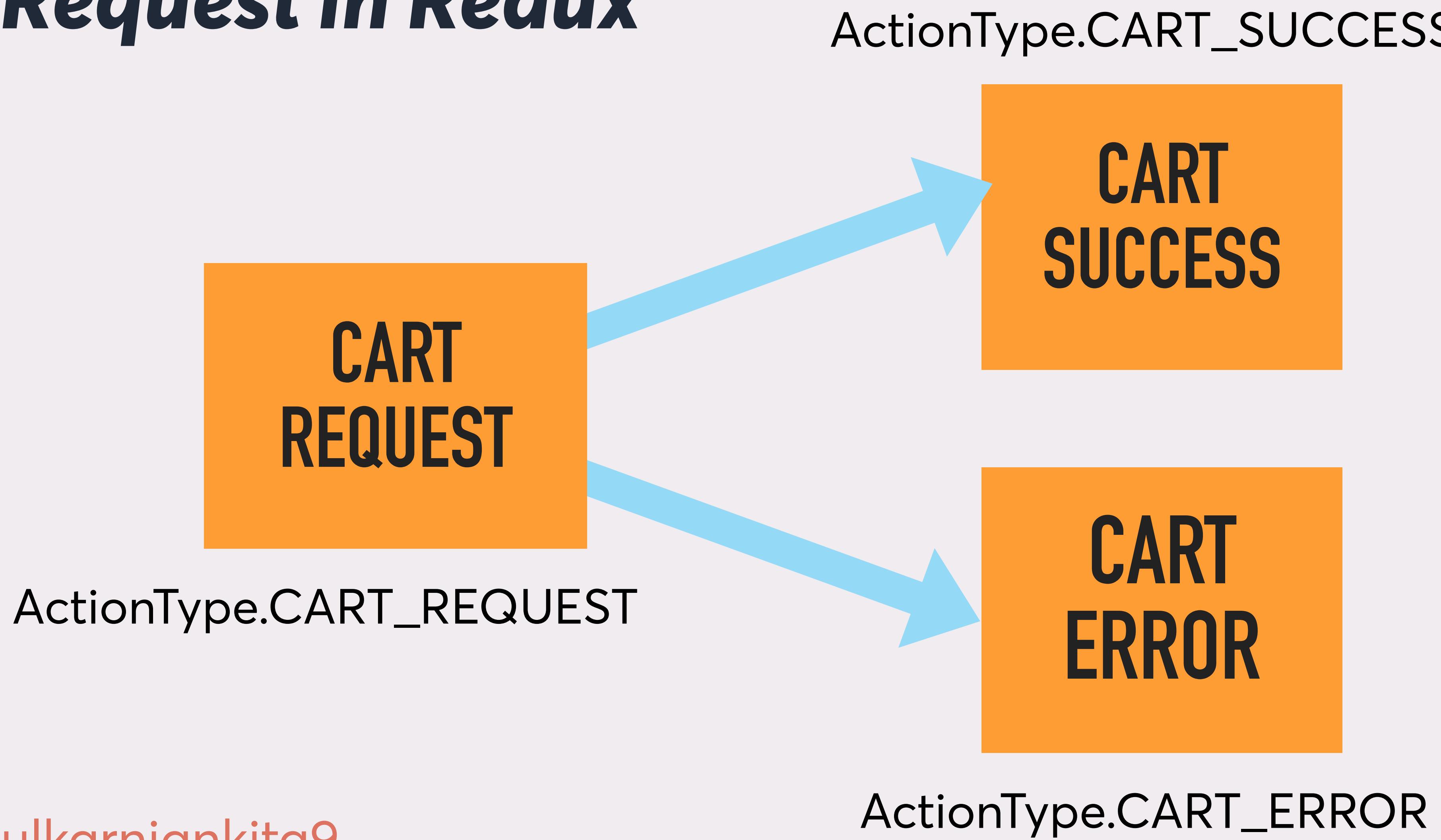


@kulkarniankita9

***Redux fixed the Majority
of our problems*** 

@kulkarniankit9

A Request in Redux



Async request in redux

(1) Listens for
CART_REQUEST action

(2) Fires
fetchCartData
function

```
// Redux Saga
import { call, put } from 'redux-saga';

export function* fetchCartData(action) {
  try {
    const data = yield call(GET_CART_QUERY);
    yield put({ type: "CART_SUCCESS", { data, isLoading: false }});
  } catch (error) {
    yield put({ type: 'CART_ERROR', { error, isLoading: false }});
  }
}

function* watchFetchCartDataQuery() {
  yield takeEvery('CART_REQUEST', fetchCartData);
}
```

Then why switch....?

@kulkarniankita9

Apollo GQL happened



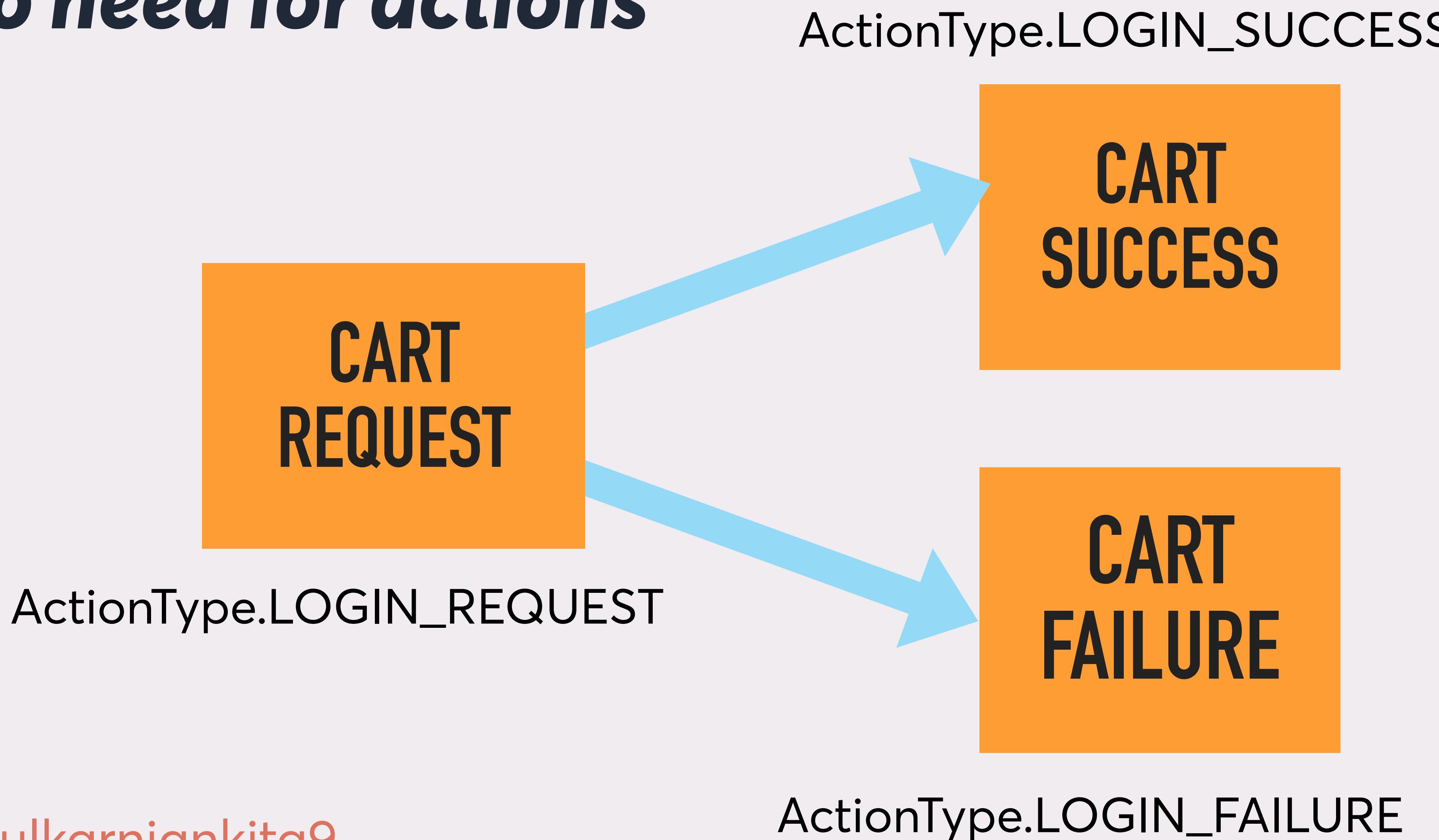
@kulkarniankita9

After Apollo GQL hooks,

```
const { useQuery } from '@apollo/client';

const { loading, error, data, refetch: refetchCart } = useQuery(GET_CART,
{ variables: { isGuest } });
```

No need for actions



WAIT, there are still some challenges... 😐

@kulkarniankita9

How do we?

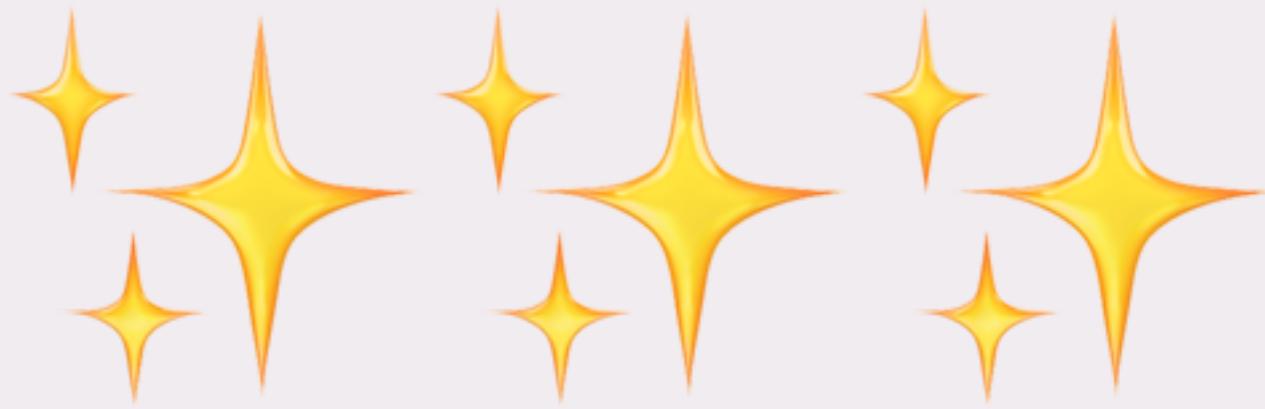
- Transform data?
- Normalize data?
- Create a million actions and reducers 😅

@kulkarniankit9



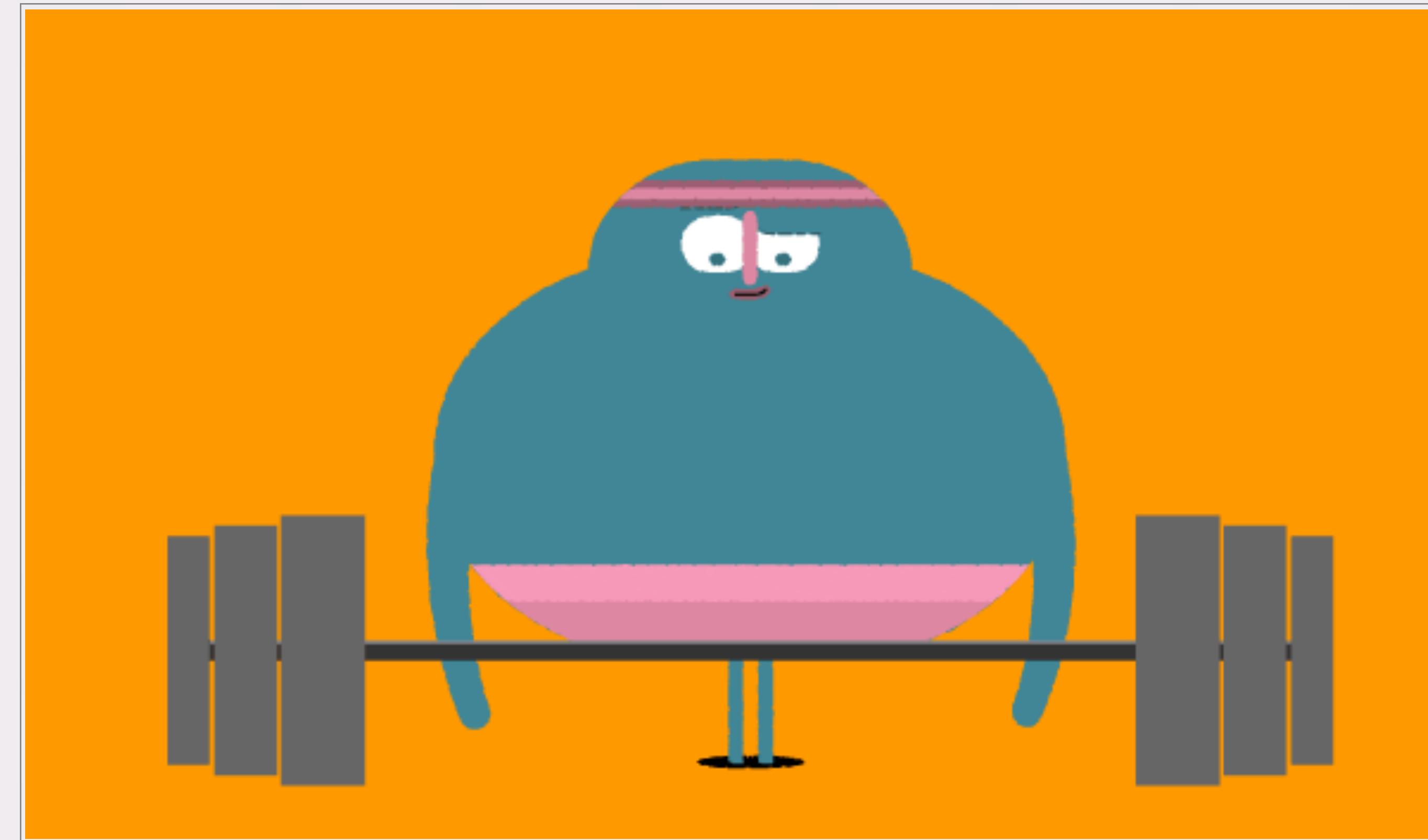
@kulkarniankita9

Apollo Client 3 is here...



@kulkarniankita9

Apollo does the heavy lifting for you 💪



@kulkarniankita9

Difference between Redux and Apollo client



@kulkarniankit9

Let's revisit state management principles

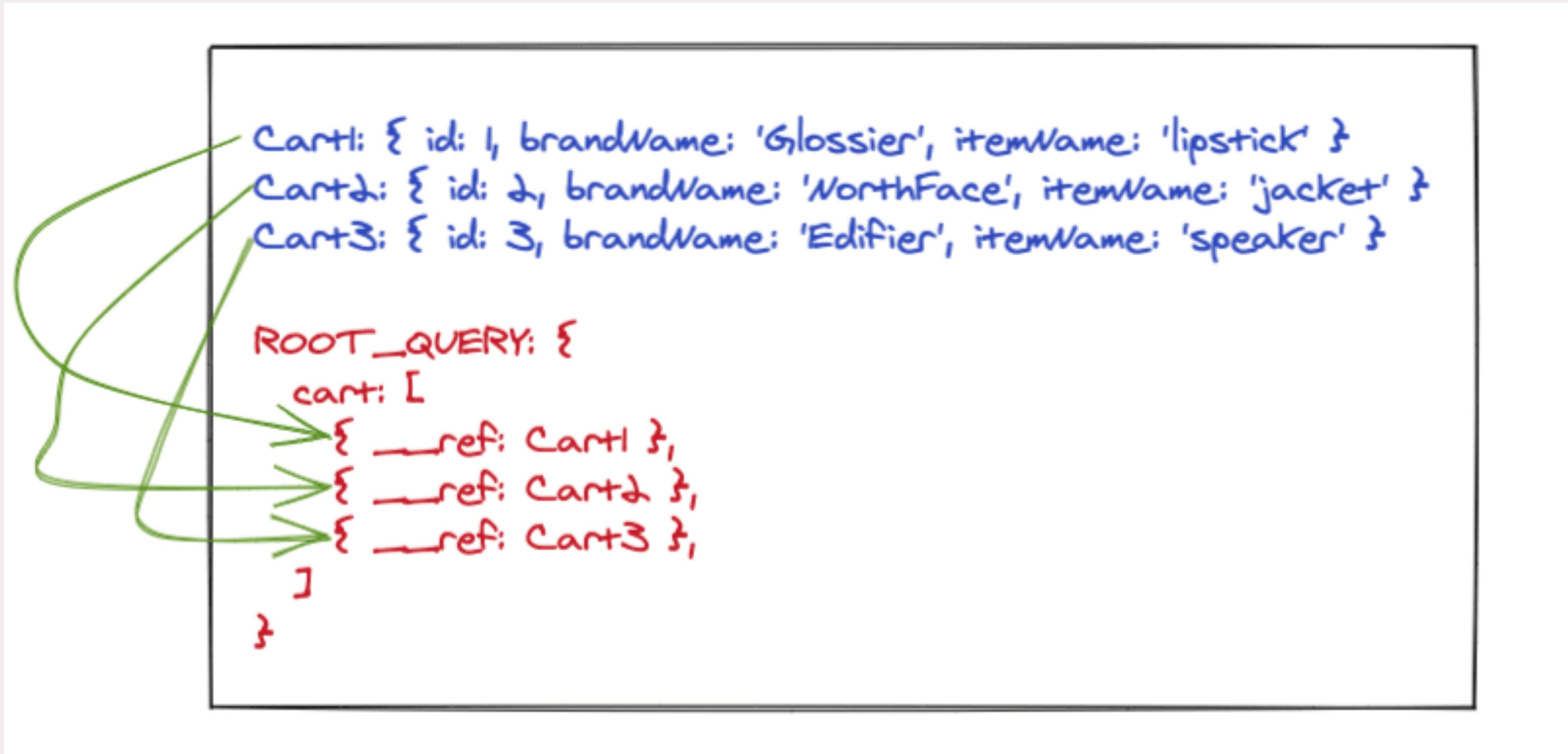
@kulkarniankita9

My data is normalized (no dupes please 🙏)



@kulkarniankit9

Root Query



@kulkarniankit9

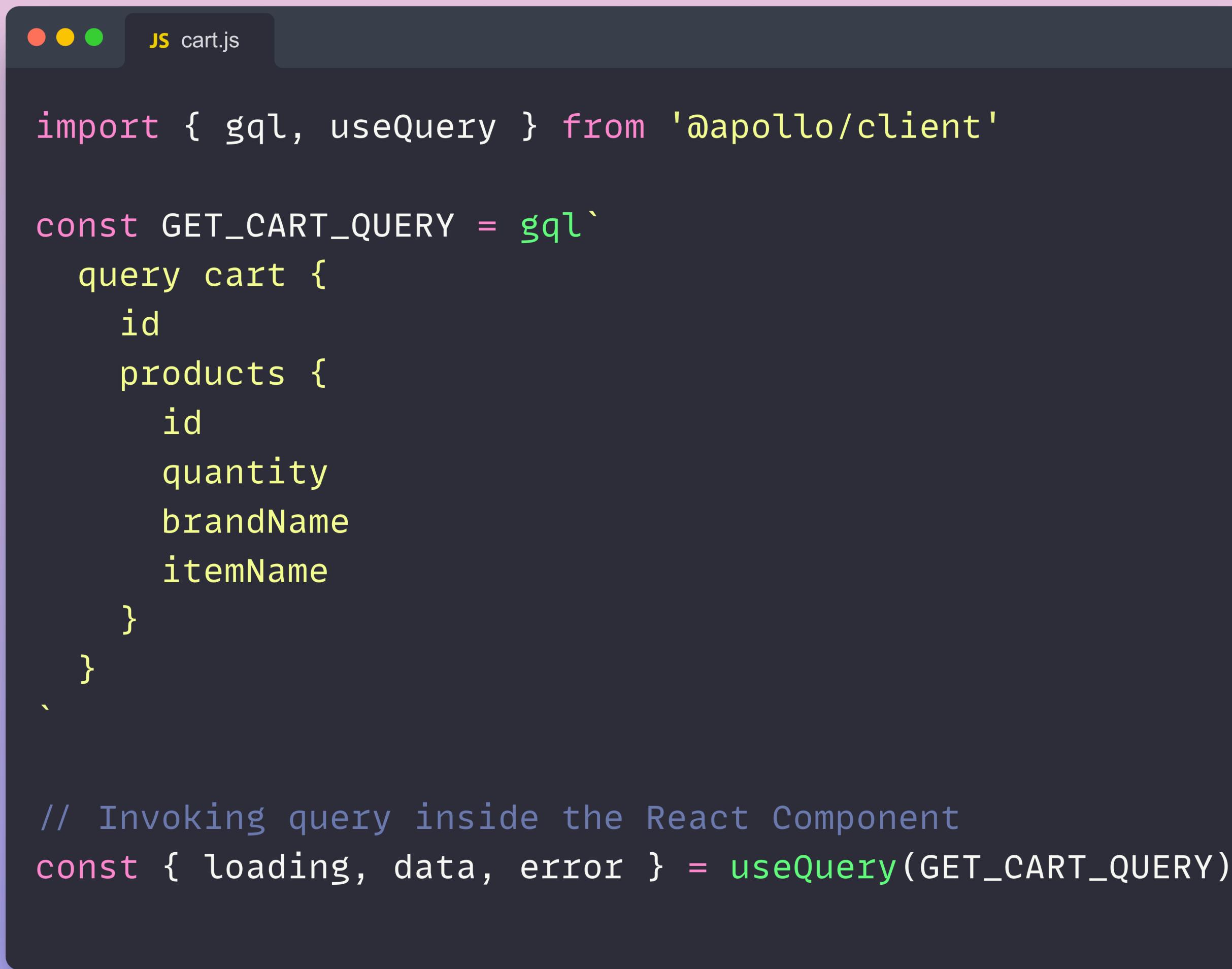
***You need to understand the internal workings of
the Apollo client***

@kulkarniankita9

Data transformations

@kulkarniankita9

Cart Query



```
import { gql, useQuery } from '@apollo/client'

const GET_CART_QUERY = gql`query cart {  id  products {    id    quantity    brandName    itemName  }}`// Invoking query inside the React Componentconst { loading, data, error } = useQuery(GET_CART_QUERY)
```



```
[  {    id: 1,    quantity: 10,    brandName: 'Glossier',    itemName: 'Lipstick',    price: {      total: 19.99    },    outOfStock: false,    availableStock: 3010  },  {    ...  }]
```

Out of stock / Low stock msg

```
const ViewStockText = () => {
  const { loading, data, error } = useQuery(GET_CART_QUERY);
  const [stockText, setStockText] = useState('');

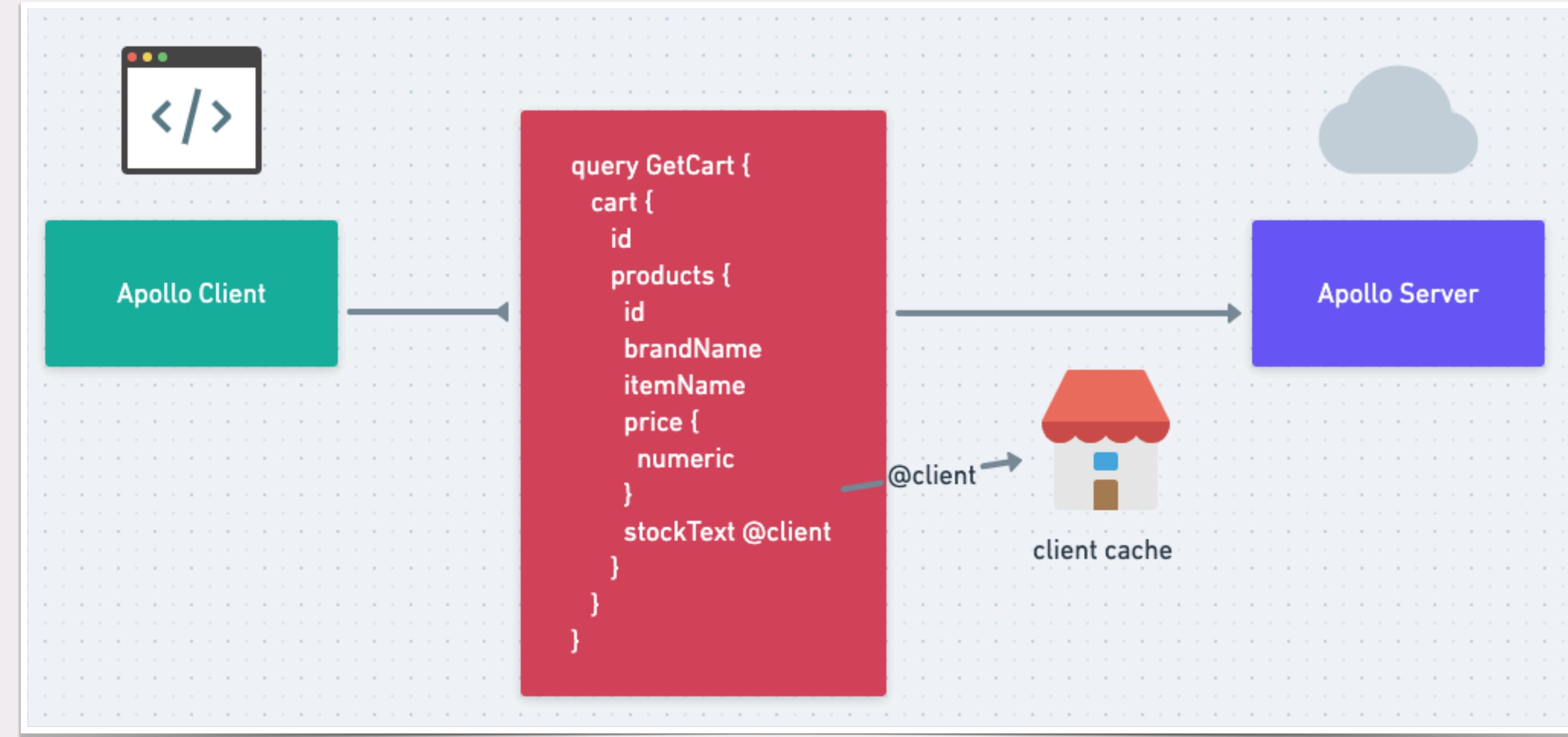
  const getOutOfStockOrLowStockText = useCallback((product) => {
    const { availableStock, quantity, outOfStock } = product;
    const lowStock = availableStock - quantity;

    if (outOfStock) {
      return 'Out of stock';
    } else if (lowStock <= 2) {
      return 'Low stock';
    }
  }, []);

  return (<div>{data?.products.map(product =>
    <p>{getOutOfStockOrLowStockText(product)}</p>
  )}</div>);
}
```

```
[{
  id: 1,
  quantity: 10,
  brandName: 'Glossier',
  itemName: 'Lipstick',
  price: {
    total: 19.99
  },
  outOfStock: true,
  availableStock: 3010
},
{
  ...
} ...more products]
```

What are Local only fields



Local only fields



```
JS cart.js
```

```
import { gql, useQuery } from '@apollo/client'

const GET_CART_QUERY = gql`  
query cart {  
  id  
  products {  
    id  
    quantity  
    brandName  
    itemName  
    → stockText @client  
  }  
}
```

Apollo Client will resolve **stockText** on the client-side

```
const { loading, data, error } = useQuery(GET_CART_QUERY)
```

Invoking query inside a React component

Local only fields



```
import { InMemoryCache, makeVar } from '@apollo/client';

const cache = new InMemoryCache({
  typePolicies: {
    Product: {
      fields: {
        stockText: {
          read(product, { readField }) {
            const { availableStock, quantity, outOfStock } = product;
            const lowStock = availableStock - quantity;

            if (outOfStock) {
              return 'Out of stock';
            } else if (lowStock ≤ 2) {
              return 'Low stock';
            }
          }
        }
      }
    }
  }
}

const client = new ApolloClient({ cache });
```

Reactive variables



Gets the
'token'
cookie

```
import { InMemoryCache, makeVar } from '@apollo/client';

const isLoggedIn = () => Cookies.get('TOKEN');

export const isUserLoggedInVar = makeVar(isLoggedIn());
```

How we create
reactive var

```
const cache = new InMemoryCache({
  typePolicies: {
    Query: {
      fields: {
        readNumOfOutofStockItems: {
          read(cart, { readField }) {
            // logic of OutofStock items
          }
        },
        isUserLoggedIn: {
          read() {
            return isUserLoggedInVar();
          }
        }
      }
    }
  }
});
```

```
const client = new ApolloClient({ cache });
```

How we declare
reactive var
here

Reactive variables



JS main.js

isLoggedInVar is a
Reactive Variable

```
const { useReactiveHook } from '@apollo/client';
const { isLoggedInVar } from './cache'
import App from 'pages/app';
import Login from 'pages/login'

const Main = () => {
  const isLoggedIn = useReactiveHook(isLoggedInVar);
  return isLoggedIn ? <App /> : <Login />
}
```

Directly use the
useReactiveHook

Updating Reactive variables

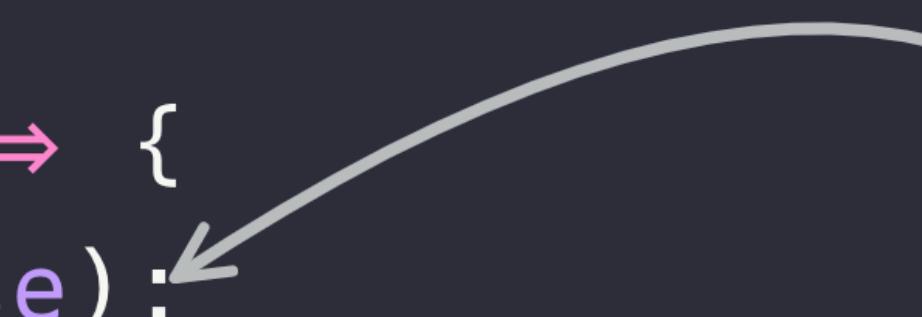


```
import React from 'react';
import { isUserLoggedInVar } from './cache';

const Logout = () => {
  const handleLogout = () => {
    isUserLoggedInVar(false);
  }

  return <button onClick={handleLogout}>Log out</button>;
}
```

This will trigger a component update



My fav ❤️



@kulkarniankita9

Keep an open mind



@kulkarniankita9

Before you go,

- All snippets and my slides are available at:
bit.ly/graphql-state-management-slides



@kulkarniankit9

To help you grow as a Developer

- Weekly Visual Snippets aka **Snacks** for GraphQL, React, Next.js
 - Latest Frontend News
 - A food snack for thought 🤔
 - Check out the [Frontend Snacks Newsletter](#)
- 👉 <https://frontendsnacks.dev/>



@kulkarniankit9

This is pretty much the end..

Questions?

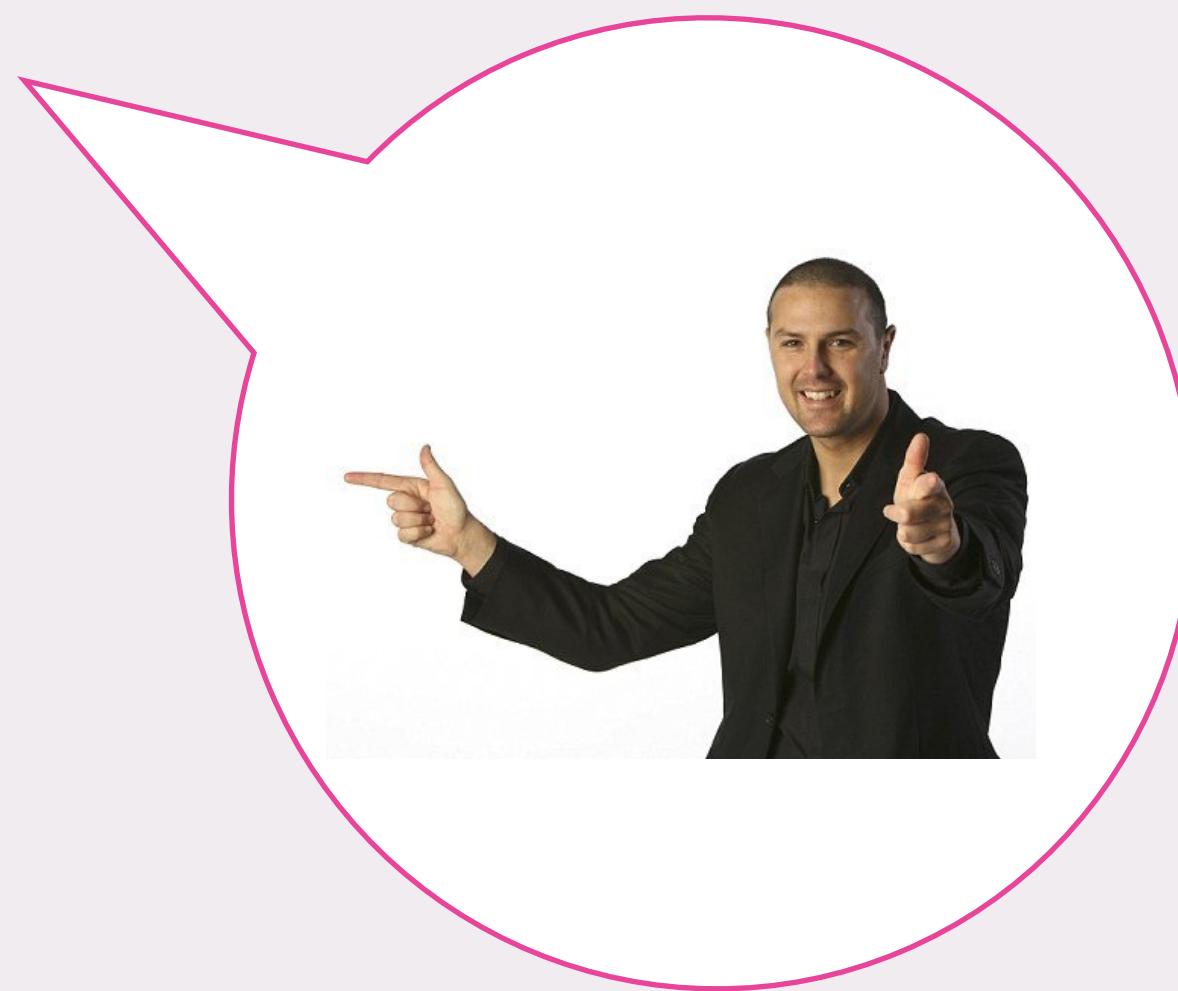
Comments?

Feedback...make it positive— 😊

Catch me after!!

@kulkarniankita9

You did it 🙌



@kulkarniankita9