Advanced Data Retrieval

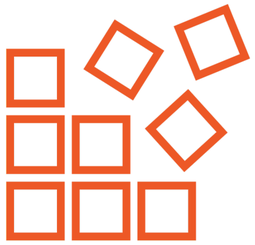
Christian Wenz
Consultant, Software Architect

@chwenz

Pain Points

Missing convenience/performance features (caching, pagination)

Clumsy state management



Ceremony

React Query

https://react-query.tanstack.com/

Console

```
npm install react-query
```

Application

```
import
  { QueryClient, QueryClientProvider }
  from "react-query";

function App() {
  return (
    <QueryClientProvider
      client={queryClient}>
      ...
    </QueryClientProvider>
  )
}
```

```
const query = useQuery(

  "events",

  () => axios.get(url)

);
```

◀ **React Query hook**

◀ **Hook key**

◀ **Async function returning Promise**

```
// query ==
   { status, data, error, isLoading }
```

◀ **Structure of returned data**

```
if (isLoading) {
  return ( <div>Loading ...</div>);
}
```

◀ **Message while loading**

```
return (
  query.data.map(...)
);
```

◀ **Output once data is available**

Pagination

Update API (page parameter)

Add page state variable

Add UI to go to another page

```
const mutation = useMutation(

  id => dispatch(addCart(id)),

  {
    onSuccess: (data, variables, context) => {}
  }

);



{ mutation.isSuccess ? <span>Success!</span> : null }
```

Mutations

Act upon completion of API calls

SWR

https://swr.vercel.app/

Console

```
npm install swr
```

Application

```
function fetcher(...args) {
  return fetch(...args).then(
    res => res.json());
};
const { data, error } =
 useSWR("/cart", fetcher);
```

```
import { mutate } from "swr";

function addCart(id) {

  fetch(
    "/cart",
    { method: "POST", body: `{"id": ${id}}` }
  )

  .then(() => { mutate("/cart") });

);
```

Mutations

Simplify the cart logic

Summary

Using React Query

Using SWR