

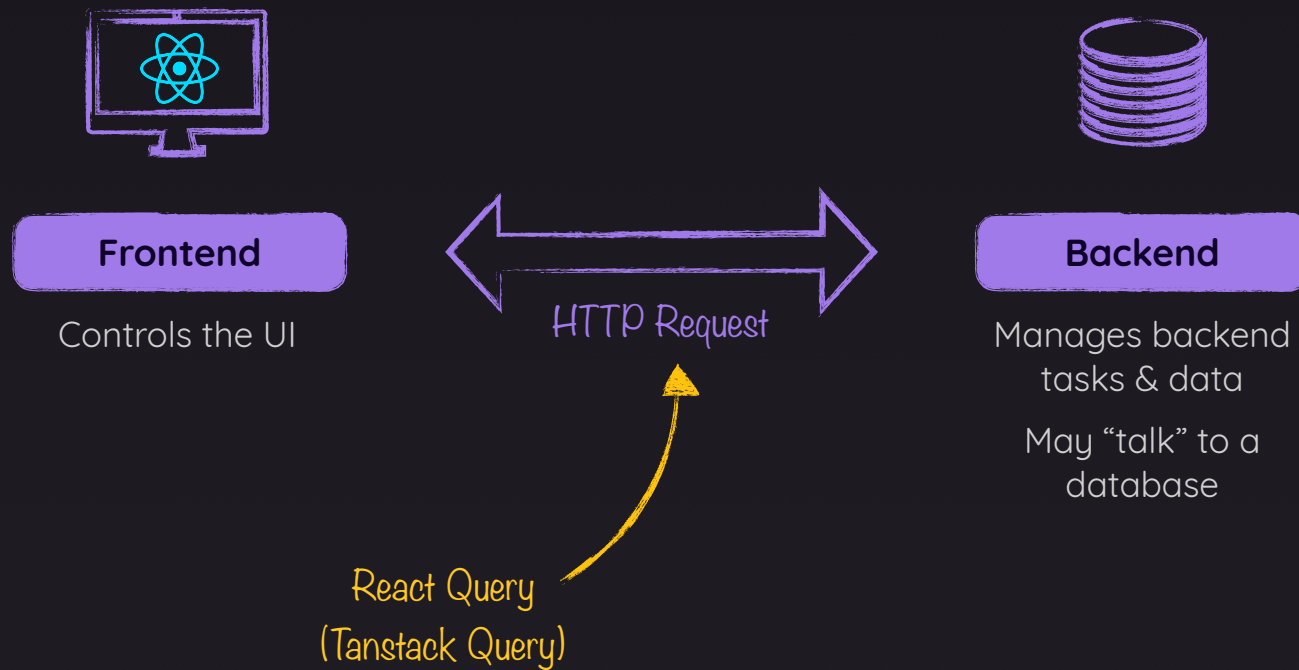
formerly React Query



Data Fetching with Tanstack Query

Sending HTTP Requests With Ease

- ▶ **What** Is Tanstack Query & **Why** Would You Use It?
- ▶ **Fetching & Mutating** Data
- ▶ **Configuring** Tanstack Query
- ▶ **Advanced Concepts:** Cache Invalidation, Optimistic Updating & More



What Is Tanstack Query?

What Is Tanstack Query?

A library that helps with sending
HTTP requests & keeping your
frontend UI in sync

You Don't Need Tanstack Query!

You Don't Need Tanstack Query!

But it can vastly simplify your code
(and your life as a developer)

Tanstack Query Does **Not** **Send HTTP** Requests

At least **not on its own**

You have to write the code that
sends the actual HTTP request

Tanstack Query then manages the
data, errors, caching & much more!

Tanstack Query Caches Query Data

```
useQuery({  
  queryKey: ['some-key'],  
  queryFn: fetchData  
});
```

Called in Component A @ 10:32



fetchData() is executed
& HTTP request is sent



Data is **received**

```
{  
  id: 'd1',  
  title: 'Some data'  
}
```

```
useQuery({  
  queryKey: ['some-key'],  
  queryFn: fetchData  
});
```

Called in Component B @ 10:34



*Cached data is reused & shown
on the screen immediately*

*Cached (stored) by Tanstack
Query*