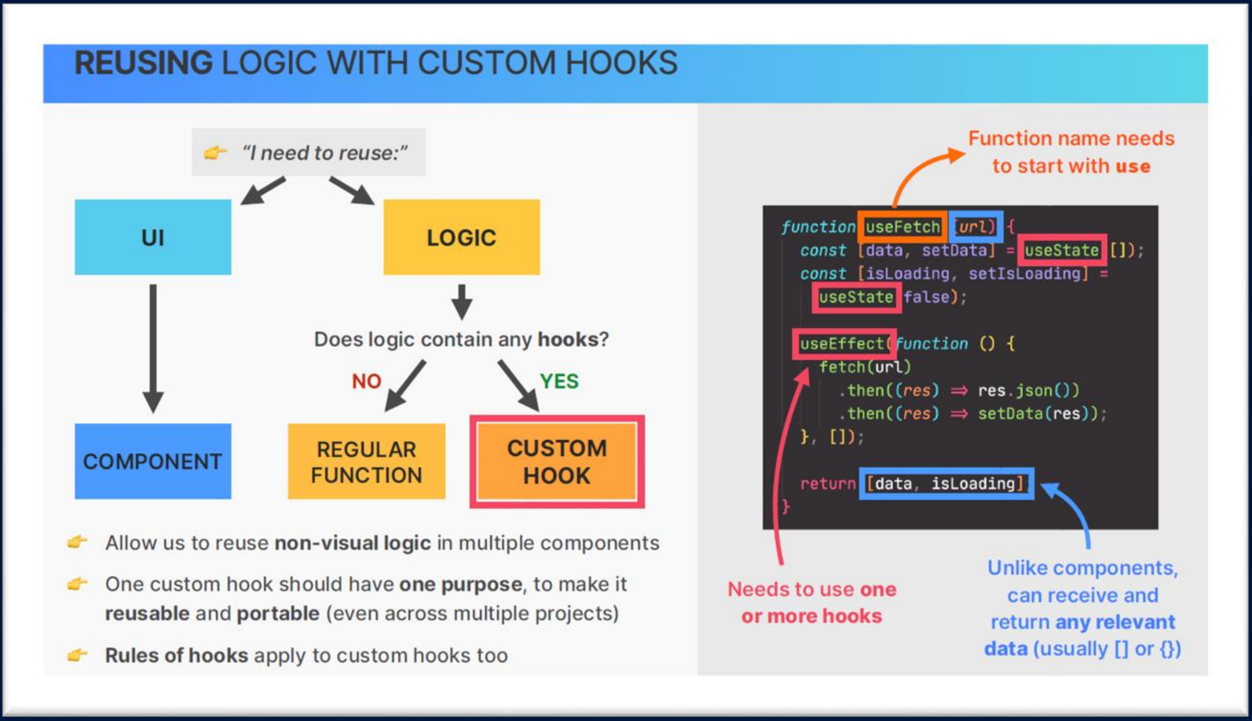
React

# CUSTOM HOOKS



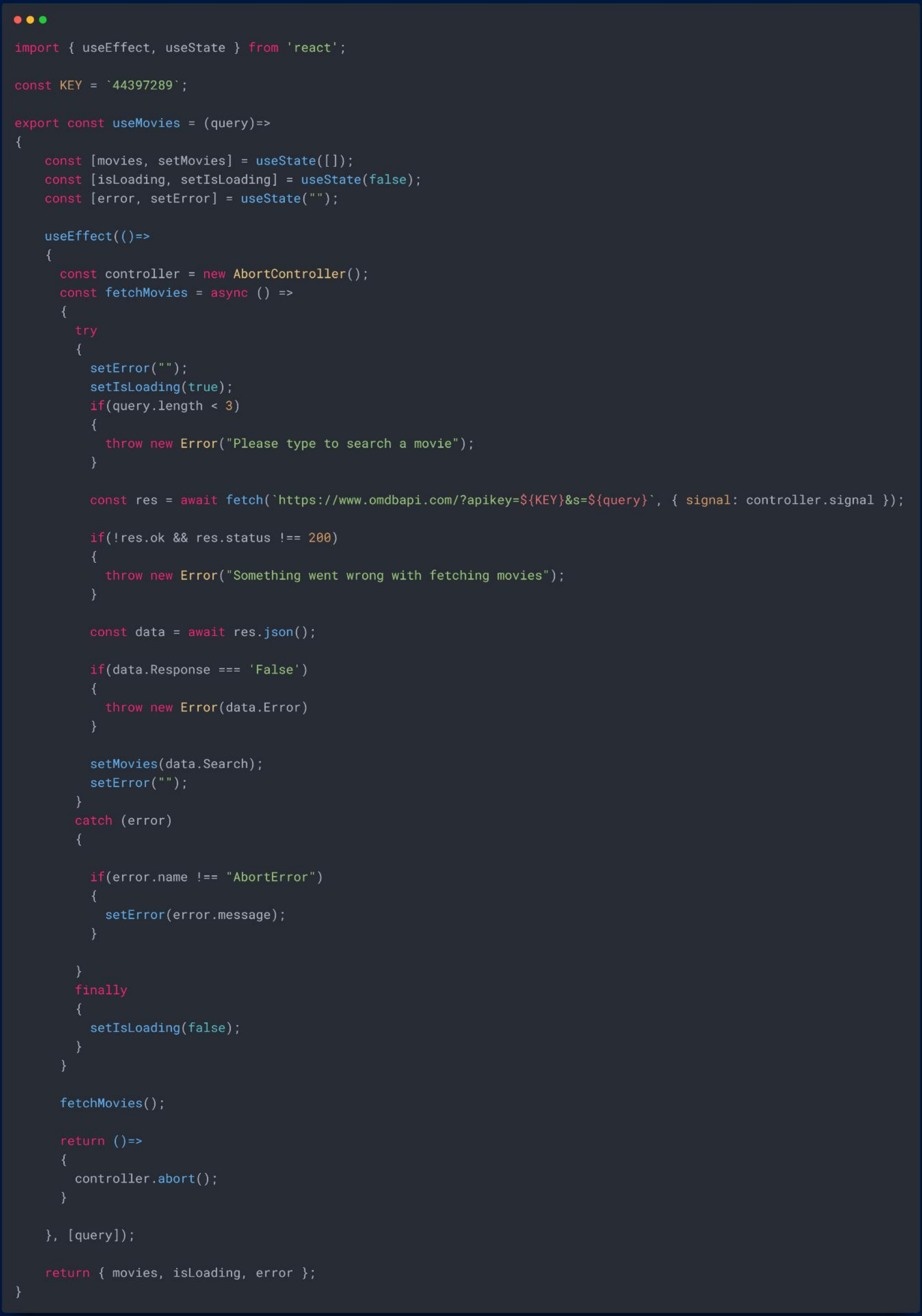
## **REUSING** LOGIC WITH CUSTOM HOOKS

*"I need to reuse:"*
UI
LOGIC
COMPONENT
Does logic contain any **hooks**?
NO
YES
REGULAR FUNCTION
CUSTOM HOOK
Function name needs to start with **use**
Needs to use **one or more hooks**
Unlike components, can receive and return **any relevant data** (usually [] or {})

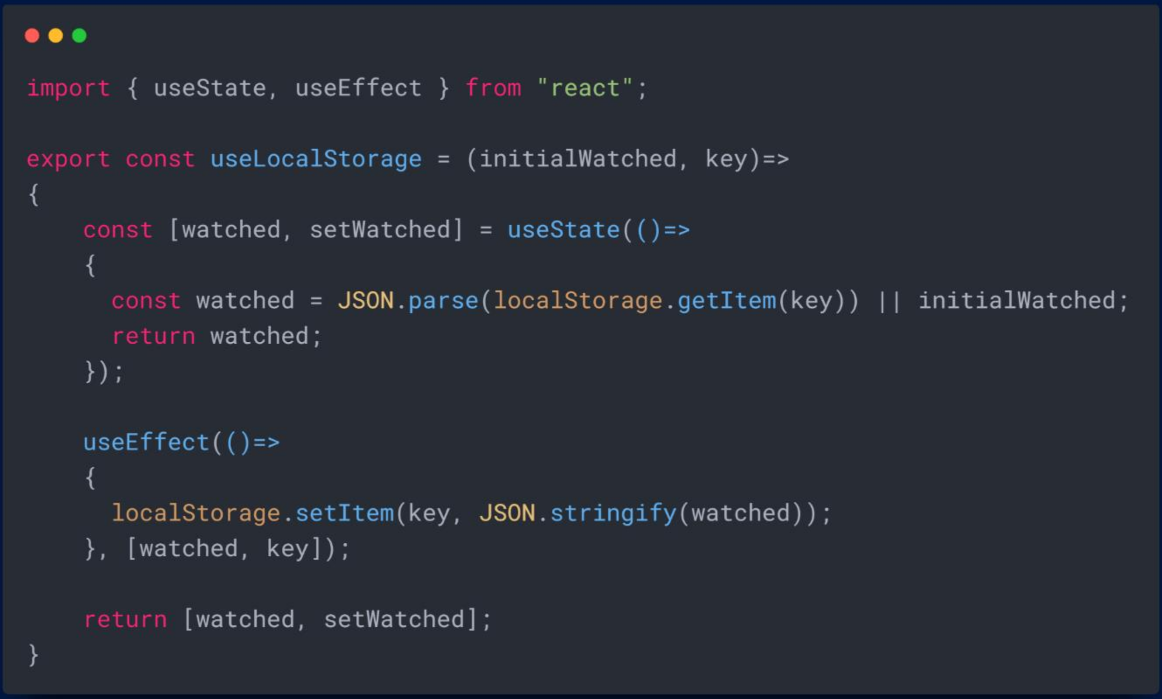
- Allow us to reuse **non-visual logic** in multiple components
- One custom hook should have **one purpose**, to make it **reusable** and **portable** (even across multiple projects)
- **Rules of hooks** apply to custom hooks too

```
function useFetch(url) {
  const [data, setData] = useState([]);
  const [isLoading, setIsLoading] =
    useState(false);

  useEffect(function () {
    fetch(url)
      .then((res) => res.json())
      .then((res) => setData(res));
  }, []);

  return [data, isLoading]
}
```

React

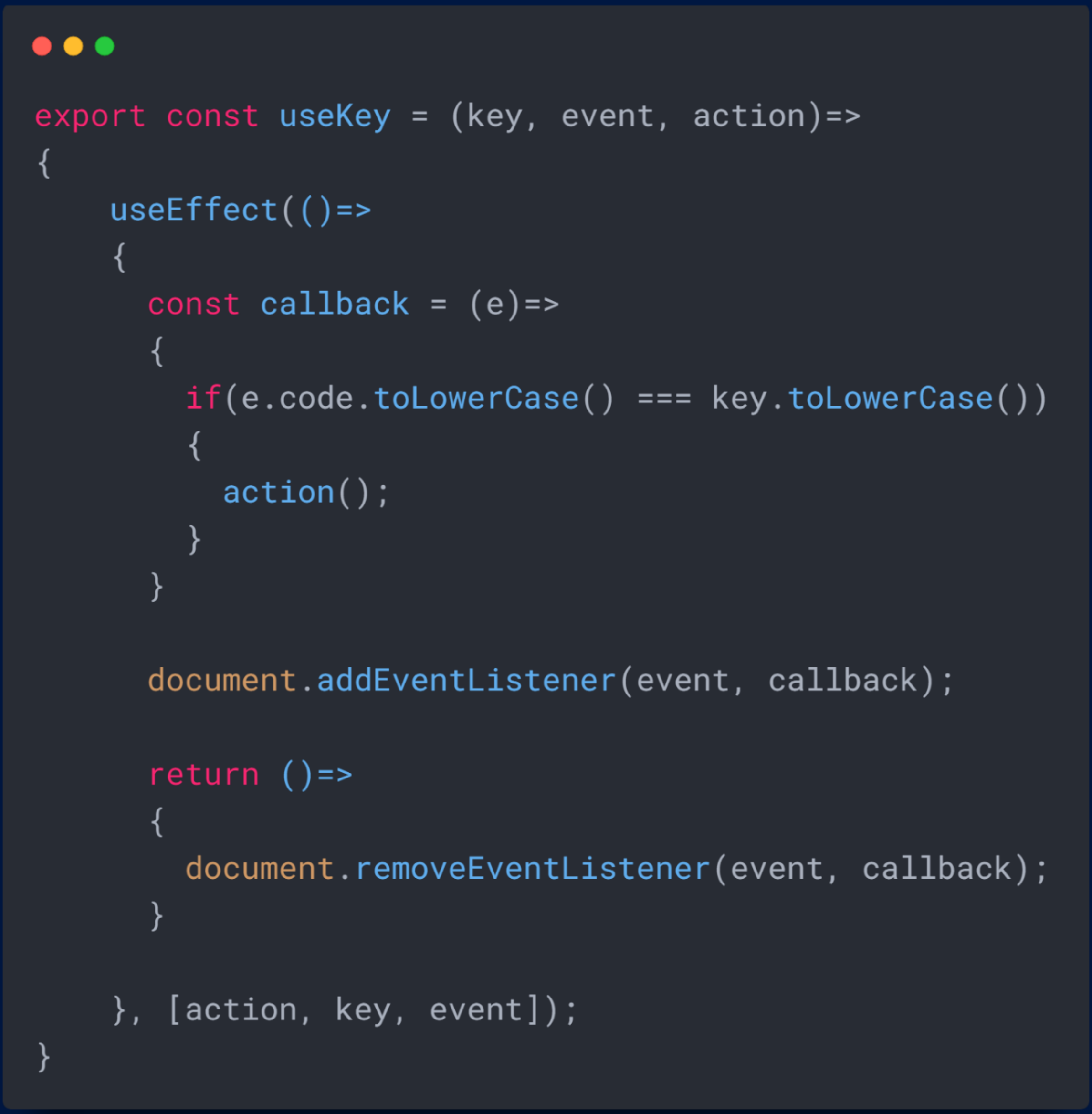


```
import { useEffect, useState } from 'react';

const KEY = `44397289`;

export const useMovies = (query)=>
{
    const [movies, setMovies] = useState([]);
    const [isLoading, setIsLoading] = useState(false);
    const [error, setError] = useState("");

    useEffect(()=>
    {
      const controller = new AbortController();
      const fetchMovies = async () =>
      {
        try
        {
          setError("");
          setIsLoading(true);
          if(query.length < 3)
          {
            throw new Error("Please type to search a movie");
          }

          const res = await fetch(`https://www.omdbapi.com/?apikey=${KEY}&s=${query}`, { signal: controller.signal });

          if(!res.ok && res.status !== 200)
          {
            throw new Error("Something went wrong with fetching movies");
          }

          const data = await res.json();

          if(data.Response === 'False')
          {
            throw new Error(data.Error)
          }

          setMovies(data.Search);
          setError("");
        }
        catch (error)
        {

          if(error.name !== "AbortError")
          {
            setError(error.message);
          }

        }
        finally
        {
          setIsLoading(false);
        }
      }

      fetchMovies();

      return ()=>
      {
        controller.abort();
      }

    }, [query]);

    return { movies, isLoading, error };
}
```



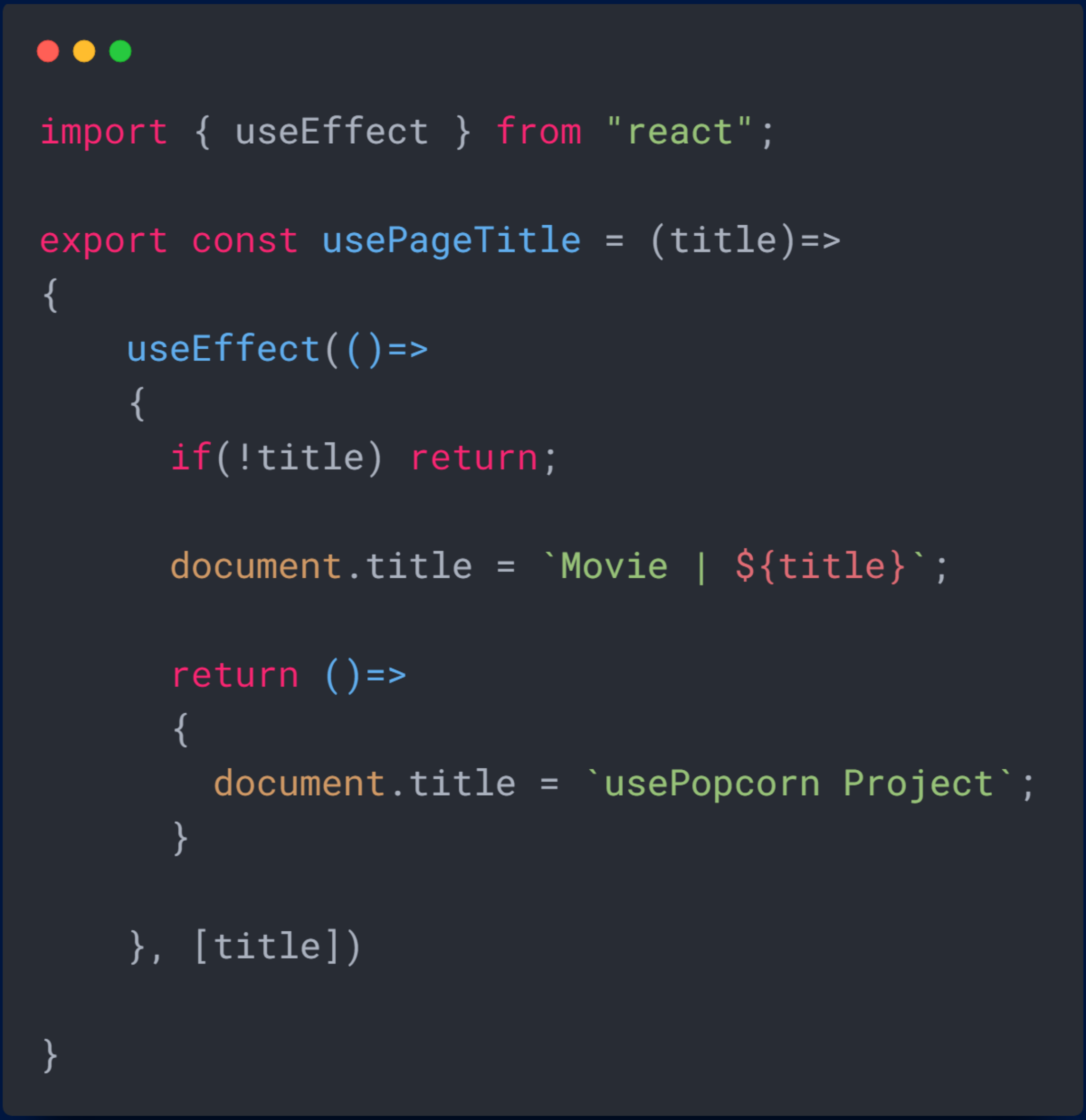
```
import { useState, useEffect } from "react";

export const useLocalStorage = (initialWatched, key)=>
{
    const [watched, setWatched] = useState(()=>
    {
      const watched = JSON.parse(localStorage.getItem(key)) || initialWatched;
      return watched;
    });

    useEffect(()=>
    {
      localStorage.setItem(key, JSON.stringify(watched));
    }, [watched, key]);

    return [watched, setWatched];
}
```

React



```javascript
export const useKey = (key, event, action)=>
{
    useEffect(()=>
    {
      const callback = (e)=>
      {
        if(e.code.toLowerCase() === key.toLowerCase())
        {
          action();
        }
      }

      document.addEventListener(event, callback);

      return ()=>
      {
        document.removeEventListener(event, callback);
      }

    }, [action, key, event]);
}
```

React

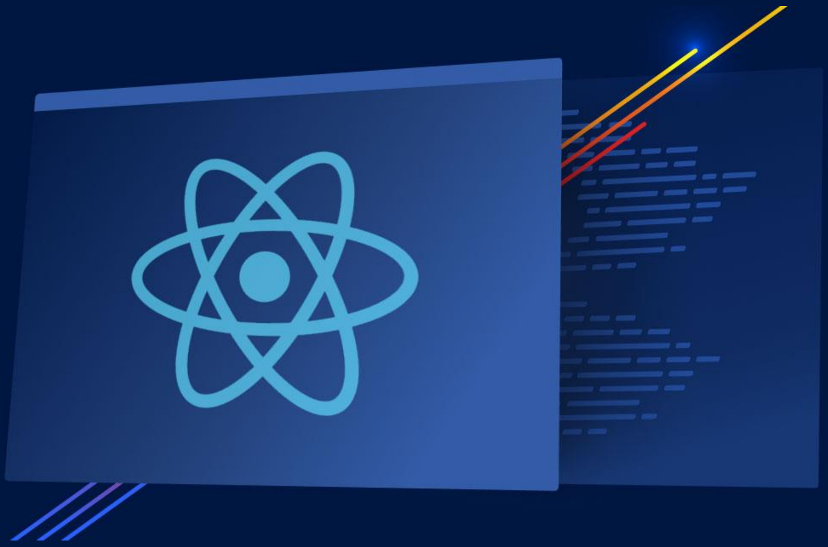


```
import { useEffect } from "react";

export const usePageTitle = (title)=>
{
    useEffect(()=>
    {
      if(!title) return;

      document.title = `Movie | ${title}`;

      return ()=>
      {
        document.title = `usePopcorn Project`;
      }

    }, [title])

}
```

React

# Do you find it helpful?

Let me know down in the comments.



Click To Follow For More On LinkedIN