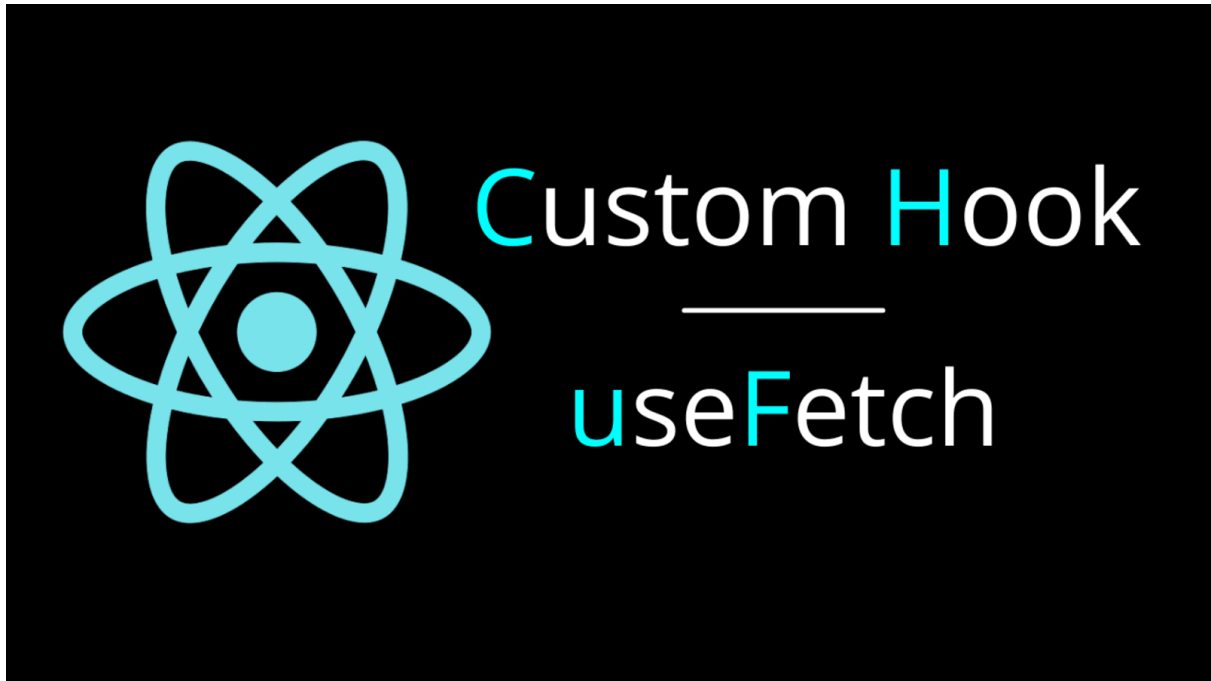


# React.Js Custom Hooks

---



Hooks are reusable functions.

When you have component logic that needs to be used by multiple components, we can extract that logic to a custom Hook.

Custom Hooks start with "use". Example: **useFetch**.

## Build a Hook

In the following code, we are **fetching** data in our Home component and **displaying** it.

We will use the **JSONPlaceholder** service to fetch fake data. This service is great for testing applications when there is no existing data.

To learn more, check out the **JavaScript** Fetch **API** section.

Use the **JSONPlaceholder** service to **fetch fake "todo"** items and display the titles on the page:

Example:

`index.js`:

```
import { useState, useEffect } from "react";
import ReactDOM from "react-dom";

const Home = () => {
  const [data, setData] = useState(null);

  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/todos")
      .then((res) => res.json())
      .then((data) => setData(data));
  }, []);
```

```
return (  
  <>  
    {data &&  
      data.map((item) => {  
        return <p key={item.id}>{item.title}</p>;  
      }}  
    </>  
  );  
};  
  
ReactDOM.render(<Home />, document.getElementById("root"));
```

The fetch logic may be needed in other components as well, so we will extract that into a custom Hook.

Move the fetch logic to a new file to be used as a custom Hook:

Example:

**useFetch.js:**

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

```
const useFetch = (url) => {  
  const [data, setData] = useState(null);
```

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

```
  return [data];  
};
```

```
export default useFetch;
```

## index.js:

```
import ReactDOM from "react-dom";
import useFetch from "../useFetch";

const Home = () => {
  const [data] = useFetch("https://jsonplaceholder.typicode.com/todos");

  return (
    <>
      {data &&
        data.map((item) => {
          return <p key={item.id}>{item.title}</p>;
        })
      }
    </>
  );
};

ReactDOM.render(<Home />, document.getElementById("root"));
```

## Example Explained

We have created a new file called **useFetch.js** containing a function called **useFetch** which contains all of the logic needed to fetch our data.

We removed the hard-coded **URL** and replaced it with a URL variable that can be passed to the custom Hook.

Lastly, we are returning our data from our Hook.

In **index.js**, we are importing our **useFetch** Hook and utilizing it like any other Hook. This is where we pass in the URL to fetch data.

Now we can reuse this custom Hook in any component to fetch data from any URL.