

30 Days Of React: Custom Hooks



LinkedIn



Follow @asabeneh

776

Author: [Asabeneh Yetayeh](#)

October, 2020

<< [Day 24](#) | [Day 26](#)>>

Custom Hooks

It is possible to make a custom hook on top of the available React hooks. For instance, when we fetch data we with use either fetch or axios to send an HTTP request and useEffect hooks to manage the React life cycle. Let's build useFetch custom hook on top of useEffect and useState.

We wrote this snippet of code in the previous section and we use useEffect hooks to fetch data from API. Now, let's convert this code to a custom hook. The naming convention for a custom hook is camelCase and it starts with the word use that is why we called our custom hook, useFetch.

```
import React, { useState, useEffect } from 'react'
import axios from 'axios'
import ReactDOM, { findDOMNode } from 'react-dom'

const Country = ({ country: { name, flag, population } }) => {
  return (
    <div className='country'>
      <div className='country_flag'>
        <img src={flag} alt={name} />
      </div>
    </div>
  )
}
```

```

    <h3 className='country_name'>{name.toUpperCase()}</h3>
    <div class='country_text'>
      <p>
        <span>Population: </span>
        {population}
      </p>
    </div>
  </div>
)
}

const App = (props) => {
  // setting initial state and method to update state
  const [data, setData] = useState([])

  useEffect(() => {
    fetchData()
  }, [])

  const fetchData = async () => {
    const url = 'https://restcountries.eu/rest/v2/all'
    try {
      const response = await fetch(url)
      const data = await response.json()
      setData(data)
    } catch (error) {
      console.log(error)
    }
  }

  return (
    <div className='App'>
      <h1>Fetching Data Using Hook</h1>
      <h1>Calling API</h1>
      <div>
        <p>There are {data.length} countries in the api</p>
        <div className='countries-wrapper'>
          {data.map((country) => (
            <Country country={country} />
          ))}
        </div>
      </div>
    </div>
  )
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Create a file name useFetch.js, and import useState and useEffect. Then transfer the state, useEffect and fetchData function part of the above code to the useFetch.js.

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

const useFetch = () => {
  const [data, setData] = useState([])

  const fetchData = async () => {
    const url = 'https://restcountries.eu/rest/v2/all'
    try {
      const response = await fetch(url)
      const data = await response.json()
      setData(data)
    } catch (error) {
      console.log(error)
    }
  }

  useEffect(() => {
    fetchData()
  }, [])
}
```

Then let's make the useFetch function to take a parameter. When we fetch data the only thing which changes is the API therefore let's pass a URL parameter for the function.

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

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

  const fetchData = async () => {
    try {
      const response = await fetch(url)
      const data = await response.json()
      setData(data)
    } catch (error) {
      console.log(error)
    }
  }

  useEffect(() => {
    fetchData()
  }, [])
}

export default useFetch
```

With the above code, we should manage to fetch the data but it is advisable to put the function in the useEffect and let's move the function code the useEffect.

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

export const useFetch = (url) => {
  const [data, setData] = useState([])

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await fetch(url)
        const data = await response.json()
        setData(data)
      } catch (error) {
        console.log(error)
      }
    }
    fetchData()
  }, [url])

  return data
}

export default useFetch
```

Now, let's combine everything and make it work.

```
// index.js

import React, { useState, useEffect } from 'react'
import axios from 'axios'
import ReactDOM, { findDOMNode } from 'react-dom'
import useFetch from './useFetch'

const Country = ({ country: { name, flag, population } }) => {
  return (
    <div className='country'>
      <div className='country_flag'>
        <img src={flag} alt={name} />
      </div>
      <h3 className='country_name'>{name.toUpperCase()}</h3>
      <div class='country_text'>
        <p>
          <span>Population: </span>
          {population}
        </p>
      </div>
    </div>
  )
}

const App = (props) => {
  const url = 'https://restcountries.eu/rest/v2/all'
```

```
const data = useFetch(url)

return (
  <div className='App'>
    <h1>Custom Hooks</h1>
    <h1>Calling API</h1>
    <div>
      <p>There are {data.length} countries in the api</p>
      <div className='countries-wrapper'>
        {data.map((country) => (
          <Country country={country} />
        ))}
      </div>
    </div>
  </div>
)

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

The useState and useEffect hooks are most common React hooks which you use on daily bases. In addition to the basic hook, there are additional hooks which are not used often. You do not have to know how to use all the hooks. The useState, useEffect and useRef are very important hooks and it is recommended to know how to use them.

Exercises

Note: Continue building the countries application

1 Build the following application using [countries API](#). [DEMO](#)

🎉 CONGRATULATIONS ! 🎉

[<< Day 24](#) | [Day 26>>](#)