

# 30 Days Of React: Fetch and Axios



LinkedIn



Follow @asabeneh

776

Author: [Asabeneh Yetayeh](#)

October, 2020

<< [Day 17](#) | [Day 19](#) >>

- [Fetch and Axios](#)
  - [Fetch](#)
  - [Axios](#)
- [Exercises](#)
  - [Exercises: Level 1](#)
  - [Exercises: Level 2](#)
  - [Exercises: Level 3](#)

## Fetch and Axios

### Fetch

Currently, JavaScript provides a fetch API to make HTTP requests. Fetch might not be supported by all browsers therefore we have install addition package for browser supports. However, if we use Axios we do not need to use additional package for browser support. Axios code seems neater than fetch. In this section we will see the difference between fetch and axios. May be if you want to know the browser support of fetch you check out on [caniuse](#) website. As of today, it has 95.62% browser support.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'

const Country = ({
  country: { name, capital, flag, languages, population, currency },
}) => {
  const formattedCapital =
    capital.length > 0 ? (
      <>
        <span>Capital: </span>
        {capital}
      </>
    ) : (
      ''
    )
  const formatLanguage = languages.length > 1 ? `Languages` : `Language`
  console.log(languages)
  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>{formattedCapital}</p>
        <p>
          <span>{formatLanguage}: </span>
          {languages.map((language) => language.name).join(', ')}
        </p>
        <p>
          <span>Population: </span>
          {population.toLocaleString()}
        </p>
        <p>
          <span>Currency: </span>
          {currency}
        </p>
      </div>
    </div>
  )
}

class App extends Component {
  state = {
    data: [],
  }

  componentDidMount() {
    const url = 'https://restcountries.eu/rest/v2/all'
    fetch(url)
      .then((response) => {
        return response.json()
      })
  }
}

```

```

        .then((data) => {
            console.log(data)
            this.setState({
                data,
            })
        })
        .catch((error) => {
            console.log(error)
        })
    }

    render() {
        return (
            <div className='App'>
                <h1>React Component Life Cycle</h1>
                <h1>Calling API</h1>
                <div>
                    <p>There are {this.state.data.length} countries in the api</p>
                    <div className='countries-wrapper'>
                        {this.state.data.map((country) => (
                            <Country country={country} />
                        ))}
                    </div>
                </div>
            </div>
        )
    }
}

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

```

We can implement async and await and make the above code short and clean.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'

const Country = ({
    country: { name, capital, flag, languages, population, currency },
}) => {
    const formattedCapital =
        capital.length > 0 ? (
            <>
                <span>Capital: </span>
                {capital}
            </>
        ) : (
            ''
        )
    const formatLanguage = languages.length > 1 ? `Languages` : `Language`
    console.log(languages)
    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>{formattedCapital}</p>
        <p>
          <span>{formatLanguage}: </span>
          {languages.map((language) => language.name).join(', ')}
        </p>
        <p>
          <span>Population: </span>
          {population.toLocaleString()}
        </p>
        <p>
          <span>Currency: </span>
          {currency}
        </p>
      </div>
    </div>
  )
}

class App extends Component {
  state = {
    data: [],
  }

  componentDidMount() {
    this.fetchCountryData()
  }

  fetchCountryData = async () => {
    const url = 'https://restcountries.eu/rest/v2/all'
    const response = await fetch(url)
    const data = await response.json()
    this.setState({
      data,
    })
  }

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

```

```

        </div>
      </div>
    )
  }
}

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

```

If we use `async` and `await` we handle the error using `try` and `catch`. Let's apply a `try catch` block in the above code.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'

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>
        <p>
          <span>Currency: </span>
          {currency}
        </p>
      </div>
    </div>
  )
}

class App extends Component {
  state = {
    data: [],
  }

  componentDidMount() {
    this.fetchCountryData()
  }

  fetchCountryData = async () => {
    const url = 'https://restcountries.eu/rest/v2/all'
    try {
      const response = await fetch(url)
      const data = await response.json()
      this.setState({

```

```

        data,
      })
    } catch (error) {
      console.log(error)
    }
  }
}

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

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

```

Now, let's see how to do the same API call using axios.

How can do fetch if we have multiple API two call ?

## Axios

Axios is a third party package and we need to install it using npm. It is the most popular way to make HTTP requests(GET, POST, PUT, PATCH, DELETE). In this example, we will cover only a get request.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import axios from 'axios'

const Country = ({
  country: { name, capital, flag, languages, population, currency },
}) => {
  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>
)
}

class App extends Component {
  state = {
    data: [],
  }

  componentDidMount() {
    const url = 'https://restcountries.eu/rest/v2/all'
    axios
      .get(url)
      .then((response) => {
        this.setState({
          data: response.data,
        })
      })
      .catch((error) => {
        console.log(error)
      })
  }

  render() {
    return (
      <div className='App'>
        <h1>React Component Life Cycle</h1>
        <h1>Calling API</h1>
        <div>
          <p>There are {this.state.data.length} countries in the api</p>
          <div className='countries-wrapper'>
            {this.state.data.map((country) => (
              <Country country={country} />
            ))}
          </div>
        </div>
      </div>
    )
  }
}

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

```

Let's also implement the axios fetching using async and await.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import axios from 'axios'

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>
  )
}

class App extends Component {
  state = {
    data: [],
  }

  componentDidMount() {
    this.fetchCountryData()
  }

  fetchCountryData = async () => {
    const url = 'https://restcountries.eu/rest/v2/all'
    try {
      const response = await axios.get(url)
      const data = await response.data
      this.setState({
        data,
      })
    } catch (error) {
      console.log(error)
    }
  }

  render() {
    return (
      <div className='App'>
        <h1>React Component Life Cycle</h1>
        <h1>Calling API</h1>
        <div>
          <p>There are {this.state.data.length} countries in the api</p>
          <div className='countries-wrapper'>
            {this.state.data.map((country) => (
              <Country country={country} />
            ))}
          </div>
        </div>
      </div>
    )
  }
}

```



```
        </div>
      </div>
    </div>
  )
}
}

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

As you have seen, there is no much difference between fetch and axios. But I recommend you to use more axios than fetch because of browser support and easy of use.

## Exercises

### Exercises: Level 1

1. What is HTTP request?
2. What are the most common HTTP requests?
3. What is fetch?
4. What is axios?
5. What is the difference between fetch and axios?
6. Do you prefer fetch to axios for make HTTP requests?

### Exercises: Level 2

1. Find the average metric weight and life span of cats in the following [API](#). There are 67 breeds of cats in the API.



### Exercises: Level 3

1. How many countries do have cat breeds?
2. Which country has the highest number of cat breeds?
3. Arrange countries in ascending order based on the number of cat breeds they have?

🎉 CONGRATULATIONS ! 🎉

<< [Day 17](#) | [Day 19](#) >>