The React Data Fetching Cheatsheet

React Playground: https://codesandbox.io/s/elegant-ardinghelli-u83v0?file=/src/App.js

Use with Fetch API

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

export default function App() {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    fetch("https://randomuser.me/api")
      .then((response) => {
        if (response.ok) {
          return response.json();
        }
        throw response;
      })
      .catch((error) => {
        console.error("Error fetching data: ", error);
        setError(error);
      })
      .finally(() => {
        setLoading(false);
      });
  }, []);

  if (loading) return "Loading...";
  if (error) return "Error!";

  return (
    <>
      <img src={data.results[0].picture.medium} alt="random user" />
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </>
  );
}
```

Use with axios

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

export default function App() {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    axios("https://randomuser.me/api")
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error("Error fetching data: ", error);
        setError(error);
      })
      .finally(() => {
        setLoading(false);
      });
  }, []);

  if (loading) return "Loading...";
  if (error) return "Error!";

  return (
    <>
      <img src={data.results[0].picture.medium} alt="random user" />
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </>
  );
}
```

Use with async / await syntax

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

export default function App() {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    getData()
  }, []);

  async function getData() {
    await axios("https://randomuser.me/api")
```

```
    .then((response) => {
      setData(response.data);
    })
    .catch((error) => {
      console.error("Error fetching data: ", error);
      setError(error);
    })
    .finally(() => {
      setLoading(false);
    });
  }

  if (loading) return "Loading...";
  if (error) return "Error!";

  return (
    <>
      <img src={data.results[0].picture.medium} alt="random user" />
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </>
  );
}
```

Use with Custom React Hook (useFetch)

```
import React from "react";
import useFetch from "react-fetch-hook";

export default function App() {
  const { isLoading, error, data } = useFetch("https://randomuser.me/api");

  if (isLoading) return "Loading...";
  if (error) return "Error!";

  return (
    <>
      <img src={data.results[0].picture.medium} alt="random user" />
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </>
  );
}
```

Use with React Query

```
// src/index.js
import { StrictMode } from "react";
```

```
import ReactDOM from "react-dom";
import { QueryClientProvider, QueryClient } from 'react-query'
import { ReactQueryDevtools } from 'react-query/devtools'

import App from "./App";

const queryClient = new QueryClient()

const rootElement = document.getElementById("root");
ReactDOM.render(
  <QueryClientProvider client={queryClient}>
    <App />
    <ReactQueryDevtools />
  </QueryClientProvider>,
  rootElement
);
```

```
// src/App.js
import React from "react";
import { useQuery } from "react-query";

export default function App() {
  const { isLoading, error, data } = useQuery("random-user", () =>
    fetch("https://randomuser.me/api").then((res) => res.json())
  );

  if (isLoading) return "Loading...";
  if (error) return "Error!";

  return (
    <>
      <img src={data.results[0].picture.medium} alt="random user" />
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </>
  );
}
```