Front End

Class 14

September 29, 2021

Agenda

- Kahoot
- Review
- Studio Review plus studio

Connecting a Backend & Database

Sending Http Requests

ACADEMIND

Module Content

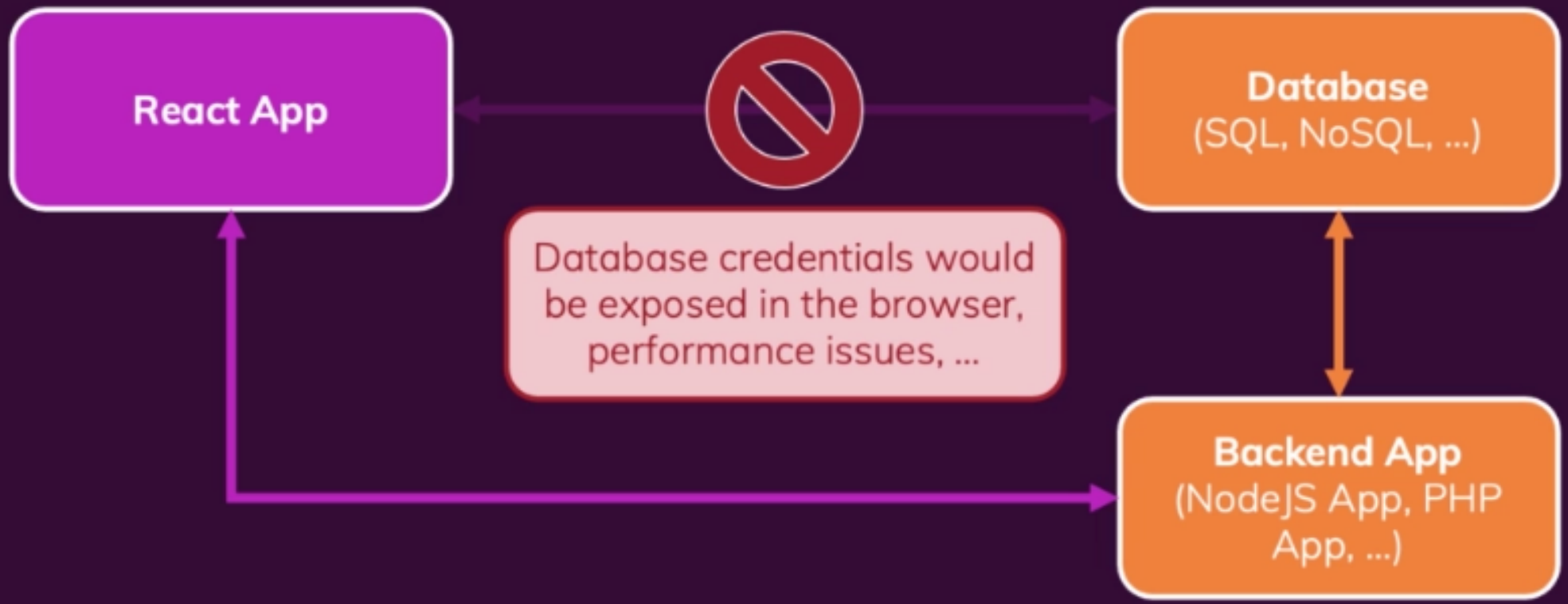
How do React Apps Interact with Databases?

Sending Http Requests & Using Responses

Handling Errors & Loading State

ACADEMIND

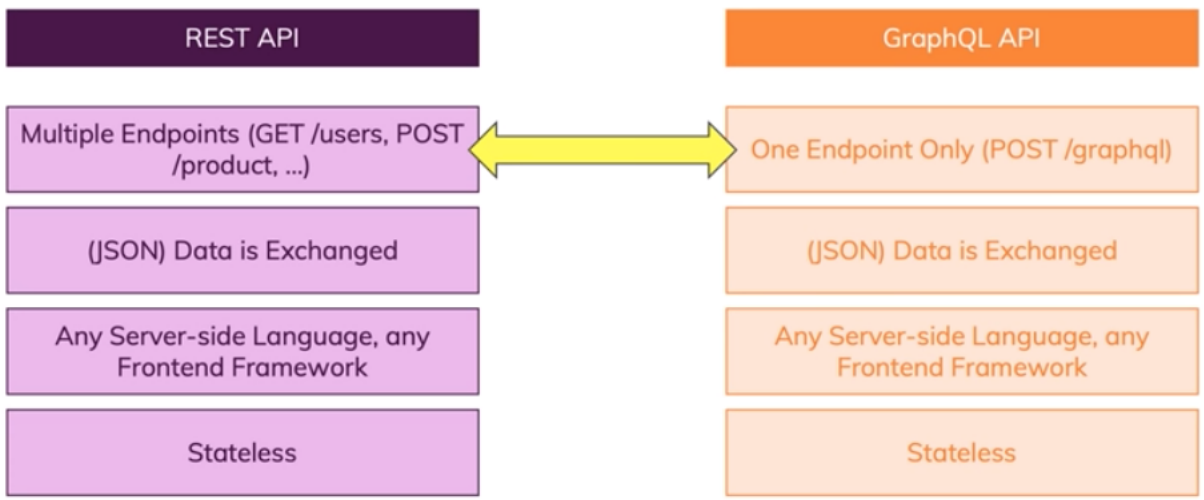
Browser-side Apps Don't Directly Talk To Databases



SWITCH TO REST PRESENTATION

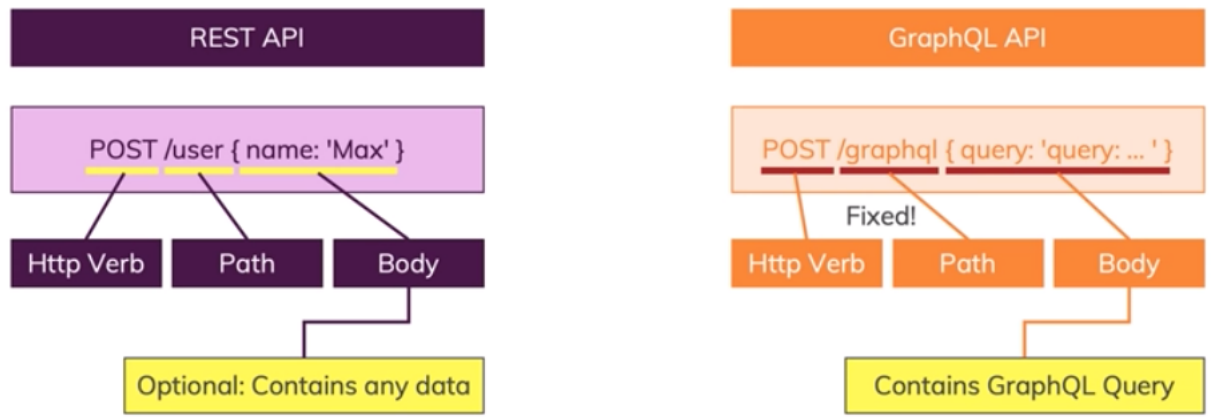
ACADEMIND

What?



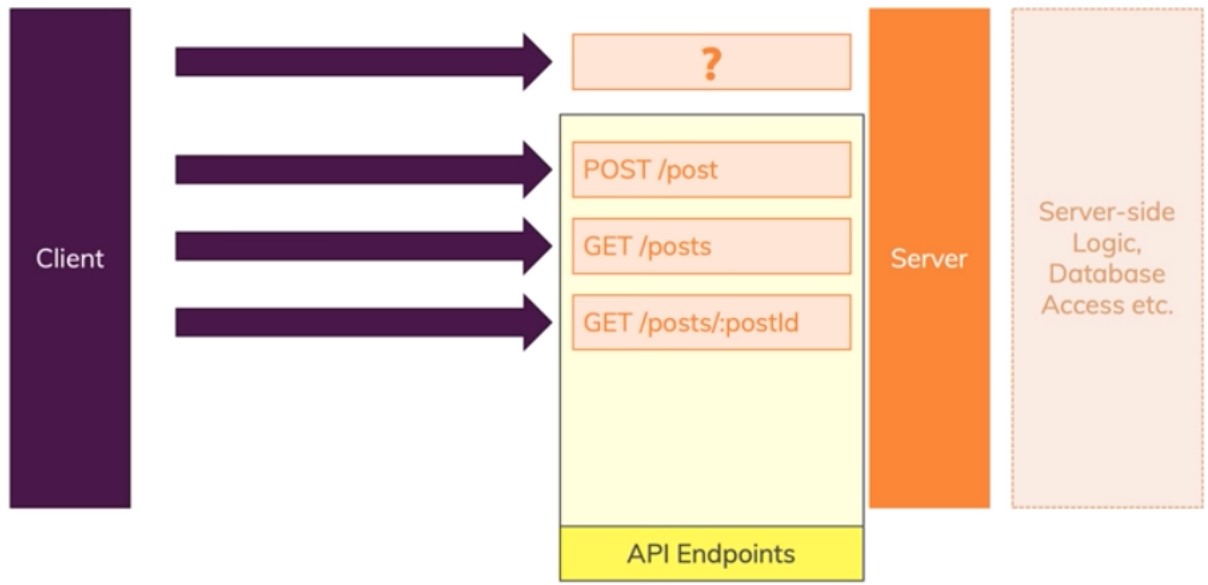
ACADEMIND

URL-Driven vs Query-Language



ACADEMIND

What's a REST API?



ACADEMIND

REST & Http Methods (Http Verbs)

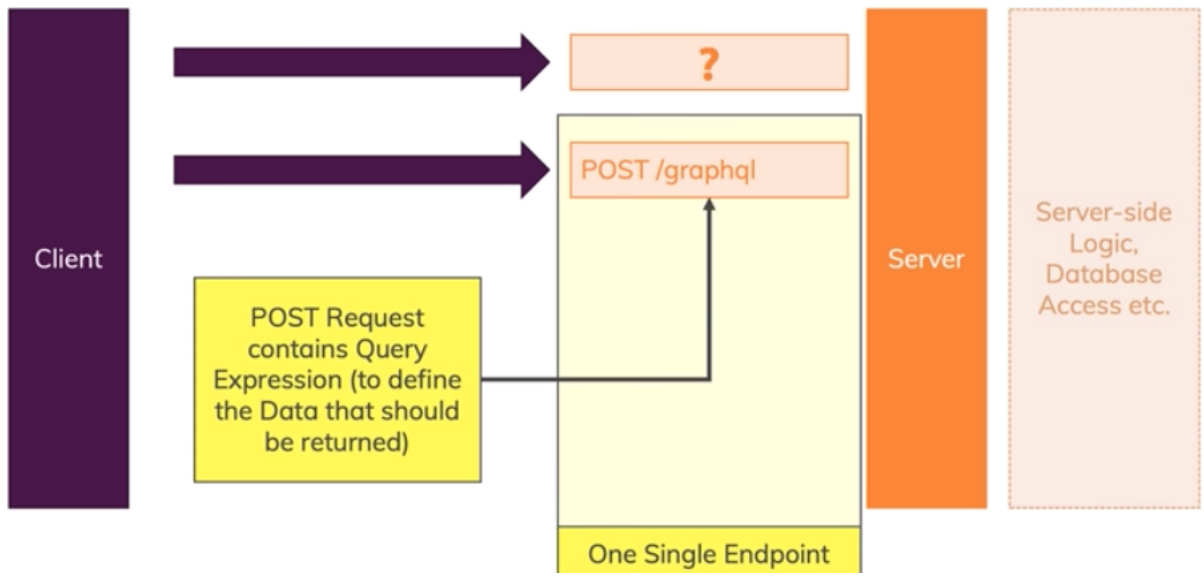
More than just GET & POST

GET	POST	PUT
Get a Resource from the Server	Post a Resource to the Server (i.e. create or append Resource)	Put a Resource onto the Server (i.e. create or overwrite a Resource)

PATCH	DELETE	OPTIONS
Update parts of an existing Resource on the Server	Delete a Resource on the Server	Determine whether follow-up Request is allowed (sent automatically)

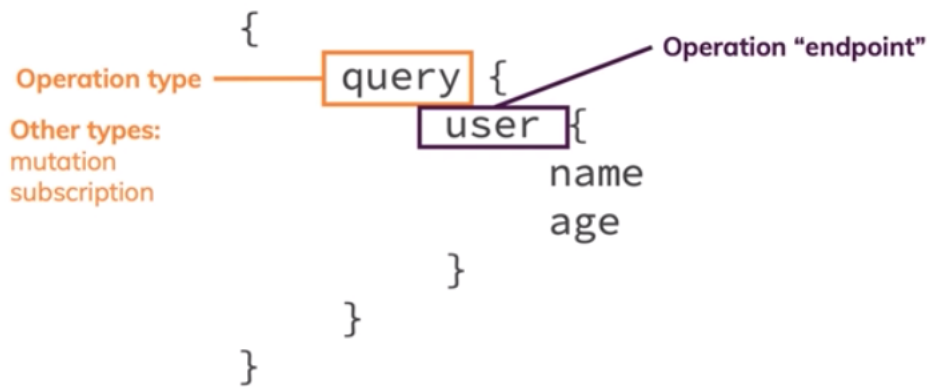
ACADEMIND

How does GraphQL Work?



ACADEMIND

A GraphQL Query



Instead use a dummy api with dummy request.

API – Application programming Interface

- Clearly defined rules to achieve results/do tasks
- http requests
- REST versus GraphQL
- Send request to URL to get data

REST slides + POSTMAN demo

Still writing JS code inside the React app.

Axios is a package to send and receive http responses.

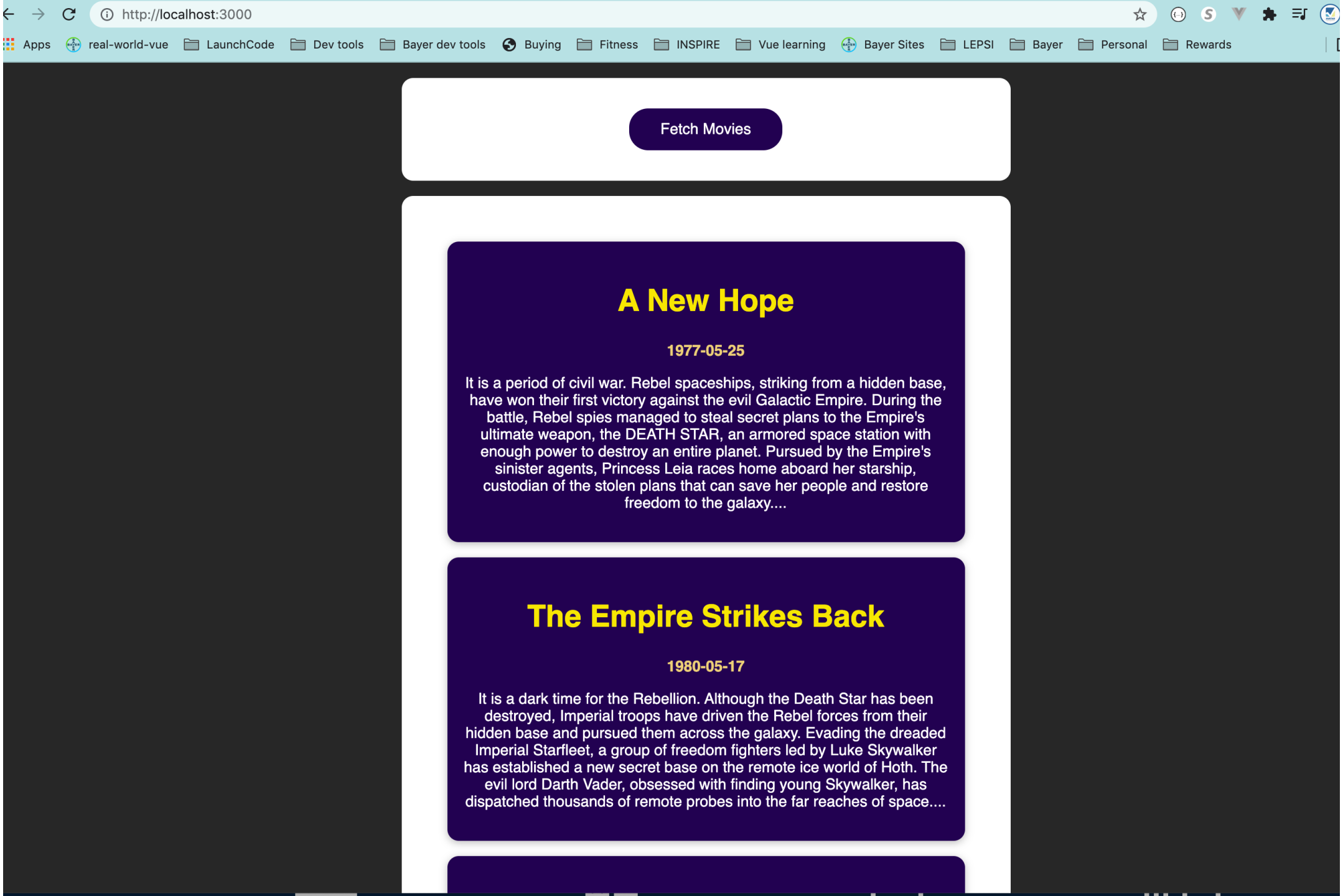
Fetch is the build in JS. Allows to fetch and send data as well.
Default is get method

Fetch returns a promise that is an object that eventually return data.
That is what async is about.

JSON -> can be converted to JS object

```
function fetchMoviesHandler() {
  fetch('https://swapi.dev/api/films/')
    .then((response) => {
      return response.json();
    })
    .then((data) => {
      const transformedMovies = data.results.map((movieData) => {
        return {
          id: movieData.episode_id,
          title: movieData.title,
          openingText: movieData.opening_crawl,
          releaseDate: movieData.release_date,
        };
      });
      setMovies(transformedMovies);
    })
    .catch();
}
```

```
async function fetchMoviesHandler() {
  const response = await fetch('https://swapi.dev/api/films/');
  const data = await response.json();

  const transformedMovies = data.results.map((movieData) => {
    return {
      id: movieData.episode_id,
      title: movieData.title,
      openingText: movieData.opening_crawl,
      releaseDate: movieData.release_date,
    };
  });
  setMovies(transformedMovies);
}
```


http://localhost:3000
Apps
real-world-vue
LaunchCode
Dev tools
Bayer dev tools
Buying
Fitness
INSPIRE
Vue learning
Bayer Sites
LEPSI
Bayer
Personal
Rewards
Fetch Movies
A New Hope
1977-05-25
It is a period of civil war. Rebel spaceships, striking from a hidden base, have won their first victory against the evil Galactic Empire. During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet. Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship, custodian of the stolen plans that can save her people and restore freedom to the galaxy....
The Empire Strikes Back
1980-05-17
It is a dark time for the Rebellion. Although the Death Star has been destroyed, Imperial troops have driven the Rebel forces from their hidden base and pursued them across the galaxy. Evading the dreaded Imperial Starfleet, a group of freedom fighters led by Luke Skywalker has established a new secret base on the remote ice world of Hoth. The evil lord Darth Vader, obsessed with finding young Skywalker, has dispatched thousands of remote probes into the far reaches of space....

Loading state when data loads
Need a spinner or something similar

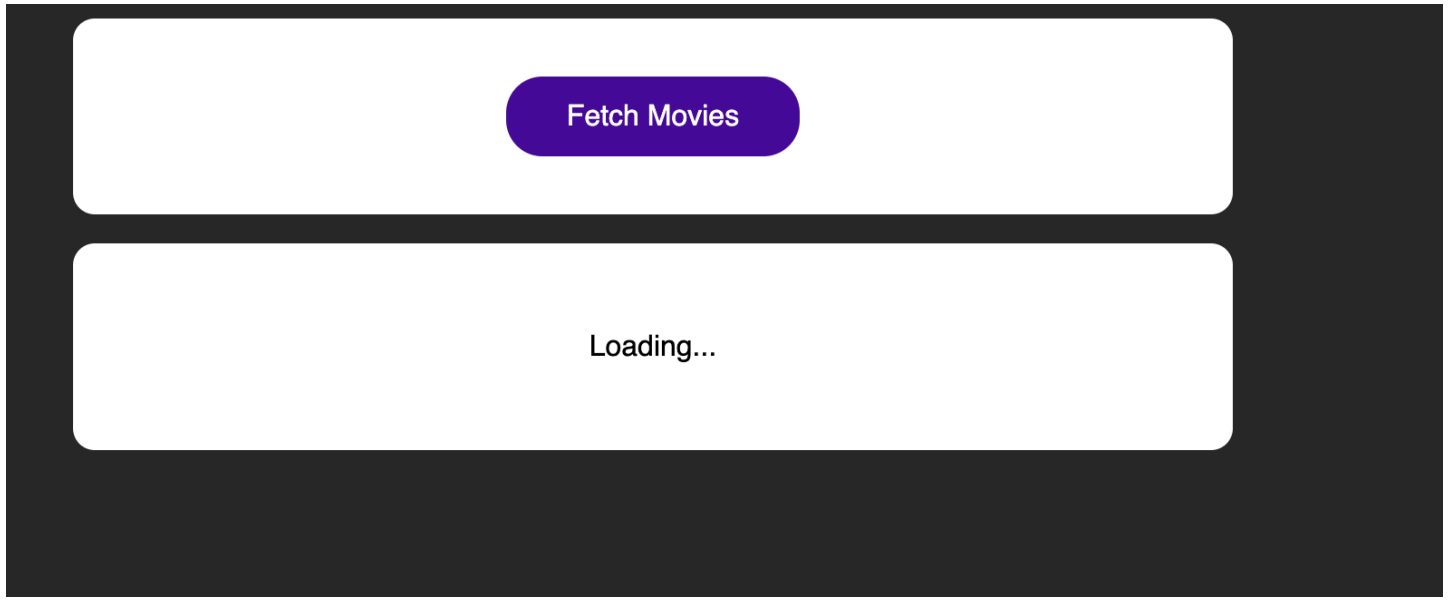
Managing state.

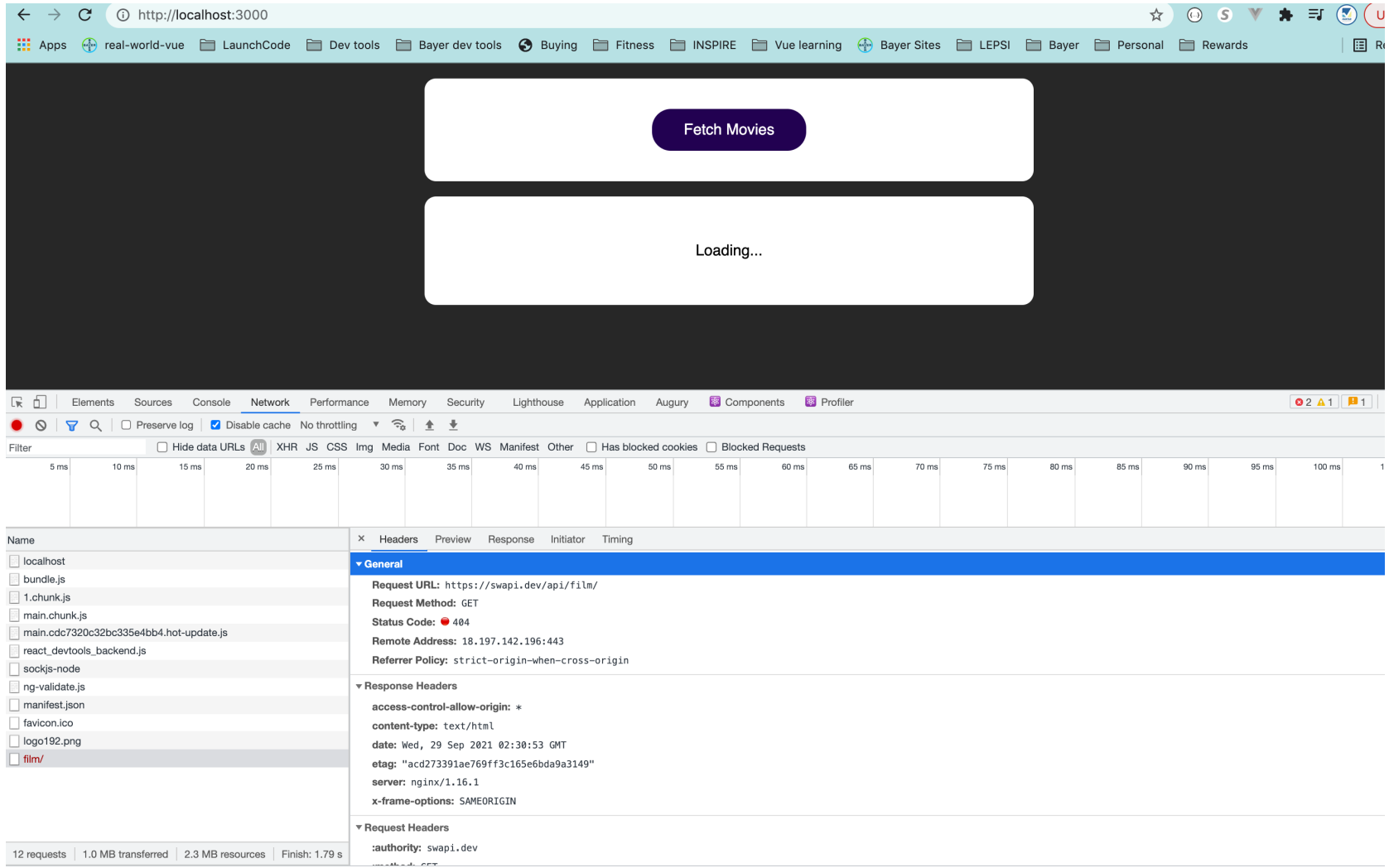
```jsx
import './App.css';

function App() {
  const [movies, setMovies] = useState([]);
  const [isLoading, setIsLoading] = useState(false);

  async function fetchMoviesHandler() {
    setIsLoading(true);
    const response = await fetch('https://swapi.dev/api/films/');
    const data = await response.json();
    const transformedMovies = data.results.map((movieData) => {
      return {
        id: movieData.episode_id,
        title: movieData.title,
        openingText: movieData.opening_crawl,
        releaseDate: movieData.release_date,
      };
    });
    setMovies(transformedMovies);
    setIsLoading(false);
  }

  return (
    <React.Fragment>
      <section>
        <button onClick={fetchMoviesHandler}>Fetch Movies</button>
      </section>
      <section>
        {!isLoading && movies.length > 0 && <MoviesList movies={movies} />}
        {!isLoading && movies.length === 0 && <p> Found no movies </p>}
        {isLoading && <p>Loading...</p>}
      </section>
    </React.Fragment>
  );
}

export default App;
```

Handling HTTP Errors

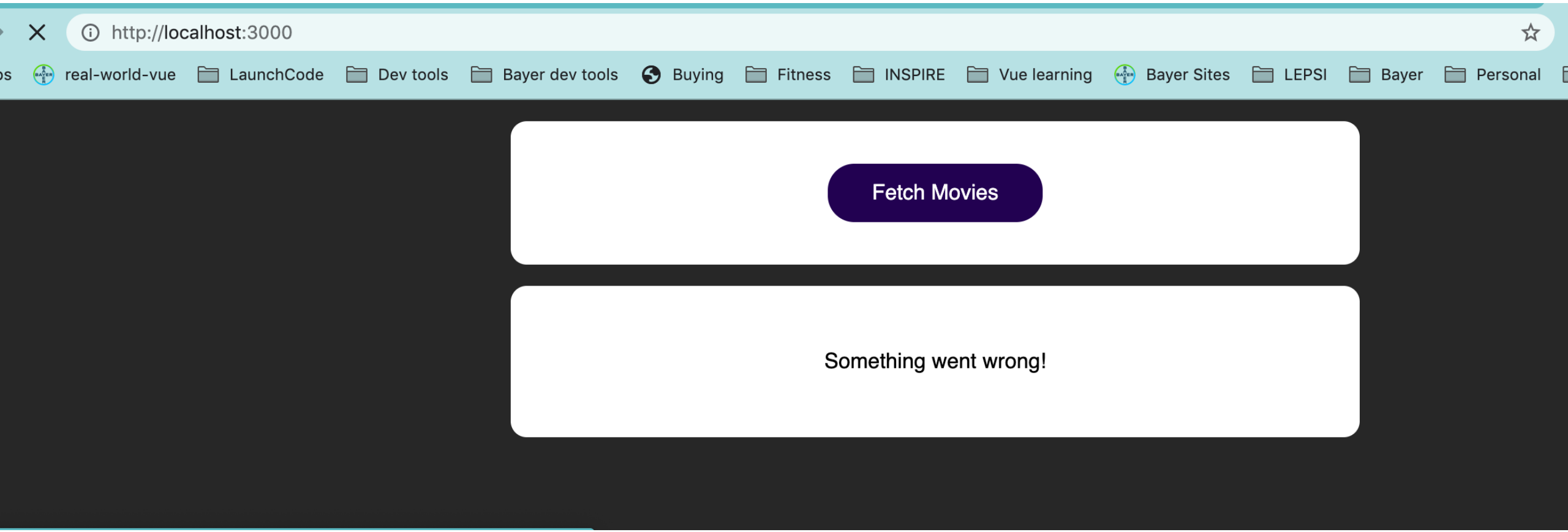
https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Invalid URL




http://localhost:3000
Apps
real-world-vue
LaunchCode
Dev tools
Bayer dev tools
Buying
Fitness
INSPIRE
Vue learning
Bayer Sites
LEPSI
Bayer
Personal
Rewards
Fetch Movies
Loading...
Elements
Sources
Console
Network
Performance
Memory
Security
Lighthouse
Application
Augury
Components
Profiler
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Has blocked cookies
Blocked Requests
Name
localhost
bundle.js
1.chunk.js
main.chunk.js
main.cdc7320c32bc335e4bb4.hot-update.js
react_devtools_backend.js
sockjs-node
ng-validate.js
manifest.json
favicon.ico
logo192.png
film/
12 requests
1.0 MB transferred
2.3 MB resources
Finish: 1.79 s
Headers
Preview
Response
Initiator
Timing
General
Request URL: https://swapi.dev/api/film/
Request Method: GET
Status Code: 404
Remote Address: 18.197.142.196:443
Referrer Policy: strict-origin-when-cross-origin
Response Headers
access-control-allow-origin: *
content-type: text/html
date: Wed, 29 Sep 2021 02:30:53 GMT
etag: "acd273391ae769ff3c165e6bda9a3149"
server: nginx/1.16.1
x-frame-options: SAMEORIGIN
Request Headers
:authority: swapi.dev

```
  const [movies, setMovies] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [error, setError] = useState(null);

  async function fetchMoviesHandler() {
    setIsLoading(true);
    setError(null);
    try {
      const response = await fetch('https://swapi.dev/api/film/');
      if (!response.ok) {
        throw new Error('Something went wrong!');
      }
      const data = await response.json();

      const transformedMovies = data.results.map((movieData) => {
        return {
          id: movieData.episode_id,
          title: movieData.title,
          openingText: movieData.opening_crawl,
          releaseDate: movieData.release_date,
        };
      });
      setMovies(transformedMovies);
    } catch (error) {
      setError(error.message);
    }
    setIsLoading(false);
  }

  return (
    <React.Fragment>
      <section>
        <button onClick={fetchMoviesHandler}>Fetch Movies</button>
      </section>
      <section>
        {!isLoading && movies.length > 0 && <MoviesList movies={movies} />}
        {!isLoading && !error && movies.length === 0 && (
          <p> Found no movies </p>
        )}
        {isLoading && <p>Loading...</p>}
        {!isLoading && error && <p>{error}</p>}
      </section>
    </React.Fragment>
  );
}
```


http://localhost:3000
real-world-vue
LaunchCode
Dev tools
Bayer dev tools
Buying
Fitness
INSPIRE
Vue learning
Bayer Sites
LEPSI
Bayer
Personal
Fetch Movies
Something went wrong!

```jsx
let content = <p>Found no movies.</p>;
if (movies.length > 0) {
  content = <MoviesList movies={movies} />;
}
if (error) {
  content = <p>{error}</p>;
}
if (isLoading) {
  content = <p>Loading...</p>;
}
return (
  <React.Fragment>
    <section>
      <button onClick={fetchMoviesHandler}>Fetch Movies</button>
    </section>
    <section>{content}</section>
  </React.Fragment>
);
```

useRequests

Fetch data when component is loading

```
  setIsLoading(true);
  setError(null);
  try {
    const response = await fetch('https://swapi.dev/api/films/');
    if (!response.ok) {
      throw new Error('Something went wrong!');
    }
    const data = await response.json();

    const transformedMovies = data.results.map((movieData) => {
      return {
        id: movieData.episode_id,
        title: movieData.title,
        openingText: movieData.opening_crawl,
        releaseDate: movieData.release_date,
      };
    });
    setMovies(transformedMovies);
  } catch (error) {
    setError(error.message);
  }
  setIsLoading(false);
}, []);

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

let content = <p>Found no movies.</p>;
if (movies.length > 0) {
  content = <MoviesList movies={movies} />;
}
if (error) {
  content = <p>{error}</p>;
}
if (isLoading) {
  content = <p>Loading...</p>;
}
return (
  <React.Fragment>
    <section>
      <button onClick={fetchMoviesHandler}>Fetch Movies</button>
    </section>
    <section>{content}</section>
  </React.Fragment>
);
```

Sending data

Firebase is a free google service product for backend, such as a database.

DB with URL

Change url in app to firebase URL to send data

BOOTSTRAP

- https://bdpastl.github.io/

- https://bdpastl.github.io/classes/BOOTSTRAP!.html

- https://bdpastl.github.io/classes/BuildBootstrap.html
- https://bdpastl.github.io/classes/advanced/Styling.html

Studio Review + Studio