UPDATING COMPONENT STATE

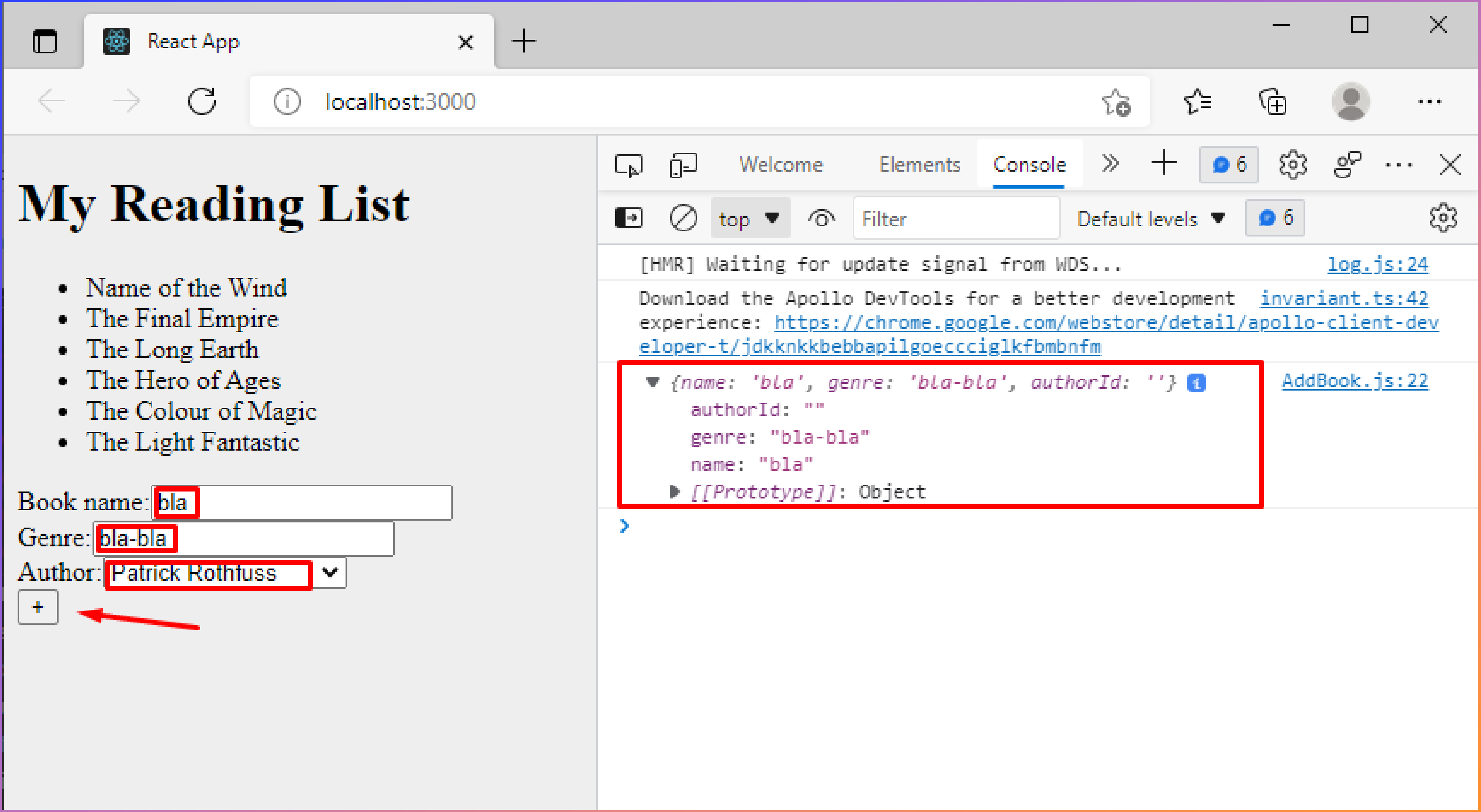
Hey gang, in this GraphQL tutorial we'll see how to keep track of the information a user enters into our form by pushing those values to our component state.

AddBook.js

```
import React from "react";
import { useQuery } from "@apollo/client";
import { getAuthorsQuery } from '../queries/queries';

function DisplayAuthors(){...}

function AddBook() {

    const [book, setBook] = React.useState({ name: "", genre: "", authorId: "" });
    const handleSubmit = e => {
        e.preventDefault();
        console.log(book);
    }

    return (
        <form id="add-book" onSubmit={handleSubmit}>
            <div className="field">
                <label>Book name:</label>
                <input value={book.name} onChange={e => setBook({ ...book, name: e.target.value })} type="text" />
            </div>
            <div className="field">
                <label>Genre:</label>
                <input value={book.genre} onChange={e => setBook({ ...book, genre: e.target.value })} type="text" />
            </div>
            <div className="field">
                <label>Author:</label>
                <select value={book.authorId} onChange={e => setBook({ ...book, authorId: e.target.value })} >
                    { DisplayAuthors() }
                </select>
            </div>
            <button>+</button>
        </form>
    )
}

export default AddBook;
```

Insert code

Modify code



My Reading List

- Name of the Wind
- The Final Empire
- The Long Earth
- The Hero of Ages
- The Colour of Magic
- The Light Fantastic

Book name: bla

Genre: bla-bla

Author: Patrick Rothfuss

+

```
[HMR] Waiting for update signal from WDS...                         log.js:24
Download the Apollo DevTools for a better development          invariant.ts:42
experience: https://chrome.google.com/webstore/detail/apollo-client-dev
eloper-t/jdkknkkbebbapilgoeccciglkfbmbnfm
▼ {name: 'bla', genre: 'bla-bla', authorId: ''}                   AddBook.js:22
    authorId: ""
    genre: "bla-bla"
    name: "bla"
  ▶ [[Prototype]]: Object
```