DMIT2008 Assignment 4a: React State, REST APIs and useEffect

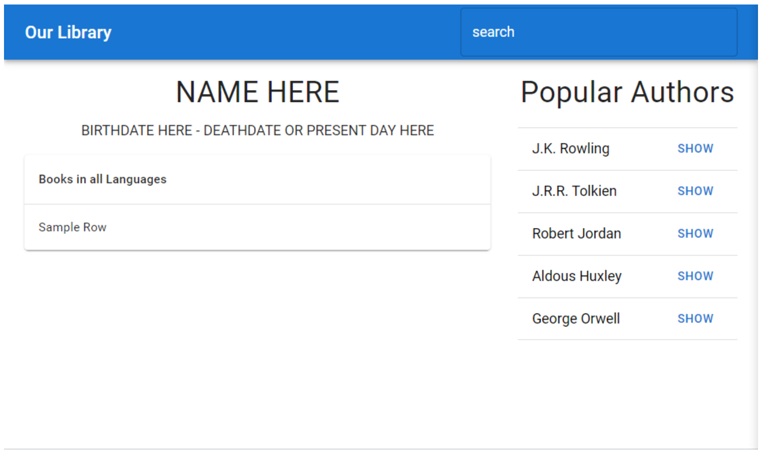
Introduction

This assignment will test everything you've learned so far up to and including the section "Components and Lifecycle" section of our course. Every frontend framework is going to have some sort of component lifecycle so this is a really important concept for us to know and understand. Secondly whether it's RESTful APIs that you're using to communicate to a server or GraphQL or something else, you'll be communicating with servers over the internet and if you're going to need to display this information to the user so you'll have to some sort of state.

This assignment will test all of your knowledge by using the Internet Archives "Open Library" Authors api that you can access here https://openlibrary.org/dev/docs/api/authors.

Overview of functionality

- The project starter looks like the image below once you install all of the dependencies and run the next.js project.



Our Library

search

NAME HERE

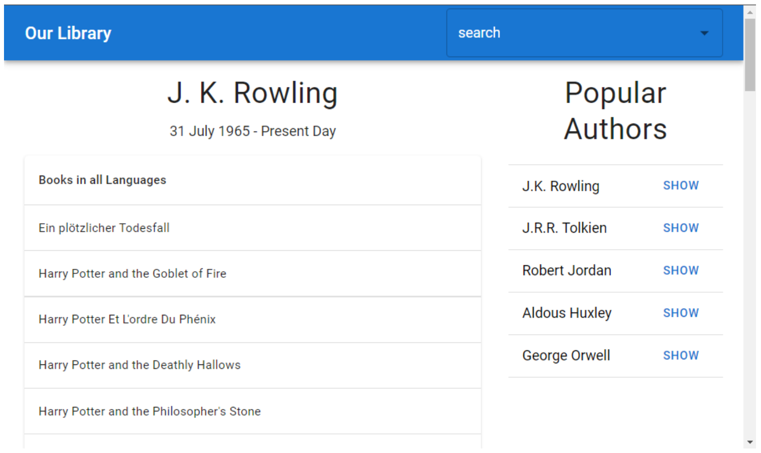
BIRTHDATE HERE - DEATHDATE OR PRESENT DAY HERE

Books in all Languages
Sample Row

Popular Authors

J.K. Rowling	SHOW
J.R.R. Tolkien	SHOW
Robert Jordan	SHOW
Aldous Huxley	SHOW
George Orwell	SHOW

- Once the project is complete here is the sample functionality that should happen **when the page is loaded.**



Our Library

search

J. K. Rowling

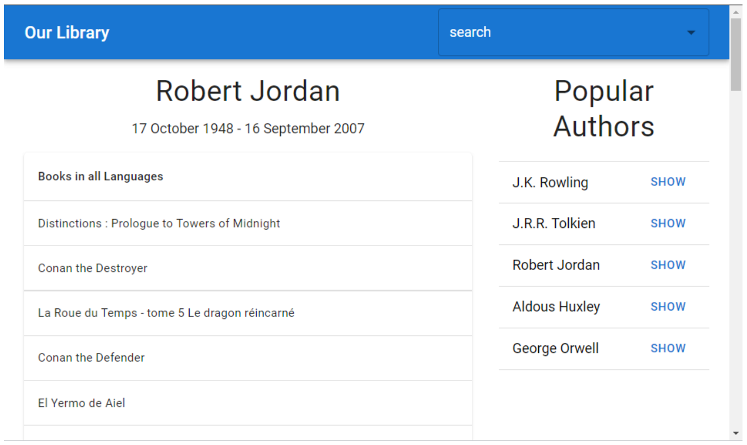
31 July 1965 - Present Day

Books in all Languages
Ein plötzlicher Todesfall
Harry Potter and the Goblet of Fire
Harry Potter Et Lordre Du Phénix
Harry Potter and the Deathly Hallows
Harry Potter and the Philosopher's Stone

Popular Authors

J.K. Rowling	SHOW
J.R.R. Tolkien	SHOW
Robert Jordan	SHOW
Aldous Huxley	SHOW
George Orwell	SHOW

- Once you click one of the favourite authors “SHOW” button (below we click “Robert Jordan”) the following should display.



Our Library

search

Robert Jordan

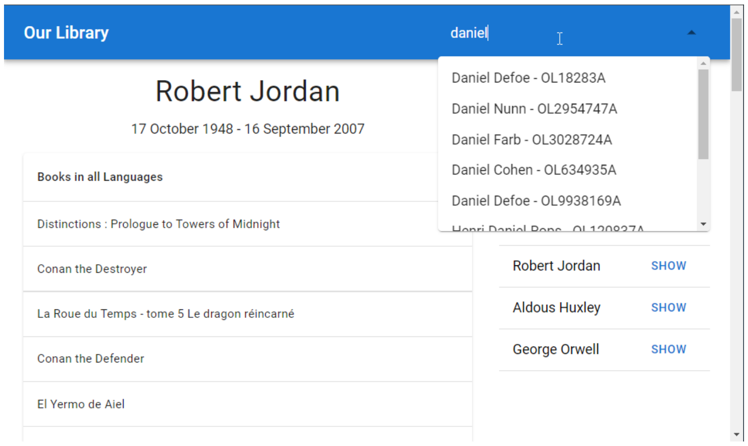
17 October 1948 - 16 September 2007

Books in all Languages
Distinctions : Prologue to Towers of Midnight
Conan the Destroyer
La Roue du Temps - tome 5 Le dragon réincarné
Conan the Defender
El Yermo de Aiel

Popular Authors

J.K. Rowling	SHOW
J.R.R. Tolkien	SHOW
Robert Jordan	SHOW
Aldous Huxley	SHOW
George Orwell	SHOW

- If you try to do the bonus you should be able to search any author, and the drop down should populate. For example if you type “Daniel” it should populate like so.



Our Library

daniel

Daniel Defoe - OL18283A
Daniel Nunn - OL2954747A
Daniel Farb - OL3028724A
Daniel Cohen - OL634935A
Daniel Defoe - OL9938169A

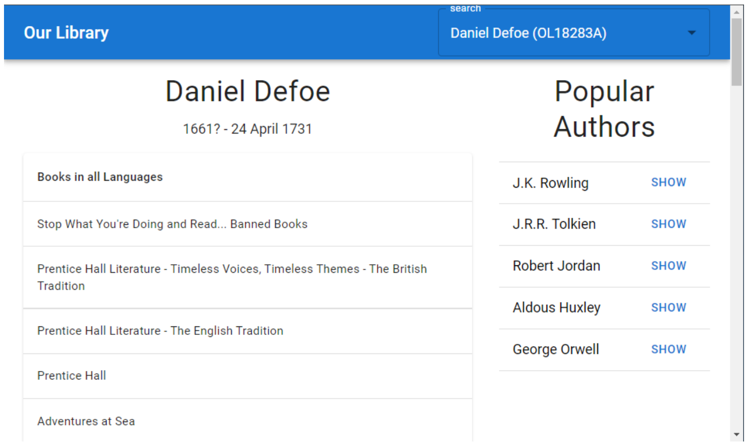
Robert Jordan

17 October 1948 - 16 September 2007

Books in all Languages
Distinctions : Prologue to Towers of Midnight
Conan the Destroyer
La Roue du Temps - tome 5 Le dragon réincarné
Conan the Defender
El Yermo de Aiel

Robert Jordan	SHOW
Aldous Huxley	SHOW
George Orwell	SHOW

- Part two of the bonus is if you click one of these values in the drop down say the first (“Daniel Defoe” in the last image) the author information should load on the page like so:



Our Library

search

Daniel Defoe (OL18283A)

Daniel Defoe

1661? - 24 April 1731

Books in all Languages
Stop What You're Doing and Read... Banned Books
Prentice Hall Literature - Timeless Voices, Timeless Themes - The British Tradition
Prentice Hall Literature - The English Tradition
Prentice Hall
Adventures at Sea

Popular Authors

J.K. Rowling	SHOW
J.R.R. Tolkien	SHOW
Robert Jordan	SHOW
Aldous Huxley	SHOW
George Orwell	SHOW

Required Tasks

- Install and run the starter project.
- Create a stateful variable named “authorKey” using useState and set the original value to “OL23919A” (Note: this is the author key to J.K Rowling).
- On the “Button” component of the “Popular authors list” change the “authorKey” stateful variable to the key of that specific author.
    - If you are confused about this please refer to the “utils/constants/popular_authors.js” file. You are expected to use a one line in the “onClick” for this.
- Observe the changes on the “authorKey” (think dependency array) using useEffect and load the “authorData” in its’ own stateful variable.
    - This means you’ll need to create a “authors.js” file in “utils/api” (and use base.js in that file for the BASE_URL)
    - You will use the endpoint specified in the **"Data on individual authors"** section at this url https://openlibrary.org/dev/docs/api/authors in your “getAuthor” function which will take **one parameter which is the author key.**
    - Use this function in your useEffect (that listens to the “authorKey”) to load the data and save it to a stateful variable say “authorData” (or try to choose a good name for it).
- Use the fetched **"Data on individual authors"** section https://openlibrary.org/dev/docs/api/authors (see previous point) data to display the following author information:
    - Information to display:
        - Author Name (where it says “NAME HERE”)
        - Author Birthdate
        - Author Death Date (if passed otherwise show present day)
    - Create a “AuthorInfo” component that will contain all of the JSX to display the other information. It should take on prop which should be the stateful variable of your author information. Use this component in your pages’ JSX.
        - Delete all unused MUI Components from index.js
- Observe the changes on the “authorKey” (think dependency array) using useEffect and load the “authorWorks” (or “authorBooks”) in its’ own stateful variable.
    - In the “utils/api/authors.js” create a new function that **will one parameter which is the author key**, this function will use the endpoint specified in the **“Works by an Author”** section https://openlibrary.org/dev/docs/api/authors in your “getBooks” function (or whatever you call it).
    - Use this “getBooks” function in a **second separate** that listens to the “authorKey” to save the books data into its’ own stateful variable say “booksData”.
- Use the fetched **“Works by an Author”** section https://openlibrary.org/dev/docs/api/authors (see previous point) data to display all of the books in a table.
    - Display the title of the book where “Sample Row” is displayed (you might have to refer to the Table MUI documentation).
    - Create a “BooksTable” component that will contain the entire table (including the TableContainer component). It should take on prop which should be the stateful variable of your book list. Use this component in your pages’ JSX.
        - Delete all unused MUI Components from index.js

- Create a component Named "PopularAuthorList" that will contain the entire popular author list, you'll probably need to pass the "setter" function for your "authorKey"
- **BONUS (don't ask your instructor about how to do this, it's supposed to be hard)**
  - Listen to changes of the "search" stateful variable using useEffect
  - Call the search api and populate the authors (use the "Searching for Authors" section in https://openlibrary.org/dev/docs/api/authors)
  - Set the author key by any random author that you've searched.

Marking key

Tasks	Grade	Marks	Total
Popular Author List and Author key setting • Author key stateful variable created. • Popular Author List Click successfully updates the author key. • PopularAuthorList Component created correctly with proper props, unused MUI components removed from index.js		**1** **3** **3**	
Author Info component, data, and useEffect • Author data stateful variable created. • "getAuthor" function uses correct endpoint, takes one "authorKey" parameter defined and exported properly. • getAuthor imported properly in the page. • useEffect used with correct dependency array. The "getAuthor" function used correctly sets the stateful value of authorData (or whatever its' called) • AuthorInfo Component created correctly with proper props, unused MUI components removed from index.js Note: Functionality should look like the sample functionality images.		**1** **5** **1** **5** **3**	
Books Info component, data and useEffect • Book data stateful variable created. • "getBooks" function uses correct endpoint, takes one "authorKey" parameter defined and exported properly. • getBooks imported properly in the page. • useEffect used with correct dependency array. The "getBooks" function used correctly sets the stateful value of booksData (or whatever its' called) • BooksTable Component created correctly with proper props, unused MUI components removed from index.js Note: Functionality should look like the sample functionality images.		**1** **5** **1** **5** **3**	
Author Searching Bonus • useEffect used with search as the dependency array • RESTful API function correctly defined • Correctly updates the authorKey and the author page is updated accordingly. Note: Functionality should look like the sample functionality images.		**[3]**	
Formatting and Style • Code Formatting and Style (run "npm run lint") and fix any errors that come up. • Node_modules folder and next folder are included (if you included them it's negative marks on your assignment)		**-3** **-3**	

Marking Rubric

Marks	5 Marks Criteria
5	**Task was completed with the highest of proficiency adhering to best practices and followed subject matter guidelines all tasks were completed to a professional standard.**
4	**Task was completed well some minor mistakes. Well above average work shows good understanding of the task and high degree of competence**
3	**Satisfactory work some features missing or incorrectly implemented. Show a moderate level of understanding in the task with room for improvement.**
2	**Below average work. Task was poorly complete. Show understanding of the task and the requirements to implement but implementation was poorly executed.**
1	**Some of the task was completed. Showed a lack of understanding in the subject matter and very poorly executed**
0	**Not completed.**

Marks	**3 Marks Criteria**
3	**Proficient shows a high degree of competence in completing task.**
2	**Capable above average degree of competence in completing task**
1	**Satisfactory shows a satisfactory degree of competence in completing task.**
0	**Shows a limited degree of competence in completing task.**

Marks	**1 Marks Criteria**
1	**Task Completed satisfactorily**
0	**Task was not executed.**