My Walkthrough steps for making React js App by JavaScript Matery: Made by Luke Yeo

**Note:**

* Videolink:<https://www.youtube.com/watch?v=b9eMGE7QtTk&list=PLNr6y7fJuf_f9wCIPQTun4pMosf5e4fFk&index=1>
* Note: This is to learn and have an experience for building a App using React JS
* THIS APP WILL BE LOCALLY RUN ON PC, NOT PUBLISH IN PUBLIC

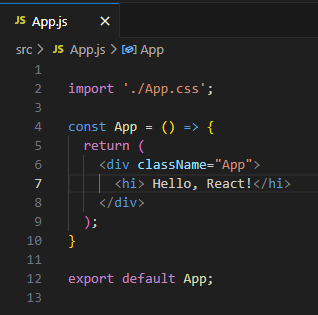
**Set up:**

1. Download node JS
2. Run admin for powershell and run: set-executionpolicy remotesigned
3. Open VScode , open a empty folder
4. Open a terminal and type in: npx create-react-app ./
5. To run the app, in the terminal type: npm start

**What each folder means:**

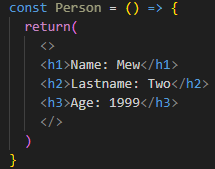
1. Package.json: dependencies and packages from Json and react
2. Node\_modules: the code for packages
3. Src: where all the logic for our apps goes.
4. Public: where the html file is
5. In src, there is a file call “index.js”, it is where the REACTDOM is, where the app will run

**In APP.JS, Learning Basics of Node.js/React:**

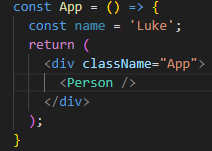
1. Delete all of useless stuff we don’t need inside the <div> and change function App() to const: 
2. This is a react fragment:



1. This is a Component:

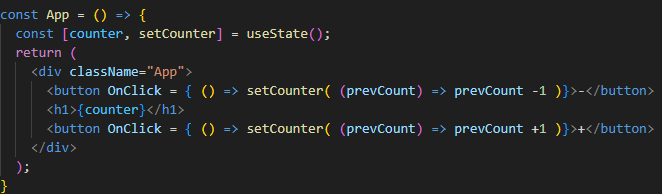


1. To call the Component, in which you can call multiple times:

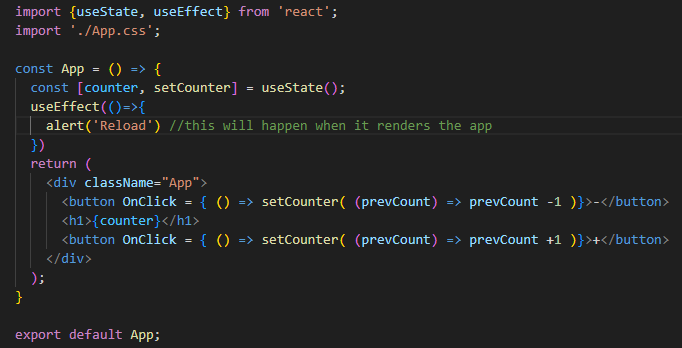


1. Using useState (call a hook) to make a counter, OnClick is an event:

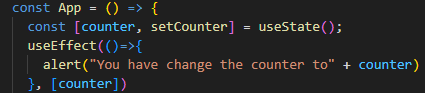




1. Using useEffect (another type of hook), e.g. after a change , we reload the app :



1. Dependency for useEffect, e.g. when we change the counter:



Note: Never change useState directly, like Counter = 100

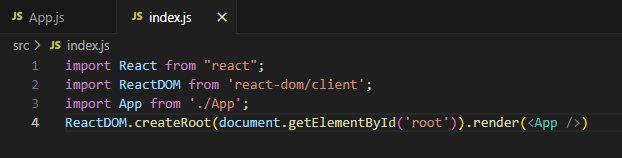
**Key learning points:**

* react jsx allows you to use JavaScript inside your “html” like a templating language. Expressions inside { } are evaluated as JavaScript code.
* We can conditionally render HTML components in jsx by placing them in curly braces to be evaluated as js
* Props passes to components through html attributes in jsx. The component catches the props object in its function definition (using functional components as class bases components are deprecated)
* Managing state variables with the usestate hook which returns a tuple, the variable that represents state item and a setter method. The setter method MUST be used to change the value of the state variable in order for react components to see the change
* You apply css files by importing them on top of your react component
* useEffect hook allows you to trigger a callback when a state variable changes (must be changed with the state setter given by useState). You can give useEffect array of dependencies (state variables), if empty array then useEffect triggered when the component loads only

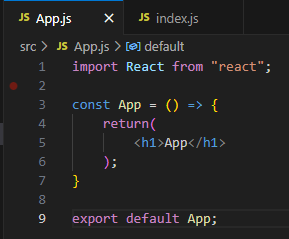
**Building of actual App from scratch (MOVIE WEBSITE, MOBILE FRIENDLY):**

1. Rename old src that we use above as to learn
2. Create a new folder call src and new files in src. Then import stuff that we will need in each of the files:

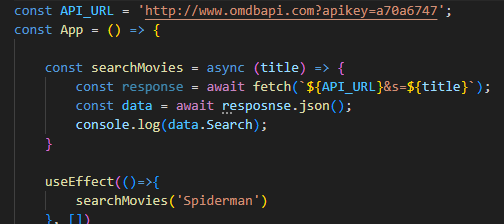
* Index.js :



* App.js:

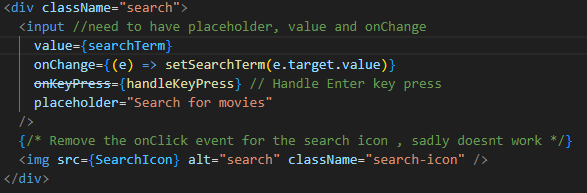


1. Go to <https://omdbapi.com/apikey.aspx> , press free and enter email, activate key and then use the key
2. Make the search function using await,fetch and setEffects.:



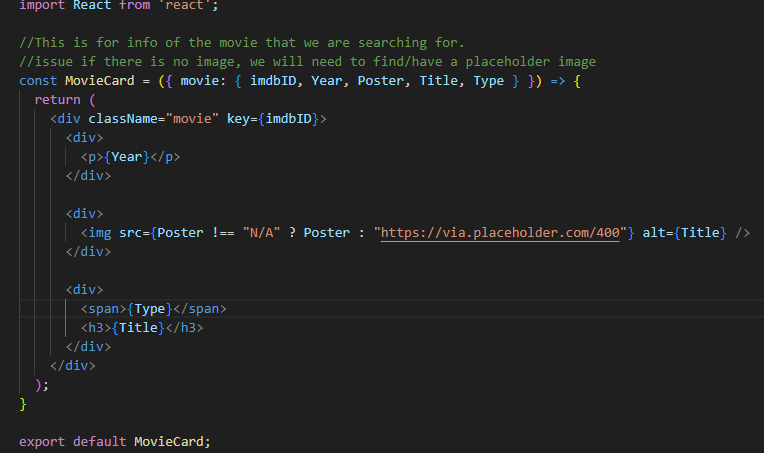
1. Note: THE APP.CSS FILE IS GIVEN BY JAVASCRIPT MASTERY, import it to App.js
2. Note: THE SEARCH.SVG FILE IS GIVEN BY JAVASCRIPT MASTERY, DOES NOT WORK, HENCE I HAVE CHANGE IT INTO PRESSING THE ENTER BUTTON INSTEAD.
3. The input for the search:

* Input in json needs Placeholder, value and onChange
* The (e) is for Enter on our keyboard

****

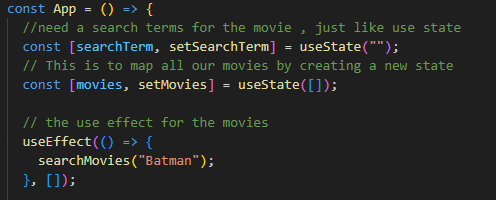
1. Check that it works using the console on the app itself, not in vscode (the f11)
2. Getting the information of the movie that we are searching for:

* Create a new file
* Remember to import react and export the file too
* Use dictionary to get what we want
* Find and Check the movie by its ID on OMDB
* Check if there is an image too

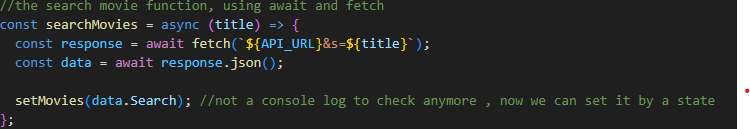


1. Dealing with the “Search” effect.

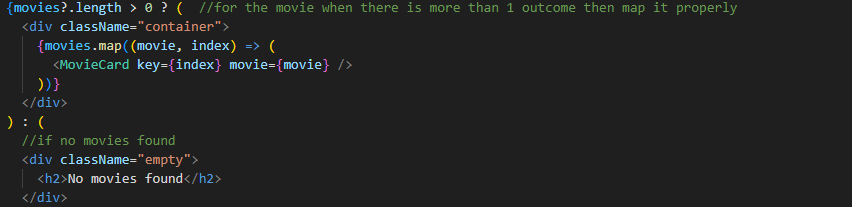
* Some issues: what if more than 1 movie
* What if there is no movie
* Previously, we were using the console log to check, now we need to display and on the App itself
  1. So we need to put a state, as a Map to all the movies that we may find



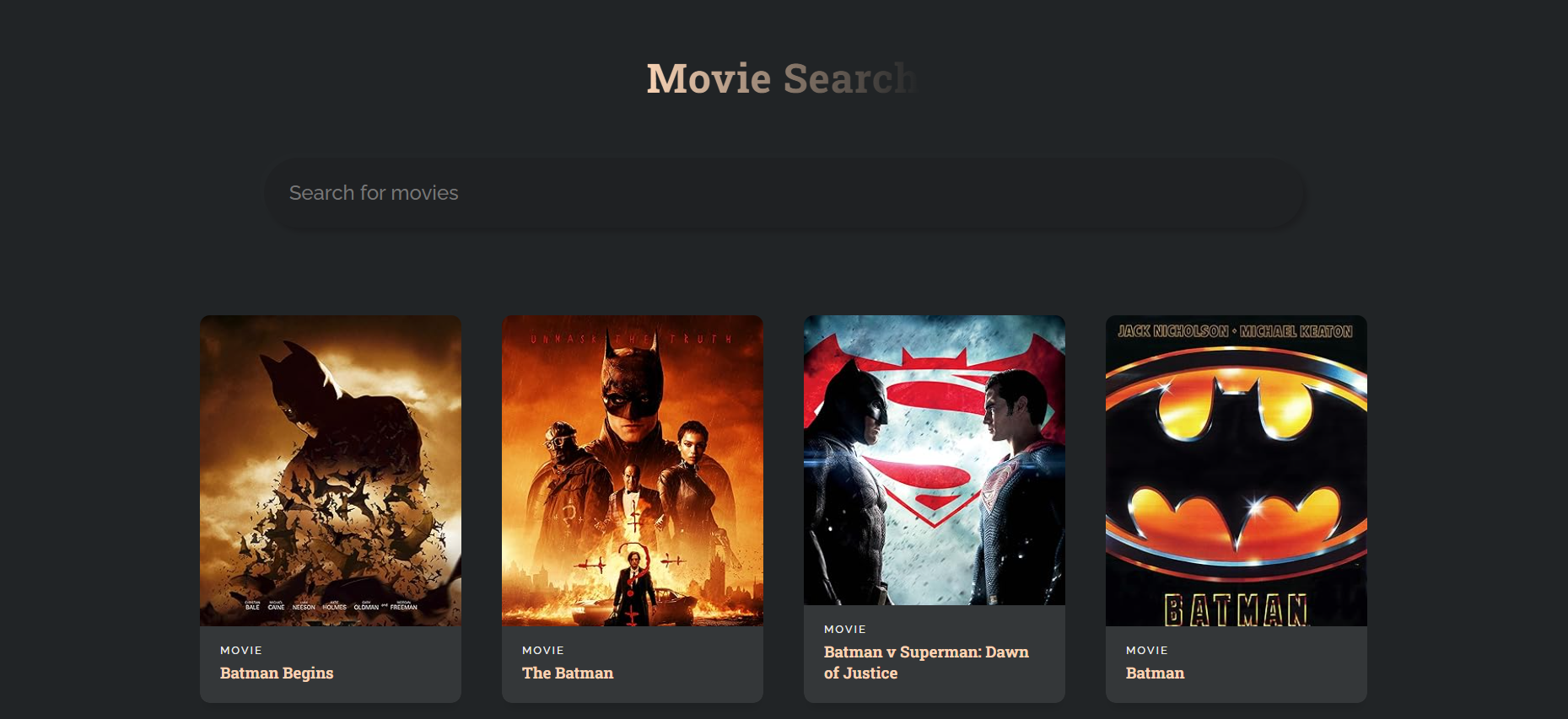
* 1. We change from console log to check to setMovies()

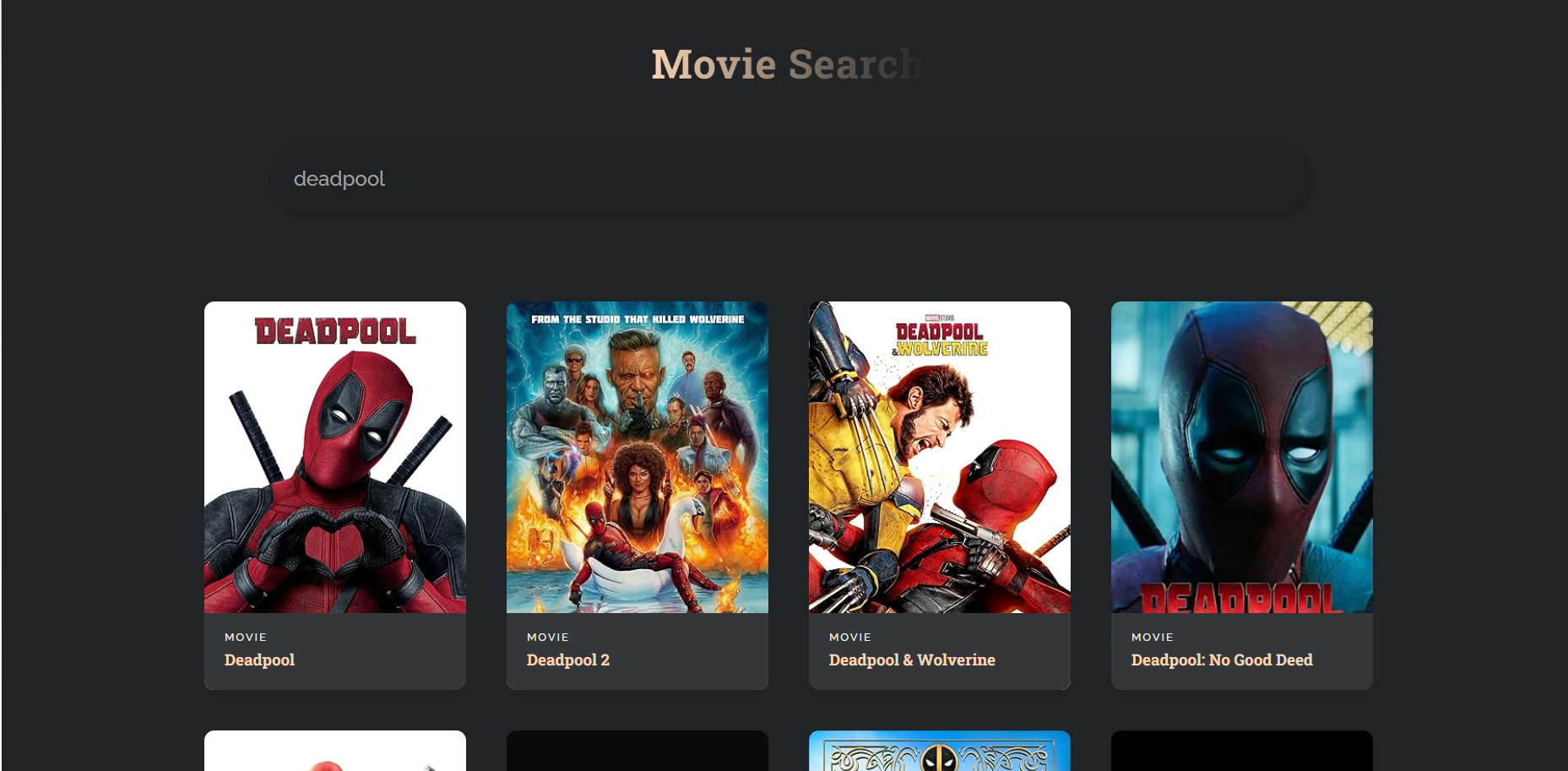


* 1. The Map display to show the movie



**Final product:**

****

****