

## 2 StephenGrider **AJAX IN REACT**

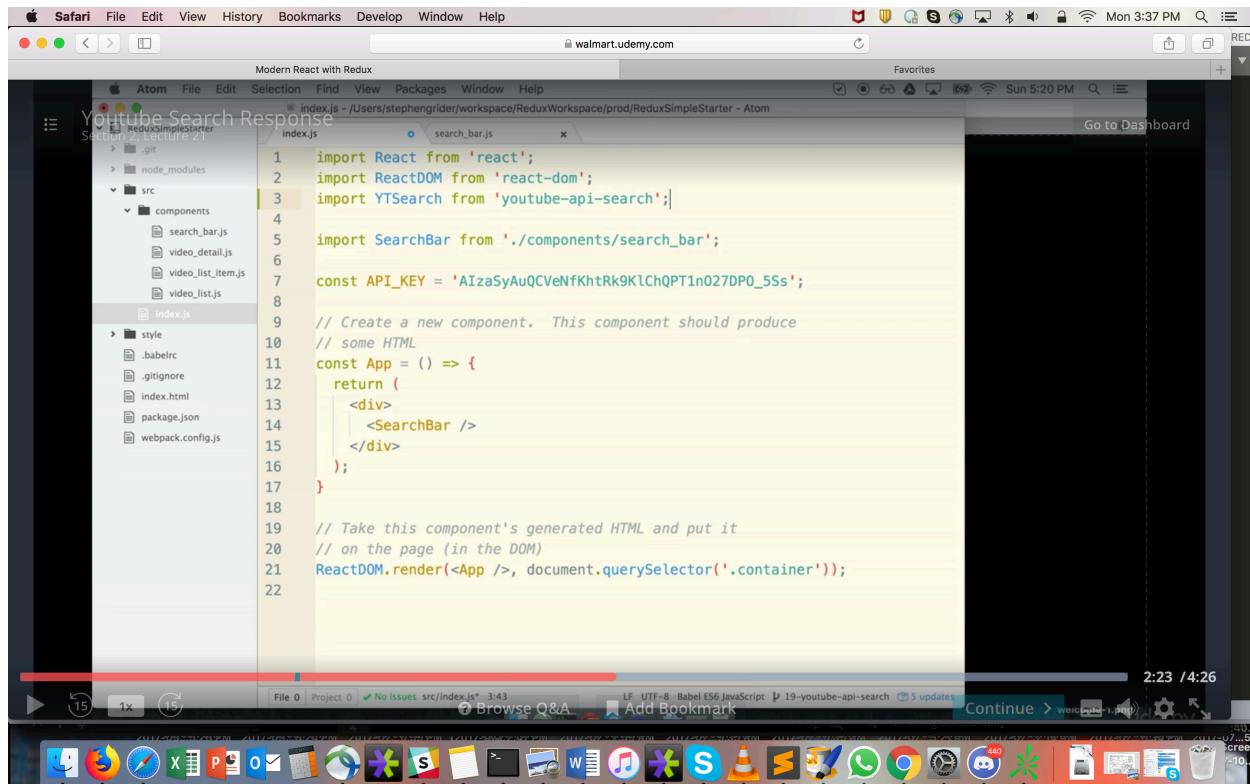
<https://github.com/StephenGrider>

<https://github.com/StephenGrider/ReduxCasts>

### **DOWNTWARDS DATA FLOW**

**only MOST parent component should fetch the data via API/FLUX/REDUX FRAMEWORDK**

import youtube-api-search from node\_modules



The screenshot shows a Mac desktop environment. At the top is a Safari browser window titled "YouTube Search Response" with the URL "walmart.udemy.com". Below it is an Atom code editor window titled "Modern React with Redux" with the file "index.js" open. The code in "index.js" is as follows:

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4
5 import SearchBar from './components/search_bar';
6
7 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_5Ss';
8
9 // Create a new component. This component should produce
10 // some HTML.
11 const App = () => {
12   return (
13     <div>
14       <SearchBar />
15     </div>
16   );
17 }
18
19 // Take this component's generated HTML and put it
20 // on the page (in the DOM)
21 ReactDOM.render(<App />, document.querySelector('.container'));
```

The Atom interface also shows a sidebar with project files like ".git", "node\_modules", "src" (containing "components", "search\_bar.js", "video\_detail.js", "video\_list\_item.js", "video\_list.js"), "style", ".babelrc", ".gitignore", "index.html", "package.json", and "webpack.config.js". The status bar at the bottom of the screen shows "File 0 Project 0 No Issues src/index.js\* 3:43 LF UTF-8 Babel ES6 JavaScript 19-youtube-api-search 5 updates". The dock at the bottom contains icons for various Mac applications.

YTSearch function called with with arguments

1 { objet of api key and search keyword }

2 a callback function

YTSearch ( { } , **function(data) {}** );

The screenshot shows the Atom code editor interface. The left sidebar displays a project structure for 'ReduxSimpleStarter' with files like .git, node\_modules, src (containing components, style, .babelrc, .gitignore, index.html, package.json, and webpack.config.js), and index.js. The main editor pane shows the content of index.js:

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4
5 import SearchBar from './components/search_bar';
6
7 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_5Ss';
8
9 YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
10   console.log(data);
11 });
12
13 // Create a new component. This component should produce
14 // some HTML
15 const App = () => {
16   return (
17     <div>
18       <SearchBar />
19     </div>
20   );
21 }
22
23 // Take this component's generated HTML and put it
24 // on the page (in the DOM)
25 ReactDOM.render(<App />, document.querySelector('.container'));
```

The status bar at the bottom indicates the file is 'index.js' at line 34, with 5 updates available.

The screenshot shows a Chrome browser window with the address bar set to 'localhost:8080'. The developer tools are open, specifically the Elements tab, which is displaying the structure of a complex JSON object. The object appears to be a search response from YouTube, containing details like the video title ('5 Types of Surfboards | Surfboard Basics'), channel information ('UCSpVHeDGr9UbREhRca0qwsA'), and snippet details. The object is deeply nested with multiple levels of objects and arrays. The status bar at the bottom shows the URL 'localhost:8080' and the time '4:03 / 4:26'.

## Now convert function based component App to Class based component

1. first we need to import {Component}
2. add class (small c) keyword
3. remove = () => part of function component then extends (not extend) that Component (capital C)
4. add render() {} method return the jsx.

The screenshot shows a Mac desktop environment. At the top, a Safari browser window is open to [walmart.udemy.com](https://walmart.udemy.com). The page displays a search interface with a search bar containing 'surfboards' and the text 'searching' below it. Below the browser is an Atom code editor window titled 'Refactoring Functional Components to Class Com...'. The editor shows two files: 'index.js' and 'search\_bar.js'. The 'index.js' file contains the following code:

```
1 import React, { Component } from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhTrk9KlChQPT1n027DP0_5Ss';
6
7 YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
8   console.log(data);
9 });
10
11 class App extends Component {
12   render() {
13     return (
14       <div>
15         <SearchBar />
16       </div>
17     );
18   }
19 }
20
21 ReactDOM.render(<App />, document.querySelector('.container'));
```

The 'search\_bar.js' file is also visible in the editor. The status bar at the bottom of the screen shows the file path as 'index.js ~ /Users/stephennginder/kspace/dukuWorkspace/prod/ReduxSimpleStarter - Atom', the file name as 'index.js', and the line number as '16:13'. The system tray at the bottom includes icons for Mail, Finder, and other system applications.

error instead of extends we are using extend  
error can be seen on console of npm start process

A screenshot of a Mac desktop showing a terminal window and a file browser. The terminal window, titled 'npm (node)', displays the output of an 'npm start' command. The output shows the hash 'd437a155a1da4cdfeeeb', the version 'webpack 1.12.9', and the time '25ms'. It then lists the assets generated by webpack, including 'bundle.js' and a chunk for 'index.js'. Following this, an error message is shown: 'Module build failed: SyntaxError: /Users/stephengrider/workspace/ReduxWorkspace/prod/ReduxSimpleStarter/src/index.js: Unexpected token (11:10)'. The file browser on the left shows the directory structure of 'ReduxSimpleStarter', including 'src' and 'index.js' files.

A screenshot of a Mac desktop showing a browser window with developer tools open. The browser window is titled 'localhost:8080' and shows a blank page. The developer tools console tab is active, displaying an error message: 'Uncaught Error: Cannot find module "./src/index.js"'. The browser's toolbar at the bottom includes icons for various applications like Mail, Finder, and Safari.

extends (not extend)

The screenshot shows a Mac desktop with a Safari browser window open to a Udemy course page. The browser's status bar indicates it's Mon 4:15 PM. The Atom code editor window is in the foreground, showing the file `index.js` with the following content:

```
1 import React, { Component } from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPT1n027DPO_5Ss';
6
7 YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
8   console.log(data);
9 });
10
11 class App extends Component {
12   render() {
13     return (
14       <div>
15         | <SearchBar />
16       </div>
17     );
18   }
19 }
20
21 ReactDOM.render(<App />, document.querySelector('.container'));
22
```

whenever used search any word we have to set the **changed YTSearch data to state**

1. That means we have to add constructor function
2. called `super(props)`
3. and initialize the state with empty object

A screenshot of a Mac desktop environment. At the top is the OS X menu bar with options like Apple, Safari, File, Edit, View, History, Bookmarks, Develop, Window, Help. Below it is the Dock with various application icons. The main window is an Atom editor showing code for a React application. The file being edited is `index.js` from a project named "Refactoring Functional Components to Class Components". The code defines a class-based component `App` that includes a search bar and a video list. A sidebar on the right shows a file tree and a log output from the terminal.

```
1 import React, { Component } from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPT1n027DP0_5Ss';
6
7 YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
8   console.log(data);
9 });
10
11 class App extends Component {
12   constructor(props) {
13     super(props);
14
15     this.state = { };
16   }
17
18   render() {
19     return (
20       <div>
21         | <SearchBar />
22       </div>
23     );
24   }
25 }
```

we will create videos as state property value will be empty array

A screenshot of a Mac desktop environment, similar to the one above but with a different window configuration. The Atom editor is still open in the background, showing the same `index.js` file. In the foreground, a new window is open in the Safari browser, displaying the Udemy website. The title bar of the browser window says "Modern React with Redux". The browser interface includes a back/forward navigation bar, a search bar, and a status bar at the bottom.

```
1 import React, { Component } from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPT1n027DP0_5Ss';
6
7 YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
8   console.log(data);
9 });
10
11 class App extends Component {
12   constructor(props) {
13     super(props);
14
15     this.state = { videos: [] };
16   }
17
18   render() {
19     return (
20       <div>
21         | <SearchBar />
22       </div>
23     );
24   }
25 }
```

now we will move search function to constructor  
so user can see immediately result when page is loaded with default value of term if  
YTSearch

The screenshot shows a Mac desktop environment. In the top left, a Safari browser window is open to 'walmart.udemy.com'. In the center, an Atom code editor window displays 'index.js' from a 'ReduxSimpleStarter' project. The code imports 'YTSearch' and 'SearchBar' components, sets up a class 'App' with a constructor and render method, and includes a call to 'ReactDOM.render'. On the right, a terminal window titled 'Stephen' shows the command 'bundle.js:85'. The bottom of the screen features a dock with various application icons.

```
index.js
1 import YTSearch from 'youtube-api-search';
2 import SearchBar from './components/search_bar';
3 const API_KEY = 'AIzaSyAuQCVeNfKhRk9K1ChQPT1n027DPO_55s';
4
5 class App extends Component {
6   constructor(props) {
7     super(props);
8
9     this.state = { videos: [] };
10
11    YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
12      console.log(data);
13    });
14
15  }
16
17  render() {
18    return (
19      <div>
20        |   <SearchBar />
21      </div>
22    );
23  }
24
25}
26
27 ReactDOM.render(<App />, document.querySelector('.container'));
28
```

we will use this.setState({ videos: data});

A screenshot of a Mac desktop. At the top is a Safari browser window titled "Modern React with Redux" showing a file named "index.js". Below it is an Atom code editor window also titled "index.js". The code in the editor is:

```
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_5Ss';
6
7 class App extends Component {
8   constructor(props) {
9     super(props);
10
11     this.state = { videos: [] };
12
13     YTSearch({key: API_KEY, term: 'surfboards'}, function(data) {
14       this.setState({ videos: data });
15     });
16   }
17
18   render() {
19     return (
20       <div>
21         | <SearchBar />
22       </div>
23     );
24   }
25
26
27 ReactDOM.render(<App />, document.querySelector('.container'));
28
```

The system tray at the bottom shows various application icons.

## Remove function keyword and user arrow function

A screenshot of a Mac desktop. At the top is a Safari browser window titled "Modern React with Redux" showing a file named "index.js". Below it is an Atom code editor window also titled "index.js". The code in the editor has been modified:

```
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_5Ss';
6
7 class App extends Component {
8   constructor(props) {
9     super(props);
10
11     this.state = { videos: [] };
12
13     YTSearch({key: API_KEY, term: 'surfboards'}, (data) => {
14       this.setState({ videos: data });
15     });
16   }
17
18   render() {
19     return (
20       <div>
21         | <SearchBar />
22       </div>
23     );
24   }
25
26
27 ReactDOM.render(<App />, document.querySelector('.container'));
28
```

The system tray at the bottom shows various application icons.

rename date to vedio

```
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPT1n027DP0_5Ss';
6
7 class App extends Component {
8   constructor(props) {
9     super(props);
10
11     this.state = { videos: [] };
12
13     YTSearch({key: API_KEY, term: 'surfboards'}, (videos) => {
14       this.setState({ videos: videos });
15     });
16   }
17
18   render() {
19     return (
20       <div>
21         | <SearchBar />
22       </div>
23     );
24   }
25 }
```

**When we have object like this when a key and value is identical than we can use es6 syntax than we can just use key {key}**

The screenshot shows a Mac desktop environment. At the top, a Safari browser window is open to [walmart.udemy.com](https://walmart.udemy.com), displaying a dashboard with a message about unread emails. Below the browser is an Atom code editor window titled "Modern React with Redux". The editor shows the file "index.js" with code for a React application. The code imports ReactDOM, YTSearch, and SearchBar components, and defines a class-based component "App" that makes a call to YTSearch with the API key and term "surfboards", then renders a 

containing a  component. To the right of the code editor is a terminal window with the title "bundle.js:85" and some log output. The desktop dock at the bottom contains icons for various applications like Finder, Mail, and the App Store.

```
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_55s';
6
7 class App extends Component {
8   constructor(props) {
9     super(props);
10
11     this.state = { videos: [] };
12
13     YTSearch({key: API_KEY, term: 'surfboards'}, (videos) => {
14       this.setState({ videos });
15       // this.setState([ videos: videos])
16     });
17   }
18
19   render() {
20     return (
21       <div>
22         | <SearchBar />
23       </div>
24     );
25   }
26 }
27
```

Now create Video List

**It will be functional Component**

**This component does not need any state**

**it does not record any user action**

**it does not re render in any fashion**

```
import react
```

A screenshot of a Mac desktop. At the top is a Safari browser window showing the Udemy website. Below it is an Atom code editor window with three tabs: index.js, video\_list.js, and search\_bar.js. The video\_list.js tab contains the following code:

```
import React from 'react';
```

The code editor's status bar shows "File 0 | Project 0 | No Issues | src/components/video\_list.js | 3:1". To the right of the code editor is a video player interface with a play button, volume controls, and a progress bar showing "0:50 / 7:42". Below the video player is a toolbar with various application icons. On the far left, there's a vertical sidebar with project files and a file viewer.

## create video list function Component

A screenshot of a Mac desktop, similar to the one above, showing the same video player and code editor setup. The video player is at "0:57 / 7:42". The code editor's status bar now shows "File 0 | Project 0 | No Issues | src/components/video\_list.js | 4:3". The video\_list.js tab now contains the following code:

```
import React from 'react';  
const VideoList = () => {  
}
```

The code editor's status bar also includes "LF | UTF-8 | Babel ES6 JavaScript | 22-props | 5 updates". The rest of the desktop environment, including the sidebar and toolbar, remains the same.

return <ul></ul>  
because it will be list of videos

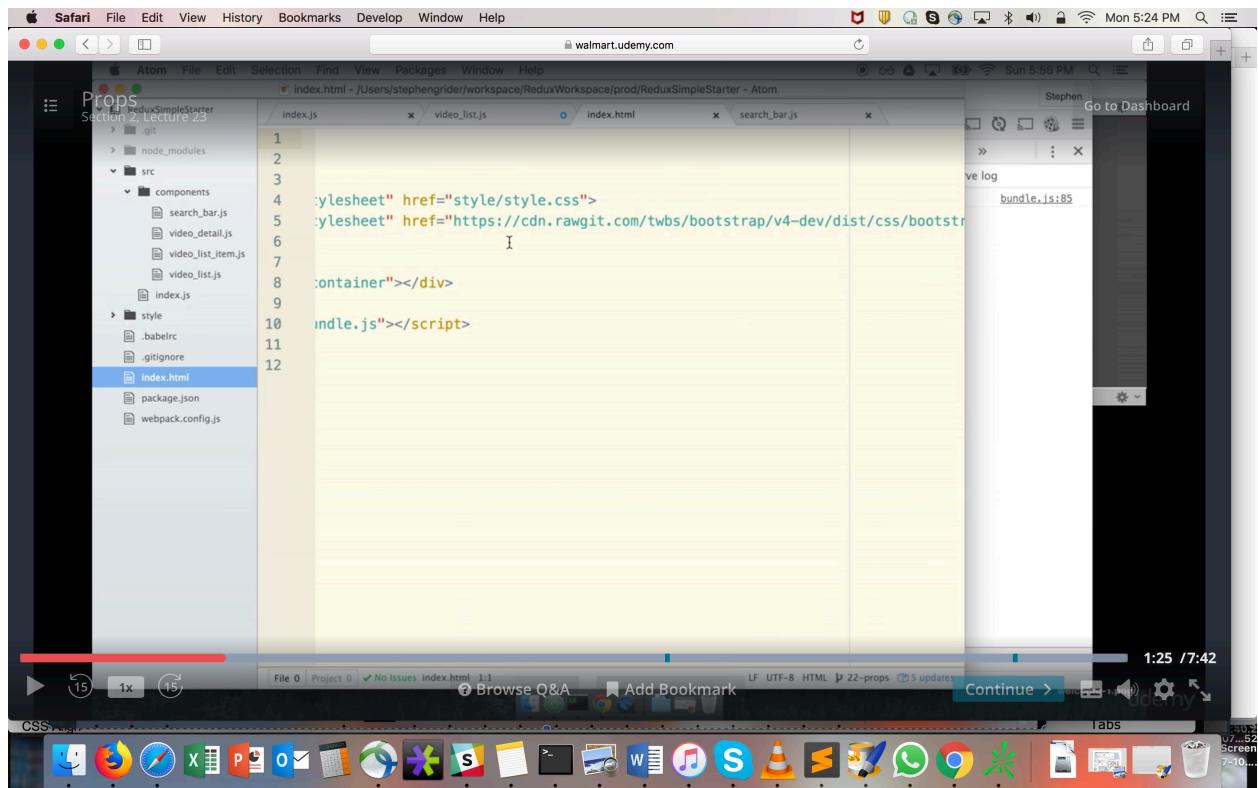
The screenshot shows a Mac desktop environment. In the foreground, an Atom code editor window is open, displaying a file named 'video\_list.js'. The code contains a single function definition:

```
1 import React from 'react';
2
3 const VideoList = () => {
4   return (
5     <ul>
6       </ul>
7   );
8 }
```

The Atom interface includes a sidebar with project files like 'index.js', 'search\_bar.js', and 'video\_list\_item.js'. The status bar at the bottom shows file statistics: 'File 0 | Project 0 | No issues | src/components/video\_list.js\* 6-9 | ? Browse Q&A | Add Bookmark | LF | UTF-8 | Babel ES6 JavaScript | 22-props | 5-updates'.

In the background, a Safari browser window is visible, showing a video player interface with a progress bar at 1:06 / 7:42. The video title is 'Go to Dashboard'.

v4 version of bootstrap css already included in index.html



we use **className** for **css class** in es6  
because **Class** keyword is already reserved for **Class based components**.

The screenshot shows the Atom IDE interface. The left sidebar displays a project structure for 'Props Section 2, Lecture 23' with files like index.js, search\_bar.js, and video\_list.js. The main editor window contains the following code:

```
1 import React from 'react';
2
3 const VideoList = () => {
4   return (
5     <ul className="list-group">
6       </ul>
7     );
8 }
```

The code is being typed into the editor. The browser tab at the top is labeled 'walmart.udemy.com'. The bottom status bar shows the file path 'src/components/video\_list.js' and the line number 'S-20'.

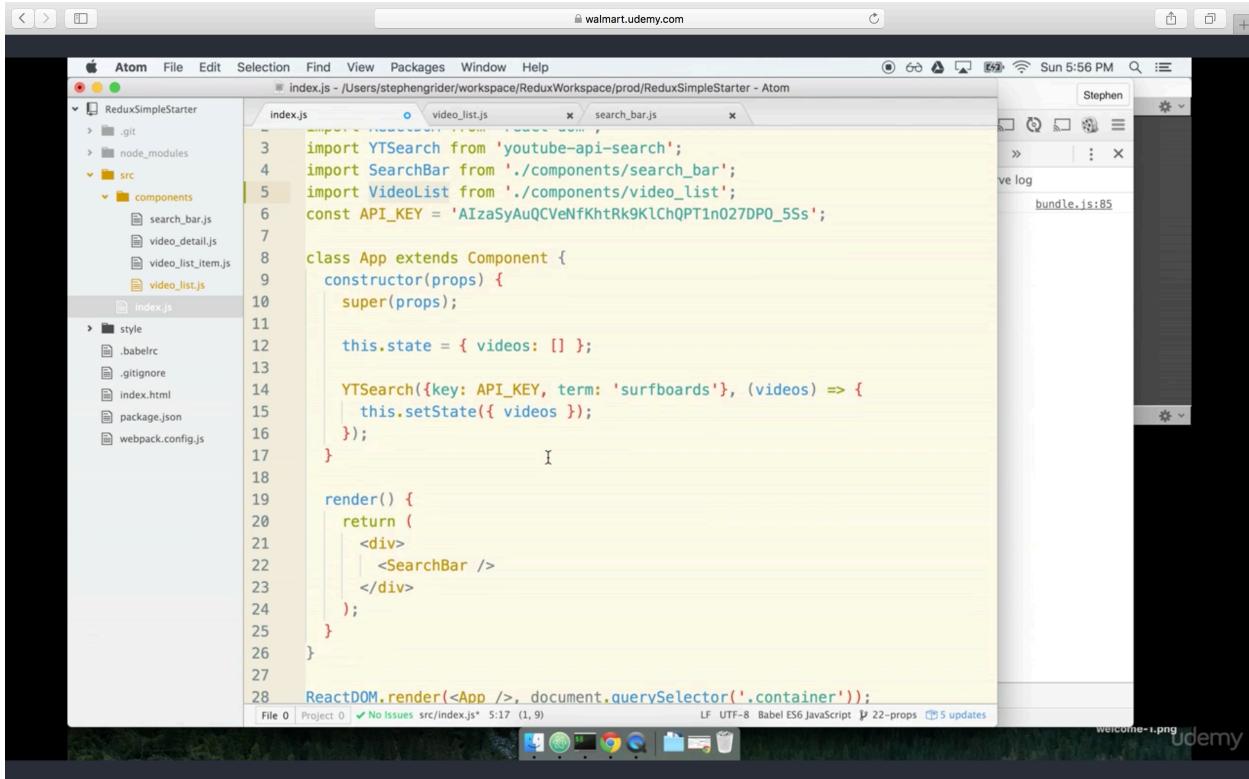
we will set className col-md-4 bootstrap css class

The screenshot shows the Atom IDE interface with the same project structure. The main editor window now contains the following code:

```
1 import React from 'react';
2
3 const VideoList = () => {
4   return (
5     <ul className="col-md-4 list-group">
6       </ul>
7     );
8 }
9
10
11 export default VideoList;
```

The 'className' attribute in the ul tag has been updated to 'col-md-4 list-group'. The browser tab at the top is labeled 'walmart.udemy.com'. The bottom status bar shows the file path 'src/components/video\_list.js' and the line number '10-25'.

import video\_list to index.js



The screenshot shows the Atom code editor interface. The file tree on the left shows a project structure for 'ReduxSimpleStarter' with files like .git, node\_modules, src (containing components, style, .babelrc, .gitignore, index.html, package.json, and webpack.config.js), and index.js. The main editor window displays the 'index.js' file with the following code:

```
3 import YTSearch from 'youtube-api-search';
4 import SearchBar from './components/search_bar';
5 import VideoList from './components/video_list';
6 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPT1n027DP0_5Ss';
7
8 class App extends Component {
9   constructor(props) {
10     super(props);
11
12     this.state = { videos: [] };
13
14     YTSearch({key: API_KEY, term: 'surfboards'}, (videos) => {
15       this.setState({ videos });
16     });
17   }
18
19   render() {
20     return (
21       <div>
22         <SearchBar />
23       </div>
24     );
25   }
26 }
27
28 ReactDOM.render(<App />, document.querySelector('.container'));
```

The status bar at the bottom indicates 'File 0 Project 0 No Issues src/index.js 5:17 (1, 9)' and 'LF UTF-8 Babel ES6 JavaScript 22-props 5 updates'. A browser window in the background shows 'walmart.udemy.com'.

add it to app render method

The screenshot shows the Atom IDE interface with the following details:

- File Explorer:** On the left, it shows a project structure for "ReduxSimpleStarter" with files like index.js, video\_list.js, search\_bar.js, and components.
- Code Editor:** The main editor area contains the following code for `index.js`:

```
const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPT1n027DP0_5Ss';

class App extends Component {
  constructor(props) {
    super(props);

    this.state = { videos: [] };

    YTSearch({key: API_KEY, term: 'surfboards'}, (videos) => {
      this.setState({ videos });
    });
  }

  render() {
    return (
      <div>
        <SearchBar />
        <VideoList />
      </div>
    );
  }
}

ReactDOM.render(<App />, document.querySelector('.container'));
```

- Terminal:** At the bottom, the terminal shows "File 0 | Project 0 | No Issues src/index.js 23:22".
- Right Panel:** A sidebar titled "Stephen" displays a "bundle.js:85" log entry.

passing video data to VideoList

1 we will create a **property videos in jsx** .  
passing data like this refer to passing props

2 we will set `{this.state.videos}` to `videos` props  
since we will be referencing a **javascript variable** we will user curly braces to pass data from state.

```
Props
Section 2, Lecture 23
Atom File Edit Selection Find View Packages Window Help
index.js video_list.js search_bar.js
index.js
8 class App extends Component {
9   constructor(props) {
10     super(props);
11
12     this.state = { videos: [] };
13
14     YTSearch({key: API_KEY, term: 'surfboards'}, (videos) => {
15       this.setState({ videos });
16     });
17
18   }
19
20   render() {
21     return (
22       <div>
23         <SearchBar />
24         <VideoList videos={this.state.videos} />
25       </div>
26     );
27   }
28
29   ReactDOM.render(<App />, document.querySelector('.container'));
30 }
```

In functional component those **props** is argument

In **class components** props will be available using **this.props** any where in class

```
import React from 'react';
const VideoList = (props) => {
  return (
    <ul className="col-md-4 list-group">
    </ul>
  );
};

export default VideoList;
```

```
import React from 'react';
const SearchBar = (props) => {
  return (
    <input type="text" placeholder="Search..." />
  );
};

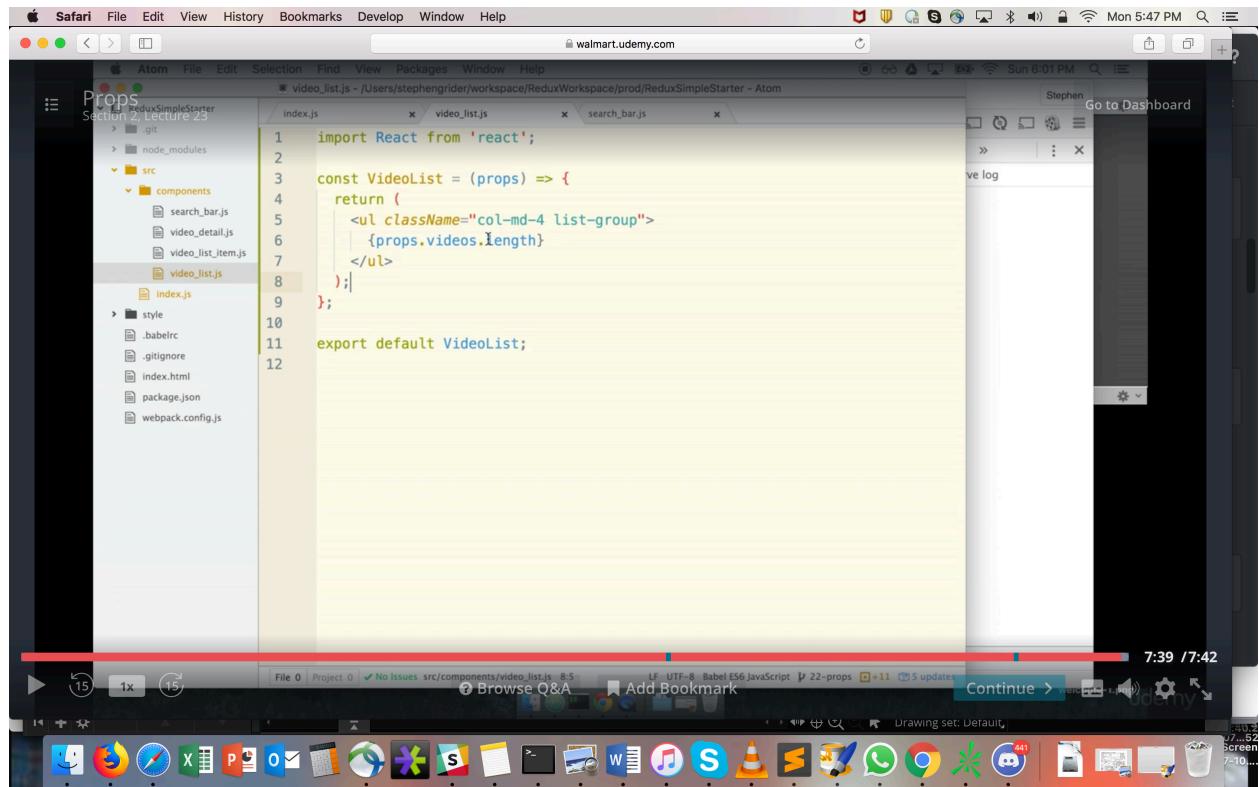
export default SearchBar;
```

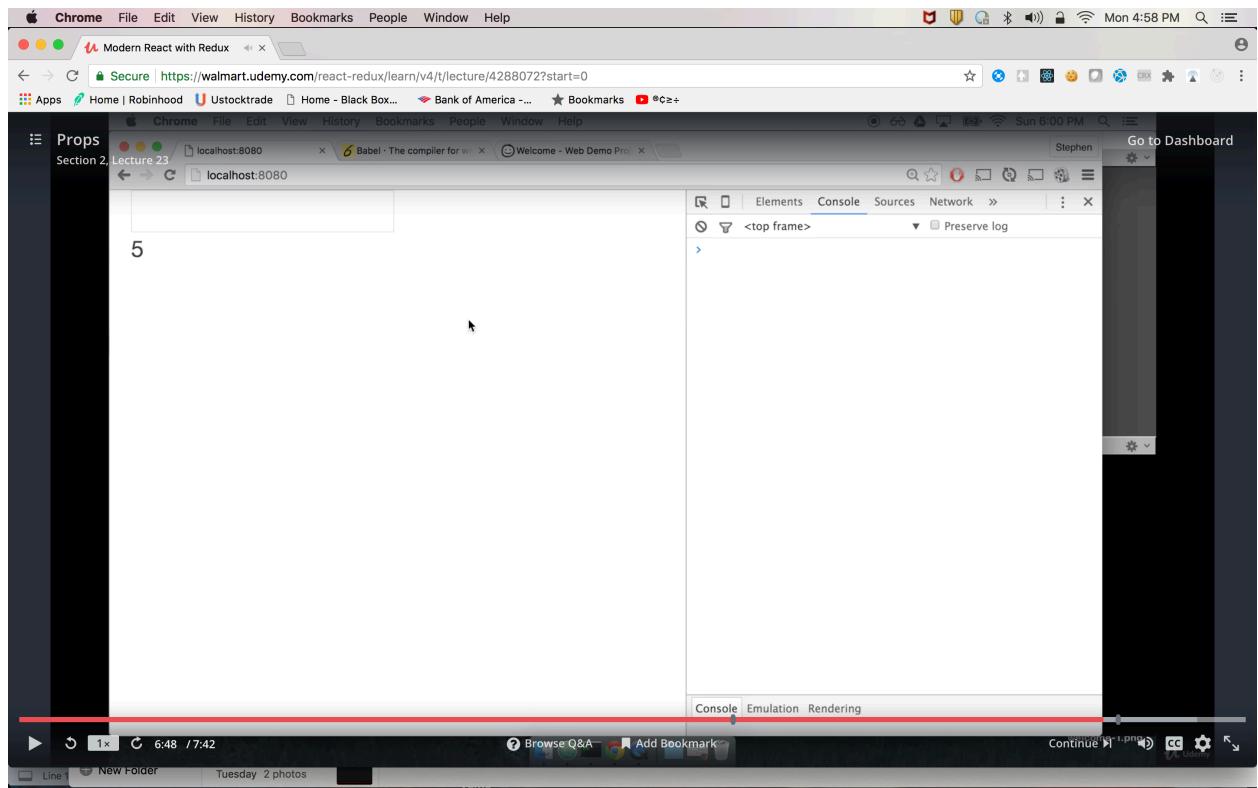
```
import React from 'react';
const VideoList = (props) => {
  const videos = props.videos;
  return (
    <ul className="col-md-4 list-group">
    </ul>
  );
};

export default VideoList;
```

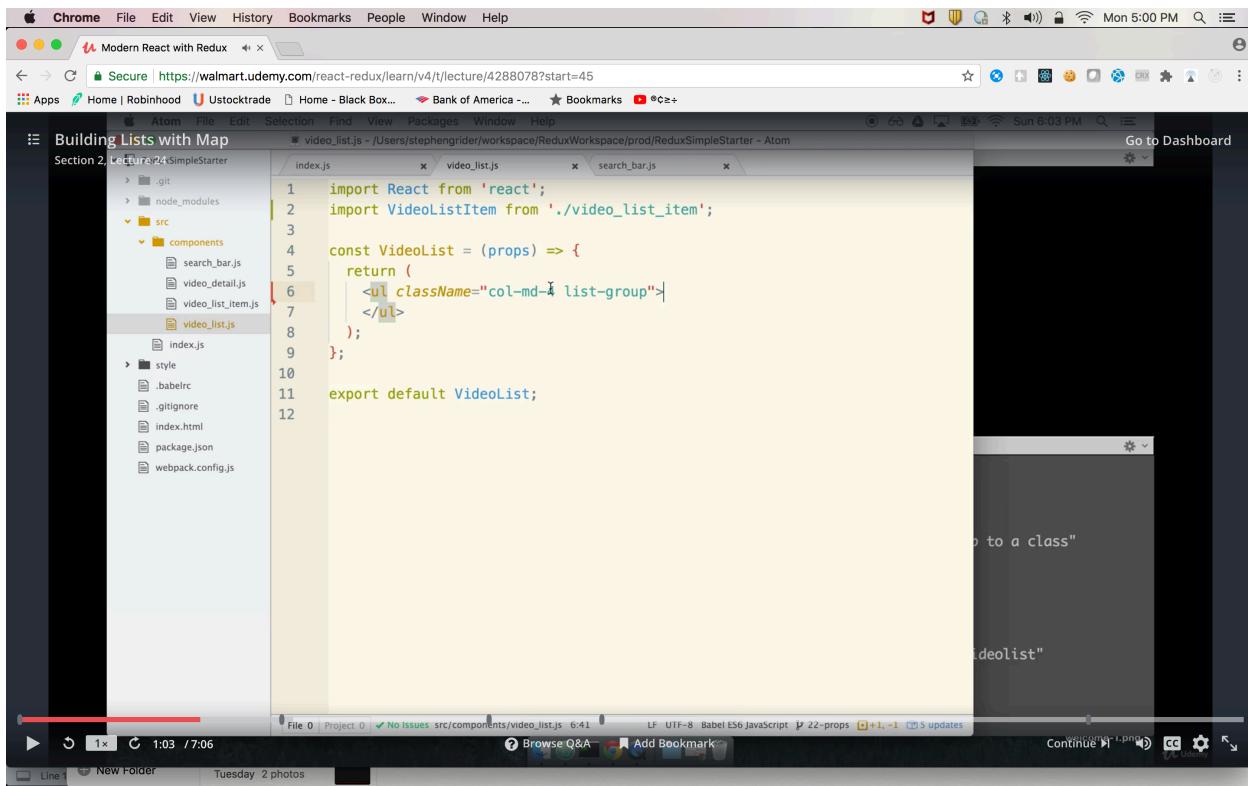
```
import React from 'react';
const SearchBar = (props) => {
  return (
    <input type="text" placeholder="Search..." />
  );
};

export default SearchBar;
```





remove length check  
now import VideoListIem



=====

**array.filter filter is a function (which is property of a array )**  
**array.filter calls a callback function which return a new array filled with all array elements that PASS A TEST ( return (age > 18) )**

```
var ages = [32,33,16,40]
ages.filter( (age) => age > 18)
```

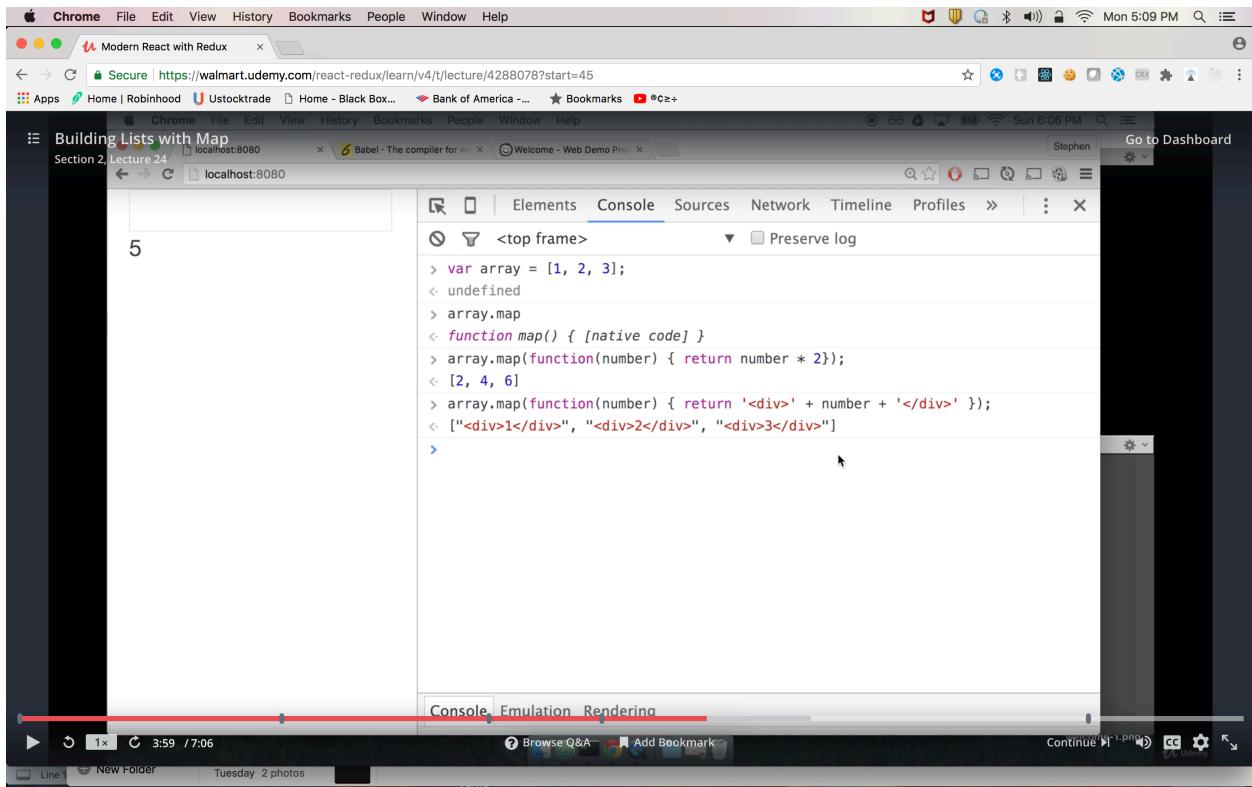
=====

=====

to iterate through list of videos we will use map (we should avoid for loop as much as possible)

**array.map map is a function (which is property of a array )**  
**array.map calls a callback function which return a new array where EACH INDEX IS RETURN value of function ( return age\* 2 ).**

**array.map is really fantastic when building a list of jsx in react.**



=====

=====

**array.forEach is a function (which is property of a array )**

The forEach() method calls a **callback function once for each element in an array, in order.** **does not return any thing**

**Note:** forEach() does not execute the function for array elements without values.

### **harmlessRansomNote Algorithm**

**made a notes out of magazine  
we will use object as hashtable**

```
function harmlessRansomNote(noteText,magazineText) {  
// split by space char will give array of words (which we are doing here)  
// split by empty string will give array of character (next is Palindrome we will use  
there)
```

```
var noteArr = noteText.split(' ');
var magazineTextArr = magazineText.split(' ');
var magazineObj = {};

// FUNCTION WILL PRINT HOW MANY WORDS ARE THERE IN //SENTENCE
magazineTextArr.forEach( (word) => {
    if(!magazineObj[word]) {
        magazineObj[word] = 0
    }
    magazineObj[word]++;
} );
}
```

```
harmlessRansomNote (' ','this is all the magazine text in the magazine');
{this:1
is:1
all:1
the:2
magazine:2
text:1
in:1}
```

```
=====
=====
```

```
now lets create video_list_item
just return a <li> Video </li>
```

to iterate through list of videos we will use map (we should avoid for loop as much as possible)

```
1 import React from 'react';
2
3 const VideoListItem = (props) => {
4   return <li>Video</li>;
5 }
6
7 export default VideoListItem;
```

at top props.videos.map

The screenshot shows a Mac OS X desktop environment. A Chrome browser window is open to a Udemy course page titled "Modern React with Redux". The URL is <https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288078?start=45>. Below the browser, an Atom code editor is running. The file tree on the left shows a project structure with files like index.js, video\_list.js, and video\_list\_item.js. The main editor area contains the following code for the VideoList component:

```
1 import React from 'react';
2 import VideoListItem from './video_list_item';
3
4 const VideoList = (props) => {
5   props.videos.map(())
6
7   return (
8     <ul className="col-md-4 list-group">
9       </ul>
10    );
11  );
12
13  export default VideoList;
14
```

for each element of videos  
we have function that will be called with single video

```
import React from 'react';
import VideoListItem from './video_list_item';

const VideoList = (props) => {
  props.videos.map((video) => {
    return (
      <ul className="list-group">
        <li>{video.title}</li>
      </ul>
    );
  });
}

export default VideoList;
```

we will define a arrow function

we will return `VideoListItem`

and we will pass a video as property name.

```
import React from 'react';
import VideoListItem from './video_list_item';

const VideoList = (props) => {
  props.videos.map((video) => {
    return <VideoListItem video={video} />
  });

  return (
    <ul className="col-md-4 list-group">
      </ul>
  );
};

export default VideoList;
```

we will need to save the reference of array that get returned  
const videoItems = result of map of function

```
import React from 'react';
import VideoListItem from './video_list_item';

const VideoList = (props) => {
  const videoItems = props.videos.map((video) => {
    return <VideoListItem video={video} />
  });

  return (
    <ul className="col-md-4 list-group">
      {videoItems}
    </ul>
  );
};

export default VideoList;
```

now we will videoltems add as jsx

since we are referencing a js variable videoltems we will use curly braces {videoltems}

```
import React from 'react';
import VideoListItem from './video_list_item';

const VideoList = (props) => {
  const videoItems = props.videos.map((video) => {
    return <VideoListItem video={video} />
  });

  return (
    <ul className="col-md-4 list-group">
      {videoItems}
    </ul>
  );
};

export default VideoList;
```

```
Warning: Each child in an array or iterator should have a unique "key" prop. Check the render method of 'VideoList'. See https://fb.me/react-warning-keys for more information.
```

we need add a key as unique property to resolve error

Atom File Edit View Packages Window Help

Secure | https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288082?start=45

Mon 6:04 PM Sun 6:13 PM

List Item Keys

Section 2, Lecture 24: List Item Keys

index.js video\_list.js video\_list\_item.js search\_bar.js

```
1 import React from 'react';
2 import VideoListItem from './video_list_item';
3
4 const VideoList = (props) => {
5   const videoItems = props.videos.map((video) => {
6     return <VideoListItem key={video.etag}>{video}</VideoListItem>;
7   });
8
9   return (
10     <ul className="col-md-4 list-group">
11       {videoItems}
12     </ul>
13   );
14 };
15
16 export default VideoList;
17
```

File 0 Project 0 No Issues src/components/video\_list.js b:43 L 111-8 Babel ES6 JavaScript P 24-list-item-keys □ +1, -1 □ 5 updates Continue □ t.png CC Settings ↗

New Folder

