PROJECT

Video Player

You just learned your first *programming pattern*. Let’s put it to use!

For this project, you’ll make three React components work together to create a responsive video player. Let’s get started!

If you get stuck during this project or would like to see an experienced developer work through it, click **“Get Unstuck“** to see a **project walkthrough video**.

Tasks

10/10 Complete

Mark the tasks as complete by checking them off

1.

Click Save, and take a look at your video player in the browser. It looks pretty good! But if you try interacting with it, you’ll find that there’s zero functionality.

Take a look at the `App` component class. This class has one property stored as `state`: a `src` containing the address of a video file. `App`‘s job is to pass this `src` down to a stateless component, and to pass *the ability to change the* `src` down to a different stateless component.

Passing `src` is the easier part, so let’s do that first.

Inside of `App`‘s *render* function, give `<Video />` an *attribute*. Make this attribute’s *name* `src`, and the attribute’s *value* equal to the `src` property stored in `this.state`.

Hint

In **App.js**:

```
<Video src={this.state.src} />
```

2.

Let’s make `<Video />` play its passed-in video file!

Select **Video.js**. In `Video`‘s render function, give `<video />` a `src` attribute. Make `src` equal to the passed-in video file.

Hint
In **Video.js**:

```
<video controls autostart autoPlay muted
src={this.props.src}/>
```

3.
Alright, the video player works! Now let’s make the *menu* work as well.

You’ve made `App` pass the `src` down to `<Video />`. Now `App` needs to pass the ability to *change* the `src` down to `<Menu />`. If you want to pass the ability to *change* a piece of `state`, then first you need to define a function that calls `this.setState`.

In **App.js**, give `App` a new property named `chooseVideo`. Set `chooseVideo`‘s value equal to a *function* with one parameter, named `newVideo`.

`chooseVideo` is going to get passed a string: either `'fast'`, `'slow'`, `'cute'`, or `'eek'`. It will use this string to choose a new src, which it will use to update `this.state.src`.

In the body of `chooseVideo`, call `this.setState`. Set `this.state.src` equal to `VIDEOS[newVideo]`.
Hint
In **App.js**:

```
chooseVideo(newVideo) {
  this.setState({
    src: VIDEOS[newVideo]
  });
}

render () {
```

4.
If you pass `chooseVideo` to `<Menu />`, then you will give `<Menu />` the ability to *update* `<App />`‘s state.

In `App`‘s render function , give `Menu` a `chooseVideo` attribute. Set `chooseVideo`‘s *value* equal to the `chooseVideo` function.
Hint
In **App.js**:

```
<Menu chooseVideo={this.chooseVideo} />
```

5.

Currently, if you pass `.chooseVideo()` to `Menu` the value of `this` will be incorrect when called. In the `constructor` of `App`, `bind` `.chooseVideo()` to the current value of `this` and store it in `this.chooseVideo`.

Hint

In **App.js**:

```
constructor(props) {
  super(props);
  this.chooseVideo = this.chooseVideo.bind(this);
}
```

6.

Alright, now you just have to attach this passed-in function to an event listener!

Select **Menu.js**. In `Menu`‘s render function, give `<form></form>` an `onClick` attribute.

Set `onClick`‘s *value* equal to the passed-in `chooseVideo` function.

Hint

In **Menu.js**:

```
<form onClick={this.props.chooseVideo}>
```

7.

Try selecting a video in the browser.

It doesn’t work! Do you know why not?

`chooseVideo` expects a *string* as an argument. But event handlers are automatically passed *event objects,* not strings.

You need to *wrap* `chooseVideo` in a new function that can take an event object as an argument.

Give `Menu` a new property, before the render function, named `handleClick`. Set `handleClick` equal to a function with one parameter named `e`.

Inside of the body of `handleClick`, declare a new variable named `text`. Set `text` equal to `e.target.value`. This will equal the text of a clicked radio button.

After declaring this `text` variable, create a new line. On your new line, call the passed-in `chooseVideo` function. Pass in `text` as an argument.
Hint
In **Menu.js**:

```
handleClick(e) {
  var text = e.target.value;
  this.props.chooseVideo(text);
}

render() {
```

8.
Currently, the value of `this` will be incorrect when you call `.handleClick()`.

Create a `constructor` for `Menu`, and in its body, call `super(props)`. Then, `bind` `.handleClick()` to the current value of `this` and store it in `this.handleClick`.
Hint
In **Menu.js**:

```
constructor(props) {
  super(props);
  this.handleClick = this.handleClick.bind(this);
}
```

9.
Only one more step! You need to use your new wrapper function as an *event handler*.

In `Menu`‘s render function’s `return` statement, replace `{this.props.chooseVideo}` with `{this.handleClick}`.
Hint
In **Menu.js**:

```
<form onClick={this.handleClick}>
```

10.
Great job!

`App` passes down `this.state.src` to `Video`. `Video` uses this info to display the chosen video.

`App` also passes down *the ability to change* `this.state.src` to `Menu`. `Menu` uses this ability to let a user to select a new video.

You’ve put together a responsive video player, and done it in a way that you will often find in the real world!

App.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Video } from './Video';
import { Menu } from './Menu';

const VIDEOS = {
  fast: 'https://content.codecademy.com/courses/React/react_video-fast.mp4',
  slow: 'https://content.codecademy.com/courses/React/react_video-slow.mp4',
  cute: 'https://content.codecademy.com/courses/React/react_video-cute.mp4',
  eek: 'https://content.codecademy.com/courses/React/react_video-eek.mp4'
};

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

    this.state = { src: VIDEOS.fast };
    this.chooseVideo = this.chooseVideo.bind(this);
  }

  chooseVideo(newVideo) {
    this.setState({
      src: VIDEOS[newVideo]
    });
  }


  render() {
    return (
      <div>
        <h1>Video Player</h1>
        <Menu
          chooseVideo={this.chooseVideo}
        />
        <Video
```

```
          src={this.state.src}
        />
      </div>
    );
  }
}

ReactDOM.render(
  <App />,
  document.getElementById('app')
);
```

Video.js

```
import React from 'react';

export class Video extends React.Component {
  render() {
    return (
      <div>
        <video
          controls autostart autoPlay muted
          src={this.props.src}
        />
      </div>
    );
  }
}
```

Menu.js

```
import React from 'react';

export class Menu extends React.Component {
  constructor(props) {
    super(props);
    this.handleClick = this.handleClick.bind(this);
  }

  handleClick(e) {
    const text = e.target.value;
    this.props.chooseVideo(text);
  }

  render() {
    return (
      <form onClick={this.handleClick}>
        <input type="radio" name="src" value="fast" /> fast
        <input type="radio" name="src" value="slow" /> slow
        <input type="radio" name="src" value="cute" /> cute
        <input type="radio" name="src" value="eek" /> eek
      </form>
    );
  }
}
```