

## **</> 008-Videos-hooks**

# Contents

## 📁 008-Videos-hooks

📄 package.json

### 📁 public

📄 index.html

📄 manifest.json

### 📁 src

📄 index.js

### 📁 apis

📄 youtube.js

### 📁 components

📄 App.js

📄 SearchBar.js

📄 VideoDetail.js

📄 VideoItem.css

📄 VideoItem.js

📄 VideoList.js

### 📁 hooks

📄 useVideos.js

# package.json

/package.json

```
{
  "name": "videos",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^4.2.4",
    "@testing-library/react": "^9.5.0",
    "@testing-library/user-event": "^7.2.1",
    "axios": "^0.18.0",
    "react": "^16.13.1",
    "react-dom": "^16.13.1",
    "react-scripts": "3.4.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

# index.html

/public/index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
  <meta name="theme-color" content="#000000">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/semantic-
ui/2.4.1/semantic.min.css">
  <!--
    manifest.json provides metadata used when your web app is added to the
    homescreen on Android. See https://developers.google.com/web/
fundamentals/web-app-manifest/
  -->
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
  <!--
    Notice the use of %PUBLIC_URL% in the tags above.
    It will be replaced with the URL of the `public` folder during the build.
    Only files inside the `public` folder can be referenced from the HTML.

    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
    work correctly both with client-side routing and a non-root public URL.
    Learn how to configure a non-root public URL by running `npm run build`.
  -->
  <title>React App</title>
</head>

<body>
  <noscript>
    You need to enable JavaScript to run this app.
  </noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
</body>

</html>
```

## manifest.json

/public/manifest.json

```
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

## index.js

/src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './components/App';

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

# youtube.js

/src/apis/youtube.js

```
import axios from 'axios';

const KEY = 'AIzaSyDVfeUxsJzp8Vs0xnFGlhC2zrllcqbC4zc';

export default axios.create({
  baseURL: 'https://www.googleapis.com/youtube/v3',
  params: {
    part: 'snippet',
    maxResults: 5,
    key: KEY,
  },
});
```

# App.js

/src/components/App.js

```
import React, { useState, useEffect } from 'react';
import SearchBar from '../SearchBar';
import VideoList from '../VideoList';
import VideoDetail from '../VideoDetail';
import useVideos from '../hooks/useVideos';

const App = () => {
  const [selectedVideo, setSelectedVideo] = useState(null);
  const [videos, search] = useVideos('buildings');

  useEffect(() => {
    setSelectedVideo(videos[0]);
  }, [videos]);

  return (
    <div className="ui container">
      <SearchBar onFormSubmit={search} />
      <div className="ui grid">
        <div className="ui row">
          <div className="eleven wide column">
            <VideoDetail video={selectedVideo} />
          </div>
          <div className="five wide column">
            <VideoList onVideoSelect={setSelectedVideo} videos={videos} />
          </div>
        </div>
      </div>
    </div>
  );
};

export default App;
```



# SearchBar.js

/src/components/SearchBar.js

```
import React, { useState } from 'react';

const SearchBar = ({ onFormSubmit }) => {
  const [term, setTerm] = useState('');

  const onSubmit = (event) => {
    event.preventDefault();

    onFormSubmit(term);
  };

  return (
    <div className="search-bar ui segment">
      <form onSubmit={onSubmit} className="ui form">
        <div className="field">
          <label>Video Search</label>
          <input
            type="text"
            value={term}
            onChange={(event) => setTerm(event.target.value)}
          />
        </div>
      </form>
    </div>
  );
};

export default SearchBar;
```

## VideoDetail.js

/src/components/VideoDetail.js

```
import React from 'react';

const VideoDetail = ({ video }) => {
  if (!video) {
    return <div>Loading...</div>;
  }

  const videoSrc = `https://www.youtube.com/embed/${video.id.videoId}`;

  return (
    <div>
      <div className="ui embed">
        <iframe title="video player" src={videoSrc} />
      </div>
      <div className="ui segment">
        <h4 className="ui header">{video.snippet.title}</h4>
        <p>{video.snippet.description}</p>
      </div>
    </div>
  );
};

export default VideoDetail;
```

## Videoltem.css

/src/components/Videoltem.css

```
.video-item {  
  display: flex !important;  
  align-items: center !important;  
  cursor: pointer;  
}  
  
.video-item.item img {  
  max-width: 180px;  
}
```

## VideoItem.js

/src/components/VideoItem.js

```
import './VideoItem.css';
import React from 'react';

const VideoItem = ({ video, onVideoSelect }) => {
  return (
    <div onClick={() => onVideoSelect(video)} className="video-item item">
      <img
        alt={video.snippet.title}
        className="ui image"
        src={video.snippet.thumbnails.medium.url}
      />
      <div className="content">
        <div className="header">{video.snippet.title}</div>
      </div>
    </div>
  );
};

export default VideoItem;
```

## VideoList.js

/src/components/VideoList.js

```
import React from 'react';
import VideoItem from '../VideoItem';

const VideoList = ({ videos, onVideoSelect }) => {
  const renderedList = videos.map(video => {
    return (
      <VideoItem
        key={video.id.videoId}
        onVideoSelect={onVideoSelect}
        video={video}
      />
    );
  });

  return <div className="ui relaxed divided list">{renderedList}</div>;
};

export default VideoList;
```

## useVideos.js

/src/hooks/useVideos.js

```
import { useState, useEffect } from 'react';
import youtube from '../apis/youtube';

const useVideos = (defaultSearchTerm) => {
  const [videos, setVideos] = useState([]);

  useEffect(() => {
    search(defaultSearchTerm);
  }, [defaultSearchTerm]);

  const search = async (term) => {
    const response = await youtube.get('/search', {
      params: {
        q: term,
      },
    });

    setVideos(response.data.items);
  };

  return [videos, search];
};

export default useVideos;
```