Source Code

# Section 7, Lecture 53

The downloadable project is in the resources folder.

The source code itself, if you would rather copy and paste it, is down below:

/AddPhoto

1. import React, { Component } from 'react';
2. import { Link } from 'react-router-dom'
3. import serializeForm from 'form-serialize'
4. class Single extends Component {
5. handleSubmit = (e) =**>** {
6. e.preventDefault()
7. const values = serializeForm(e.target, { hash: true })
8. if (values["imageLink"] && values["description"]) {
9. values.id = Number(new Date())
10. this.props.addPicture(values.id,values.imageLink,values.description)
11. this.props.history.push('/')
12. }
13. }
14. render() {
15. return (
16. **<div** className="add-post"**>**
17. **<div** className="form" onSubmit={this.handleSubmit}**>**
18. **<form** className="add-form"**>**
19. **<input** name="imageLink" placeholder="Link"**/>**
20. **<input** name="description" placeholder="Description"**/>**
21. **<button>**Post**</button>**
22. **</form>**
23. **</div>**
24. **</div>**
25. );
26. }
27. }
28. export default Single;

**/App**

1. import { bindActionCreators } from 'redux';
2. import { connect } from 'react-redux';
3. import \* as actionCreators from '../redux/action'
4. import Main from './Main'
5. import {withRouter } from 'react-router';
6. function mapStateToProps(state, ownProps) {
7. return {
8. posts: state
9. }
10. }
11. function mapDispachToProps(dispatch) {
12. return bindActionCreators(actionCreators, dispatch);
13. }
14. const App = withRouter(connect(mapStateToProps, mapDispachToProps)(Main));
15. export default App;

**Main**

1. import React, {Component} from 'react';
2. import {Link} from 'react-router-dom'
3. import PhotoWall from '../Components/PhotoWall'
4. import AddPost from '../Components/AddPost'
5. import { Route,Switch } from 'react-router-dom'
6. class Main extends Component {
7. render() {
8. return (
9. **<div>**
10. **<h1** className="font-face"**>**
11. **<Link** to="/"**>**Photowall**</Link>**
12. **</h1>**
13. **<Switch>**
14. **<Route** exact path='/' render={(params) =**>** (**<PhotoWall** {...this.props}{...params}**/>**)}**/>**
15. **<Route** path='/create' render={(params) =**>** (**<AddPost** {...this.props}{...params}**/>**)}**/>**
16. **</Switch>**
17. **</div>**
18. );
19. }
20. }
21. export default Main;

/Photo

1. import React, { Component } from 'react';
2. import { Link } from 'react-router-dom'
3. class Photo extends Component {
4. render() {
5. console.log(this.props)
6. const { post, i } = this.props;
7. return (
8. **<figure** className="grid-figure"**>**
9. **<div** className="grid-photo-wrap"**>**
10. **<img** src={post.imageLink} alt={post.description} className="grid-photo" **/>**
11. **</div>**
12. **<figcaption>**
13. **<p>**{post.description}**</p>**
14. **<div** className="control-buttons"**>**
15. **<button** className="Remove" onClick={() =**>** {
16. this.props.history.push('/')
17. this.props.removePicture(i)
18. }}**>**Remove**</button>**
19. **</div>**
20. **</figcaption>**
21. **</figure>**
22. );
23. }
24. }
25. export default Photo;

/Photowall:

1. import React, { Component } from 'react';
2. import Photo from './Photo'
3. import { Link } from 'react-router-dom'
4. class PhotoWall extends Component {
5. render() {
6. return (
7. **<div>**
8. **<Link** to='/create' className="addpicture"**></Link>**
9. **<div** className="photo-grid"**>**
10. {console.log(this.props)}
11. {this.props.posts.sort(function(x, y){return y.id - x.id;}).map((post, i) =**>** {
12. return **<Photo** key={i} post={post} i={i} {...this.props}**/>**
13. })}
14. **</div>**
15. **</div>**
16. );
17. }
18. }
19. export default PhotoWall;

**/Posts**

1. const posts = [{
2. id: "0",
3. description: "beautiful landscape",
4. imageLink: "https://image.jimcdn.com/app/cms/image/transf/none/path/sa6549607c78f5c11/image/i4eeacaa2dbf12d6d/version/1490299332/most-beautiful-landscapes-in-europe-lofoten-european-best-destinations-copyright-iakov-kalinin.jpg" +
5. "3919321\_1443393332\_n.jpg"
6. }, {
7. id: "1",
8. description: "Aliens???",
9. imageLink: "https://img.purch.com/rc/640x415/aHR0cDovL3d3dy5zcGFjZS5jb20vaW1hZ2VzL2kvMDAwLzA3Mi84NTEvb3JpZ2luYWwvc3BhY2V4LWlyaWRpdW00LWxhdW5jaC10YXJpcS1tYWxpay5qcGc=" +
10. "08323785\_735653395\_n.jpg"
11. }, {
12. id: "2",
13. description: "On a vacation!",
14. imageLink: "https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2017/08/24/104670887-VacationExplainsTHUMBWEB.1910x1000.jpg"
15. }]
16. export default posts;

/Action

1. export function addPicture(id, imageLink, description) {
2. return {
3. type: 'ADD\_PICTURE',
4. id,
5. imageLink,
6. description
7. }
8. }
10. export function removePicture(i) {
11. return {
12. type: 'REMOVE\_PICTURE',
13. i
14. }
15. }

/Reducer

1. import {combineReducers } from 'redux';
2. import \_posts from '../data/posts';
3. const postReducer = function posts(state = \_posts, action) {
4. console.log(action.type)
5. switch(action.type) {
6. case 'REMOVE\_PICTURE' :
7. return [...state.slice(0,action.i),...state.slice(action.i + 1)]
8. case 'ADD\_PICTURE' :
9. return [{"id": action.id, "imageLink": action.imageLink, "description": action.description},...state]
10. default: return state
11. }
12. }
13. export default postReducer;

/stylesheet

1. /\*
2. Variables
3. \*/
4. html {
5. font-size: 10px;
6. font-family: sans-serif;
7. }
8. p {
9. font-size: 1.6rem;
10. line-height: 1.5;
11. }
12. h1 {
13. font-family: billabong, 'billabongregular';
14. text-align: center;
15. font-weight: 100;
16. font-size: 13rem;
17. margin: 2rem 0;
18. letter-spacing: -1px;
19. color: black;
20. text-decoration: none;
21. }
22. h1 a {
23. color: black;
24. text-decoration: none;
25. }
26. h1 a:focus {
27. outline: 0;
28. }
29. @font-face {
30. font-family: 'billabongregular';
31. src: url("https://cdn.rawgit.com/milktronics/beaglegr.am/master/public/fonts/billabong-webfont.eot");
32. src: url("https://cdn.rawgit.com/milktronics/beaglegr.am/master/public/fonts/billabong-webfont.eot?#iefix") format('embedded-opentype'), url("https://cdn.rawgit.com/milktronics/beaglegr.am/master/public/fonts/billabong-webfont.woff") format('woff'), url("https://cdn.rawgit.com/milktronics/beaglegr.am/master/public/fonts/billabong-webfont.ttf") format('truetype'), url("https://cdn.rawgit.com/milktronics/beaglegr.am/master/public/fonts/billabong-webfont.svg#billabongregular") format('svg');
33. font-weight: normal;
34. font-style: normal;
35. }
36. body {
37. background: offwhite;
38. }
39. .photo-grid {
40. display: flex;
41. flex-wrap: wrap;
42. max-width: 1000px;
43. margin: 0 auto;
44. }
45. .grid-figure,
46. .single-photo {
47. flex-basis: calc(33.333% - 4rem);
48. flex-grow: 1;
49. flex-shrink: 0;
50. margin: 0 2rem 2rem 2rem;
51. padding: 2rem;
52. border: 1px solid #d3d3d3;
53. background: #fff;
54. position: relative;
55. }
56. .single-photo {
57. max-width: 1000px;
58. margin: 0 auto;
59. display: flex;
60. background: #fff;
61. }
62. .single-photo .grid-figure {
63. margin: 0 2rem 0 0;
64. border: 0;
65. padding: 0;
66. flex: 1 0 60%;
67. max-width: 60%;
68. }
69. .single-photo .comments {
70. flex: 1 0 40%;
71. max-width: 40%;
72. }
73. .single-photo .grid-photo {
74. width: 100%;
75. margin: 0;
76. }
77. .grid-photo {
78. width: calc(100% + 4rem);
79. margin-left: -2rem;
80. margin-top: -2rem;
81. max-width: none;
82. }
83. .remove-comment {
84. background: none;
85. border: 0;
86. line-height: 1;
87. opacity: 0;
88. }
89. .remove-comment:hover {
90. color: #f00;
91. }
92. .comment{
93. width: 100%
94. }
95. .comment p {
96. border-bottom: 1px solid #d3d3d3;
97. padding: 0.5rem 0;
98. }
99. .comment-form input,
100. .comment-form textarea {
101. width: 100%;
102. border: 0;
103. font-size: 1.3rem;
104. padding: 1rem 0;
105. border-bottom: 1px solid #d3d3d3;
106. outline: none;
107. resize: vertical;
108. }
109. .grid-photo-wrap {
110. position: relative;
111. }
112. .remove {
113. background-size: contain;
114. font-size: 2rem;
115. padding: 1rem;
116. position: absolute;
117. color:red;
118. left: 50%;
119. top: 50%;
120. pointer-events: none;
121. }
122. .addpicture {
123. background: url("https://image.flaticon.com/icons/svg/60/60740.svg") center no-repeat;
124. background-size: contain;
125. padding: 40px;
126. display: table;
127. margin: 20px auto;
128. border: none;
129. }
130. /\*
131. Buttons
132. \*/
133. .control-buttons {
134. margin: 0 auto;
135. display: flex;
136. align-items: center;
137. justify-content: center;
138. }
139. button, .button {
140. border: 2px solid #f2f2f2;
141. background: none;
142. flex-basis: 48%;
143. display: inline-block;
144. line-height: 2;
145. text-decoration: none;
146. padding: 5px;
147. text-align: center;
148. font-size: 15px;
149. color: red;
150. transition: all 0.2s;
151. box-sizing: padding-box;
153. }
154. button:hover,.button:hover, button:focus, .button:focus {
155. border-color: #00f;
156. outline: 0;
157. }
158. /\*
159. Cowboy style speech bubble - you should probably use an SVG for this if you are doing more icons.
160. \*/
161. .speech-bubble {
162. width: 1.5rem;
163. height: 1.25rem;
164. background: #00f;
165. display: inline-block;
166. border-radius: 50%;
167. position: relative;
168. }
169. .speech-bubble:after {
170. display: inline-block;
171. position: absolute;
172. content: '';
173. width: 0;
174. height: 0;
175. border-style: solid;
176. border-width: 0 1.25rem 1.25rem 0;
177. border-color: transparent #00f transparent transparent;
178. top: 30%;
179. left: 0;
180. }
181. /\*Add Post css\*/
182. .add-post {
183. width: 360px;
184. padding: 0 0 0;
185. margin: auto;
186. }
187. .form input {
188. font-family: sans-serif;
189. outline: 0;
190. background: #f2f2f2;
191. width: 100%;
192. border: 0;
193. margin: 0 0 15px;
194. padding: 15px;
195. box-sizing: border-box;
196. font-size: 14px;
197. }
198. .form button {
199. font-family: "Roboto", sans-serif;
200. text-transform: uppercase;
201. outline: 0;
202. background:#065784;
203. width: 100%;
204. border: 0;
205. padding: 15px;
206. color: #FFFFFF;
207. font-size: 14px;
208. -webkit-transition: all 0.3 ease;
209. transition: all 0.3 ease;
210. cursor: pointer;
211. }
212. .form button:hover,.form button:active,.form button:focus {
213. background: #7B8589;
214. }

/Index.js

1. import React from 'react';
2. import ReactDOM from 'react-dom';
3. import App from './Components/App'
4. import { BrowserRouter } from 'react-router-dom'
5. import './styles/stylesheet.css'
6. import {Provider} from 'react-redux'
7. import { createStore } from 'redux';
8. import rootReducer from './redux/reducer';
9. const store = createStore(rootReducer)
10. ReactDOM.render(**<Provider** store={store}**>**
11. **<BrowserRouter>**
12. **<App/>**
13. **</BrowserRouter>**
14. **</Provider>**,
15. document.getElementById('root'));
16. /index.html  
      
    <!DOCTYPE html**>**
17. **<html** lang="en"**>**
18. **<head>**
19. **<meta** charset="utf-8"**>**
20. **<meta** name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"**>**
21. **<meta** name="theme-color" content="#000000"**>**
22. <!--
23. manifest.json provides metadata used when your web app is added to the
24. homescreen on Android. See https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/
25. --**>**
26. **<link** rel="manifest" href="%PUBLIC\_URL%/manifest.json"**>**
27. **<link** rel="shortcut icon" href="%PUBLIC\_URL%/favicon.ico"**>**
28. <!--
29. Notice the use of %PUBLIC\_URL% in the tags above.
30. It will be replaced with the URL of the `public` folder during the build.
31. Only files inside the `public` folder can be referenced from the HTML.
32. Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will
33. work correctly both with client-side routing and a non-root public URL.
34. Learn how to configure a non-root public URL by running `npm run build`.
35. --**>**
36. **<title>**React App**</title>**
37. **</head>**
38. **<body>**
39. **<noscript>**
40. You need to enable JavaScript to run this app.
41. **</noscript>**
42. **<div** id="root"**></div>**
43. <!--
44. This HTML file is a template.
45. If you open it directly in the browser, you will see an empty page.
46. You can add webfonts, meta tags, or analytics to this file.
47. The build step will place the bundled scripts into the **<body>** tag.
48. To begin the development, run `npm start` or `yarn start`.
49. To create a production bundle, use `npm run build` or `yarn build`.
50. --**>**
51. **</body>**
52. **</html>**