ROTEIRO PARA CONSUMO DE UMA API

1. Crie a pasta Consumo de API
2. Na pasta crie o arquivo index.html
3. Acesse a página do github do AXIOS <https://github.com/axios/axios>
4. A biblioteca pode ser baixada via npm
5. ‘npm install axios’
6. ou utilizando o script cdn
7. “<script src="https://unpkg.com/axios/dist/axios.min.js"></script>”
8. No arquivo index.html
9. Copia o script cdn abaixo da tag <\body.>
10. No arquivo index.html vamos criar a primeira requisição do axios, para ele carregar na página todos os dados da API games.
11. axios.get("http://localhost:45678/games").then(*response* =>{
12. console.log(*response*);
13. }).catch(*err*=>{
14. console.log(*err*);
15. Arquivo index.html configurado
16. <!DOCTYPE *html*>
17. <html *lang*="en">
18. <head>
19. <meta *charset*="UTF-8">
20. <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">
21. <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">
22. <title>Loja de Games</title>
23. </head>
24. <body>
26. </body>
27. <script *src*="https://unpkg.com/axios/dist/axios.min.js"></script>
28. <script>
29. axios.get("http://localhost:45678/games").then(*response* =>{
30. console.log(*response*);
31. }).catch(*err*=>{
32. console.log(*err*);
33. });
34. </script>
35. </html>
36. Na pasta APIDEGAMES
37. Para evitar erro de ‘cors’ na execução do consumo da API deve-se instalar a biblioteca
38. ‘npm install cors –save’ na pasta da API
39. No arquivo index.js
40. Importar biblioteca cors ‘const cors = require(‘cors’);
41. Configurando o aplicativo para executar o cors ‘app.use(cors());
42. Iniciar a API no console na pasta APIDEGAMES ‘node index.js’
43. Na pasta Consumo de API
44. No arquivo index.html
45. Vamos configurar uma <ul> para receber os títulos dos games
46. <hr>
47. <h1>Lista de games</h1>
48. <ul *id*="games">
49. </ul>
50. Vamos configurar o código para carregar dentro da ul os jogosCr
51. <!DOCTYPE *html*>
52. <html *lang*="en">
53. <head>
54. <meta *charset*="UTF-8">
55. <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">
56. <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">
57. <title>Loja de Games</title>
58. <hr>
59. <h1>Lista de games</h1>
60. <ul *id*="games">
61. </ul>
62. </head>
63. <body>
65. </body>
66. <script *src*="https://unpkg.com/axios/dist/axios.min.js"></script>
67. <script>
68. axios.get("http://localhost:45678/games").then(*response* =>{
69. //a variavel games irá receber os dados da resposta
70. var games = *response*.data;
71. //a variavel list será os itens da ul 'games'
72. var list = document.getElementById('games');
73. //percorrendo o array dos dados
74. games.forEach(*game* => {
75. console.log(*game*);
76. //criando um item de uma lista
77. var item = document.createElement('li');
79. //vamos salvar dentro do item 'li' os dados
80. item.innerHTML = *game*.id + " - " + *game*.title + " - " + *game*.price;
81. //vamos adicionar o item dentro da lista
82. list.appendChild(item);
83. });
84. }).catch(*err*=>{
85. console.log(*err*);
86. });
87. </script>
88. </html>
89. Criando inputs no head da página para cadastrar novos games
90. <head>
91. <meta *charset*="UTF-8">
92. <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">
93. <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">
94. <title>Loja de Games</title>
95. <hr>
96. <h1>Lista de games</h1>
97. <ul *id*="games">
99. </ul>
100. <hr>
101. <h4>Novo game</h4>
102. <input *type*="text" *name*="title" *id*="title" *placeholder*="Título"><br>
103. <input *type*="text" *name*="year" *id*="year" *placeholder*="Ano"><br>
104. <input *type*="number" *name*="price" *id*="price" *placeholder*="Preço"><br>
105. <button *type*="button" *onclick*="createGame()" >Criar</button>
107. </head>
108. Criar uma função para coletar os dados dentro dos inputs e salvar os novos dados utilizando o axios.
109. <script>
110. //função para cadastrar novos games
111. function createGame(){
112. //pegando o valor dos inputos
113. var titleInput = document.getElementById('title');
114. var yearInput = document.getElementById('year');
115. var priceInput = document.getElementById('price');
116. //montando um json que será enviado para a API
117. var game = {
118. title: titleInput.value,
119. year: yearInput.value,
120. price: priceInput.value
121. }
122. //rota para cadastrar um novo game, basta informar a rota e os dados
123. axios.post("http://localhost:45678/game",game).then(*response*=>{
124. //irá verificar se foi realizado o cadastro
125. if(*response*.status == 200 ){
126. console.log("Game cadastrado");
127. alert("Game cadastrado");
128. }
129. }).catch(*err*=>{
130. console.log(*err*);
131. })
132. }
133. //o axios irá executar a url para buscar todos os games
134. axios.get("http://localhost:45678/games").then(*response* =>{
135. //a variavel games irá receber os dados da resposta
136. var games = *response*.data;
137. //a variavel list será os itens da ul 'games'
138. var list = document.getElementById('games');
139. //percorrendo o array dos dados
140. games.forEach(*game* => {
141. //console.log(game);
142. //criando um item de uma lista
143. var item = document.createElement('li');
145. //vamos salvar dentro do item 'li' os dados
146. item.innerHTML = *game*.id + " - " + *game*.title + " - " + *game*.price;
147. //vamos adicionar o item dentro da lista
148. list.appendChild(item);
149. });
150. }).catch(*err*=>{
151. console.log(*err*);
152. });
154. </script>
155. ATRIBUTOS PERSONALIZAVEIS
156. Podemos atribuir atributos personalizados no html, basta nomearmos como data-“atributo”
157. No forEach de games vamos inserir um novo atributo ao item
158. ‘item.setAttribute
159. Faça a alteração no arquivo index.html conforme abaixo:
160. //percorrendo o array dos dados
161. games.forEach(*game* => {
162. //console.log(game);
163. //criando um item de uma lista
164. var item = document.createElement('li');
166. //setando o atributo data-id ao item
167. item.setAttribute("data-id",*game*.id);
168. //setando o atributo data-title ao item
169. item.setAttribute('data-title',*game*.title);
170. //setando o atributo data-year ao item
171. item.setAttribute('data-year',*game*.year);
172. //setando o atributo data-price ao item
173. item.setAttribute('data-price',*game*.price);
174. //vamos salvar dentro do item 'li' os dados
175. item.innerHTML = *game*.id + " - " + *game*.title + " - " + *game*.price;
176. //vamos adicionar o item dentro da lista
177. list.appendChild(item);
178. });
179. Criando um botão ao lado de cada item para apagar o item se necessário.
180. //percorrendo o array dos dados
181. games.forEach(*game* => {
182. //console.log(game);
183. //criando um item de uma lista
184. var item = document.createElement('li');
186. //setando o atributo data-id ao item
187. item.setAttribute("data-id",*game*.id);
188. //setando o atributo data-title ao item
189. item.setAttribute('data-title',*game*.title);
190. //setando o atributo data-year ao item
191. item.setAttribute('data-year',*game*.year);
192. //setando o atributo data-price ao item
193. item.setAttribute('data-price',*game*.price);
194. //vamos salvar dentro do item 'li' os dados
195. item.innerHTML = *game*.id + " - " + *game*.title + " - " + *game*.price;
196. //criando o botão ao lado de cada item
197. var deleteItem = document.createElement("button");
199. //adicionando o botão na página com o caption Deletar
200. deleteItem.innerHTML = "Deletar";
201. //adicionando o botão dentro do item
202. item.appendChild(deleteItem)
203. //vamos adicionar o item dentro da lista
204. list.appendChild(item);
205. });
206. ------ SEM COMMIT ----
207. Criado função para deletar um item e configurando o item da lista e rota delete no axios.
208. //função para deletar um item
209. function deletarItem(*listItem*){
210. //capturando o id do item
211. var id = *listItem*.getAttribute('data-id');
212. //rota utilizando o axios para deletar o item
213. axios.delete("http://localhost:45678/game/"+id).then(*response*=>{
214. alert("Game deletetado");
215. }).catch(*err*=>{
216. console.log(*err*);
217. });
218. }