

todo

src/App.js

```
1  import React from 'react';
2  // import logo from './logo.svg';
3  // import './App.css';
4  import './todo.css'
5
6  class App extends React.Component {
7    constructor () {
8      super()
9      this.state = {
10        curretTodoName: '',
11        currendEditId: '',
12        // 全选全不选
13        isAllSelect: '',
14        filter: '',
15        //
16        todos: [
17          {
18            id: 1,
19            name: '吃饭饭',
20            check: false
21          },
22          {
23            id: 2,
24            name: '睡觉觉',
25            check: true
26          },
27          {
28            id: 3,
29            name: '打豆豆',
30            check: false
31          },
32        ]
33      }
34    }
35    componentDidMount () {
36      // 获取路由 监听路由
37      this.renderByHash()
38    }
39    // 渲染任务列表
40    // 列表渲染
41    renderTodos () {
42      return this.state.todos.map( item => {
43        if (this.state.filter === '#/active' && item.check) return null
44        if (this.state.filter === '#/completed' && !item.check) return null
45        return (
```

```

46     // 动态class
47     <li className={ [item.check ? 'completed' : null, this.state.currentEditId
=== item.id ? 'editing' : null].join(' ') } key={ item.id }>
48         <div className="view">
49             <input className="toggle" type="checkbox" checked={ item.check }
50                 onChange={ e => this.changeCheckBox(e, item) } />
51             <label onDoubleClick={ e => this.setCurrentEditId(e, item) } >{
item.name }</label>
52             <button className="destroy" onClick={ e => this.removeTodo(item.id)
}></button>
53         </div>
54         <input id="input1" ref={ `eidInput_${item.id}` } className="edit" value={
item.name } onBlur={ e=> this.setCurrentEditId(e, item, -1) } onChange={ e=>
this.editDoto(e, item) } autoFocus />
55     </li>
56 )
57 })
58 }
59 // 14: 根据hash来过滤输出
60 renderByHash () {
61     // 监视锚点后续改变
62     window.addEventListener('hashchange', ()=>{
63         this.setState({
64             filter: window.location.hash
65         })
66     })
67     this.setState({
68         filter: window.location.hash
69     })
70 }
71 // 赋值当前任务标题
72 setCurrentName (e) {
73     this.setState({
74         curretTodoName: e.target.value
75     })
76 }
77 // 添加任务
78 addTodo (e) {
79     e.preventDefault()
80     console.log('1')
81     if (this.state.curretTodoName.trim() === '') {
82         return false
83     }
84     let minId = -1
85     this.state.todos.forEach(item => {
86         if (item.id > minId) {
87             minId = item.id
88         }
89     })
90     minId++
91     this.state.todos.push({
92         id: minId,
93         name: this.state.curretTodoName,

```

```

94     check: false
95   })
96   // 更新视图层
97   this.setState({
98     currentTodoName: ''
99   })
100 }
101 // 删除任务
102 removeTodo (id) {
103   let todoIndex = this.state.todos.findIndex( item => {
104     return item.id === id
105   })
106   this.state.todos.splice(todoIndex, 1)
107   this.setState({})
108 }
109 // 改变checkbox
110 changeCheckBox (e, todo) {
111   todo.check = e.target.checked
112   this.setState({})
113 }
114 // 设置当前编辑的任务
115 setCurrentEditId (e, todo, cancel) {
116   if (cancel === -1) {
117     this.setState({
118       currentEditId: -1
119     })
120     return
121   }
122   // 更新后 才能使用DOM
123   // 回调 refs
124   this.setState({
125     currentEditId: todo.id
126   }, () => {
127     this.refs[`eidInput_${todo.id}`].focus()
128   })
129 }
130 // 对当前的todo 进行编辑
131 // 直接就是双向的绑定 为什么要写这么多代码
132 editDoto (e, todo) {
133   todo.name = e.target.value
134   this.setState({})
135 }
136 // 全选
137 allSelect (e) {
138   this.state.todos.forEach( item => {
139     item.check = e.target.checked
140   })
141   this.setState({
142     isAllSelect: e.target.checked
143   })
144 }
145 render () {
146   return (

```

```

147     <React.Fragment>
148       <section className="todoapp">
149         <header className="header">
150           <h1>todos</h1>
151           <form onSubmit={ e=> this.addToDo(e) }>
152             <input className="new-todo" value={ this.state.curretTodoName }
onChange={ e=> this.setCurrentName(e) } placeholder="what needs to be done?"
autoFocus />
153           </form>
154         </header>
155         { /* This section should be hidden by default and shown when there are todos
*/}
156         <section className="main">
157           <input id="toggle-all" className="toggle-all" type="checkbox" checked={
this.state.isAllSelect } onChange={ e=> this.allSelect(e) } />
158           <label htmlFor="toggle-all">Mark all as complete</label>
159           <ul className="todo-list">
160             { this.renderTodos() }
161           </ul>
162         </section>
163         { /* This footer should hidden by default and shown when there are todos */}
164         <footer className="footer">
165           { /* This should be `0 items left` by default */}
166           <span className="todo-count">
167             <strong>{
168               this.state.todos.filter(item => {
169                 return !item.check
170               }).length
171             }</strong>
172             item left
173           </span>
174           { /* Remove this if you don't implement routing */}
175           <ul className="filters">
176             <li>
177               <a className={ this.state.filter === '/' ? 'selected' : null }
href="/">All</a>
178             </li>
179             <li>
180               <a className={ this.state.filter === '/active' ? 'selected' :
null } href="/active">Active</a>
181             </li>
182             <li>
183               <a className={ this.state.filter === '/completed' ? 'selected'
: null } href="/completed">Completed</a>
184             </li>
185           </ul>
186           { /*Hidden if no completed items are left ↓ */}
187           <button className="clear-completed" onClick={ e=> {
188             this.setState({
189               todos: this.state.todos.filter(item => {
190                 return !item.check
191               })
192             })

```

```
193         } }>Clear completed</button>
194     </footer>
195 </section>
196 <footer className="info">
197     <p>Double-click to edit a todo</p>
198     { /* Remove the below line ↓ */ }
199     <p>Template by <a href="http://sindresorhus.com">Sindre Sorhus</a></p>
200     { /* Change this out with your name and url ↓ */ }
201     <p>Created by <a href="http://todomvc.com">you</a></p>
202     <p>Part of <a href="http://todomvc.com">TodoMVC</a></p>
203 </footer>
204 </React.Fragment>
205 )
206 }
207
208 }
209
210 export default App;
211
```