

```
1  import {makeHttpRequest} from './helper.js';
2
3  export class EditableField {
4      constructor(id,property,field,editButton,currentValue,emptyValue) {
5          this.id = id;
6          this.field = field;
7          this.editButton = editButton;
8          this.currentValue = currentValue;
9          this.emptyValue = emptyValue;
10         this.property = property;
11         this.isEditing = false;
12
13         // Populate the field
14         field.innerHTML = currentValue ? currentValue : emptyValue;
15
16         // Setup the edit button
17         this.editButton.addEventListener('click', this.handleEditStart.bind(this));
18     }
19
20     handleEditStart() {
21         this.isEditing = true;
22         this.editButton.classList.add('hide');
23
24         this.field.innerHTML = this.currentValue;
25         this.field.classList.add('editing');
26         this.field.contentEditable = true;
27         this.placeCaretAtEnd(this.field);
28
29         this.addSaveCancelButtons();
30     }
31
32     addSaveCancelButtons() {
33
34         // Create a div to put the buttons in
35         this.buttonDiv = document.createElement('div');
36
37         // Create the save button
38         let saveBut = document.createElement('button');
39         saveBut.innerHTML = "Save";
40         this.buttonDiv.appendChild(saveBut)
41
42         // Create the cancel button
43         let cancelBut = document.createElement('button');
44         cancelBut.innerHTML = "Cancel";
45         this.buttonDiv.appendChild(cancelBut)
46
47         // Add event listeners
48         saveBut.addEventListener('click', this.handleSave.bind(this))
49         cancelBut.addEventListener('click', this.handleCancel.bind(this))
50
51         // Add the buttons beneath the text field
52         this.field.parentNode.insertBefore(this.buttonDiv, this.field.nextSibling);
53
54     }
55
56     handleSave() {
57
58         this.isEditing = false;
59         this.editButton.classList.remove('hide');
60         this.buttonDiv.innerHTML = "";
61     }
```

```
62     if (this.field.textContent == "") {
63         this.field.innerHTML = this.emptyValue;
64         this.currentValue = "";
65     }
66     else {
67         this.currentValue = this.field.innerHTML;
68     }
69
70     this.field.classList.remove('editing');
71     this.field.contentEditable = false;
72
73
74     let data = {};
75     data[this.property] = this.currentValue;
76
77     makeHttpRequest(`api/logs/${this.id}`, 'PATCH', JSON.stringify(data), 'application/json')
78     .then(res => {
79         console.log(res);
80     })
81
82 }
83
84 handleCancel() {
85
86     this.isEditing = false;
87     this.editButton.classList.remove('hide');
88     this.buttonDiv.innerHTML = "";
89
90     this.field.innerHTML = this.currentValue ? this.currentValue : this.emptyValue;
91     this.field.classList.remove('editing');
92     this.field.contentEditable = false;
93
94 }
95
96
97
98 // From https://stackoverflow.com/questions/4233265/contenteditable-set-caret-at-the-end-of-the-text-cross-
99 browser
100 placeCaretAtEnd(el) {
101     el.focus();
102     if (typeof window.getSelection != "undefined"
103         && typeof document.createRange != "undefined") {
104         var range = document.createRange();
105         range.selectNodeContents(el);
106         range.collapse(false);
107         var sel = window.getSelection();
108         sel.removeAllRanges();
109         sel.addRange(range);
110     } else if (typeof document.body.createTextRange != "undefined") {
111         var textRange = document.body.createTextRange();
112         textRange.moveToElementText(el);
113         textRange.collapse(false);
114         textRange.select();
115     }
116 }
117 }
```