

Page de configuration

Contexte:

Pour le bon fonctionnement des produits et services proposés par mon entreprise, il est nécessaire que le client puisse procéder à quelques configurations. Il m'a donc chargé de m'occuper de la page de configuration en respectant l'esthétique globale du site.

Compétence travaillées:

Le langage utilisé est React (bibliothèque JS). Git permettait le suivi du projet pour mon responsable et de me faire de retour.. J'ai beaucoup appris sur les formulaire et les appels à API.

Annexes:

```
49 <Card body>
50   <CardBody>
51     <CardTitle tag="h4">
52       {t(Reset of the configuration)}
53     </CardTitle>
54     <CardSubtitle tag="h5" className="mb-2 text-muted">
55       {t(Use the button below to reset the KNX co..)}
56     </CardSubtitle>
57     <Button className="btn-round h5" color="primary" onClick={handleOpen}>
58       >
59     </Button>
60     {t(Factory reset)}
61   </CardBody>
62   <Modal>
63     centered
64     size="lg"
65     toggle={toggleOpen}
66     isOpen={open}
67   </Modal>
68   <ModalHeader toggle={toggleOpen}>
69     {t(Confirmation)}
70   </ModalHeader>
71   <ModalBody>
72     {t(NOTE: This will overwrite the factory de..)}
73   </ModalBody>
74   <ModalFooter>
75     <Button
76       color="primary"
77       onClick={() => { doOnValidation(); handleClose() }}
78     >
79     {t(Confirm)}
80   </ModalFooter>
81 </Card>
82
```

```
4 Button,
5 Input,
6 Col,
7 Form,
8 FormGroup,
9 FormFeedback,
10 Label,
11 } from "reactstrap";
12
13 export default function ConfigBody(props) {
14
15   let validateNbChild = (routerChildAddr, nbRouterChild) => {
16     const routerChildLastDigit = parseInt(routerChildAddr.split(".")[2])
17     nbRouterChild = parseInt(nbRouterChild)
18
19     if (routerChildLastDigit + nbRouterChild - 1 < 0 || routerChildLastDigit +
20       return false;
21     }
22     if (nbRouterChild < 2 || nbRouterChild > 30) {
23       return false;
24     }
25     return true
26   }
27   const { t } = useTranslation();
28
29   let validateData = (data) => {
30
31     //TODO : Handle case where router and routerChild Address are overlapping
32
33     const knxAddressRegex = /^(1[0-5]||[0-9]?)\.(1[0-5]||[0-9]?)\.(25[0-5]|2[0-4]
34     props.setNewData({
35       ...data, valid: {
36         routerPhysicalAddr: knxAddressRegex.test(data.routerPhysicalAddr) && da
```


Configuration of the KNX router

Use the form below to change the address of the Knx router, as well as the configuration of the physical addresses distributed by the router

V.1.2.9


Physical address of router

11128




Physical address of router

11129



Number of addresses available for distribution

30



Modify

Reset of the configuration

Use the button below to reset the KNx configuration of the Mixibus

Factory reset


Configuration of the KNX router

Use the form below to change the address of the Knx router, as well as the configuration of the physical addresses distributed by the router

V.1.2.9


Physical address of router

11128




Physical address of router

11129



Number of addresses available for distribution

30



Confirm

Reset of the configuration

Use the button below to reset the KNx configuration of the Mixibus

Factory reset