

# Page de mise à jour

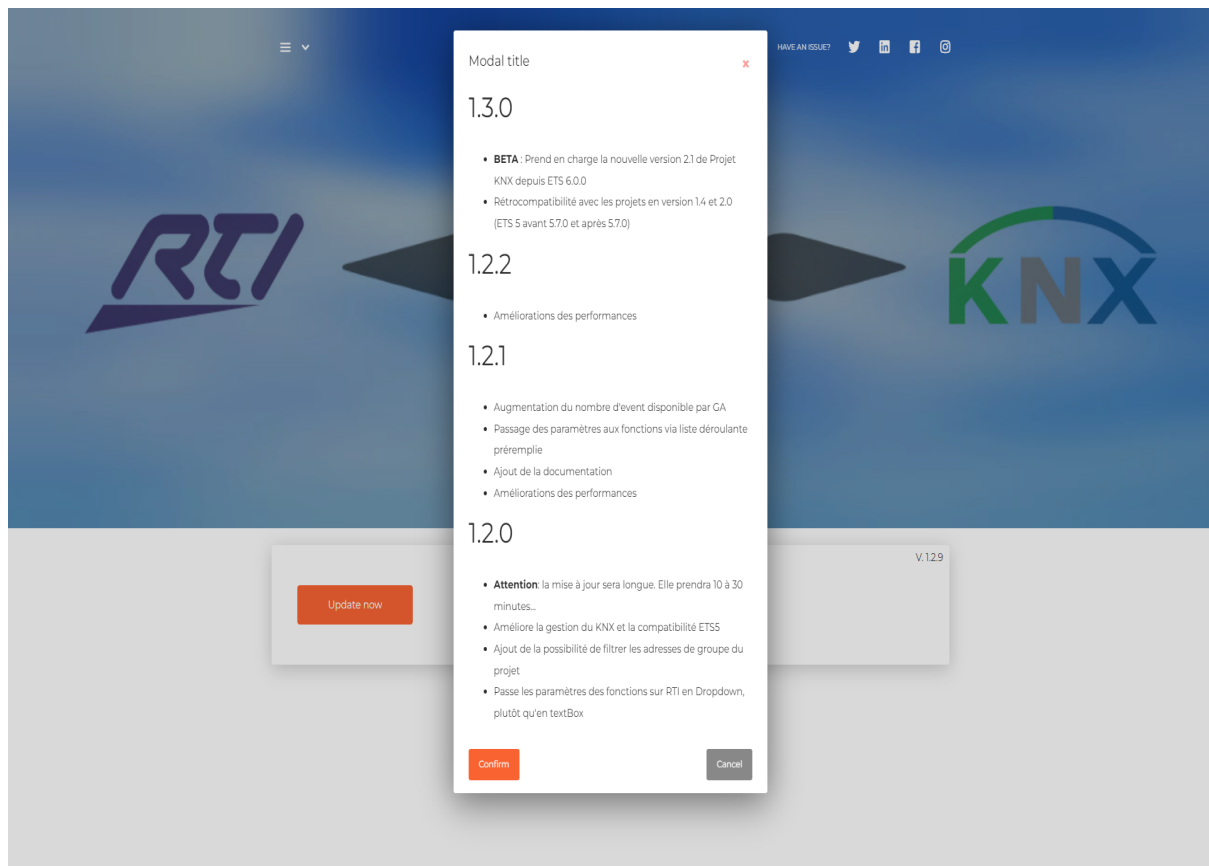
## Contexe:

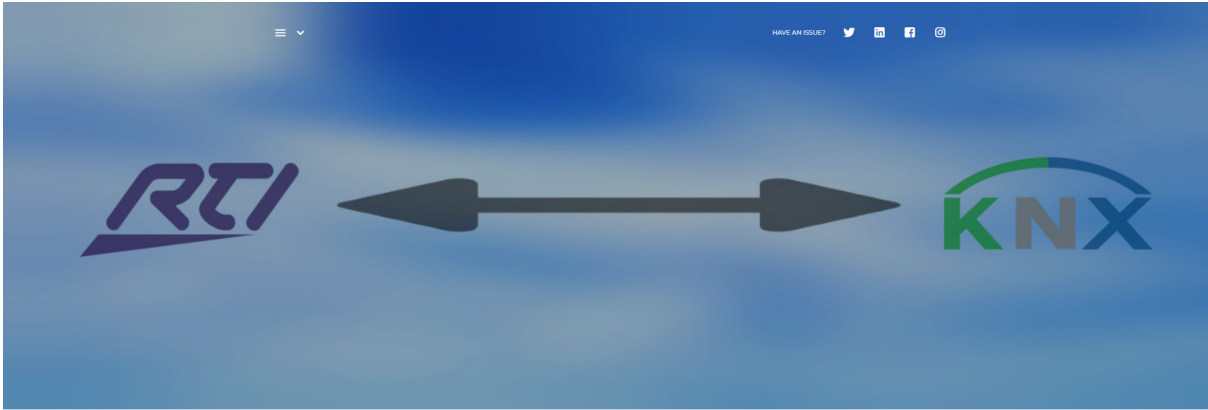
Le domaine de la domotique évolue continuellement et très vite. Le site nécessite donc de faire appel à beaucoup d'API et il est important pour la stabilité et la sécurité du client et de ses appareils/configuration d'avoir une version à jour.

## Compétences travaillées:

C'était l'une de mes première mission dans ce projet avec React donc j'en ai beaucoup appris sur ce langage. Git permettait le suivi du projet pour mon responsable et de me faire de retour. J'ai appris à faire appel à un API.

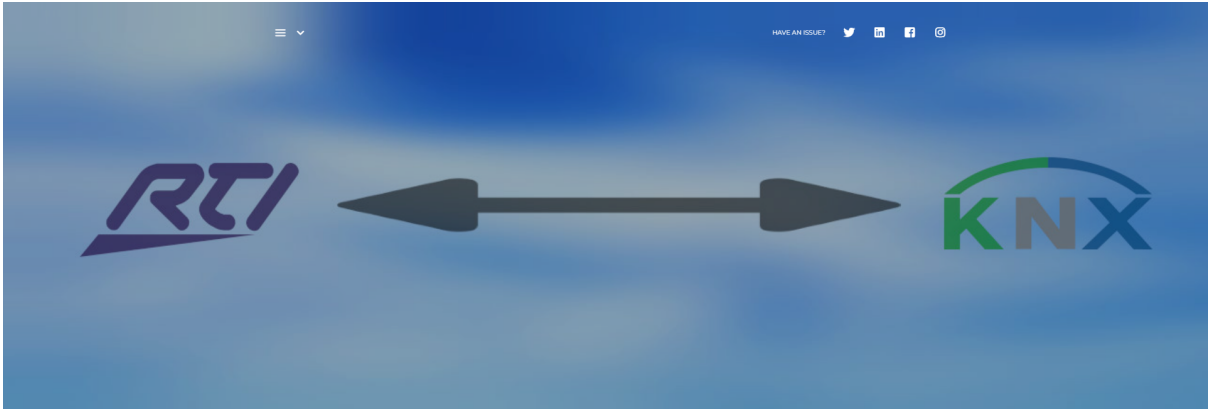
## annexes :





Update now

V.12.9



Up to date

V.12.9

```

8 export default function CurrentVersion(props) {
9   const URL_API_CURRENT_VERSION = URLS.URL_API_CURRENT_VERSION;
10   const URL_API_STABLE_VERSION = URLS.URL_API_STABLE_VERSION;
11
12   const { enqueueSnackbar } = useSnackbar();
13
14   // TODO : comparer stable et current pour afficher si une nouvelle maj est dispo
15   let [currentVersion, setCurrentVersion] = useState('');
16   let [stableVersion, setStableVersion] = useState('');
17
18   useEffect(() => {
19     if(props.current == null || props.stable == null) {
20       reloadData();
21     } else {
22       setCurrentVersion(props.current)
23       setStableVersion(props.stable)
24     }
25   }, [])
26
27   useEffect(() => {
28     setCurrentVersion(props.current)
29     setStableVersion(props.stable)
30   }, [props.current, props.stable])
31
32   let refreshCurrentFromApi = () => {
33     return new Promise((resolve, reject) => {
34       fetch(URL_API_CURRENT_VERSION)
35         .then(result => {
36           return result.text()
37         })
38         .then(result => {
39           setCurrentVersion(result);
40           return resolve();
41         })
42     })
43   }
44
45   props.stable === props.current
46   ? <div>
47     <Button
48       color="success"
49       block
50       disabled
51       className="btn btn-primary btn-lg btn-block">
52       {/* TODO: ajouter une icone check avant le texte */}
53       {/* TODO: changer la taille/police du texte */}
54       {t(bp to date)}
55     </Button>
56   </div>
57   :
58   <div>
59     <Button
60       color=""
61       block
62       className="btn btn-primary btn-lg btn-block "
63       onClick={handleOpen}
64     >
65       {t(update now)}
66     </Button>
67   </div>
68 }
69
70 <Modal
71   isOpen={open} toggle={toggleOpen}
72 >
73   <ModalHeader toggle={toggleOpen}>
74     {t(modal title)}
75   </ModalHeader>
76   <ModalBody>
77     <div className="">

```