

```
1  # GROUPOMANIA
2
3  ## Créer un réseau social d'entreprise
4
5  *Projet 7 de la formation Développeur web d'OpenClassRooms.*
6
7  Le projet consiste à construire un réseau social interne pour les employés de
8  Groupomania.
9  Le but de cet outil est de faciliter les interactions entre les collègues.
10
11 ## Compétences évaluées
12
13 ##### La création de l'application fait appel aux compétences suivantes:
14
15 - Implémenter un stockage de données sécurisé en utilisant SQL
16 - Personnaliser le contenu envoyé à un client web
17 - Authentifier un utilisateur et maintenir sa session
18 - Gérer un stockage de données à l'aide de SQL
19
20 ## Comment utiliser l'application?
21
22 ##### Les installations et commandes suivantes sont nécessaires:
23
24 - Installer Node, Sass, Npm, mySql sur votre poste de travail
25 - Télécharger ou cloner le projet
26 - Vous devez créer 3 base de données Sql sur votre poste de travail (groupomania,
27   groupomania_test et groupomania_production) en les laissant dans le dossier par
28   défaut de vos bases de données MySql.
29 - Pour vous connecter à cette base de données, vous devez renseigner son nom dans le
30   fichier backend/config/config.json
31 - Vous devez aussi mettre vos informations de connection mysql à cet endroit, dans
32   mon cas sqluser/password.
33 - Une fois cela fait, vous pouvez créer les tables avec les modèles Sequelize que
34   j'ai concus. La commande pour ca est: npx sequelize db:migrate (depuis le terminal
35   backend).
36 - Certaines colonnes des tables ne seront pas utilisées car par manque de temps je
37   n'ai pas pu développer toutes les fonctionnalités que j'aurais voulues.
38 - Pour profiter pleinement de actions possibles dans l'application, il faut créer un
39   administrateur (qui fait office de modérateur). Pour cela, inscrivez vous de manière
40   classique sur l'application. Puis dans votre terminal mySql, et taper: UPDATE 'users'
41   SET 'isAdmin' = '1' WHERE 'id'=1 (si c'est le premier utilisateur que vous créez)
42 - Aller dans le frontend (via un terminal) et faire `npm install`
43 - Faire `npm run serve`, le serveur se lance sur http://localhost:8080
44 - Aller dans le backend (via un terminal) et faire `npm install`
45 - Toujours dans le dossier backend, créer un fichier .env et insérer dedans
46   (TOKEN='Random_secret_token')
47 - Faire `nodemon`. L'écoute se fait sur le port 3000 et vous pouvez vous rendre à
48   présent sur l'adresse du localhost pour tester l'application.
49
50 ##### Sécurisation de l'application et des données?
51
52 - bcrypt (hash et salage du mot de passe utilisateur)
53 - jsonwebtoken (sécurisation de l'authentification pour les requêtes)
54 - Base de données mySQL sur laquelle les informations sensibles ne sont pas stockées
55   sans cryptage
56 - Création d'une API REST avec le framework Express
57 - Vérification du jeton et du propriétaire pour la suppression de contenu
58 - Jeton avec une durée d'expiration de 12h
59
60 ## Environnement de développement
61
62 #####
63
64 - Framework VUE JS (VUE CLI, VUE ROUTER, VUE X)
65 - Node / Sass / Npm
```

```
53 - Sequelize / Axios
54 - Nodemon
55 - MySQL
56
57 #### Pour en savoir plus sur la sécurité informatique (développement/production/
   utilisation):
58
59 - [Framework Express: meilleures pratiques en production](https://expressjs.com/fr/
   advanced/best-practice-security.html)
60 - [GitHub Advisory Database](https://github.com/advisories)
61 - [OWASP Top 10 Web Application Security Risks](https://owasp.org/www-project-top-
   ten/)
```

```
1  .DS_Store
2  node_modules
3  /dist
4
5
6  /backend/images
7
8  # local env files
9  .env.local
10 .env.*.local
11
12 # Log files
13 npm-debug.log*
14 yarn-debug.log*
15 yarn-error.log*
16 pnpm-debug.log*
17
18 # Editor directories and files
19 .idea
20 .vscode
21 *.suo
22 *.ntvs*
23 *.njsproj
24 *.sln
25 *.sw?
```

```
1  # groupomania
2
3  VUE 2
4  VUE CLI 4.5.15
5  DART SASS
6  VUE ROUTEUR
7  VUE X
8
9  ## Project setup
10  ```
11  npm install
12  ```
13
14  ### Compiles and hot-reloads for development
15  ```
16  npm run serve
17  ```
18
19  ### Compiles and minifies for production
20  ```
21  npm run build
22  ```
23
24  ### Lints and fixes files
25  ```
26  npm run lint
27  ```
28
29  ### Customize configuration
30  See [Configuration Reference](https://cli.vuejs.org/config/).
```

```
1  {
2    "name": "groupomania",
3    "version": "0.1.0",
4    "private": true,
5    "scripts": {
6      "serve": "vue-cli-service serve",
7      "build": "vue-cli-service build",
8      "lint": "vue-cli-service lint"
9    },
10   "dependencies": {
11     "axios": "^0.24.0",
12     "core-js": "^3.6.5",
13     "vue": "^2.6.11",
14     "vue-router": "^3.2.0",
15     "vuex": "^3.4.0"
16   },
17   "devDependencies": {
18     "@vue/cli-plugin-babel": "~4.5.0",
19     "@vue/cli-plugin-eslint": "~4.5.0",
20     "@vue/cli-plugin-router": "~4.5.0",
21     "@vue/cli-plugin-vuex": "~4.5.0",
22     "@vue/cli-service": "~4.5.0",
23     "babel-eslint": "^10.1.0",
24     "eslint": "^6.7.2",
25     "eslint-plugin-vue": "^6.2.2",
26     "sass": "^1.26.5",
27     "sass-loader": "^8.0.2",
28     "vue-template-compiler": "^2.6.11"
29   }
30 }
```

```
1  <!DOCTYPE html>
2  <html lang="">
3    <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width,initial-scale=1.0">
7      <link rel="icon" href="<%= BASE_URL %>favicon.ico">
8      <title>Groupomania, réseau social</title>
9    </head>
10   <body>
11     <noscript>
12       <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work
properly without JavaScript enabled. Please enable it to continue.</strong>
13     </noscript>
14     <div id="app"></div>
15     <!-- built files will be auto injected -->
16   </body>
17 </html>
```

```
1  @import "variables";
2  @import "mixins";
3
4  h1 {
5    font-style: italic;
6    font-weight: bold;
7    text-align: left;
8    padding: 0 1rem;
9  }
10
11 h2 {
12   text-align: left;
13   padding-left: 1rem;
14   text-decoration: underline;
15 }
16
17 body {
18   margin: 0;
19 }
20
21 .footer {
22   background-color: #f2f2f2;
23   padding: 1rem;
24   margin-top: 5rem;
25   display: flex;
26   justify-content: space-around;
27   flex-wrap: wrap;
28   &__content {
29     padding: 1rem;
30   }
31 }
32
33 .data {
34   display: flex;
35   flex-direction: column;
36   align-items: center;
37
38   &__button {
39     background-color: $soft-grey;
40     max-width: 150px;
41     padding: 1rem;
42     margin: 1rem;
43     color: black;
44     @include shadow;
45     align-self: center;
46   }
47
48   &__container {
49     border: 2px solid;
50     border-radius: 10px;
51     margin: 1rem 0;
52     padding: 1rem;
53     width: 80%;
54
55     @include tablet {
56       width: 50%;
57     }
58   }
59 }
```

```
1
2 @mixin flex-column {
3     display: flex;
4     flex-wrap: wrap;
5     flex-direction: column;
6 }
7
8 @mixin desktop {
9     @media screen and (min-width: 1020px) {
10         @content;
11     }
12 }
13
14 @mixin tablet {
15     @media screen and (min-width: 420px) {
16         @content;
17     }
18 }
19
20 @mixin soft-typo {
21     font-size: 1rem;
22     background-color: $soft-grey;
23     padding: 1rem;
24     font-size: 1rem;
25     text-shadow: 1px 0 1px rgba(0, 0, 0, 0.3);
26 }
27
28 @mixin bold-typo {
29     font-size: 1rem;
30     font-weight: 800;
31     text-shadow: 1px 0 1px rgba(0, 0, 0, 0.3);
32     background-color: $mid-grey;
33     padding: 1rem;
34 }
35
36 @mixin shadow {
37     border-radius: 1rem;
38     box-shadow: 0.7rem 0.7rem 1rem rgb(200, 200, 200);
39     &:hover {
40         box-shadow: 0.7rem 0.7rem 1rem rgb(130, 130, 130);
41         transition: all 500ms;
42     }
43 }
```



```
1  $rouge: #ec4e48;
2  $bleu: #838bcc;
3  $vert: #3cd6b0;
4
5  $soft-grey: #f7f7f7;
6  $mid-grey: #ededed;
7  $hard-grey: #353535;
8
9  $fa-font-path: "../webfonts";
```

```
1  import Vue from 'vue'
2  import App from './App.vue'
3  import router from './router'
4  import store from './store'
5
6  Vue.config.productionTip = false
7
8  new Vue({
9    router,
10   store,
11   render: h => h(App)
12 }).$mount('#app')
```

```
1 <template>
2   <div id="app">
3     <router-view/>
4   </div>
5 </template>
6
7 <style lang="scss">
8   #app {
9     font-family: Avenir, Helvetica, Arial, sans-serif;
10    -webkit-font-smoothing: antialiased;
11    -moz-osx-font-smoothing: grayscale;
12    text-align: center;
13    color: #2c3e50;
14  }
15 </style>
```

```
1  <template>
2    <div>
3      <Header />
4      <Navbar />
5      <h2>Informations du compte</h2>
6      <div class="data">
7        <div class="data__container">
8          <section>
9            <div class="account__Message">
10              <p>
11                Prénom: <span class="precision-typo">{{ firstName }}</span>
12              </p>
13            </div>
14            <div class="account__Message">
15              <p>
16                Nom: <span class="precision-typo"> {{ lastName }} </span>
17              </p>
18            </div>
19            <div class="account__Message">
20              <p>
21                Email: <span class="precision-typo"> {{ email }} </span>
22              </p>
23            </div>
24            <div class="account__Message">
25              <p>
26                Identifiant utilisateur:
27                <span class="precision-typo"> {{ id }} </span>
28              </p>
29            </div>
30            <div v-if="isAdmin === 'false'" class="account__Message">
31              <p>Vous n'êtes pas administrateur du site.</p>
32              <p>
33                Vous pouvez le contacter par mail à l'adresse
34                <span class="precision-typo">admin@admin.fr</span>
35              </p>
36            </div>
37            <button @click="deleteMyAccount()" class="data__button">
38              Supprimer mon compte
39            </button>
40          </section>
41        </div>
42      </div>
43      <Footer />
44    </div>
45  </template>
46
47  <script>
48    import Header from "../components/Header.vue";
49    import Navbar from "../components/Navbar.vue";
50    import Footer from "@components/Footer.vue";
51
52    import axios from "axios";
53
54    export default {
55      name: "Account",
56      components: {
57        Header,
58        Navbar,
59        Footer,
60      },
61      data() {
62        return {
63          isAdmin: localStorage.getItem("isAdmin"),
64          firstName: "",
65          lastName: "",
66          email: "",
```

```
67     id: localStorage.getItem("userId"),
68   };
69 },
70
71   created() {
72     this.getAccount();
73   },
74   methods: {
75     getAccount() {
76       const id = localStorage.getItem("userId");
77
78       axios
79         .get("http://localhost:3000/api/user/" + id, {
80           headers: {
81             Authorization: "Bearer " + localStorage.getItem("token"),
82           },
83         })
84         .then((response) => {
85           this.email = response.data.email;
86           this.firstName = response.data.firstName;
87           this.lastName = response.data.lastName;
88         })
89
90         .catch((error) => {
91           console.log(error);
92         });
93     },
94
95     deleteMyAccount() {
96       const userId = localStorage.getItem("userId");
97
98       axios
99         .delete("http://localhost:3000/api/user/" + userId, {
100           headers: {
101             "Content-Type": "application/json",
102             Authorization: "Bearer " + localStorage.getItem("token"),
103           },
104         })
105         .then((res) => {
106           console.log("compte supprimé");
107           localStorage.clear();
108           alert("Compte supprimé");
109           this.$router.push("/");
110
111           return res.status(200).json({ message: "Compte supprimé" });
112         })
113         .catch((error) => {
114           console.log(error);
115         });
116     },
117   },
118 };
119 </script>
120
121 <style scoped lang="scss">
122   @import "../styles/main.scss";
123
124   .account {
125     background-color: #f2f2f2;
126     border: 3px solid;
127     border-radius: 10px;
128     margin: 1rem 0;
129     padding: 1rem;
130     width: 80%;
131
132     @include tablet {
```

```
133     width: 50%;
134   }
135
136   &__Message {
137     background-color: #f2f2f2;
138     border: 1px solid;
139     margin: 1rem 0;
140   }
141
142
143   }
144 </style>
```

```

1  <template>
2    <div>
3      <Header />
4      <Navbar />
5      <h2>Les derniers utilisateurs inscrits</h2>
6      <div class="data">
7        <div class="data__container" v-for="user in accounts" :key="user.id">
8          <div class="accounts--info--precision">
9            Prénom:
10             <span class="precision-typo">{{ user.firstName }}</span>
11          </div>
12          <div class="accounts--info--precision">
13            Nom:
14             <span class="precision-typo">{{ user.lastName }}</span>
15          </div>
16          <div class="accounts--info--precision">
17            Identifiant:
18             <span class="precision-typo">{{ user.id }} </span>
19          </div>
20          <div class="accounts--title">
21            Inscrit depuis le:
22             <span class="precision-typo">{{ user.createdAt }} </span>
23          </div>
24          <div class="accounts--content">
25            Email:
26
27             <span class="precision-typo">{{ user.email }} </span>
28          </div>
29          <button
30            v-if="isAdmin === 'true'"
31            @click="deleteAccount(user)"
32            class="data__button"
33          >
34            Supprimer le compte
35          </button>
36          <!-- <button
37            @click="getOneAccount(user)"
38            class="accounts--info--deletePost"
39          >
40            Voir le compte
41          </button> -->
42        </div>
43      </div>
44    </div>
45  </template>
46
47  <script>
48    import axios from "axios";
49    import Header from "@components/Header.vue";
50    import Navbar from "@components/Navbar.vue";
51
52    export default {
53      name: "AllAccounts",
54      components: {
55        Header,
56        Navbar,
57      },
58      data() {
59        return {
60          accounts: [],
61          user: "",
62          isAdmin: localStorage.getItem("isAdmin"),
63        };
64      },
65
66      created() {

```

```
67     this.getAllAccounts();
68   },
69   methods: {
70     getAllAccounts() {
71       axios
72         .get("http://localhost:3000/api/user/", {
73           headers: {
74             Authorization: "Bearer " + localStorage.getItem("token"),
75           },
76         })
77         .then((res) => {
78           console.log("ALLACCOUNTS consultés");
79
80           this.accounts = res.data;
81         })
82         .catch((error) => console.log(error));
83     },
84
85     // getOneAccount(user) {
86     //   axios
87     //     .get("http://localhost:3000/api/user/" + user.id, {
88     //       headers: {
89     //         Authorization: "Bearer " + localStorage.getItem("token"),
90     //       },
91     //     })
92     //     .then((response) => {
93     //       this.email = response.data.email;
94     //       this.firstName = response.data.firstName;
95     //       this.lastName = response.data.lastName;
96     //       this.userId = response.data.id;
97     //       this.$router.push('/user/' + user.id);
98     //     })
99     //     .catch((error) => {
100     //       console.log(error);
101     //     });
102     // },
103
104     deleteAccount(user) {
105       axios
106         .delete("http://localhost:3000/api/user/" + user.id, {
107           headers: {
108             "Content-Type": "application/json",
109             Authorization: "Bearer " + localStorage.getItem("token"),
110           },
111         })
112         .then((res) => {
113           console.log("Compte supprimé");
114           if (this.$route.path === "/allAccounts") {
115             window.location.reload();
116           } else this.$router.push("/allAccounts");
117           res.status(200).json({ message: "Post supprimé" });
118         })
119         .catch((error) => {
120           console.log(error);
121         });
122     },
123   },
124 },
125 };
126 </script>
127
128 <style scoped lang="scss">
129 @import "../styles/main.scss";
130
131 .accounts {
```



```
133
134   &--info {
135     @include flex-column;
136     font-weight: bold;
137     font-size: 1rem;
138     padding: 1rem;
139     color: $rouge;
140
141   }
142   &--title {
143     @include bold-typo;
144   }
145   &--content {
146     @include soft-typo;
147   }
148 }
149 </style>
```

```
1  <template>
2    <div>
3      <Header />
4      <Navbar />
5      <div class="notOnMobile">
6        <div class="notOnMobile__title"><h2>Poster votre article</h2></div>
7        <div class="notOnMobile__createPost"><CreatePost /></div>
8      </div>
9      <h2>Les derniers articles publiés</h2>
10
11    <section class="data">
12      <div class="data__container" v-for="post in posts" :key="post.id">
13        <div class="posts--info">
14          <div class="posts--info--precision">
15            Article numéro:
16            <span class="precision-tylo">{{ post.id }} </span>
17          </div>
18          <div class="posts--info--precision">
19            Posté par:
20            <span class="precision-tylo">{{ post.userId }}</span>
21          </div>
22          Le:
23          <span class="precision-tylo">{{ post.createdAt }} </span>
24        </div>
25      </div>
26      <div class="posts--title">{{ post.title }}</div>
27      <div class="posts--content">{{ post.content }}</div>
28      <button
29        v-if="isAdmin === 'true' || post.userId == userId"
30        @click="deletePost(post)"
31        class="data__button"
32      >
33        Supprimer
34      </button>
35      <!-- @click="showComment = !showComment"
36        <button
37
38          @click="getAllComments(post)"
39          class="show-comment"
40        >
41          Voir les commentaires
42        </button>
43        <div v-show="showComment" class="comment">
44          Affichage?
45        </div> -->
46    </div>
47  </section>
48 </div>
49 </template>
50
51 <script>
52 import Header from "@components/Header.vue";
53 import Navbar from "@components/Navbar.vue";
54 import CreatePost from "../components/CreatePost.vue";
55
56 import axios from "axios";
57
58 export default {
59   name: "AllPosts",
60   components: {
61     Header,
62     CreatePost,
63     Navbar,
64   },
65   data() {
66     return {
```

```
67     posts: [],
68     post: "",
69     isAdmin: localStorage.getItem("isAdmin"),
70     id: "",
71     userId: localStorage.getItem("userId"),
72     //showComment: false,
73     //comments: [],
74   };
75 },
76
77   created() {
78     this.getAllPosts();
79     //this.getAllComments();
80   },
81
82   methods: {
83     // getAllComments(post) {
84     //   const userId = localStorage.getItem("userId");
85     //   console.log("userId", userId);
86     //   console.log("post", post.id);
87
88     //   axios
89     //     .get("http://localhost:3000/api/comment/", +post.id, {
90     //       headers: {
91     //         Authorization: "Bearer " + localStorage.getItem("token"),
92     //       },
93     //       body: {
94     //         userId: userId,
95     //       },
96     //     })
97     //     .then((res) => {
98     //       console.log("COMMENTS consultés");
99     //       this.comments = res.data;
100    //     })
101
102    //     .catch((error) => {
103    //       console.log(error);
104    //     });
105    //   },
106
107    getAllPosts() {
108      axios
109        .get("http://localhost:3000/api/post/", {
110          headers: {
111            Authorization: "Bearer " + localStorage.getItem("token"),
112          },
113        })
114        .then((res) => {
115          console.log("ALLPOSTS consultés");
116          this.posts = res.data;
117        })
118
119        .catch((error) => {
120          console.log(error);
121        });
122    },
123
124    deletePost(post) {
125      axios
126        .delete("http://localhost:3000/api/post/" + post.id, {
127          headers: {
128            "Content-Type": "application/json",
129            Authorization: "Bearer " + localStorage.getItem("token"),
130          },
131        })
132        .then((res) => {
```

```
133         console.log("Post supprimé");
134         if (this.$route.path === "/allPosts") {
135             window.location.reload();
136         } else this.$router.push("/allPosts");
137         res.status(200).json({ message: "Post supprimé" });
138     })
139     .catch((error) => {
140         console.log(error);
141     });
142     },
143 };
144 };
145 </script>
146
147 <style lang="scss">
148 @import "./styles/main.scss";
149
150 .notOnMobile {
151     display: none;
152
153     @include desktop {
154         display: flex;
155         flex-direction: column;
156
157         &__createPost {
158             display: flex;
159             justify-content: center;
160         }
161     }
162 }
163
164 .posts {
165     &--info {
166         @include flex-column;
167         font-weight: bold;
168         font-size: 1rem;
169         padding: 1rem;
170         color: $rouge;
171     }
172     &--title {
173         @include bold-typo;
174     }
175     &--content {
176         @include soft-typo;
177     }
178 }
179
180 .precision-typo {
181     @include bold-typo;
182     padding: 0;
183     color: black;
184 }
185 </style>
```

```
1 <template>
2   <div>
3     <Header />
4     <div class="home">
5       <div class="welcomeAnim">
6         
11        
16      </div>
17    </div>
18    <section>
19      <div class="welcomeMessage">
20        Groupomania vous souhaite une Bonne Année 2022! Nous avons pour cette
21        occasion le plaisir de vous proposer un réseau social afin que chaque
22        collaborateur puisse interagir avec les autres, partager, proposer...
23        N'attendez plus pour vous inscrire et commencer à en profiter!
24      </div>
25    </section>
26    <Footer />
27  </div>
28 </template>
29
30 <script>
31 // @ is an alias to /src
32 import Header from "@components/Header.vue";
33 import Footer from "@components/Footer.vue";
34
35 export default {
36   name: "Home",
37   components: {
38     Header,
39     Footer,
40   },
41 };
42 </script>
43
44 <style scoped lang="scss">
45 @import "../../styles/main.scss";
46
47 .welcomeAnim {
48   @include desktop {
49     display: flex;
50     justify-content: space-around;
51   }
52 }
53
54 .welcomeMessage {
55   font-size: 1rem;
56   font-weight: bold;
57   margin: 0 2rem 2rem 2rem;
58
59   @include tablet {
60     font-size: 1.5rem;
61     margin: 0 5rem 5rem 2rem;
62   }
63 }
64
65 .home {
66   &__bigLogo {
```

```
67     display: none;
68     @include desktop {
69         max-width: 400px;
70         width: 50%;
71         margin: 5rem;
72         display: flex;
73         animation: spinner-rotate-logo 3000ms linear infinite;
74     }
75 }
76 &__smallLogo {
77     width: 100%;
78     max-height: 200px;
79     animation: spinner-opacity2-logo 4000ms linear infinite;
80
81     @include tablet {
82         animation: spinner-opacity-logo 3000ms linear infinite;
83         max-width: 400px;
84         width: 25%;
85         align-self: center;
86     }
87 }
88 }
89
90 @keyframes spinner-rotate-logo {
91     0% {
92         transform: rotate(0deg);
93     }
94
95     100% {
96         transform: rotate(360deg);
97     }
98 }
99
100 @keyframes spinner-opacity-logo {
101     0% {
102         opacity: 0;
103     }
104
105     50% {
106         opacity: 1;
107     }
108
109     100% {
110     }
111 }
112 @keyframes spinner-opacity2-logo {
113     0% {
114         opacity: 0;
115     }
116
117     100% {
118         opacity: 1;
119     }
120 }
121 </style>
```

```
1  <template>
2    <div>
3      <Header />
4      <div class="login">
5        <form @submit.prevent="login" class="login_form">
6          <div class="login_form_input">
7            <label for="email" class="login_form_input_label">Email</label>
8            <input type="email" v-model="email" id="email" name="email" />
9          </div>
10
11          <div class="login_form_input">
12            <label for="password" class="login_form_input_label">
13              >Mot de passe</label>
14            >
15            <input
16              type="password"
17              v-model="password"
18              id="password"
19              name="password"
20            />
21          </div>
22
23          <button class="login_form_button">Valider</button>
24        </form>
25      </div>
26      <Footer />
27    </div>
28  </template>
29
30  <script>
31    // @ is an alias to /src
32    import Header from "@components/Header.vue";
33    import Footer from "@components/Footer.vue";
34    import axios from "axios";
35
36    export default {
37      name: "Login",
38      components: {
39        Header,
40        Footer,
41      },
42      data() {
43        return {
44          email: "",
45          password: "",
46        };
47      },
48      methods: {
49        login() {
50          axios
51            .post("http://localhost:3000/api/user/login", {
52              email: this.email,
53              password: this.password,
54            })
55            .then((res) => {
56              localStorage.setItem("userId", res.data.userId);
57              localStorage.setItem("token", res.data.token);
58              localStorage.setItem("isAdmin", res.data.isAdmin); // faire une méthode
59              isAdminByToken ?
60                this.$router.push("/allPosts");
61            })
62            .catch((error) => {
63              console.log(error);
64            });
65        },
66      },
67    };
68  </script>
```

```
66   };
67 </script>
68
69 <style scoped lang="scss">
70 @import "../styles/main.scss";
71
72 .login {
73   display: flex;
74   justify-content: center;
75   align-items: center;
76   margin: 3rem 0 15rem 0; // voir pour une position absolu du footer? également sur
    toutes les pages.
77
78   @include desktop {
79     margin-top: 10rem;
80   }
81
82   &__form {
83     @include flex-column;
84     background-color: #f2f2f2;
85     align-items: center;
86     width: 85%;
87     border: 2px solid #3f3d56;
88     border-radius: 5px;
89     padding: 1.5rem 0;
90
91     @include desktop {
92       width: 50%;
93       padding: 3rem 0;
94       border: 3px solid #3f3d56;
95     }
96
97     &__input {
98       display: flex;
99       flex-direction: column;
100      margin-bottom: 2rem;
101      width: 70%;
102      &__label {
103        text-align: start;
104        font-weight: bold;
105        margin-bottom: 0.5rem;
106      }
107    }
108    &__button {
109      border: 1px solid $hard-grey;
110      border-radius: 5px;
111      font-weight: bold;
112      padding: 0.9rem;
113      &:hover,
114      &:focus {
115        border: 1px solid $hard-grey;
116        background-color: $rouge;
117        cursor: pointer;
118      }
119    }
120  }
121 }
122 </style>
```



```
1  <template>
2    <div>
3      <Header />
4      <Navbar />
5      <h2>Poster votre article</h2>
6      <div class="data">
7        <CreatePost />
8      </div>
9      <Footer />
10   </div>
11 </template>
12
13 <script>
14 import Header from "../components/Header.vue";
15 import Navbar from "../components/Navbar.vue";
16 import CreatePost from "../components/CreatePost.vue";
17 import Footer from "@components/Footer.vue";
18
19
20 export default {
21   name: "Post",
22   components: {
23     Header,
24     Navbar,
25     CreatePost,
26     Footer
27   },
28 };
29 </script>
30
31 <style lang="scss">
32 @import "../styles/main.scss";
33
34
35 </style>
```

```
1 <template>
2   <div>
3     <Header />
4     <div class="signup">
5       <form @submit.prevent="signup" class="signup_form">
6         <div class="signup_form_input">
7           <label for="firstName" class="signup_form_input_label">
8             Prénom</label>
9           <input
10             type="text"
11             v-model="firstName"
12             id="firstName"
13             name="firstName"
14           />
15         </div>
16
17         <div class="signup_form_input">
18           <label for="lastName" class="signup_form_input_label">Nom</label>
19           <input type="text" v-model="lastName" id="lastName" name="lastName" />
20         </div>
21
22         <div class="signup_form_input">
23           <label for="email" class="signup_form_input_label">Email</label>
24           <input type="email" v-model="email" id="email" name="email" />
25         </div>
26
27         <div class="signup_form_input">
28           <label for="password" class="signup_form_input_label">
29             Mot de passe</label>
30           <input
31             type="password"
32             v-model="password"
33             id="password"
34             name="password"
35           />
36         </div>
37
38         <button class="signup_form_button">Valider</button>
39       </form>
40     </div>
41     <Footer />
42   </div>
43 </template>
44
45 <script>
46 // @ is an alias to /src
47 import Header from "@components/Header.vue";
48 import Footer from "@components/Footer.vue";
49 import axios from "axios";
50
51 export default {
52   name: "Signup",
53   components: {
54     Header,
55     Footer,
56   },
57   data() {
58     return {
59       firstName: "",
60       lastName: "",
61       email: "",
62       password: "",
63     };
64   },
65 };
66
```

```
67     methods: {
68       signup() {
69         axios
70           .post("http://localhost:3000/api/user/signup", {
71             firstName: this.firstName,
72             lastName: this.lastName,
73             email: this.email,
74             password: this.password,
75           })
76           .then(() => {
77             this.$router.push("/login");
78           })
79           .catch((error) => {
80             console.log(error);
81           });
82       },
83     },
84   };
85 </script>
86
87 <style scoped lang="scss">
88 @import "./styles/main.scss";
89
90 .signup {
91   display: flex;
92   justify-content: center;
93   align-items: center;
94   margin: 3rem 0 5rem 0;
95
96   @include desktop {
97     margin-top: 10rem;
98   }
99
100   &__form {
101     @include flex-column;
102     background-color: #f2f2f2;
103     align-items: center;
104     width: 85%;
105     border: 2px solid #3f3d56;
106     border-radius: 5px;
107     padding: 1.5rem 0;
108
109     @include desktop {
110       width: 50%;
111       padding: 3rem 0;
112       border: 3px solid #3f3d56;
113     }
114
115     &__input {
116       display: flex;
117       flex-direction: column;
118       margin-bottom: 2rem;
119       width: 70%;
120       &__label {
121         text-align: start;
122         font-weight: bold;
123         margin-bottom: 0.5rem;
124       }
125     }
126
127     &__button {
128       border: 1px solid $hard-grey;
129       border-radius: 5px;
130       font-weight: bold;
131       padding: 0.9rem;
132       &:hover,
133       &:focus {
```

```
133         border: 1px solid $hard-grey;
134         background-color: $rouge;
135         cursor: pointer;
136     }
137 }
138 }
139 }
140 </style>
```

```
1  import Vue from 'vue'
2  import Vuex from 'vuex'
3
4  Vue.use(Vuex)
5
6  export default new Vuex.Store({
7    state: {
8      status: "Non connecté",
9      organismeDeFormation: "OpenClassRooms",
10
11    },
12    mutations: {
13    },
14    actions: {
15    },
16    getters: {
17      copyright: (state) => {
18        const currentYear = new Date().getFullYear()
19
20        return `Copyright ${state.organismeDeFormation} $
21      {currentYear}`
22    },
23
24    },
25    modules: {
26    }
27  })
```

```
1  import Vue from 'vue'
2  import VueRouter from 'vue-router'
3  import Home from '../views/Home.vue'
4
5  Vue.use(VueRouter)
6
7  const routes = [
8    {
9      path: '/',
10     name: 'Home',
11     component: Home
12   },
13   {
14     path: '/signup',
15     name: 'Signup',
16     component: () => import('../views/Signup.vue')
17   },
18   {
19     path: '/login',
20     name: 'Login',
21     component: () => import('../views/Login.vue')
22   },
23   {
24     path: '/user/:id',
25     name: 'Account',
26     component: () => import('../views/Account.vue')
27   },
28   {
29     path: '/allAccounts',
30     name: 'AllAccounts',
31     component: () => import('../views/AllAccounts.vue')
32   },
33   {
34     path: '/post',
35     name: 'Post',
36     component: () => import('../views/Post.vue')
37   },
38   {
39     path: '/allPosts',
40     name: 'AllPosts',
41     component: () => import('../views/AllPosts.vue')
42   }
43 ]
44
45
46 const router = new VueRouter({
47   routes
48 })
49
50 export default router
```

```
1 <template>
2   <div class="data__container">
3     <form @submit.prevent="createPost" class="createpost__items">
4       <label> Titre: <input v-model="title" /> </label>
5       <br /><br />
6       <label> Contenu: </label>
7       <br /><br />
8       <textarea
9         v-model="content"
10        placeholder="Que souhaitez vous partager?"
11        rows="5"
12        cols="5"
13      ></textarea>
14
15      <input type="file" ref="file" @change="uploadPhoto" />
16      <div v-if="imageUrl.length > 0">
17        
18      </div>
19      <button class="createpost data__button" type="submit">Publier</button>
20    </form>
21  </div>
22 </template>
23
24 <script>
25 import axios from "axios";
26
27 export default {
28   name: "CreatePost",
29   components: {},
30   data() {
31     return {
32       title: this.title,
33       content: this.content,
34       userId: "",
35       image: "",
36       imageUrl: "",
37     };
38   },
39
40   methods: {
41     uploadPhoto(event) {
42       this.image = event.target.files[0];
43       console.log("image", this.image);
44       this.imageUrl = URL.createObjectURL(this.image);
45       console.log("imageUrl", this.imageUrl);
46     },
47
48     createPost() {
49       const formData = new FormData();
50
51       formData.append("title", this.title);
52       formData.append("content", this.content);
53       formData.append("attachment", this.image);
54
55       axios
56         .post("http://localhost:3000/api/post", formData, {
57           headers: {
58             "Content-Type": "multipart/form-data",
59             Authorization: "Bearer " + localStorage.getItem("token"),
60           },
61         })
62         .then((res) => {
63           console.log("POST effectué");
64           if (this.$route.path === "/allPosts") {
65             window.location.reload();
66           }
67         });
68     }
69   }
70 }
```

```
67         } else this.$router.push("/allPosts");
68         res.status(200).json({ message: "Post effectué" });
69     })
70     .catch((error) => {
71         console.log(error);
72     });
73     },
74 },
75 };
76 </script>
77
78 <style lang="scss" scoped>
79 @import "../styles/main.scss";
80
81 .createpost {
82     &__items {
83         @include flex-column;
84     }
85 }
86 </style>
```



```
1 <template>
2   <footer class="footer">
3     <div class="footer__content">A propos</div>
4     <div class="footer__content">{{ copyright }}</div>
5     <div class="footer__content">Assistance</div>
6   </footer>
7 </template>
8
9 <script>
10  import { mapGetters, mapState } from "vuex";
11
12  export default {
13    name: "Footer",
14
15    computed: {
16      ...mapGetters({
17        copyright: "copyright",
18      }),
19      ...mapState({
20        organismeDeFormation: "OpenClassRooms",
21      }),
22    },
23  }
24 </script>
25
26 <style lang="scss">
27  @import "../styles/main.scss";
28 </style>
```

```
1 <template>
2   <div class="navbar">
3     <router-link to="/">
4       
9     </router-link>
10
11     
16     <div class="navbar__container">
17       <router-link v-if="!id" class="navbar__container--link" to="/signup">
18         S'inscrire</router-link>
19       >
20
21       <router-link v-if="!id" class="navbar__container--link" to="/login">
22         Se connecter</router-link>
23       >
24       <a
25         v-if="id"
26         class="navbar__container--link"
27         :href="'http://localhost:8080/#/user/' + id"
28         >Mon compte</a>
29       >
30       <button
31         v-if="id"
32         @click="showModal = !showModal"
33         class="navbar__container--link"
34       >
35         Déconnexion
36       </button>
37       <LogoutModale v-show="showModal" class="modal" />
38     </div>
39   </div>
40 </template>
41
42 <script>
43 import LogoutModale from "@components/LogoutModale.vue";
44
45 export default {
46   name: "Navbar",
47   components: {
48     LogoutModale,
49   },
50
51   data: () => ({
52     id: localStorage.getItem("userId"),
53     showModal: false,
54   }),
55 };
56 </script>
57
58 <style lang="scss">
59 @import "../../styles/main.scss";
60
61 .navbar {
62   display: flex;
63   align-items: center;
64   justify-content: space-between;
65   flex-wrap: wrap;
66 }
```

```
67     &__container {
68         display: flex;
69         flex-direction: column;
70
71         @include tablet {
72             flex-direction: row;
73         }
74
75         &--Link {
76             color: black;
77             text-decoration: none;
78             font-weight: bold;
79             padding: 0.5rem;
80
81             @include tablet {
82                 margin: 1rem 0;
83             }
84
85             @include desktop {
86                 margin: 1rem;
87                 padding: 0.9rem;
88             }
89
90             &:hover,
91             &:focus {
92                 cursor: pointer;
93                 border: 1px;
94                 background-color: $rouge;
95                 border-radius: 5px;
96             }
97         }
98     }
99     &__smallLogo {
100         max-height: 100px;
101     }
102 }
103 </style>
```

```
1 <template>
2   <div>
3     <div>Il est {{ heureCourante() }}</div>
4   </div>
5 </template>
6
7 <script>
8   export default {
9     name: "HeureCourante",
10
11     created() {
12       setInterval(() => this.$forceUpdate(), 10000); //tester avec le statut connecté
13     },
14     methods: {
15       heureCourante() {
16         let now = new Date();
17         let heure = ("0" + now.getHours()).slice(-2);
18         let minute = ("0" + now.getMinutes()).slice(-2);
19         return `${heure}h${minute}`;
20       },
21     },
22   };
23 </script>
24
25 <style lang="scss">
26   @import "../styles/main.scss";
27
28
29 </style>
```

```
1 <template>
2   <div id="myModal" class="modal">
3     <div class="modal-content">
4       <div class="modalLineOne">
5         <div>Etes-vous certain de vouloir vous déconnecter?</div>
6         <span @click="closeModal" class="close">&times;</span>
7       </div>
8       <div class="modalLineTwo">
9         <button @click="disconnect">OUI</button>
10        <button @click="closeModal">NON</button>
11      </div>
12    </div>
13  </div>
14 </template>
15
16 <script>
17 export default {
18   name: "LogoutModale",
19
20   methods: {
21     disconnect() {
22       if (this.$route.path === "/") {
23         localStorage.clear();
24         window.location.reload();
25       } else localStorage.clear();
26       this.$router.push("/");
27     },
28
29     closeModal() {
30       var modal = document.getElementById("myModal");
31       modal.style.display = "none";
32     },
33   },
34 };
35 </script>
36
37 <style scoped lang="scss">
38 @import "./styles/main.scss";
39
40
41 .modal {
42
43   position: fixed;
44   z-index: 1;
45   left: 0;
46   top: 0;
47   width: 100%;
48   height: 100%;
49   overflow: auto;
50   background-color: rgb(0, 0, 0);
51   background-color: rgba(0, 0, 0, 0.4);
52 }
53
54
55 .modal-content {
56   background-color: #fefefe;
57   margin: 15% auto;
58   padding: 20px;
59   border: 1px solid #888;
60   width: 80%;
61   color: black;
62   font-weight: bold;
63 }
64
65
66 .close {
```

```
67     color: #aaa;
68     float: right;
69     font-size: 28px;
70     font-weight: bold;
71   }
72
73   .close:hover,
74   .close:focus {
75     color: black;
76     text-decoration: none;
77     cursor: pointer;
78   }
79
80   .modalLineOne {
81     display: flex;
82     justify-content: space-between;
83   }
84
85   .modalLineOne div {
86     align-self: center;
87   }
88
89   .modalLineTwo {
90     display: flex;
91     justify-content: space-around;
92   }
93 </style>
```

```
1 <template>
2   <div class="options">
3     <div class="options__container--clock">
4       <HeureCourante />
5     </div>
6     <a
7       class="options__container--link"
8       :href="'http://localhost:8080/#/user/' + id"
9     >Gérer mon compte</a>
10   >
11   <router-link class="options__container--link" to="/post"
12     >Poster un article</router-link>
13   >
14   <router-link class="options__container--link" to="/allPosts"
15     >Tous les articles</router-link>
16   >
17   <router-link class="options__container--link" to="/allAccounts"
18     >Tous les utilisateurs</router-link>
19   >
20 </div>
21 </template>
22
23 <script>
24 import HeureCourante from "@components/HeureCourante.vue";
25
26 export default {
27   name: "Navbar",
28   components: {
29     HeureCourante,
30   },
31
32   data: () => ({
33     id: localStorage.getItem("userId"),
34   }),
35 };
36 </script>
37
38 <style scoped lang="scss">
39 @import "../styles/main.scss";
40
41 .options {
42   background-color: #f2f2f2;
43   display: flex;
44   justify-content: center;
45   flex-wrap: wrap;
46   width: 100%;
47
48   &__container {
49
50     &--link {
51       color: black;
52       text-decoration: none;
53       margin: 0.3rem;
54       font-weight: bold;
55       padding: 0.5rem;
56
57       @include tablet {
58         margin: 1rem;
59         padding: 0.9rem;
60       }
61     }
62   }
63
64   &--clock {
65     color: black;
66     margin: 0.3rem;
67     padding: 0.5rem;
```

```
67     @include tablet {
68       margin: 1rem;
69       padding: 0.9rem;
70     }
71   }
72 }
73 }
74
75 p {
76   margin: 0.5rem;
77 }
78 </style>
```



```
1  const express = require('express');
2  const path = require('path');
3
4  require('dotenv').config();
5
6  const app = express();
7
8  const userRoutes = require('./routes/user');
9  const postRoutes = require('./routes/post');
10 const commentRoutes = require('./routes/comment');
11
12 app.use((req, res, next) => {
13   res.setHeader('Access-Control-Allow-Origin', '*');
14   res.setHeader('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content,
Accept, Content-Type, Authorization');
15   res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, PATCH,
OPTIONS');
16   next();
17 });
18
19 app.use(express.json()); // Permet de ne plus utiliser body-parser qui est inclut à
présent dans Express
20 app.use(express.urlencoded({ extended: true }));
21
22 app.use('/images', express.static(path.join(__dirname, 'images')));
23
24 app.use('/api/user', userRoutes);
25 app.use('/api/post', postRoutes);
26 app.use('/api/comment', commentRoutes);
27
28
29 module.exports = app;
```

```
1  const http = require('http');
2  const app = require('./app');
3
4  const normalizePort = val => {
5    const port = parseInt(val, 10);
6
7    if (isNaN(port)) {
8      return val;
9    }
10   if (port >= 0) {
11     return port;
12   }
13   return false;
14 };
15 const port = normalizePort(process.env.PORT || '3000');
16 app.set('port', port);
17
18 const errorHandler = error => {
19   if (error.syscall !== 'listen') {
20     throw error;
21   }
22   const address = server.address();
23   const bind = typeof address === 'string' ? 'pipe ' + address : 'port: ' + port;
24   switch (error.code) {
25     case 'EACCES':
26       console.error(bind + ' requires elevated privileges. ');
27       process.exit(1);
28       break;
29     case 'EADDRINUSE':
30       console.error(bind + ' is already in use. ');
31       process.exit(1);
32       break;
33     default:
34       throw error;
35   }
36 };
37
38 const server = http.createServer(app);
39
40 server.on('error', errorHandler);
41 server.on('listening', () => {
42   const address = server.address();
43   const bind = typeof address === 'string' ? 'pipe ' + address : 'port ' + port;
44   console.log('Listening on ' + bind);
45 });
46
47 server.listen(port);
```

```
1  {
2    "name": "backend",
3    "version": "1.0.0",
4    "description": "",
5    "main": "server.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "dependencies": {
12     "bcryptjs": "^2.4.3",
13     "dotenv": "^10.0.0",
14     "express": "^4.17.2",
15     "jsonwebtoken": "^8.5.1",
16     "multer": "^1.4.4",
17     "mysql2": "^2.3.3",
18     "sequelize": "^7.0.0-alpha.3"
19   },
20   "devDependencies": {
21     "nodemon": "^2.0.15",
22     "sequelize-cli": "^6.3.0"
23   }
24 }
```

```
1  const express = require('express');
2  const router = express.Router();
3  const userCtrl = require('../controllers/user');
4
5  const getAuth = require('../middleware/getAuth');
6  const putDeleteAuth = require('../middleware/putDeleteAuth');
7
8  const multer = require('../middleware/multer-config');
9
10 router.post('/signup', userCtrl.signup);
11 router.post('/login', userCtrl.login);
12
13 router.get('/', getAuth, userCtrl.getAllAccounts);
14 router.get('/:id', getAuth, userCtrl.getAccount);
15
16 router.delete('/:id', getAuth, putDeleteAuth, userCtrl.deleteAccount);
17
18 //router.get('/:id', getAuth, userCtrl.getOneAccount); //facultatif
19 //router.put('/:id', userCtrl.modifyAccount); //facultatif
20
21
22 module.exports = router;
```

```
1  const express = require('express');
2  const router = express.Router();
3  const postCtrl = require('../controllers/post');
4
5  const getAuth = require('../middleware/getAuth');
6  const postAuth = require('../middleware/postAuth');
7  const putDeleteAuth = require('../middleware/putDeleteAuth');
8  const multer = require('../middleware/multer-config');
9
10 router.post('/', postAuth, multer, postCtrl.createPost);
11
12 router.get('/', getAuth, postCtrl.getAllPosts);
13 router.get('/:id', getAuth, postCtrl.getOnePost);
14
15 router.delete('/:id', getAuth, putDeleteAuth, postCtrl.deletePost);
16
17 //router.put('/:id', postCtrl.modifyPost); //facultatif
18
19
20
21 module.exports = router;
```

```
1  const express = require('express');
2  const router = express.Router();
3  const commentCtrl = require('../controllers/comment');
4
5  const getAuth = require('../middleware/getAuth');
6  const putDeleteAuth = require('../middleware/putDeleteAuth');
7
8  //router.post('/:postId', getAuth, commentCtrl.createComment); // en cours
9
10 //router.get('/:postId', getAuth, commentCtrl.getAllComments); // en cours
11
12 //router.delete('/:commentId', putDeleteAuth, commentCtrl.deleteComment); // en cours
13
14
15 module.exports = router;
```

```
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    class user extends Model {
7      /**
8       * Helper method for defining associations.
9       * This method is not a part of Sequelize lifecycle.
10      * The `models/index` file will call this method automatically.
11      */
12      static associate(models) {
13        // define association here
14
15        models.user.hasMany(models.post);
16        models.user.hasMany(models.comment);
17      }
18    };
19    user.init({
20      firstName: DataTypes.STRING,
21      lastName: DataTypes.STRING,
22      email: DataTypes.STRING,
23      password: DataTypes.STRING,
24      isAdmin: DataTypes.BOOLEAN,
25      isActive: DataTypes.BOOLEAN,
26      keepConnected: DataTypes.BOOLEAN
27    }, {
28      sequelize,
29      modelName: 'user',
30    });
31    return user;
32  };
```

```
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    class post extends Model {
7      /**
8       * Helper method for defining associations.
9       * This method is not a part of Sequelize lifecycle.
10      * The `models/index` file will call this method automatically.
11      */
12      static associate(models) {
13        // define association here
14        models.post.belongsTo(models.user, {
15          foreignKey: 'userId'
16        })
17        models.post.hasMany(models.comment);
18      }
19    };
20    post.init({
21      userId: DataTypes.INTEGER,
22      title: DataTypes.STRING,
23      content: DataTypes.STRING,
24      attachment: DataTypes.STRING
25    }, {
26      sequelize,
27      modelName: 'post',
28    });
29    return post;
30  };
```



```
1  'use strict';
2
3  const fs = require('fs');
4  const path = require('path');
5  const Sequelize = require('sequelize');
6  const basename = path.basename(__filename);
7  const env = process.env.NODE_ENV || 'development';
8  const config = require(__dirname + '/../config/config.json')[env];
9  const db = {};
10
11  let sequelize;
12  if (config.use_env_variable) {
13    sequelize = new Sequelize(process.env[config.use_env_variable], config);
14  } else {
15    sequelize = new Sequelize(config.database, config.username, config.password,
16    config);
17  }
18  fs
19    .readdirSync(__dirname)
20    .filter(file => {
21      return (file.indexOf('.') !== 0) && (file !== basename) && (file.slice(-3) ===
22      '.js');
23    })
24    .forEach(file => {
25      const model = require(path.join(__dirname, file))(sequelize,
26      Sequelize.DataTypes);
27      db[model.name] = model;
28    });
29
30  Object.keys(db).forEach(modelName => {
31    if (db[modelName].associate) {
32      db[modelName].associate(db);
33    }
34  });
35
36  db.sequelize = sequelize;
37  db.Sequelize = Sequelize;
38
39  module.exports = db;
```

```
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    class comment extends Model {
7      /**
8       * Helper method for defining associations.
9       * This method is not a part of Sequelize lifecycle.
10      * The `models/index` file will call this method automatically.
11      */
12      static associate(models) {
13        // define association here
14      }
15    };
16    comment.init({
17      userId: DataTypes.INTEGER,
18      postId: DataTypes.INTEGER,
19      content: DataTypes.STRING
20    }, {
21      sequelize,
22      modelName: 'comment',
23    });
24    return comment;
25  };
```

```
1  'use strict';
2  module.exports = {
3    up: async (queryInterface, Sequelize) => {
4      await queryInterface.createTable('comments', {
5        id: {
6          allowNull: false,
7          autoIncrement: true,
8          primaryKey: true,
9          type: Sequelize.INTEGER
10       },
11       userId: {
12         type: Sequelize.INTEGER,
13         allowNull: false,
14         references: {
15           model: "users",
16           key: "id",
17         },
18       },
19       postId: {
20         type: Sequelize.INTEGER,
21         allowNull: false,
22         references: {
23           model: "posts",
24           key: "id",
25         },
26       },
27       content: {
28         type: Sequelize.STRING,
29         allowNull: false,
30       },
31       createdAt: {
32         allowNull: false,
33         type: Sequelize.DATE
34       },
35       updatedAt: {
36         allowNull: false,
37         type: Sequelize.DATE
38       }
39     });
40   },
41   down: async (queryInterface, Sequelize) => {
42     await queryInterface.dropTable('comments');
43   }
44 };

```

```
1  'use strict';
2  module.exports = {
3    up: async (queryInterface, Sequelize) => {
4      await queryInterface.createTable('posts', {
5        id: {
6          allowNull: false,
7          autoIncrement: true,
8          primaryKey: true,
9          type: Sequelize.INTEGER
10       },
11       userId: {
12         type: Sequelize.INTEGER,
13         allowNull: false,
14         references: {
15           model: "users",
16           key: "id",
17         },
18       },
19       title: {
20         type: Sequelize.STRING,
21         allowNull: false,
22       },
23       content: {
24         type: Sequelize.STRING,
25         allowNull: false,
26       },
27       attachment: {
28         type: Sequelize.STRING,
29         allowNull: true,
30       },
31     },
32     createdAt: {
33       allowNull: false,
34       type: Sequelize.DATE
35     },
36     updatedAt: {
37       allowNull: false,
38       type: Sequelize.DATE
39     }
40   });
41 },
42 down: async (queryInterface, Sequelize) => {
43   await queryInterface.dropTable('posts');
44 }
45 };
```

```
1  'use strict';
2  module.exports = {
3    up: async (queryInterface, Sequelize) => {
4      await queryInterface.createTable('users', {
5        id: {
6          allowNull: false,
7          autoIncrement: true,
8          primaryKey: true,
9          type: Sequelize.INTEGER
10       },
11       firstName: {
12         allowNull: false,
13         type: Sequelize.STRING
14       },
15       lastName: {
16         allowNull: false,
17         type: Sequelize.STRING
18       },
19       email: {
20         allowNull: false,
21         type: Sequelize.STRING
22       },
23       password: {
24         allowNull: false,
25         type: Sequelize.STRING
26       },
27       isAdmin: {
28         allowNull: false,
29         type: Sequelize.BOOLEAN,
30         defaultValue: false
31       },
32       isActive: {
33         allowNull: false,
34         type: Sequelize.BOOLEAN,
35         defaultValue: false
36       },
37       keepConnected: {
38         allowNull: false,
39         type: Sequelize.BOOLEAN,
40         defaultValue: false
41       },
42       createdAt: {
43         allowNull: false,
44         type: Sequelize.DATE
45       },
46       updatedAt: {
47         allowNull: false,
48         type: Sequelize.DATE
49       }
50     });
51   },
52   down: async (queryInterface, Sequelize) => {
53     await queryInterface.dropTable('users');
54   }
55 };
```

```

1  const jwt = require("jsonwebtoken");
2
3  module.exports = (req, res, next) => {
4    console.log(
5      "
6    );
7
8    console.log(
9      "          PARTIE COMMUNE, CONCERNE LA
10     REQUETE
11   );
12   if (req.headers.authorization === "" || !req.headers.authorization) {
13     console.log("pas le cas req.headers.authorization");
14     console.log(req.body.headers.Authorization);
15   } else {
16     if (req.params.id === "" || !req.params.id) {
17       console.log("pas le cas req.params.id");
18
19       try {
20         console.log(
21           "
22         );
23
24         console.log(
25           "          AUTH (en cours de vérification du NIVEAU)
26         );
27         const token = req.headers.authorization.split(" ")[1];
28         const decodedToken = jwt.verify(token, process.env.TOKEN);
29         console.log("decodedToken", decodedToken);
30         const isAdmin = decodedToken.isAdmin;
31         console.log(
32           "
33         );
34         if (isAdmin) {
35           console.log(
36             "
37           );
38
39           console.log(
40             "          GET ACCESS => TOKEN OK_(ADMIN
41     ACCESS)
42           );
43           next();
44         } else {
45           console.log(
46             "
47           );
48           console.log(
49             "          GET ACCESS => TOKEN OK_(USER
50     ACCESS)
51           );
52           next();
53         }
54       } catch (error) {
55         res.status(403).json({ error: error | "Unauthorized request" });
56       }
57     }

```

```
58     if (req.params.id) {
59         console.log("cas req.params.id , req.params.id =", req.params.id);
60         next();
61     }
62 }
63 };
```

```
1  const multer = require('multer');
2
3  const MIME_TYPES = { // Dictionnaire mime types pour ceux disponibles depuis le
  frontend
4    'image/jpg': 'jpg',
5    'image/jpeg': 'jpg',
6    'image/png': 'png'
7  };
8
9  const storage = multer.diskStorage({ // Objet de configuration pour multer. Nécessite
  2 éléments: destination et filename
10    destination: (req, file, callback) => {
11      callback(null, 'images')
12    },
13    filename: (req, file, callback) => { // Générer le nouveau nom
14      const name = file.originalname.split(' ').join('_'); // Partie avant
  l'extension
15      const extension = MIME_TYPES[file.mimetype]; // Extension
16      callback(null, name + Date.now() + '.' + extension); // Nom global: partie
  avant l'extension + Timestamp créant un fichier unique + extension
17    }
18  });
19
20 module.exports = multer({ storage: storage }).single('image'); // Export du
  middleware configuré avec méthode multer précisant l'unicité du fichier et le type
  image
```



```
1  const jwt = require("jsonwebtoken");
2
3  module.exports = (req, res, next) => {
4    console.log(
5      "
6    );
7    if (req.headers.authorization === "" || !req.headers.authorization) {
8      console.log("req.headers.authorization ABSENT");
9    }
10   next();
11  };
```

```

1  const jwt = require("jsonwebtoken");
2
3  module.exports = (req, res, next) => {
4
5
6  if (req.params.id) {
7    console.log("cas req.params.id , req.params.id =", req.params.id);
8
9    try {
10     console.log(
11       "_____AUTH (en cours de vérification du NIVEAU)
12       _____"
13     );
14     const token = req.headers.authorization.split(" ")[1];
15     const decodedToken = jwt.verify(token, process.env.TOKEN);
16     console.log("decodedToken", decodedToken);
17     const isAdmin = decodedToken.isAdmin;
18     const userId = decodedToken.userId;
19     const reqId = req.params.id;
20     console.log(userId);
21     console.log(
22       "_____
23     );
24     if (isAdmin) {
25       console.log(
26         "_____AUTH NIVEAU 3 => TOKEN OK_(ADMIN
27         ACCESS)_____
28       );
29       next();
30     } else {
31       if (userId === reqId) {
32         console.log(
33           "_____AUTH NIVEAU 2 => TOKEN OK_(DELETE AND MODIF
34           ACCESS)_____
35         );
36         next();
37       } else {
38         console.log(
39           "_____AUTH NIVEAU 1 => TOKEN OK_(READ ONLY
40           ACCESS)_____
41         );
42         next();
43       }
44     }
45   } catch (error) {
46     res.status(403).json({ error: error | "Unauthorized request" });
47   }
48 }

```

```
1  const db = require("../models/index");
2  const jwt = require("jsonwebtoken");
3
4  exports.createComment = (req, res, next) => {
5    // // const token = req.body.headers.Authorization.split(" ")[1];
6    // // const decodedToken = jwt.verify(token, process.env.TOKEN);
7    // // const userId = decodedToken.userId;
8
9    // db.comment
10   //   .create({
11   //     content: req.body.content,
12   //     postId: req.body.postId,
13   //     userId: req.body.userId,
14   //   })
15   //   .then(() => res.status(201).json({ message: "Commentaire créé" }))
16   //   .catch((error) => res.status(400).json({ error: "erreur catch" }));
17
18  };
19
20  exports.getAllComments = (req, res, next) => {
21
22    // db.comment
23    //   .findAll({
24    //     order: [["createdAt", "DESC"]],
25    //     where: { postId: req.params.postId }
26    //   })
27    //   .then((comments) => {
28    //     res.status(202).json(comments);
29    //   })
30    //   .catch((error) => res.status(400).json({ error }));
31
32  };
33
34  exports.deleteComment = (req, res, next) => {
35    // const postId = JSON.parse(req.params.id);
36
37    // db.comment
38    //   .findOne({
39    //     attributes: ["id"],
40    //     where: { id: postId },
41    //   })
42    //   .then((post) => {
43    //     if (!post) {
44    //       return res.status(401).json({ error: "Post inexistant" });
45    //     }
46    //     db.post
47    //       .destroy({
48    //         attributes: ["id"],
49    //         where: { id: postId },
50    //       })
51    //       .then(() => res.status(200).json({ message: "Post supprimé" }))
52    //       .catch(() => res.status(500).json({ error: "erreur catch" }));
53    //   })
54    //   .catch((error) => res.status(500).json({ error: "erreur catch" }));
55
56  };
```

```
1  const db = require("../models/index");
2  const jwt = require("jsonwebtoken");
3  const fs = require("fs");
4
5  exports.createPost = (req, res, next) => {
6
7    console.log("req.body", req.body)
8
9    const token = req.headers.authorization.split(" ")[1];
10   const decodedToken = jwt.verify(token, process.env.TOKEN);
11   const userId = decodedToken.userId;
12
13   db.user
14     .findOne({
15       where: { id: userId },
16     })
17
18     .then((user) => {
19       if (user) {
20
21         db.post
22           .create({
23             content: req.body.content,
24             title: req.body.title,
25             userId: user.id,
26             attachment: req.file ? `${req.protocol}://${req.get('host')}/images/${
27               req.file.filename}` : req.body.image,
28           })
29           .then(() => res.status(201).json({ message: "Post créé" }))
30           .catch((error) => res.status(400).json({ error: "erreur catch" }));
31       } else {
32         return res.status(404).json({ error: "Utilisateur non trouvé" });
33       }
34     })
35     .catch((error) => res.status(501).json({ error: "erreur catch" }));
36   };
37
38   exports.getAllPosts = (req, res, next) => {
39     db.post
40       .findAll({
41         order: [["createdAt", "DESC"]],
42       })
43       .then((posts) => {
44         res.status(202).json(posts);
45       })
46       .catch((error) => res.status(400).json({ error }));
47   };
48
49   exports.getOnePost = (req, res, next) => {
50     const postId = JSON.parse(req.params.id);
51
52     db.post
53       .findOne({
54         attributes: ["id", "userId", "title", "content", "attachment"],
55         where: { id: postId },
56       })
57
58       .then((post) => {
59         if (!post) {
60           return res.status(401).json({ error: "Post inexistant" });
61         }
62         res.status(200).json(post);
63       })
64       .catch((error) => res.status(500).json({ error: "erreur" }));
65   };
66 }
```

```
66 exports.modifyPost = (req, res, next) => {};  
67  
68 exports.deletePost = (req, res, next) => {  
69   const postId = JSON.parse(req.params.id);  
70   console.log(req.params.id);  
71  
72   db.post  
73     .findOne({  
74       attributes: ["id"],  
75       where: { id: postId },  
76     })  
77     .then((post) => {  
78       if (!post) {  
79         return res.status(401).json({ error: "Post inexistant" });  
80       }  
81       db.post  
82         .destroy({  
83           attributes: ["id"],  
84           where: { id: postId },  
85         })  
86         .then(() => res.status(200).json({ message: "Post supprimé" })))  
87         .catch(() => res.status(500).json({ error: "erreur catch" }));  
88     })  
89     .catch((error) => res.status(500).json({ error: "erreur catch" }));  
90   };
```

```
1  const db = require("../models/index");
2  const bcrypt = require("bcryptjs");
3  const jwt = require("jsonwebtoken");
4
5  exports.signup = (req, res, next) => {
6    const email = req.body.email;
7    const firstName = req.body.firstName;
8    const lastName = req.body.lastName;
9    const password = req.body.password;
10
11    if (
12      email == null ||
13      email == "" ||
14      firstName == null ||
15      firstName == "" ||
16      lastName == null ||
17      lastName == "" ||
18      password == null ||
19      password == ""
20    ) {
21      return res.status(400).json({ error: "Formulaire incomplet" });
22    }
23
24    db.user
25      .findOne({
26        where: { email: req.body.email },
27      })
28      .then((user) => {
29        if (user) {
30          return res.status(401).json({ error: "Email déjà utilisé!" });
31        }
32        bcrypt
33          .hash(req.body.password, 5)
34          .then((hash) => {
35            db.user.create({
36              firstName: req.body.firstName,
37              lastName: req.body.lastName,
38              email: req.body.email,
39              password: hash,
40              isAdmin: false,
41            });
42          });
43          .then(() => res.status(201).json({ message: "Utilisateur créé" }));
44          .catch((error) => res.status(400).json({ error }));
45        });
46      .catch((error) => res.status(500).json({ error: "error" }));
47    });
48
49  exports.login = (req, res, next) => {
50
51    const email = req.body.email;
52    const password = req.body.password;
53
54    if (email == null || email == "" || password == null || password == "") {
55      return res.status(400).json({ error: "Champs manquants" });
56    }
57
58    db.user
59      .findOne({ where: { email: req.body.email } })
60      .then((user) => {
61        if (!user) {
62          return res.status(401).json({ error: "Utilisateur non trouvé" });
63        }
64      })
65    }
66  }
```

```
67     bcrypt
68     .compare(req.body.password, user.password)
69     .then((valid) => {
70         if (!valid) {
71             return res.status(401).json({ error: "Mot de passe incorrect" });
72         }
73         res.status(200).json({
74             email: user.email,
75             lastName: user.lastName,
76             firstName: user.firstName,
77             userId: user.id,
78             isAdmin: user.isAdmin,
79             token: jwt.sign(
80                 { userId: user.id, isAdmin: user.isAdmin },
81                 process.env.TOKEN,
82                 {
83                     expiresIn: "12h",
84                 }
85             ),
86         });
87     })
88     .catch((error) => res.status(500).json({ error }));
89 })
90
91 .catch((error) => res.status(500).json({ error: "erreur" }));
92 };
93
94 exports.getAccount = (req, res, next) => {
95
96     db.user
97     .findOne({
98         attributes: ["id", "firstName", "lastName", "email", "isAdmin"],
99         where: { id: req.params.id },
100     })
101     .then((user) => {
102         if (!user) {
103             return res.status(401).json({ error: "Utilisateur inexistant" });
104         }
105         res.status(200).json(user);
106     })
107     .catch((error) => res.status(500).json({ error: "erreur" }));
108 };
109
110 exports.getOneAccount = (req, res, next) => {
111     const userId = JSON.parse(req.params.id);
112     console.log("req.params.id", req.params.id);
113
114     db.user
115     .findOne({
116         attributes: ["id", "firstName", "lastName", "email"],
117         where: { id: userId },
118     })
119     .then((user) => {
120         if (!user) {
121             return res.status(401).json({ error: "Utilisateur inexistant" });
122         }
123         res.status(200).json(user);
124     })
125     .catch((error) => res.status(500).json({ error: "erreur" }));
126 };
127
128 exports.modifyAccount = (req, res, next) => {};
129
130 exports.deleteAccount = (req, res, next) => {
131     const userId = JSON.parse(req.params.id);
132     console.log("req.params.id", req.params.id);
```

```
133
134     db.user
135       .findOne({
136         attributes: ["id"],
137         where: { id: userId },
138       })
139
140     .then((user) => {
141       if (!user) {
142         return res.status(401).json({ error: "Utilisateur inexistant" });
143       }
144       db.user
145         .destroy({
146           attributes: ["id"],
147           where: { id: userId },
148         })
149         .then(() => res.status(200).json({ message: "Compte supprimé" }))
150         .catch(() => res.status(500).json({ error: "erreur" }));
151     })
152
153     .catch((error) => res.status(500).json({ error: "erreur" }));
154 };
155
156 exports.getAllAccounts = (req, res, next) => {
157   db.user
158     .findAll({
159       order: [["createdAt", "DESC"]],
160       attributes: [
161         "id",
162         "firstName",
163         "lastName",
164         "email",
165         "createdAt",
166         "isAdmin",
167       ],
168     })
169     .then((users) => res.status(202).json(users))
170     .catch((error) => res.status(400).json({ error: "erreur" }));
171 };
```



```
1  {
2    "development": {
3      "username": "sqluser",
4      "password": "password",
5      "database": "groupomania",
6      "host": "127.0.0.1",
7      "dialect": "mysql"
8    },
9    "test": {
10     "username": "sqluser",
11     "password": "password",
12     "database": "groupomania_test",
13     "host": "127.0.0.1",
14     "dialect": "mysql"
15   },
16   "production": {
17     "username": "sqluser",
18     "password": "password",
19     "database": "groupomania_production",
20     "host": "127.0.0.1",
21     "dialect": "mysql"
22   }
23 }
```