

Côte Arthur

# RAPPORT DE STAGE

effectué au sein de l'établissement

**Office Français de la Biodiversité**

**Maître de stage:** François COUILLOUD



**OFB**  
**OFFICE FRANÇAIS  
DE LA BIODIVERSITÉ**

# Sommaire

<b>Sommaire</b>	<b>1</b>
<b>Remerciements</b>	<b>2</b>
<b>Glossaire</b>	<b>3</b>
<b>Curriculum vitae</b>	<b>4</b>
<b>Présentation de l'entreprise d'accueil</b>	<b>5</b>
Situation géographique	5
Historique	7
Activités	8
Effectifs	9
Organisation de l'entreprise	9
<b>Activités confiées à l'étudiant</b>	<b>10</b>
Diagramme de Gantt	10
Analyse de la demande	11
Conception	12
Maquettage	12
Développement	16
Choix des technologies	16
Base de données	17
Back-end	18
Front-end	21
Etat d'avancement	23
Etat de l'interface	23
Fonctionnalités restant à implémenter	25
Difficultés rencontrées	25
<b>Conclusion</b>	<b>26</b>

# Remerciements

Je tiens à remercier mon maître de stage François COUILLLOUD pour sa confiance, sa disponibilité et son encadrement tout au long de ce stage.

Je remercie également Alexis Nardiello pour ses nombreux conseils et aides apportés quant au développement de l'application sans qui l'application n'aurait pas été d'une qualité équivalente.

# Glossaire

- Front-end : Partie serveur.
- Back-end : Partie client.
- API : “Application Programming Interface”. Il s’agit d’une interface par laquelle un logiciel offre des services à d’autres logiciels.
- JSON : “JavaScript Object Notation” Format de données textuelles dérivé de la notation des objets du langage JavaScript.

# Curriculum vitae

## Arthur CÔTE



06 21 87 47 50



4, rue Nestor Cornier  
38100 - Grenoble  
Né le 12/12/1999 à Grenoble



cote.arthur.clint@gmail.com

### LANGUES

- Anglais : C1
- Espagnol : A2

### FORMATION

IUT1 LP SMIN | Grenoble  
2019 - 2020  
Lycée du Grésivaudan BTS SNIR | Meylan  
2017 - 2019  
Lycée Emmanuel Mounier | Grenoble  
2014 - 2017  
Collège Charles Munch | Grenoble  
2010 - 2014

### DIPLÔMES

BTS SYSTÈMES NUMÉRIQUES  
OPTION INFORMATIQUE ET RÉSEAUX  
BACCALAURÉAT S  
SPÉCIALITÉ PHYSIQUE-CHIMIE  
BREVET DES COLLÈGES

### EXPÉRIENCE PROFESSIONNELLE

OFB - WEBDESIGNER  
20 janvier - 24 juin 2020 | Gières  
ECONOCOM - CHARGÉ DE SUPPORT  
28 mai - 06 juillet 2018 | Grenoble

### CONNAISSANCES INFORMATIQUES

#### DÉVELOPPEMENT DESKTOP

##### JAVA



##### C++



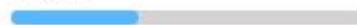
##### C



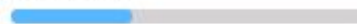
##### WINDOWS



##### LINUX



##### RESEAUX



#### DÉVELOPPEMENT MOBILE

##### ANDROID



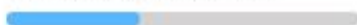
##### iOS



##### HYBRIDE



#### DÉVELOPPEMENT WEB



### CENTRES D'INTÉRÊTS

- Skateboard
- Lecture, Musique
- Jeux vidéos

# Présentation de l'entreprise d'accueil

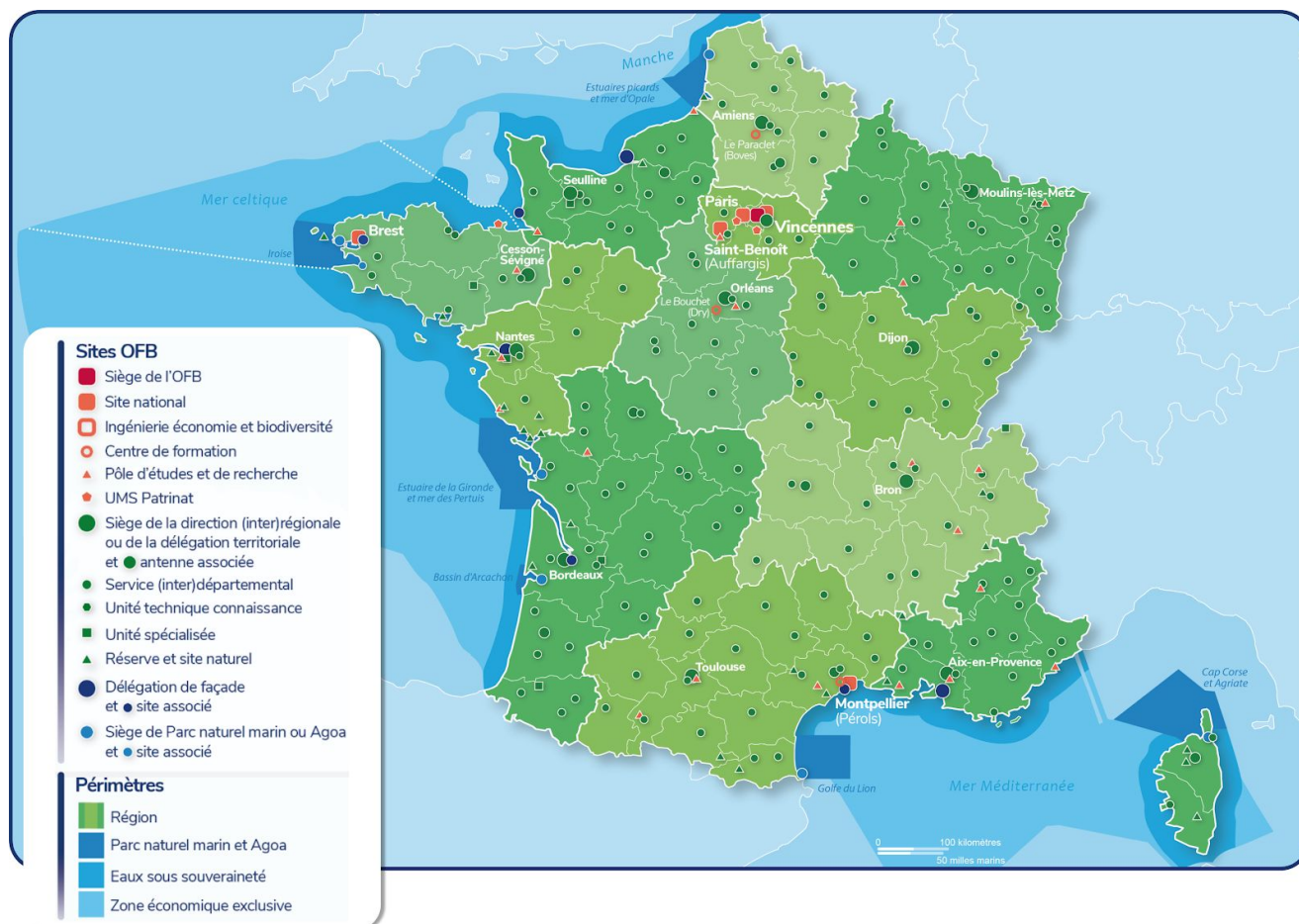
## Situation géographique

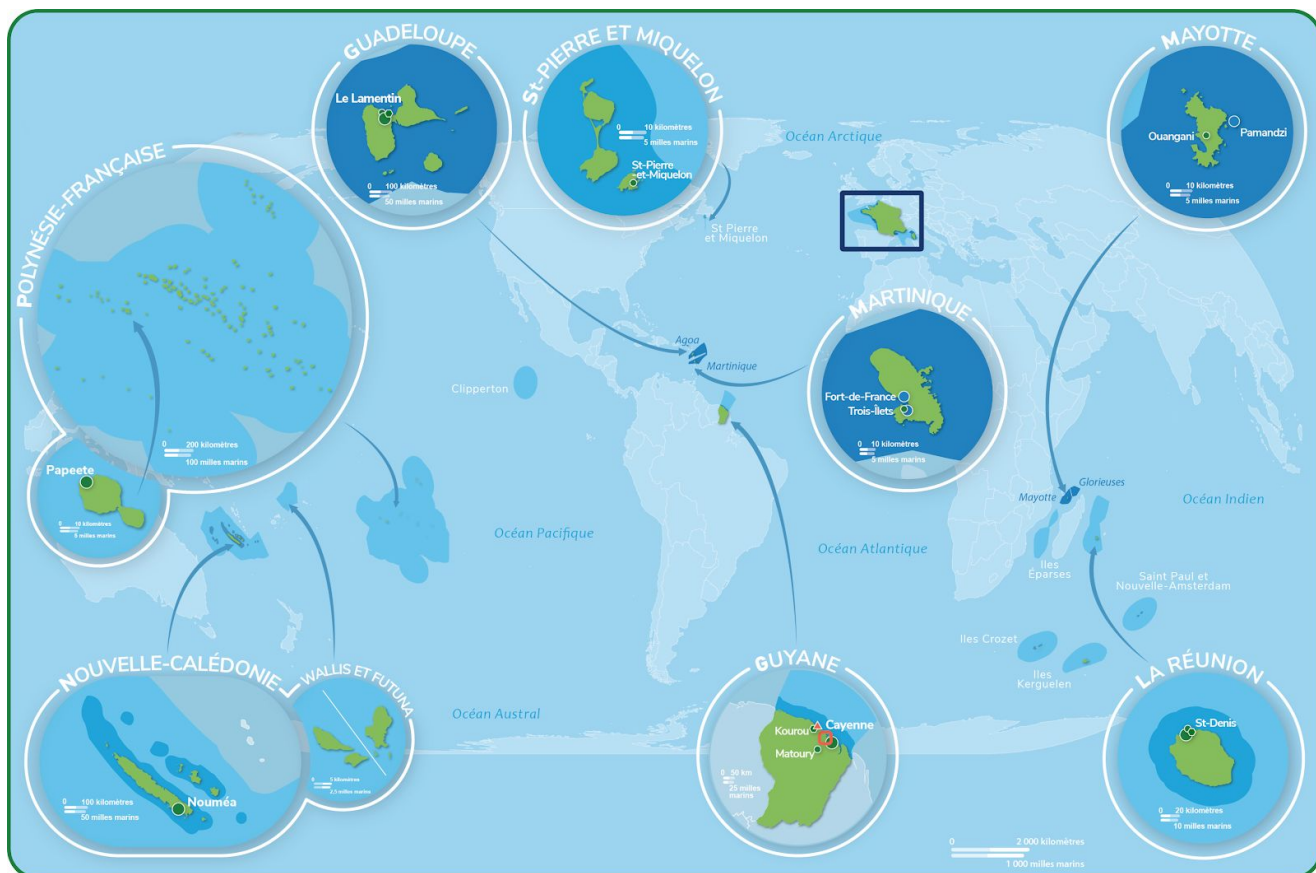
L'Office français de la biodiversité est présent sur l'ensemble du territoire français en métropole et Outre-mer.

Selon trois niveaux d'élaboration, il s'organise sous forme de matrice pour considérer tous les environnements:

- **Échelle nationale** : identification et orientation de la politique et la stratégie de l'OFB à l'échelle nationale (directions et délégations nationales).
- **Échelle régionale** où s'exercent la coordination et la déclinaison territoriale (directions régionales).
- **Échelons départementaux et locaux** : fonctionnement et mise en œuvre spécifique aux niveaux départemental et local (services départementaux, antennes de façade, parcs naturels marins, etc.).

### Implantations de l'Office français de la biodiversité





### “Un maillage au plus près des territoires” :

- 5 sites nationaux
- 12 directions régionales, 1 direction interrégionale métropolitaine et une direction pour les Outre-mer
- 90 services départementaux métropolitains et 3 services interdépartementaux
- 4 services départementaux ultra-marins, 1 brigade nature Océan indien et 1 service territorial à Saint-Pierre et Miquelon
- 3 délégations de façade maritimes métropolitaines
- 2 délégations territoriales ultramarines (Polynésie française et Nouvelle-Calédonie compétente pour Wallis et Futuna)
- 3 délégués territoriaux (Antilles, Guyane et Océan indien)
- 9 parcs naturels marins (Iroise, Mayotte, golfe du Lion, Glorieuses, estuaires picards et de la mer d’Opale, Bassin d’Arcachon, estuaire de la Gironde et mer des Pertuis, cap Corse et Agriate, Martinique)
- Le sanctuaire de mammifères marins Agoa aux Antilles
- Plusieurs pôles d’études et de recherche sur les espèces et les milieux

## Historique

L'Office français de la biodiversité est un établissement public national créé par la loi n°019-773 du 24 juillet 2019. Il contribue au suivi, à la protection et à la gestion des milieux terrestres, aquatiques et marins ainsi qu'à restaurer la biodiversité, l'équilibre et la gestion durable de l'eau en coordination avec les politiques nationales de lutte contre le réchauffement climatique.

En avril 2018, le rapport effectué par l'Inspection générale des finances et le Conseil général de l'environnement et du développement durable démontre un manque de coordinations quant aux politiques de l'eau et de la biodiversité menées jusqu'à présent. Le gouvernement prend alors l'initiative d'un projet de loi ayant pour but de faciliter et d'améliorer l'exercice de ces missions et de bâtir une synergie avec les multiples organismes.

Le projet de loi est ensuite présenté en novembre 2018 et est définitivement adopté le 16 juillet 2019.

Le 1er janvier 2020, l'Office français de la biodiversité entre officiellement en fonction fusionnant l'Agence française pour la biodiversité (AFB) et de l'Office national de la chasse et de la faune sauvage (ONCFS)



# Activités

L'OFB est chargé de 5 missions principales:

- **Connaissance et expertise :**

Le développement de la connaissance et de l'expertise sur l'ensemble des composantes de la nature constitue l'une des missions principales de l'Office français de la biodiversité. Mieux comprendre les espèces, les milieux, les services rendus par la biodiversité et les menaces qu'elle subit est primordial pour protéger le vivant.

- **Gestion et restauration des espaces protégés :**

Les pressions exercées par les activités humaines sont la principale cause de l'érosion de la biodiversité, et ce, de façon souvent irréversible. C'est pourquoi, il est apparu essentiel de protéger certains espaces jugés remarquables en termes d'espèces et habitats, de fonctions écologiques et de services rendus par les écosystèmes, mais aussi souvent de patrimoine culturel et paysager.

- **Police de l'environnement :**

Exercice des polices administrative et judiciaire relatives à l'eau (pollution de la ressource, atteinte aux zones humides ou littoral), aux espaces naturels, à la flore et la faune sauvage (espèces gibier ou protégées, lutte contre les trafics d'espèces), à la chasse (contre-braconnage, renforcement de la sécurité à la chasse) et à la pêche.

- **Mobilisation de la société :**

Mobilisation de l'ensemble de la société en faveur de la biodiversité. Son rôle de levier auprès des acteurs, décideurs et des citoyens doit contribuer à réduire les pressions exercées sur la faune, la flore et leurs habitats tout en multipliant les contributions positives. Pour ce faire, l'établissement entend faire monter le sujet biodiversité dans l'opinion publique et créer les conditions optimales pour favoriser l'engagement de tous.

- **Appui aux politiques publiques :**

Accompagner et apporter son appui aux acteurs publics pour la conception, la mise en œuvre et l'évaluation de leurs politiques.

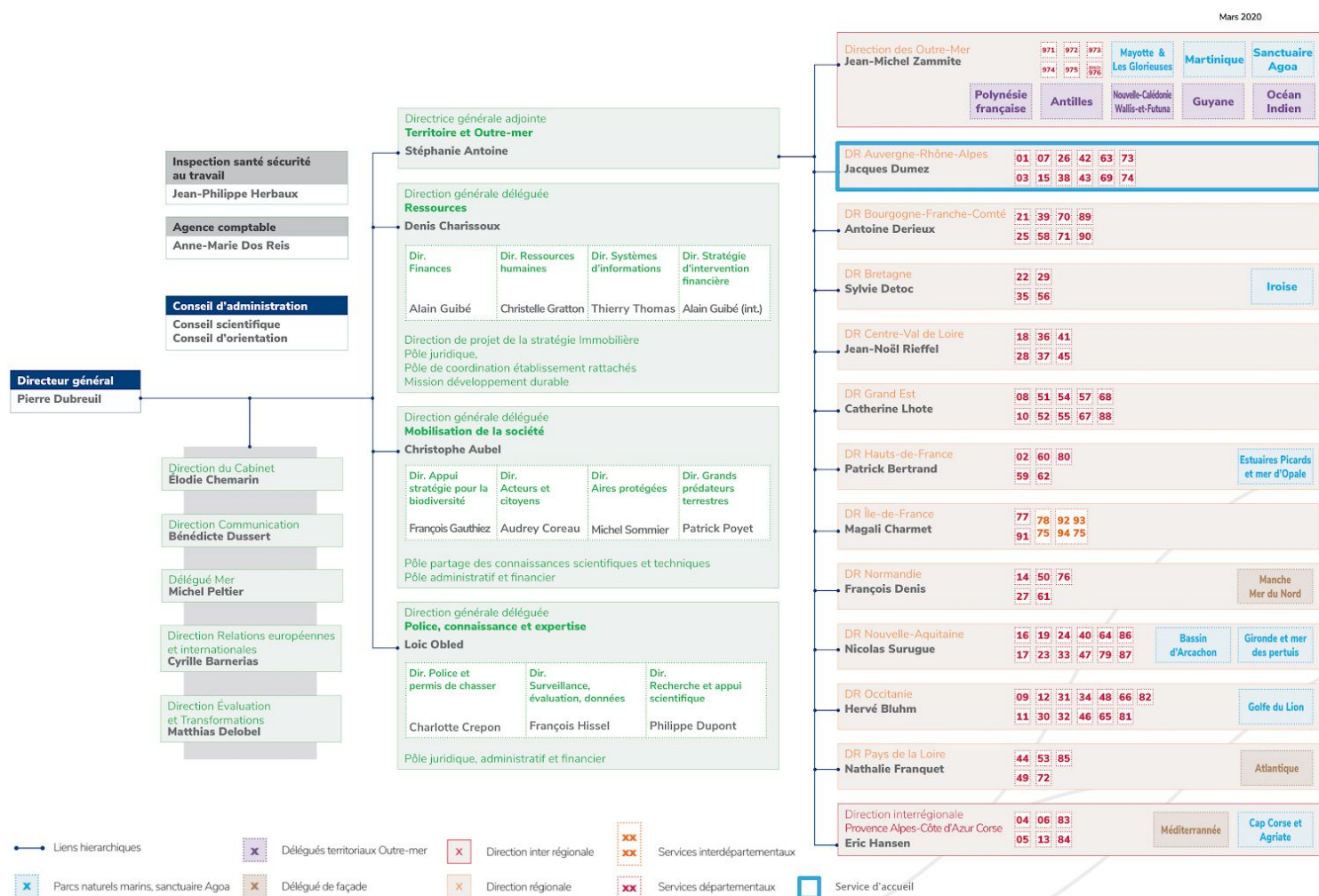
## Effectifs

La majorité des personnels de l'OFB sont des agents de terrain. Environ 2 800 agents sont répartis sur tout le territoire métropolitain et en Outre-mer :

- services centraux implantés à Auffargis (78), Brest (29), Montpellier (34), Paris (75) et Vincennes (94)
- directions régionales et les services départementaux
- 11 parcs nationaux rattachés
- 26 réserves de faune sauvage relèvent de l'OFB

## Organisation de l'entreprise

### Macro-organigramme de l'Office français pour la biodiversité





## Analyse de la demande

Le besoin soulevé par le service Auvergne-Rhône Alpes est un outil web permettant de gagner du temps en facilitant la communication et le partage de connaissances au sein de l'ensemble du service.

Jusqu'à présent, le personnel du service dispersé dans l'ensemble de la région avait recours aux mails lorsqu'il souhaitait transmettre des informations à leurs collègues. Si ce système est en soit fonctionnel, il est parsemé d'inconvénients.

En effet, tout le monde ne consulte pas l'ensemble de ses mails en permanence, d'autant plus que le nombre de mails reçus peut être très abondant.

Ainsi les informations qui intéressent quelqu'un peuvent être dissimulées parmi tous les autres mails. Il est donc difficile de retrouver une information consultée, voir archivée de manière aléatoire.

De plus le manque de structure et de lignes directrices quant au partage d'informations soulève d'autre problèmes : les documents transmis sont souvent trop copieux en informations et donc pas assez succincts pour être consultés dans leur entièreté.

L'application doit donc permettre de transmettre et consulter des informations et documents avec un format non fixe. J'ai donc proposé de garder le format PDF qui était déjà utilisé pour la plupart des documents partagés afin de faciliter à la fois l'ajout (on remplace juste l'envoi d'un document par mail par un envoi sur l'application web) et le développement de l'application (pas besoin de développer une création de documents).

L'envoi de document sera donc un simple formulaire (titre, thème...) dans lequel on ajoute le PDF souhaité.

Afin de faciliter la consultation, il faut organiser de manière pertinente les documents et permettre à l'utilisateur de filtrer ce qu'il lui est affiché.

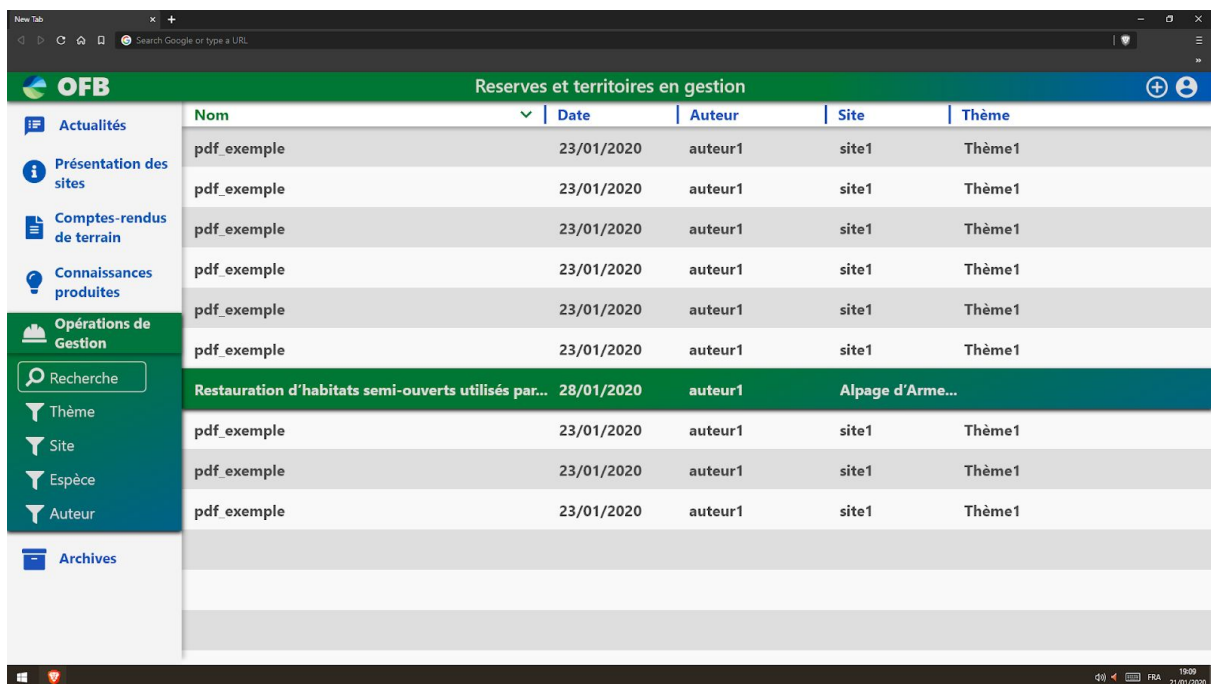
# Conception

## Maquettage

La prémisses de la maquette est simple, il s'agit d'une liste de documents affichée sous forme de table. Lorsqu'on clique sur une ligne de la liste, le document pdf s'affiche, on peut le faire défiler, zoomer, le télécharger.

On peut filtrer la liste à l'écran à l'aide d'un menu à disposition et changer l'ordre de la liste en fonction de différents champs (date, titre...).

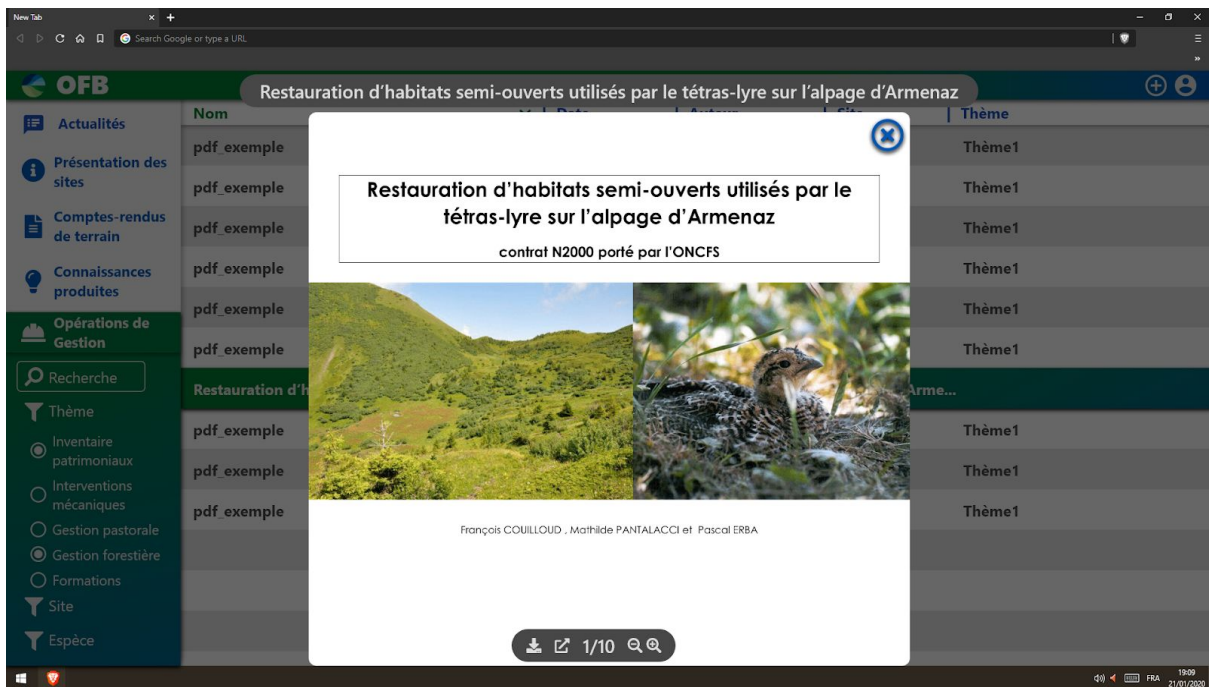
### Disposition de la page type "liste"



The screenshot displays a web application interface for "OFB Reserves et territoires en gestion". The interface includes a sidebar with navigation links: Actualités, Présentation des sites, Comptes-rendus de terrain, Connaissances produites, Opérations de Gestion, Recherche, Thème, Site, Espèce, Auteur, and Archives. The main content area shows a table with columns: Nom, Date, Auteur, Site, and Thème. The table contains several rows of data, including a highlighted row for "Restauration d'habitats semi-ouverts utilisés par..." with a date of 28/01/2020 and a site of "Alpage d'Arme...".

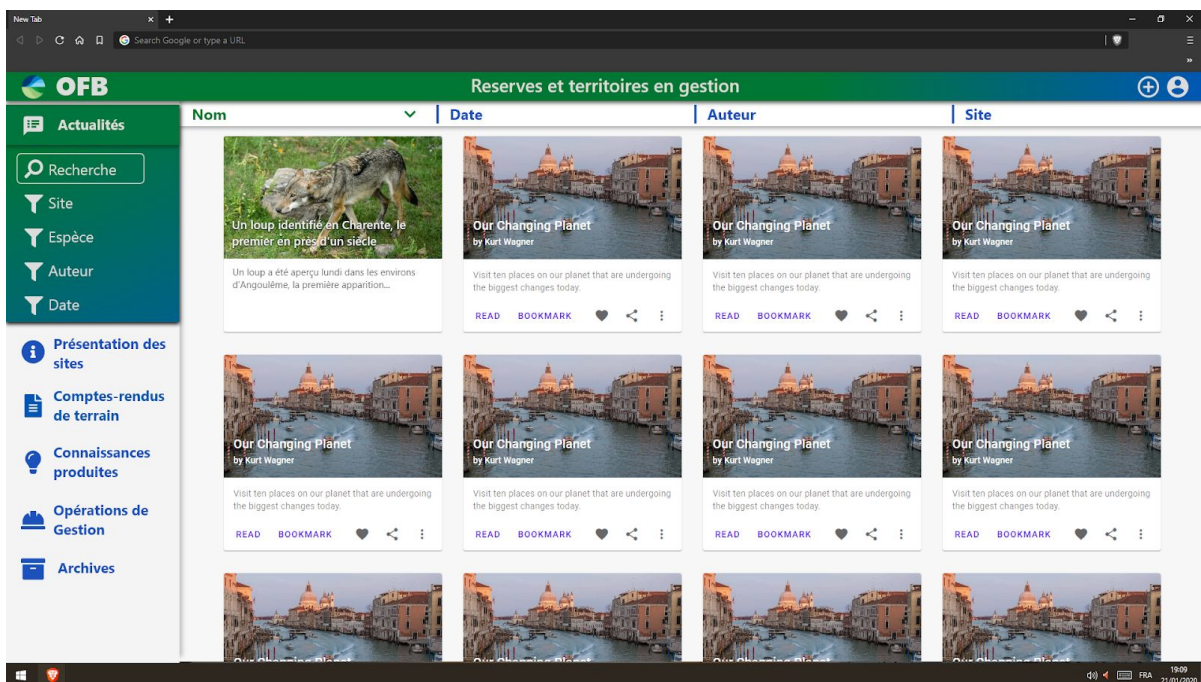
Nom	Date	Auteur	Site	Thème
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1
Restauration d'habitats semi-ouverts utilisés par...	28/01/2020	auteur1	Alpage d'Arme...	
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1
pdf_exemple	23/01/2020	auteur1	site1	Thème1

## Visualisation d'un PDF



On a ici la possibilité de consulter le PDF, zoomer, et de le télécharger. Si on clique la croix en haut à droite ou à côté du PDF, la visualisation se ferme.

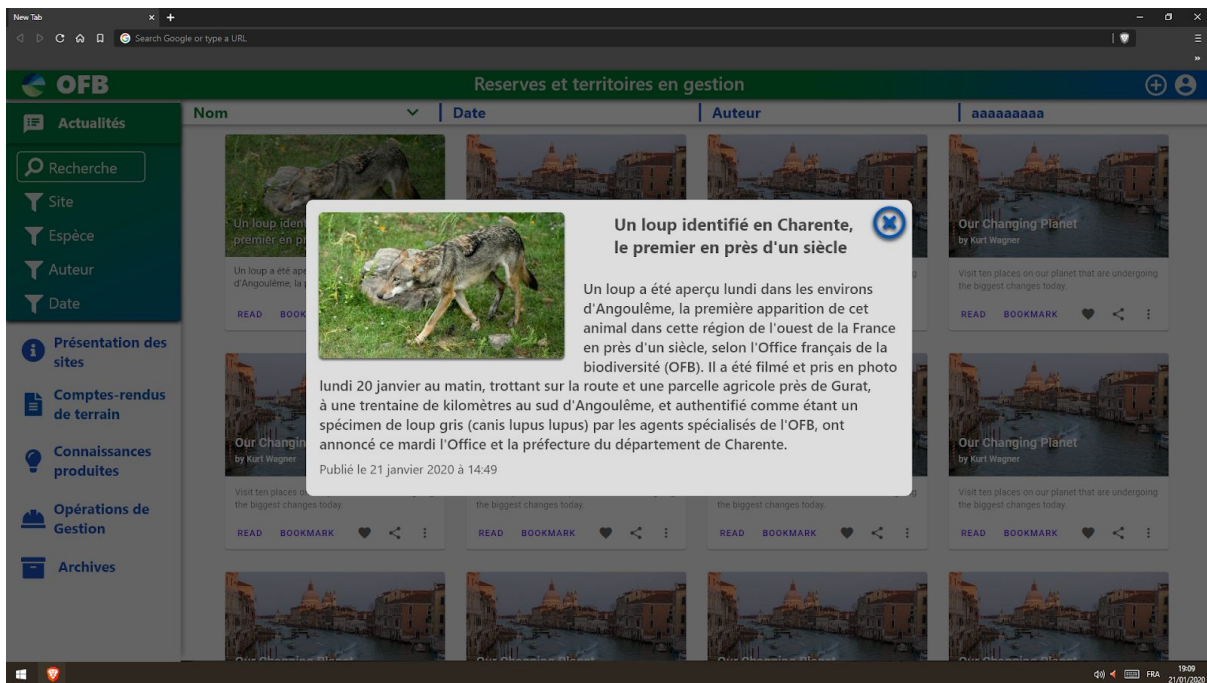
## Disposition type "carte"



Les ajouts de type "actualités" sont composés d'une photo, d'un titre et d'une description au lieu d'un pdf. J'ai donc opté pour un affichage moins condensé et plus agréable visuellement avec des "cartes".



## Visualisation d'actualité



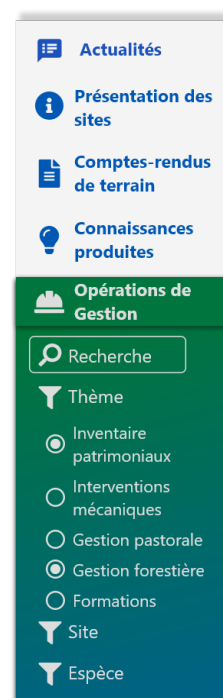
Cette visualisation est similaire à celle du PDF en plus simple, tout le contenu est directement affiché et on peut la fermer en cliquant à côté ou sur la croix en haut à droite.

Les champs situés en haut des tables permettent de changer l'ordre de l'affichage. Si on clique une 2ème fois sur un champ déjà sélectionné l'ordre est alors inversé (par exemple, date du plus récent au plus ancien et vice-versa).

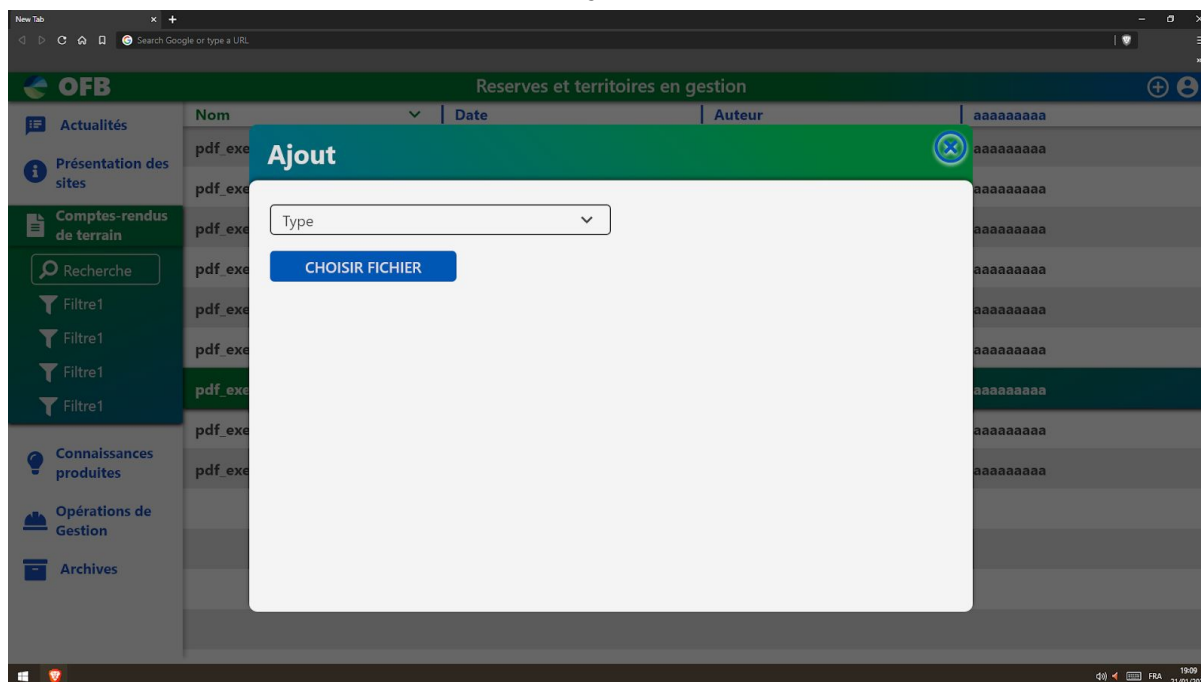
Nom	Date	Auteur	Site	Thème
-----	------	--------	------	-------

Chaque "onglet" du menu correspond à une table. Ces onglets correspondent à des "catégories" de documents.

L'onglet actuellement sélectionné est étendu afin de laisser place aux filtres. Les filtres se présentent de différentes manières en fonction de ce qui est le plus judicieux (barre de recherche, liste déroulante...).



## Ajout



A tout moment sur le site, l'utilisateur a la possibilité d'ajouter du contenu avec le bouton “+” en haut à droite de l'écran.

L'utilisateur choisi alors le type d'ajout (Actualité, Compte-rendu terrain...). Le formulaire à remplir va alors s'adapter (Présence ou non de thèmes, import d'une photo ou d'un pdf, mots clés...).

Si l'utilisateur n'est pas connecté, le bouton “+” n'est pas cliquable.



# Développement

## Choix des technologies

En ce qui concerne la base de données, je me suis orienté vers quelque chose qui m'était familier ; la base étant assez basique, n'importe quelle technologie aurait pu fonctionner, j'ai opté pour MariaDB, une base de données SQL.

Pour ce qui est du Front-end, je me suis orienté vers ReactJS pour plusieurs raisons :

- ce framework avait été étudié en cours récemment.
- il est aujourd'hui un des frameworks les plus utilisés dans le monde du travail. Il est donc très profitable de le maîtriser.
- sa popularité permet aussi de trouver des ressources et aides plus facilement.

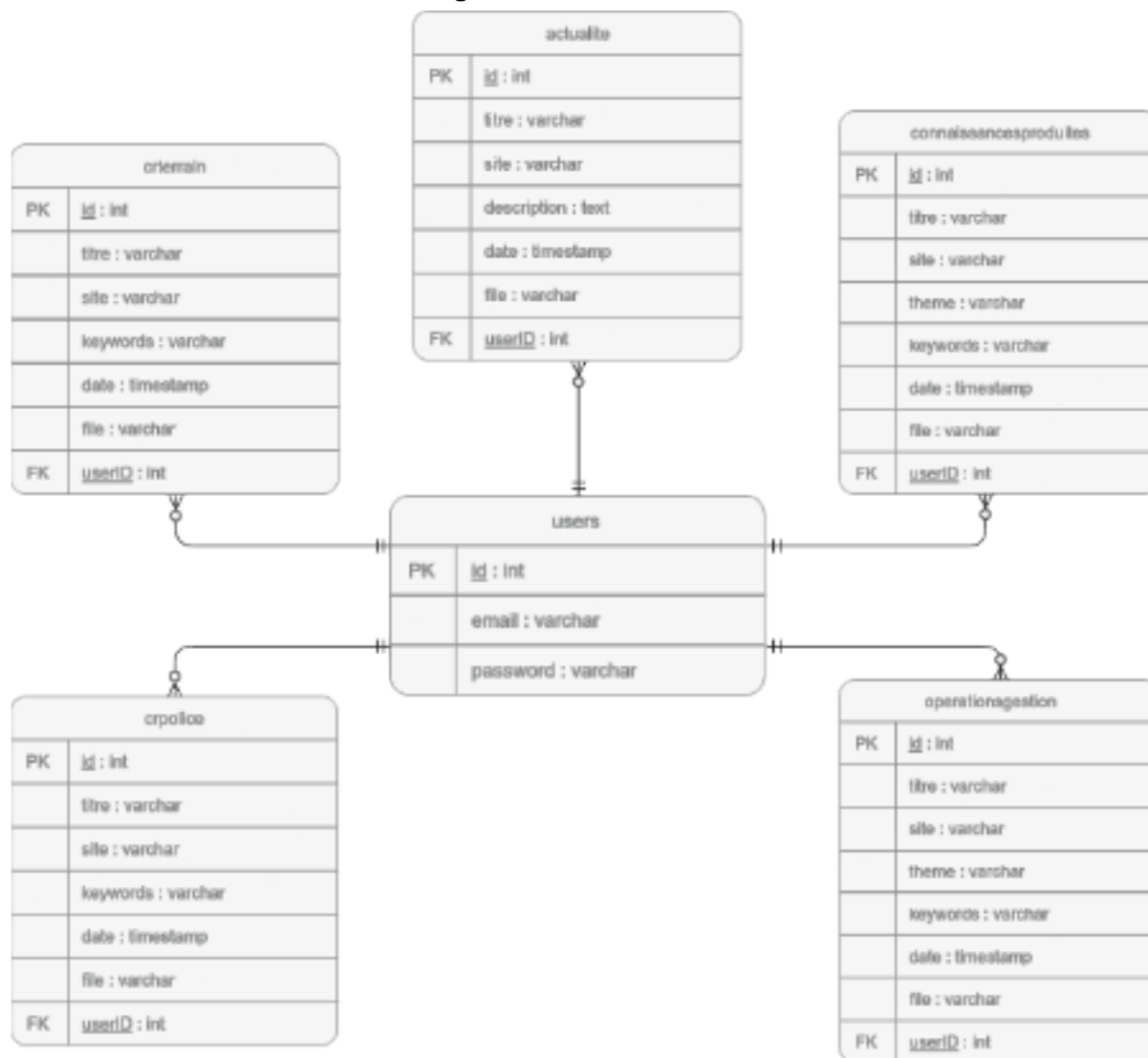
Préférant TypeScript à JavaScript je me suis orienté vers ce langage étant donné qu'il est compatible avec ReactJS.

J'ai décidé de rester en JavaScript avec NodeJS pour le Back-end car cela semblait être le choix "par défaut" pour les projets React, ainsi je disposais d'un plus grand nombre de ressources lors du développement.

J'ai utilisé Github pour la gestion de code et de sauvegarde.

## Base de données

### Diagramme entité relation



Ne connaissant pas l'intégralité des filtres au moment de la conception de la base de données, j'ai fait le choix d'une table par onglet.

Une fois les champs des tables au complet, je me suis rendu compte que ce modèle n'était pas forcément le plus optimisé. En effet, les tables "crterrain" et "crpolice" ainsi que "connaissancesproduites" et "operationsgestion" sont, d'une part identiques, et d'autre part, dans l'ensemble similaires entre elles.

Il aurait peut-être été plus judicieux d'avoir un champ "type" pour différencier les différents types d'ajout et d'avoir la possibilité de mettre les champs qui n'apparaissent pas systématiquement en "null".

Néanmoins le modèle imagé ci-dessus permet une simplification des requêtes : pas besoin de vérifier un champ "type" à chaque sélection de la table.

## Back-end

Le back-end est principalement basé sur la librairie ExpressJS. Elle permet de créer une api robuste de manière rapide et simple.

Configuration du back-end :

```
const app = express();

app.use(cors()); //autorisation du CORS (partage de ressources entre origines multiples)
app.use(express.json({ limit: '500MB' })); //autorisation du JSON lors de POST
app.use('/files', express.static('files')); //montage du dossier files
```

Ici on prépare le fonctionnement du back avec différentes autorisations nécessaires. On monte aussi le dossier *files* afin de pouvoir récupérer les fichier envoyés à l'aide de leurs noms.

Mise en place des routes de l'api :

```
app.route('/api/actualite').post(homeController.actualite);
```

On lie ici la route *www.nomdusite.com/api/actualite* avec la méthode *actualite* présente dans la classe *homeController*. Lorsque le navigateur ira chercher l'url en question, le back-end exécutera le code présent dans la méthode liée afin de renvoyer les informations souhaitées.

Méthode *actualite* :

```
public actualite(req: Request, res: Response): void
{
    let strQuery = `SELECT actualite.* FROM actualite`;

    if(req.body.auteur){ ...
    }else{ ...
    }

    if(req.body.orderBy){ //le champ orderBy est defini dans le JSON envoyé
        if(req.body.desc){ //si l'ordre est descendant
            strQuery += ` ORDER BY ${req.body.orderBy} DESC`;
        }else{
            strQuery += ` ORDER BY ${req.body.orderBy} ASC`;
        }
    }else{
        strQuery += ` ORDER BY date DESC`;
    }

    connection.query(strQuery, (err, results) => {
        if(err) { //si il y a une erreur
            res.json(err); //envoi de l'erreur
        }
        res.json(results); //envoi des resultats
    });
}
```

On prépare ici la requête SQL en fonction des filtres envoyés sous format JSON par le client.

On envoie ensuite la requête à la base de données avec la librairie mysql et on renvoie le résultat de cette requête au client.

Création de la connection avec la base de données :

```
const connection = mysql.createConnection(config.mysql);
```

Avec *config.mysql* un JSON :

```
mysql: {  
  host: 'localhost',  
  port: 3306,  
  user: 'root',  
  database: 'ofb',  
  password: ''  
}
```

L'absence de mot de passe de l'utilisateur "root" est une faille de sécurité à corriger lors du déploiement. La création d'un utilisateur autre pour l'application avec un nombre minimum de droits pour son bon fonctionnement est également à prévoir.

Sauvegarde du fichier dans le dossier monté */files* :

```
let fileName = shortid.generate();  
//Génération d'un ID unique pour le nom du fichier (ex: NwQMhVDX4)  
  
let fileType = req.body.file.split('/')[1].split(';')[0];  
//Récupération du type de fichier (pdf, jpg...)  
  
fs.writeFile(`./files/${fileName}.${fileType}`,  
req.body.file.split(';base64,').pop(),  
{encoding: 'base64'}, (err) => {  
  if(err){  
    console.log(err);  
  }  
});
```

Ce fichier sauvegardé sera alors accessible avec l'URL :  
*www.nomdusite.com/files/nomdufichier*

## Front-end

Pour les composants front-end, je me suis orienté vers Material-Ui. Il s'agit du framework interface utilisateur le plus populaire pour ReactJS. Il est non seulement visuellement impressionnant et fluide mais sa documentation est très complète et agréable à consulter. Chaque composant est présenté avec de multiples exemples pour les nombreuses variantes proposées. Une code source est en plus à disposition pour tous les exemples à la fois en JavaScript et en TypeScript.

L'application est composée d'un menu, d'une barre d'actions, de boîtes de dialogues pour les formulaires et les visualisations de documents ainsi qu'une table formée à partir d'un tableau contenant les données de la base de données.

Récupération des données sur le back-end :

```
axios.post('http://localhost:25565/api/actualite',
{headers: { 'Content-Type': 'application/json' }} )
.then(r =>
  r.data[0] ? (
    setState((prevState)=>({
      ...prevState,
      selectedTable: 'actualite',
      dataArray: r.data,
    }))
  ) : (
    setState((prevState)=>({
      ...prevState,
      selectedTable: 'actualite',
      dataArray: [],
    }))
  )
)
```

A l'aide de la librairie *axios* le client envoie une requête pour récupérer ici le contenu de la table *actualite* sans filtres. On envoie en en-tête le header JSON pour passer les CORS. Les données renvoyées sont sous forme de tableau, ce pourquoi je vérifie si le premier élément du tableau est défini avant de modifier le contenu de ma page. S'il n'est pas défini, je reviens à un état vide.

Récupération de données avec envois de filtres :

```
axios.post(`http://localhost:25565/api/${childData}`, filters,  
{headers: { 'Content-Type': 'application/json' }} )  
.then(r =>
```

Avec *filters* un JSON de la forme:

```
export type FiltersType = {  
  orderBy?: string;  
  desc?: boolean;  
  search?: string;  
  sites?: string;  
  year?: string;  
  auteur?: string;  
}
```

On envoie donc au back-end ce JSON, le back-end vérifiera alors quels sont les champs définis et en tiendra compte dans la requête envoyée à la base de données.

# Etat d'avancement

## Etat de l'interface

### Affichage sous forme de cartes

The screenshot shows the OFB interface with a green header bar containing the logo and the text "Réserves et territoires en gestion". On the left, there is a sidebar with a menu under "Actualités" and a search section. The main content area displays a card for a reserve titled "test" with a date of "19/06/2020" and an author "a@a.a". The card includes a placeholder image of a mountain range and a brief description.

### Affichage sous forme de liste

The screenshot shows the OFB interface with a green header bar containing the logo and the text "Réserves et territoires en gestion". On the left, there is a sidebar with a menu under "Comptes-rendus terrain" and a search section. The main content area displays a table with three rows of data, each representing a reserve. The columns are labeled "Titre", "Date", "Auteur", and "Site".

Titre	Date	Auteur	Site
a	15/06/2020	b@b.b	Bauges
a	15/06/2020	a@a.a	Bauges
a	15/06/2020	a@a.a	Bauges

### Fenêtre de connexion

The screenshot shows a login window titled "Connexion". It contains two input fields: "Email \*" and "Mot de passe \*". Below the fields is a blue button labeled "SE CONNECTER" with a right-pointing arrow. A close button (X) is located in the top right corner of the window.



## Ajout d'actualité

Ajout

Type \*


Actualité

Titre \*

test

Site \*

Belledonne



le-grand-colon-(a-droite-de-l'image)-vu-depuis-grenoble-photo-d-archives-le-di-michel-paul-1577988773.jpg

IMAGE

Taille maximale : 500MB

Description \*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas congue felis placerat, finibus ante sed, pharetra magna. Suspendisse fermentum velit et purus laoreet cursus. Pellentesque varius orci et ante ultrices faucibus. Suspendisse sodales ornare semper. Ut eget sem mi. Morbi aliquam sapien sit amet ante auctor viverra. Praesent cursus quis sem at tempus. Suspendisse malesuada bibendum dolor.

ENVOYER

## Ajout de connaissances produites

Ajouter

Type \*

Connaissances produites

Titre \*

test

Activités humaines

Climat

Fonctionnement démographique

Régimes alimentaires

Suivi sanitaire

Utilisation spatiale

Taille maximale : 500MB

Mots clés

Envoyer

## Visualisation d'actualité

## Visualisation de pdf

[illegible]

## Fonctionnalités restant à implémenter

- Création de comptes avec l'interface graphique
- Mise en place d'un système de pages pour accélérer l'application lorsque la taille de la base de données augmentera

## Difficultés rencontrées

Dans l'ensemble, le développement de l'application s'est déroulé sans embûches majeures.

Bien que de nombreuses difficultés se sont présentées au cours du développement, j'ai réussi à les surmonter sans perdre trop de temps notamment grâce à l'aide de connaissances plus familières avec ReactJS.

Les conditions liées au COVID-19 lors du déroulement du stage m'ont contraint à télé-travailler une bonne partie du stage limitant ainsi les échanges avec mon maître de stage et le service concerné par l'application. J'ai pu néanmoins mener à bien le développement de l'application depuis mon domicile.

# Conclusion

Pour conclure j'ai pu, lors de ce stage de fin de license professionnelle, mettre en application un grand nombre de connaissances abordées en cours. Ce stage était donc à la fois très formateur en ce qui concerne le développement web (notamment le framework ReactJS) mais aussi en gestion de projet.

La liberté offerte par mon maître de stage à la fois sur les technologies et le déroulement du développement m'a permis de réellement m'épanouir tout au long du stage.

Je suis certain que la réalisation de ce projet de A à Z, me sera très bénéfique lorsque j'entrerai sur le marché du travail.