



Développeur Web

RAPPORT TECHNIQUE

Du 17 mai au 23 juillet

Étudiant

Florian FAHY-NOURDIN

Tuteur

Stéphane DOMAS

Sommaire

Frameworks, modules et langages requis :.....	3
Fichiers :.....	3
Explication d'un schéma mongoose :.....	3
Descriptif des routes :.....	4
Comment créer une nouvelle route ?.....	5

Frameworks, modules et langages requis :

- VueJS
- Vuetify
- ExpressJS
- cors
- MongoDB
- Mongoose
- mongoose-mpath
- Javascript

Fichiers :

La hiérarchie de fichier se décompose ainsi :

- Un dossier pour les vues
 - Un dossier pour chaque schéma
 - Un fichier par page
- Un dossier pour les schémas
 - Un fichier par schéma
- Un dossier pour le serveur
 - Un fichier index.js servant à exécuter le serveur et l'API

Explication d'un schéma mongoose :

```
const enseignantSchema = new Schema({  
  nom: String,  
  prenom: String,  
  surnom: String,  
  email: String,  
  statut: { type: Schema.Types.ObjectId, ref: 'Statut' },  
});
```

Au format JSON, on met le nom du champ en clé et le type attendu en valeur. Pour permettre une référence à un autre objet en valeur, en plus de mettre le type ObjectId dans la clé « type », on spécifie quel objet on attend dans la clé « ref ».

Il est également possible d'utiliser les crochets pour spécifier que l'on utilisera ce champ comme un tableau.

Descriptif des routes :

/getElements/:idFormation	: récupère la décomposition (les éléments) d'une formation sous forme d'arbre grâce au module mpath
/addEnseignant	: Ajoute un nouvel enseignant à la BDD
/editEnseignant	: Modifie l'enseignant avec l'id passé
/addStatut	: Ajoute un statut
/editStatut	: Modifie un statut avec l'id passé
/deleteEnseignant	: Supprime l'enseignant avec l'id passé
/deleteStatut	: Supprime le statut avec l'id passé
/addProjet	: Ajoute un nouveau projet
/addEnseignantsToProjet	: Ajoute les enseignants au projet spécifié
/createFormation	: Ajoute une nouvelle formation
/createElement	: Crée un nouvel élément et l'ajoute aux fils du père
/modifyElement/:id	: Modifie l'élément avec l'id passé
/modifyVolumeHebdo	: Modifie le volume hebdomadaire de l'élément avec l'id passé

`/addIntervenantToElement` : Ajoute des intervenants (enseignants) à l'élément avec l'id passé

`/getStatuts` : Retourne la liste des statuts

`/getIntervenants/:projectId` : Retourne les intervenants (enseignants) associés au projet avec l'id passé

`/getEnseignants` : Retourne les enseignants

`/getProjets` : Retourne les projets

`/getFormation/:idFormation` : Retourne la formation avec l'id passé

Comment créer une nouvelle route ?

```
app.post('/addEnseignant', async (req, res) => {  
  new Enseignant({  
    _id: new mongoose.Types.ObjectId(),  
    nom:req.body.nom,  
    prenom:req.body.prenom,  
    surnom:req.body.surnom,  
    email:req.body.email,  
    statut:req.body.statut  
  }).save()  
})
```

```
app.post('/editEnseignant', async (req, res) => {  
  Enseignant.updateOne({_id:req.body.id}, {  
    nom:req.body.nom,  
    prenom:req.body.prenom,  
    surnom:req.body.surnom,  
    email:req.body.email,  
    statut:req.body.statut  
  })  
})
```

```
app.get('/getStatuts', async (req, res) => {  
  res.send(await Statut.find({}))  
})
```

Pour répondre à la question, étudions comment ces 3 routes sont faites.

- Pour `/addEnseignant`, on commence crée un « new Enseignant », et on lui passe en paramètres un JSON rempli par les informations envoyé depuis la page web.

Ensuite on n'oublie pas « .save() » pour que ce soit ajouté à la base de données

- Pour /editEnseignant on utilise la fonction « updateOne » du schéma Enseignant, avec comme paramètre l'id de l'enseignant en question puis les données. L'id ressemble à quelque chose du genre « 60afae5cc2f83f3e580c023d ».
- Dans /getStatus on utilise la fonction « find() » qui permet de trouver tous les statuts dans la base de données.

A noter qu'ici « {} » est vide car nous voulons tous les statuts, mais si vous cherchez un statut, vous pouvez spécifier en écrivant « {_id:<id>} » comme vu pour la fonction update et en utilisant de préférence la fonction « findOne() »