

Guide d'étapes clés - Développez le back-end d'un site de notation de livres

Comment utiliser ce document ?

Ce guide vous propose un découpage du projet en étapes. Vous pouvez suivre ces étapes selon vos besoins. Dans chacune, vous trouverez :

- des recommandations pour compléter la mission ;
- les points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet (attention, celui-ci peut varier d'un apprenant à l'autre).

Suivre ce guide vous permettra ainsi :

- d'organiser votre temps ;
- de gagner en autonomie ;
- d'utiliser les cours et ressources de façon efficace ;
- de mobiliser une méthodologie professionnelle que vous pourrez réutiliser.

Gardez en tête que votre progression sur les étapes n'est qu'une estimation, et sera différente selon votre vitesse de progression.

Recommandations générales

Dans ce projet, vous allez utiliser et mettre en place beaucoup de choses nouvelles (notamment avec Node.js, Express, MongoDB). Il est donc plus important que jamais de suivre sérieusement et méthodiquement les cours, et notamment le cours [Passez au Full Stack avec Node.js, Express et MongoDB](#), qui va vous permettre de mettre en place, étape par étape, une base de code solide que vous devrez adapter pour coller au projet (étape 1 de ce document).

Une fois le cours terminé et le code adapté, vous devrez vous employer à implémenter le nécessaire pour ajouter toute la fonctionnalité de notation des livres (étape 2 de ce document).

Étape 1 : Mise en place de l'application

Pour cette étape, vous pouvez suivre en détail le cours *Passez au Full Stack avec Node.js, Express et MongoDB*.

Soyez **extrêmement vigilant** lorsque vous allez reproduire ce qui est déroulé dans le cours. Il est très très fréquent qu'une toute petite erreur de frappe soit à l'origine du non-fonctionnement du projet. Un exemple typique d'erreur : dans le cours on parle de “**t**hing”, de “**T**hing”, de “**t**hings”, et de “**T**hings”. Une lettre peut faire toute la différence.

Lors de la mise en place du code présenté dans le cours, prenez bien le temps d'identifier ce qui doit être adapté pour le projet. Par exemple, dans le cours, on parle de “things”, alors que dans le projet nous traitons de “books” (les livres). Pensez également au fait que les propriétés associées à une “thing” ne sont pas forcément les mêmes que les propriétés associées à un “book”.

Étape 1.1 : Créez un serveur Express simple

20 % de progression

Une fois cette étape terminée, je devrais avoir :

- Un environnement de développement installé.
- Un serveur Node fonctionnel.

Recommandations :

Suivez scrupuleusement la partie 1 du cours. Cette partie va vous permettre :

- de configurer un environnement de développement avec toutes les dépendances requises ;
- de démarrer et d'utiliser un serveur Node pour gérer une première application Express ;
- et de créer de premières routes dans votre application.

Ressources :

- [Partie 1](#) du cours “Passez au Full Stack avec Node.js, Express et MongoDB”.

Étape 1.2 : Créez une API RESTful

40 % de progression

Une fois cette étape terminée, je devrais avoir :

- Une base de données MongoDB installée et connectée à mon application Express.
- Le modèle de données complet sur MongoDB.

Recommandations :

Suivez scrupuleusement la partie 2 du cours. Cette partie va vous permettre de :

- configurer et connecter une base de données à l'application Express ;
- d'utiliser Mongoose pour créer un modèle de données, afin de faciliter les opérations de la base et l'implémentation des routes CRUD, pour rendre l'application dynamique.

Ressources :

- [Partie 2](#) du cours “Passez au Full Stack avec Node.js, Express et MongoDB”.

Étape 1.3 : Mettez en place un système d'authentification sur votre application

60 % de progression

Une fois cette étape terminée, je devrais avoir :

- L'authentification des utilisateurs implémentée et fonctionnelle.

Recommandations :

Suivez scrupuleusement la partie 3 du cours. Cette partie va vous permettre de :

- stocker les informations utilisateur dans votre base ;
- implémenter le cryptage de mot de passe ;
- créer et envoyer des tokens au front-end pour authentifier les requêtes ;
- ajouter un middleware d'authentification pour sécuriser les routes dans votre API.

Ressources :

- [Partie 3](#) du cours “Passez au Full Stack avec Node.js, Express et MongoDB”.

Étape 1.4 : Ajoutez une gestion des fichiers utilisateur sur l'application

80 % de progression

Une fois cette étape terminée, je devrais avoir :

- Implémenté toutes les fonctionnalités d'affichage, création et modification des livres.

Recommandations :

- Suivez scrupuleusement la partie 4 du cours. Cette partie va vous permettre d'avoir une base de code solide.
- Une fois Multer ajouté, pensez à optimiser les images envoyées par les utilisateurs.

Ressources :

- [Partie 4](#) du cours “Passez au Full Stack avec Node.js, Express et MongoDB” ;
- [La documentation de Node JS](#) ;
- [La documentation d'Express](#) où on peut y retrouver :
 - l'onglet “getting started” pour avoir des informations sur [l'installation](#) par exemple, ou bien une [FAQ](#) ;
 - l'onglet “Guide” pour obtenir des informations à propos du [routing](#), de [l'écriture](#) ou [l'utilisation](#) de middleware ;
- [La documentation de MongoDB](#).
- Un [article de dev.to](#) sur la compression d'images avec un serveur Express, Multer et Sharp

À la fin de cette Etape 1 (de 1.1 à 1.4), vous aurez ainsi :

- créé et utilisé le serveur Node pour servir l'application Express connectée à la base de données MongoDB ;
- utilisé Mongoose pour connecter votre serveur à la base de données MongoDB et permettre les opérations du CRUD ;
- implémenté l'authentification sécurisée (token JWT) ;
- implémenté la gestion de téléchargement et d'optimisation des images.

Étape 2 : Fonctionnalité de notation

Étape 2.1 : Gestion de l'ajout d'une notation d'un livre

90% de progression

Une fois cette étape terminée, je devrais avoir :

- Implémenté la fonctionnalité de notation de livres.

Recommandations :

Prenez le temps de réfléchir à ce que la fonctionnalité de notation d'un livre va induire dans la structuration de votre projet :

- Quel est l'impact sur le modèle de données ? Il semble intéressant, pour ne pas dire indispensable, de se reporter au Model "Book" fourni dans les spécifications techniques de l'API.
- Quel impact sur les routes ?

Pour chaque livre, chaque utilisateur va pouvoir renseigner une notation. Dans le modèle "Book" on peut retrouver un tableau "ratings" qui contiendra les notes des différents utilisateurs pour un livre donné. Il sera donc nécessaire d'ajouter un élément dans ce tableau pour chaque nouvelle notation saisie.

Points de vigilance :

Attention à ne pas permettre à un même utilisateur de renseigner plusieurs notation sur un seul et même livre. Pensez donc à vérifier, dans le tableau "ratings", que l'utilisateur n'a pas déjà renseigné une notation.

Étape 2.2 : Gestion du calcul de la note moyenne d'un livre

100 % de progression

Une fois cette étape terminée, je devrais avoir :

- Implémenté la fonctionnalité calcul de la moyenne d'un livre.
- L'application finalisée.

Recommandations :

- Maintenant qu'il est possible de renseigner la notation d'un livre, il faut s'assurer de tenir à jour sa note moyenne. Pensez à bien recalculer la note moyenne associée au livre après chaque nouvelle note saisie.
- Cette note moyenne devra être stockée dans la propriété "averageRating" du modèle "Book".

Projet terminé !