

# Étapes clés pour le P6 du parcours Développeur Web

Création d'une API sécurisée pour une application d'évaluation

Dans ce document, vous trouverez un exemple des étapes clés à suivre pour mener à bien votre projet. Vous verrez :

- Quelles parties du livrable correspondent aux différentes étapes clés.
- Les méthodes pour mener à bien chaque étape.
- Les problèmes potentiels à connaître ou les erreurs à éviter.
- Le temps nécessaire estimé de l'ensemble du projet.
- Des ressources externes utiles pour chaque étape.

## Recommandations générales

À chaque étape, assurez-vous que le serveur démarre correctement, avec un message dans la console indiquant que le serveur a démarré et s'est connecté avec succès à MongoDB. Si vous recevez un message indiquant que le serveur ne parvient pas à démarrer, vérifiez qu'il n'y ait pas d'erreur dans la console du backend.

Testez votre code à chaque grande étape. Des tests fréquents peuvent vous éviter d'oublier des parties importantes du code et vous alerter sur les erreurs.

## Étape 1 : Démarrer le serveur backend

15 % d'avance

 **Partie du livrable :** Projet initialisé

 **Recommandations :**

- Suivez ces étapes :
  - Créer un projet vide pour démarrer le serveur Node.js ;
  - Installer Express ;
  - Installer Mongoose.

- À partir de la version 4.16 d'Express, bodyparser est inclus et vous n'avez pas besoin de l'installer.
  - Utilisez ( `express.json()` ) pour analyser le corps de la requête.

#### ⚠ Les problèmes à connaître :

- *Si le port 3000 est utilisé par un autre processus*, redémarrez complètement votre ordinateur (pour permettre l'utilisation du port) ou changez le port utilisé dans l'application Express.
- *Impossible de se connecter à MongoDB*. Vérifiez la chaîne de connexion, le nom d'utilisateur et le mot de passe de MongoDB et vérifiez que MongoDB Atlas (ou un service similaire) autorise toutes les adresses IP à se connecter au cluster.



#### Ressources (en anglais)

- [Utilisation d'Express](#)
- [Comment connecter et utiliser votre MongoDB avec Node.js](#)
- [Travailler avec des variables d'environnement](#)

## Étape 2 : Construire le parcours utilisateur

30 % d'avance



#### Partie du livrable : API



#### Recommandations :

- Créez les éléments suivants :
  - Modèle d'utilisateur ;
  - Parcours utilisateur ;
  - Contrôleur d'utilisateur.
- L'utilisateur est en mesure d'effectuer les opérations suivantes :
  - Créer un compte ;
  - Se connecter et disposer d'un token valide.
- Consultez l'onglet réseau de Devtools pour plus d'informations.

#### ⚠ Les problèmes à connaître :

- *Le mot de passe n'est pas haché*. Veillez à hacher le mot de passe.
- *Un utilisateur peut s'inscrire plusieurs fois avec la même adresse électronique*. Assurez-vous que le code vérifie qu'une adresse électronique est unique.



#### Ressources

- [Ce que sont les verbes de requêtes d'API.](#)

## Étape 3 : Démarrer le middleware

40 % d'avance

### Partie du livrable : API

#### Recommandations :

- Ajout de multer pour les images.
- Ajout d'authorize pour la validation des tokens.
  - Authorize doit être ajoutée avant de commencer à construire le parcours pour les sauces car l'authentification est nécessaire pour qu'un utilisateur puisse effectuer une action sur le parcours des sauces.

#### Les problèmes à connaître :

- Les images importées sont manquantes.
- Multer n'est pas correctement configuré.
- Le chemin statique n'a pas été ajouté à l'application pour fournir les images. Assurez-vous d'ajouter le chemin statique à l'application.

#### Ressources

- [Configuration de multer](#) (en anglais) .
- [Configuration de multer \(en français\)](#)

## Étape 4 : Construire la route Sauce de l'API

70 % d'avance

### Partie du livrable : API

#### Recommandations :

- Créez les éléments suivants :
  - Le Modèle Sauce ;
  - La Route Sauce ;
  - Le Contrôleur Sauce.
- Autorisez toutes les fonctions en utilisant middleware Authorize.
- L'utilisateur est en mesure d'effectuer les opérations suivantes :
  - Ajouter une nouvelle sauce ;
  - Supprimer une sauce ;
  - Voir toutes les sauces.
- Consultez l'onglet réseau de Devtools pour plus d'informations.

#### Les problèmes à connaître :

- Erreur 401 (l'utilisateur n'est pas autorisé).
- Multer ne sauvegarde pas les images.
- Les images ne sont pas affichées sur le frontend.

## Ressources

- [Ajout ou suppression d'un fichier de gitignore.](#)
- [Les méthodes des tableaux expliquées.](#)

## Étape 5 : Terminer la route Sauce de l'API

100 % d'avance

---

### Partie du livrable : API complété

#### Recommandations :

- Exécutez l'application en tant qu'utilisateur pour vérifier que toutes les fonctions ont été correctement mises en œuvre, testez :
  - Les deux types de demandes :
    - Avec un fichier présent ;
    - Sans fichier.
  - Les trois scénarios de la fonction « like » (1, 0, -1) ;
    - L'utilisateur peut liker ou ne pas aimer une sauce (ou aucun des deux)
  - Seul le propriétaire de la sauce peut modifier ou supprimer une sauce existante.

#### Les problèmes à connaître :

- Erreur 401 (l'utilisateur n'est pas autorisé).
- Multer ne sauvegarde pas les images.
- Les images ne sont pas affichées sur le frontend.
- Les données ne sont pas modifiées lorsque l'utilisateur tente de modifier une sauce existante.
- La fonction « modifier » échoue lorsqu'une image est téléchargée ou modifiée.
- La fonction « like » échoue lorsque l'utilisateur essaie de liker ou de ne pas aimer une sauce plusieurs fois.
- Le propriétaire de la sauce ne peut pas voir les boutons « modifier » et « supprimer ». L'identifiant de la Sauce doit être valide et ne pas contenir de faute de frappe, car seul le propriétaire de la Sauce peut la modifier ou la supprimer.

 **Projet terminé !**