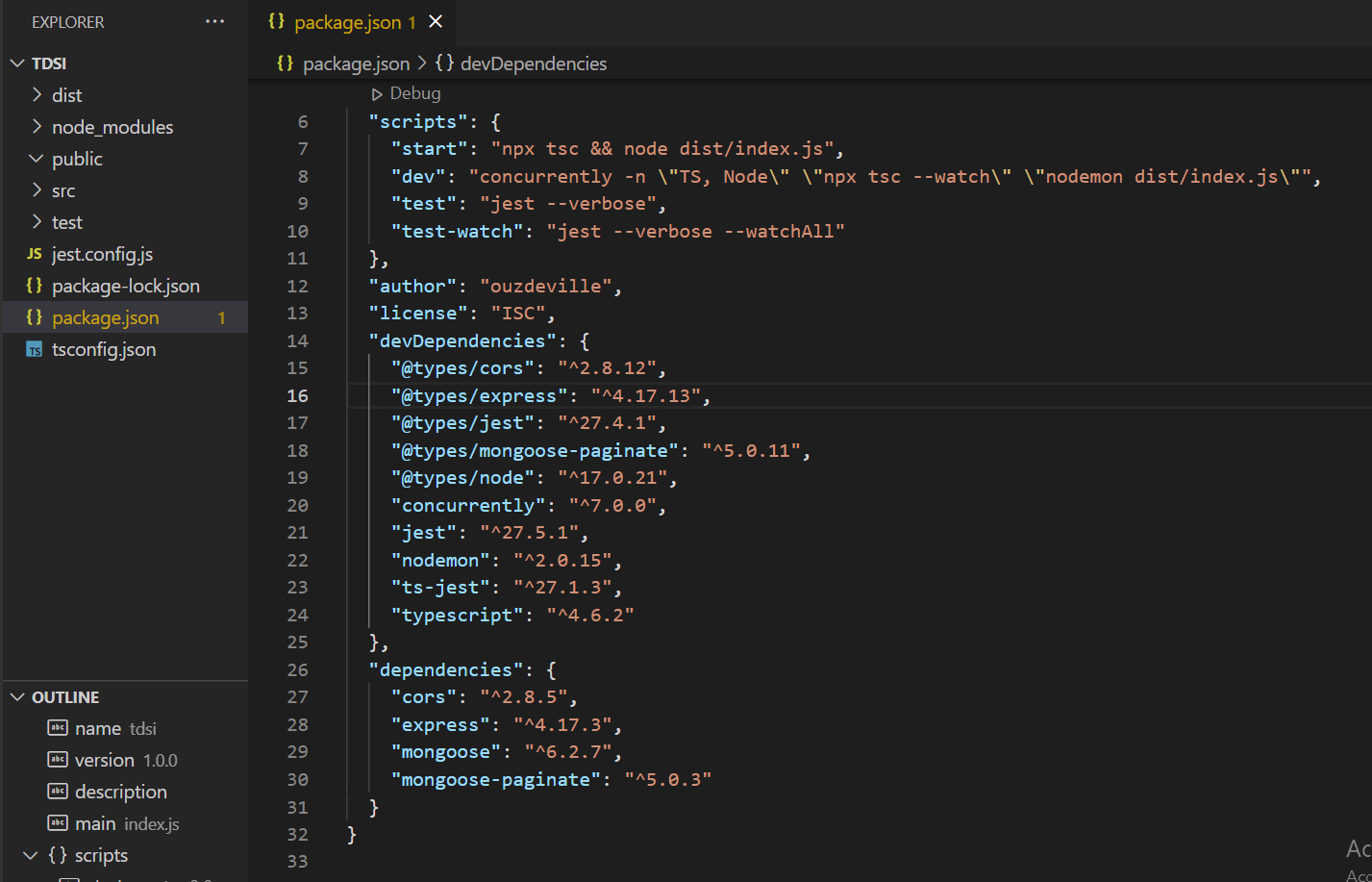
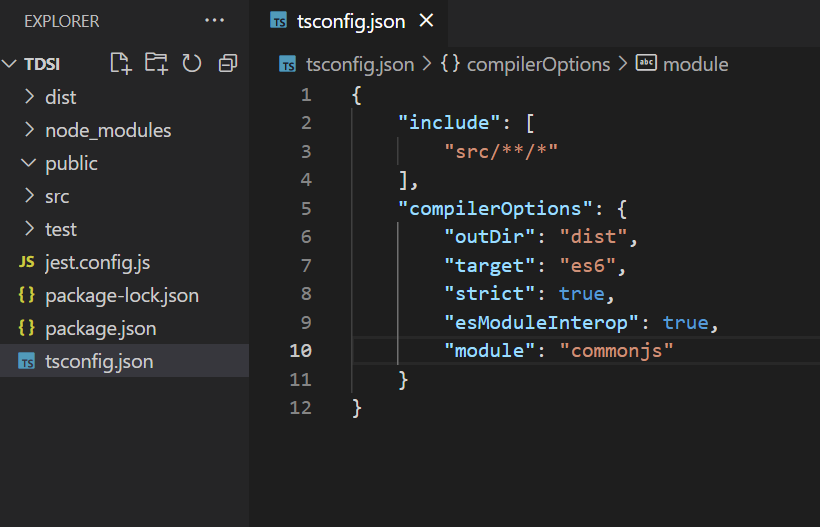
1. Packages à Installer un à un:
   1. npm install --save «**dependencie**»
   2. npm install --save-dev «**devdependencie**»
   3. Cinfiguration des Scripts pour les différents stages

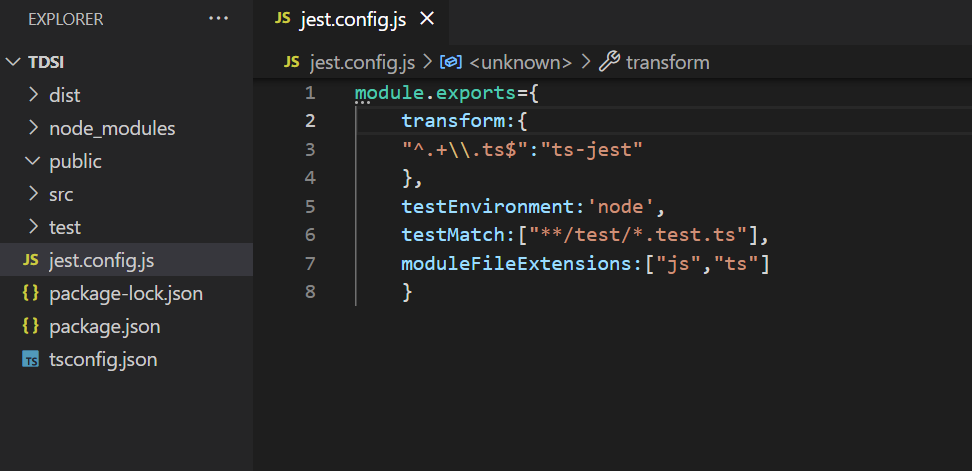


1. Configuration de TypeScript : les sous repertoire de **src** seront compilés et les fichiers js placés le répertoire **dist**

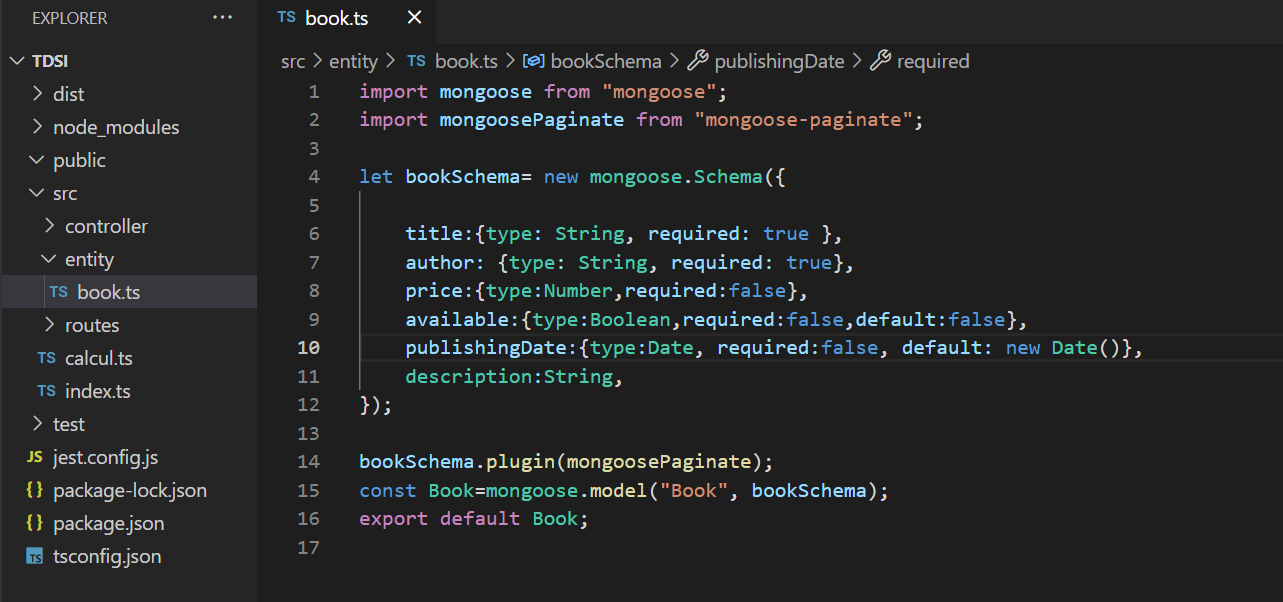


1. Configuration de **Jest** pour les **Tests unitaires**

Les fichiers dont le nom termine par **".test.ts"** dont le chemin contient **"test"**

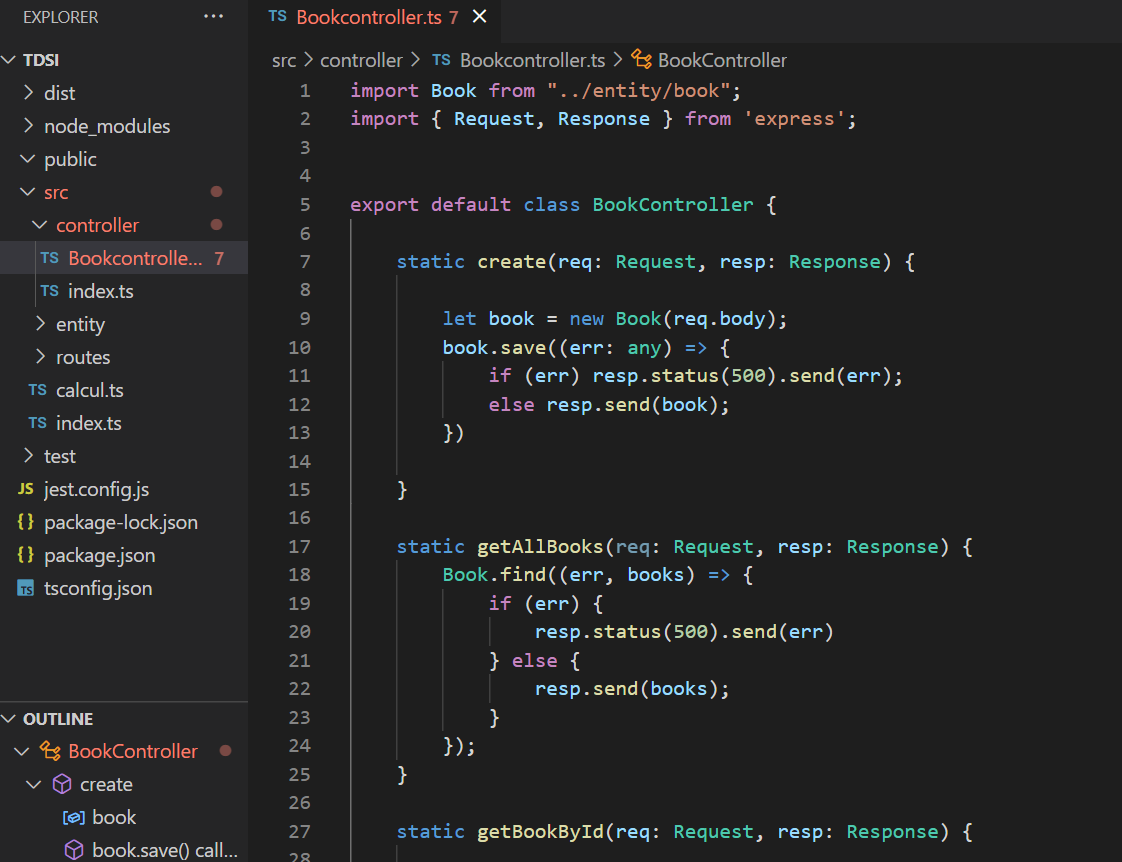
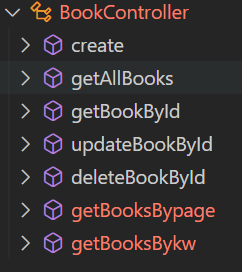


1. **Création des Entité avec Mongoose dans le répertoire entity**



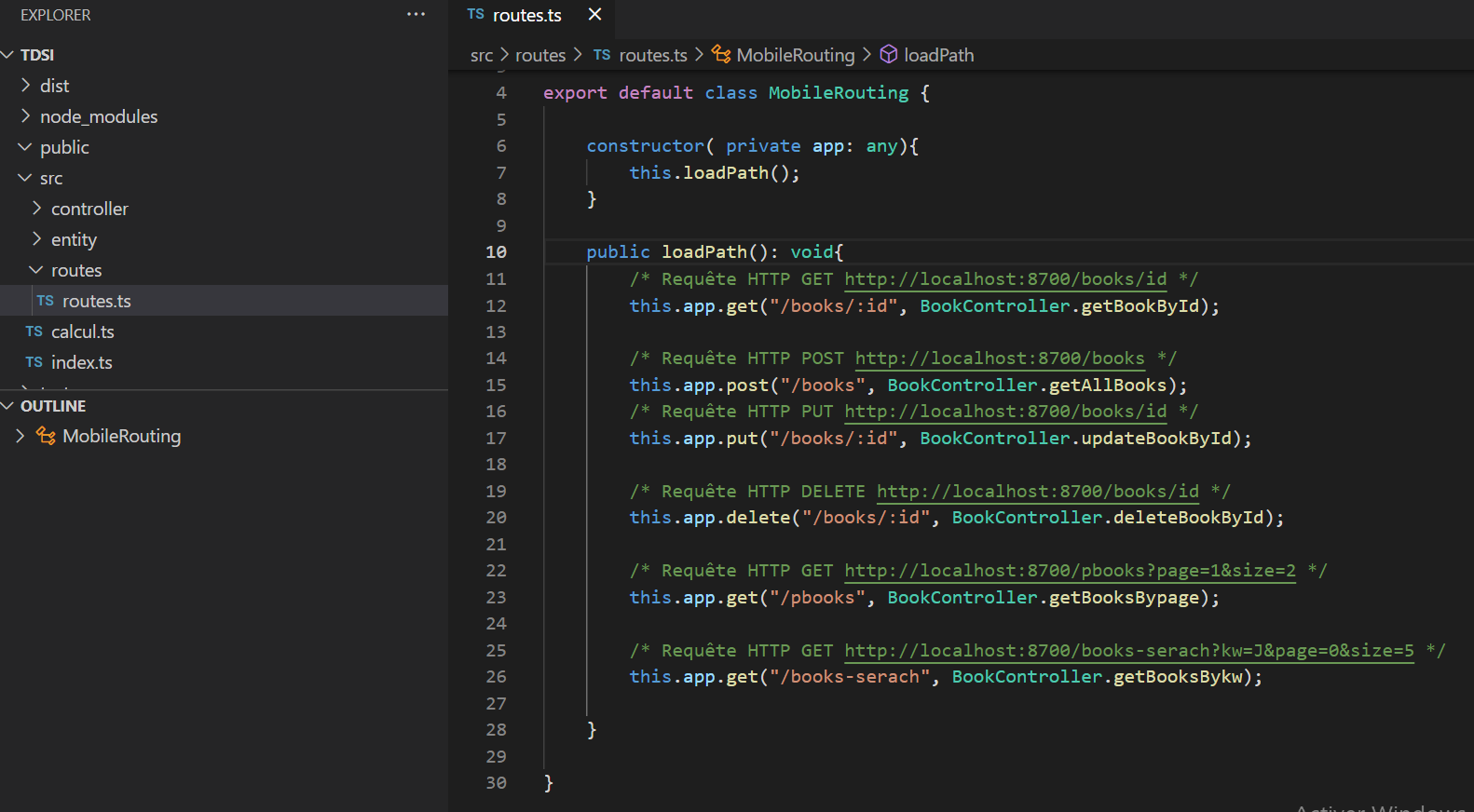
1. Création des Controller dans le répertoire **controller**

Il est préférable de créer un controller par entité et de noter que chaque méthode aura ici pour intput req and response.



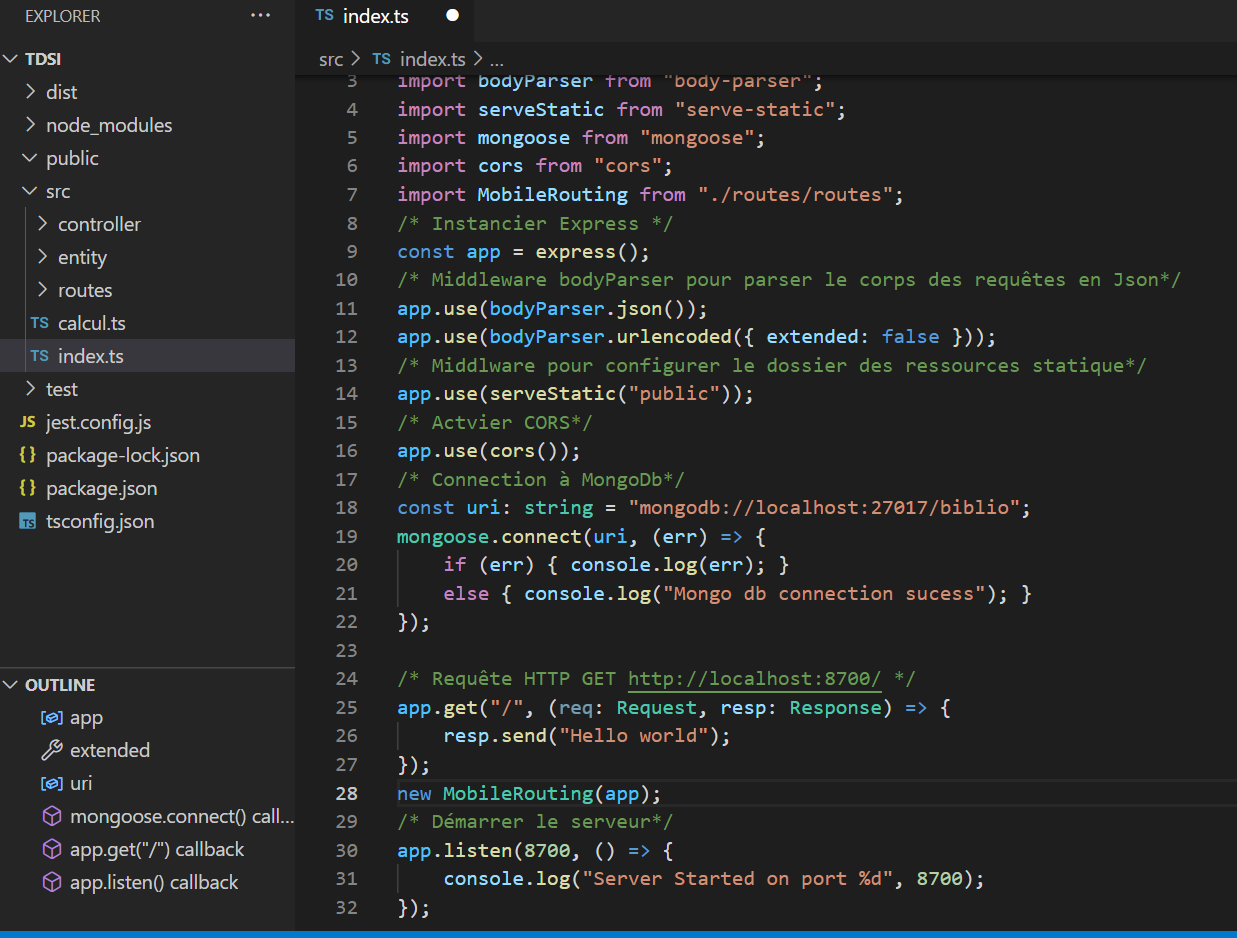
1. **Création des routes pour le Mobile dans le répertoire routes**

Comme les méthodes sont créés au niveau des controller, il faut les lier aux URLs et Au méthodes http(POST, GET, PUT, DELETE) dans des classes de routage.



1. **Intégration du routage dans l’index du backend**

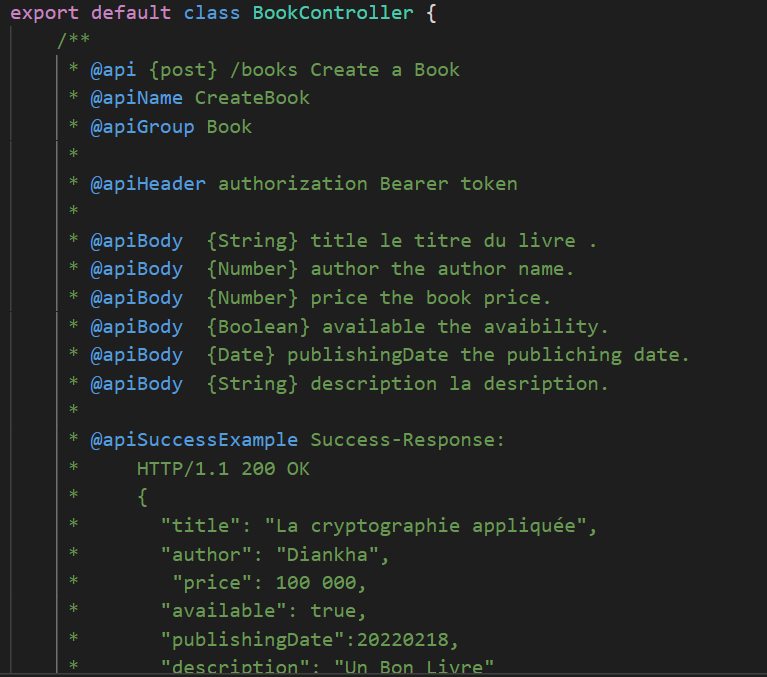
Il faut s’assurer que les routages soient chargés au lancement de l’application cf. ligne 28.

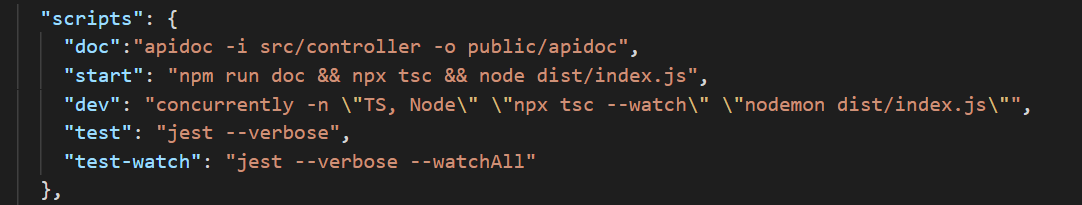


1. Documentation des API nodejs avec **apiDoc**

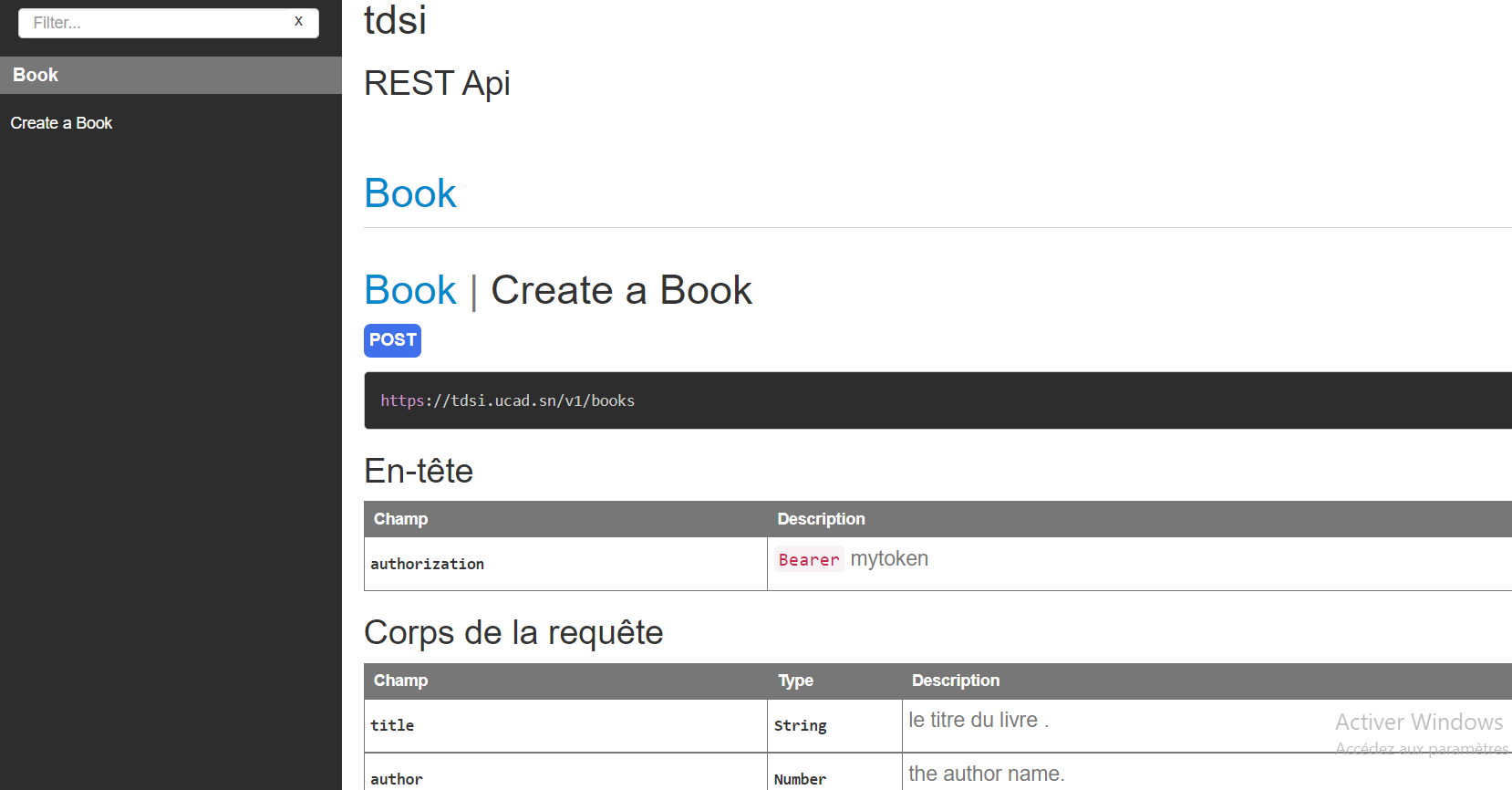
**ApiDoc** permet de faire la description des API pour faciliter leur consommassion auprès des clients. La Description permet de savoir quel méthode http utilisée, quelle url, quels params envoyés.

* 1. npm install apidoc –save

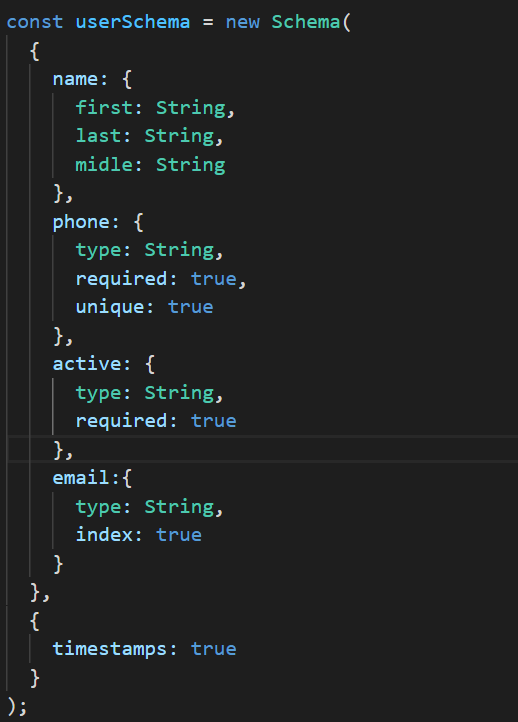
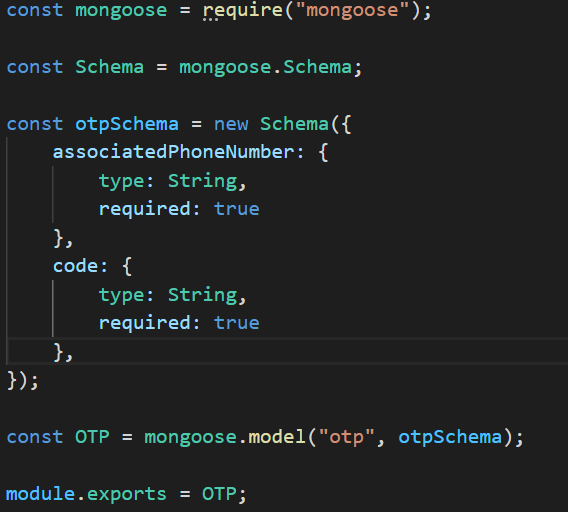




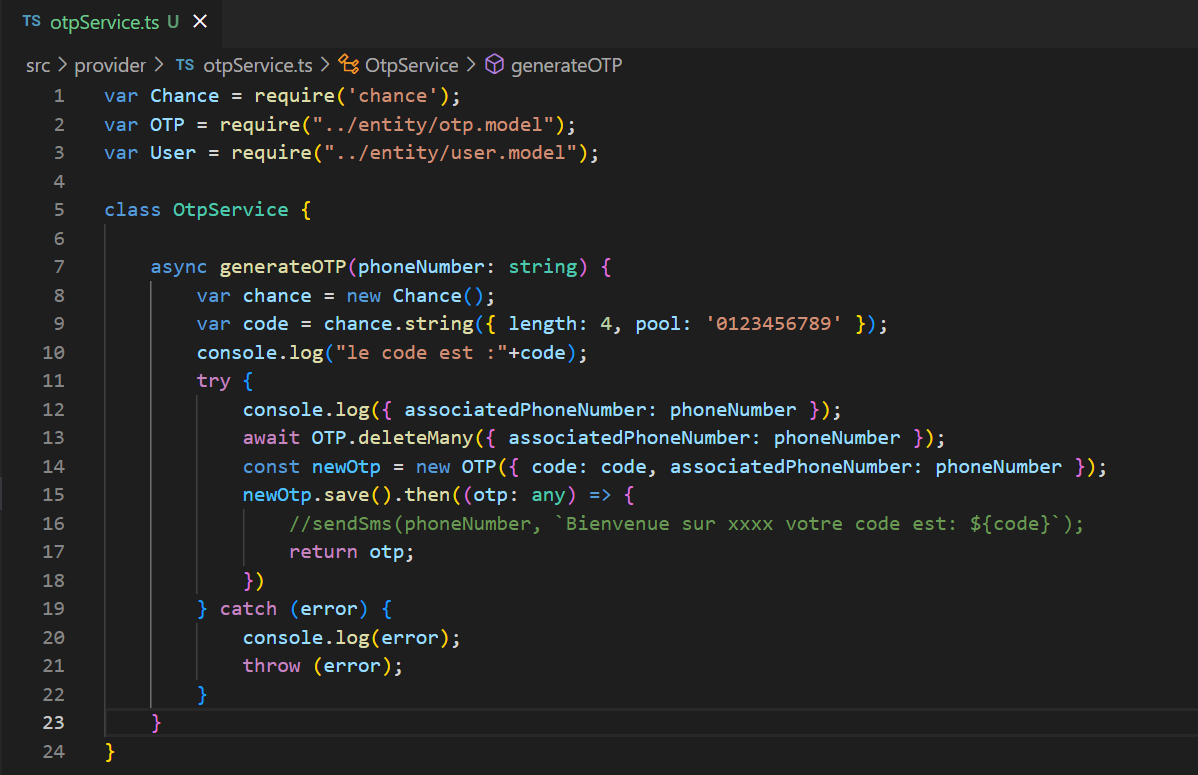
* npm run start
* <http://localhost:8700/apidoc/>



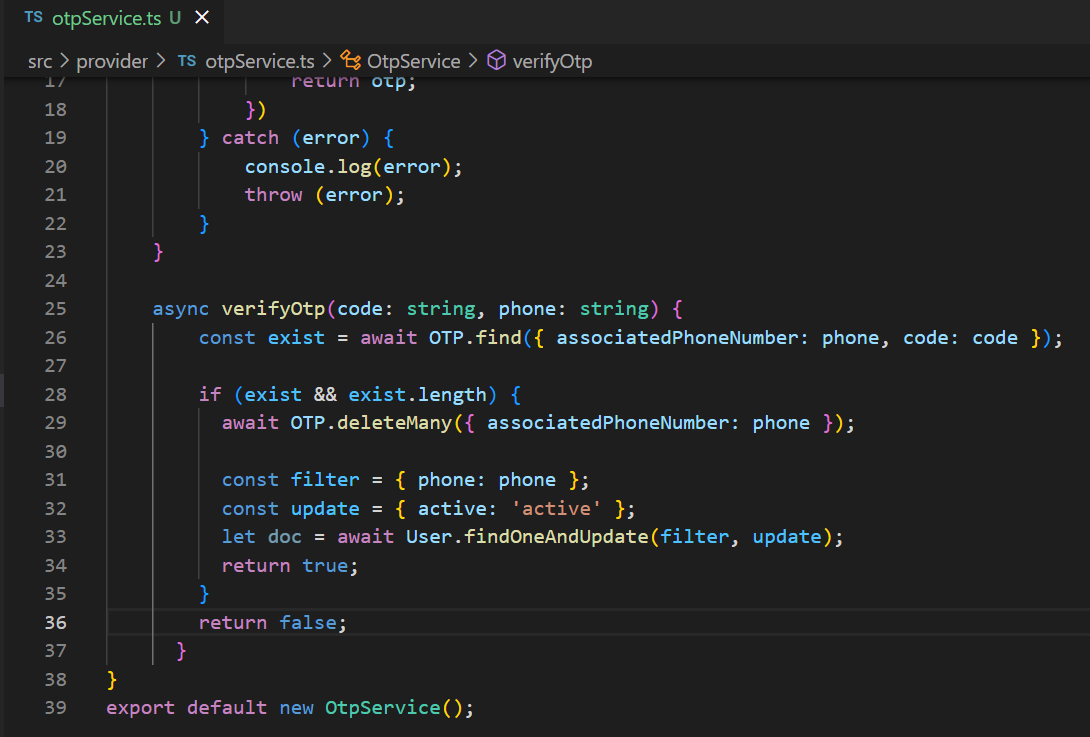
1. Intégration de Provider/Service : Exemple pour la génération de Token
   1. **Création des Entité User et OTP dans le dossier entity**

* 1. **Génération d’OTP et envoie au client dans le répertoire provider**



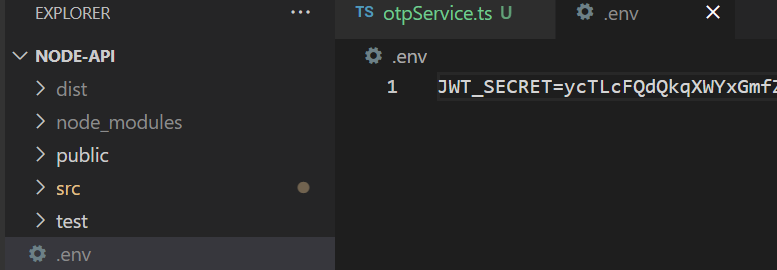
* 1. **Vérification d’OTP**



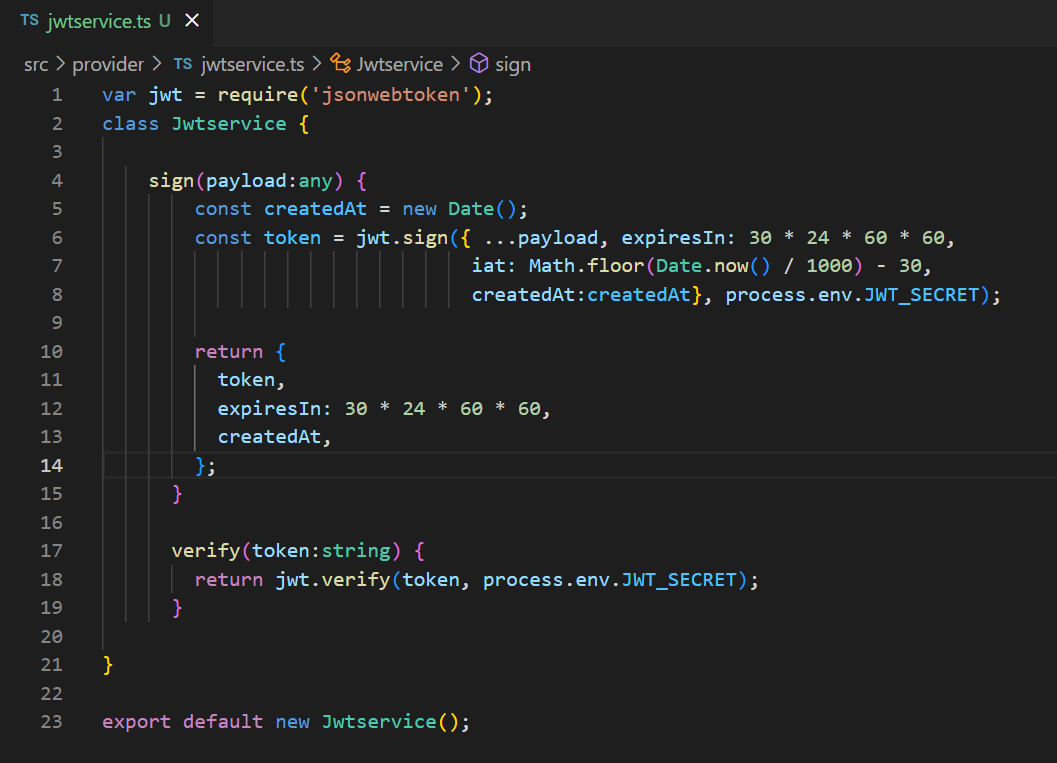
* 1. **JWT Service : sign et verify,**

npm install –save **jsonwebtoken**

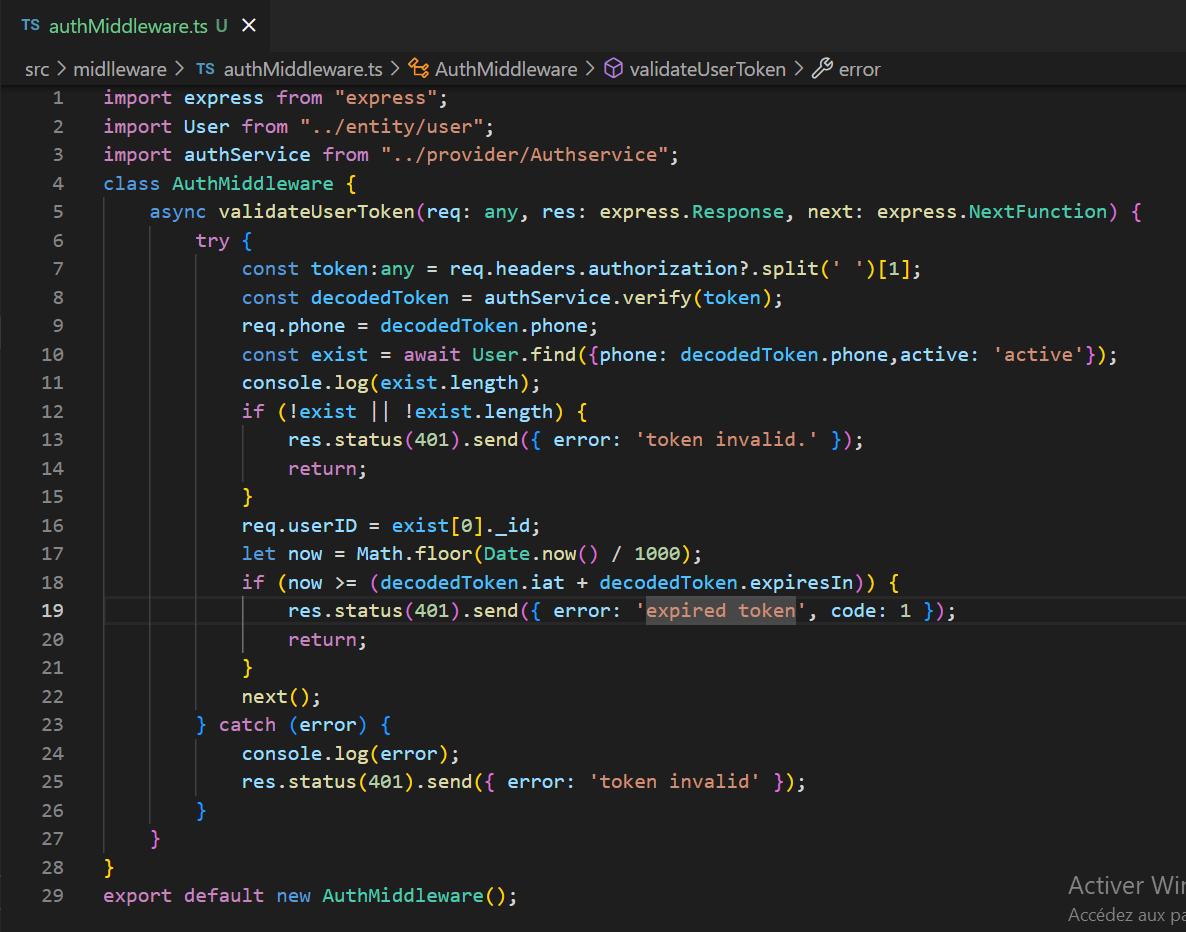
Créer un fichier "**.env**" qui contient la variable "**JWT\_SECRET**"



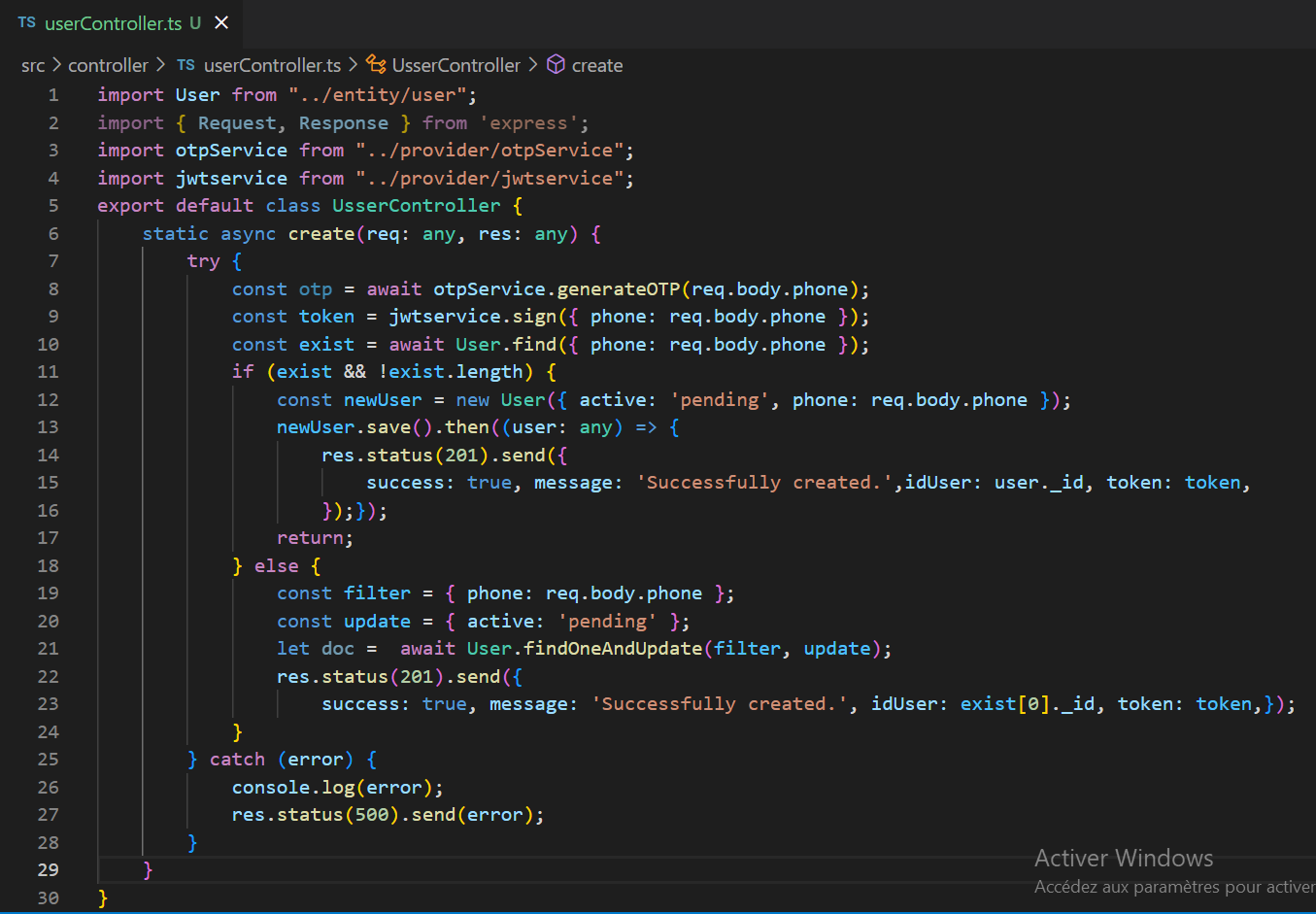
Créer la classe **Jwtservice** dans le rep **provider**



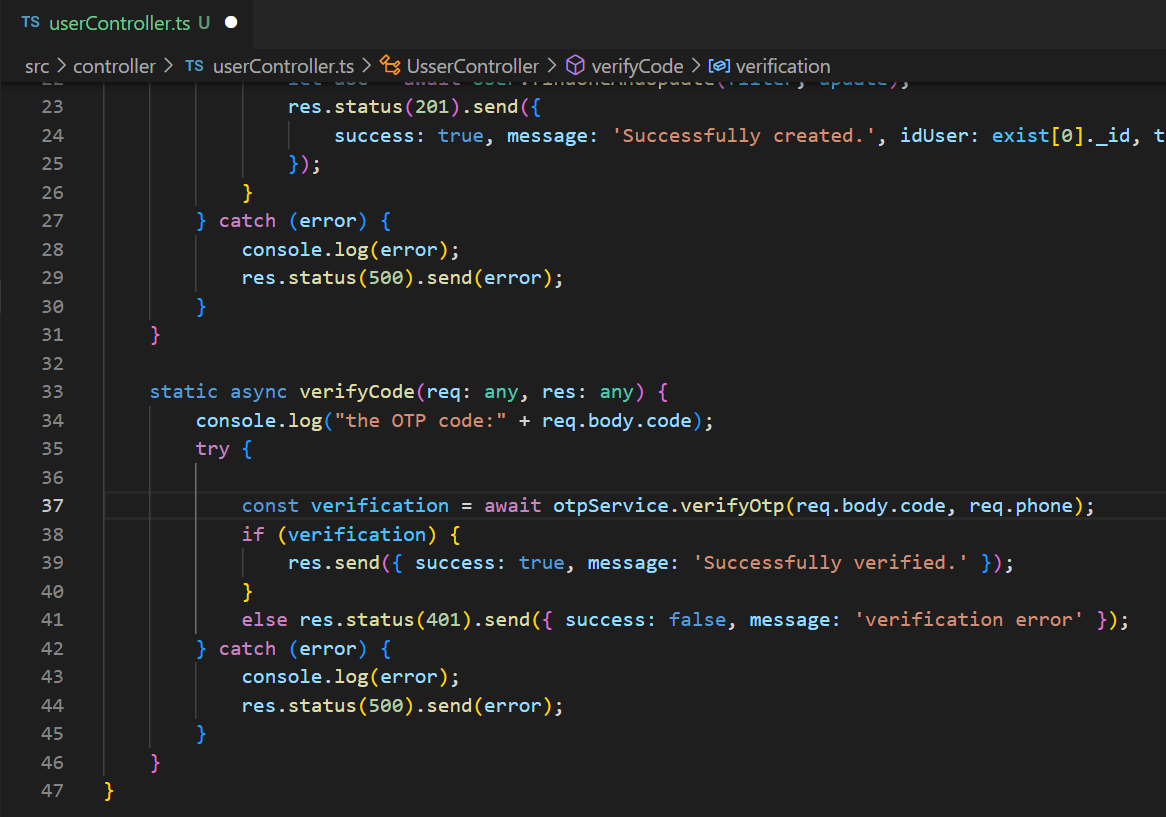
1. Intégration du middleware pour l’authentification des requêtes :
   1. Créer **authMiddleware** dans le rep **middleware** :



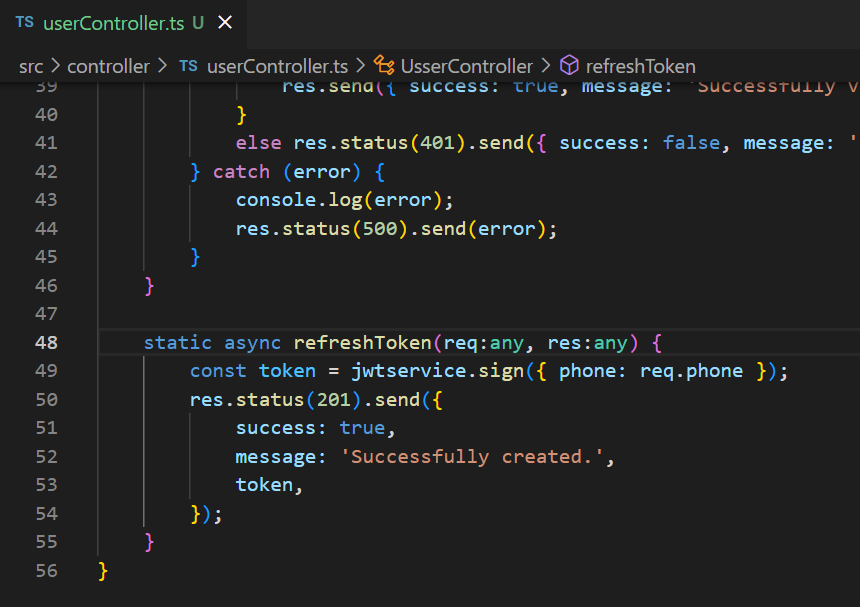
1. Inscription d’un utilisateur :
   1. Créer la classe **UserController** dans controller
   2. Génération d’un otp (avec inscription dans la BD et 1 sms send au client)
   3. Génération d’un **token jwt**
   4. **Rechercher le user à partir du phone**
      1. S’il n’existe pas créer un user, l’enregistrer dans la BD et send une réponse au client.
      2. Sinon mettre à jour le statut du client et send une réponse au client.



* 1. Vérification du code pour changer le **status** du compte



* 1. Renouvellement du token



Ne pas oublier de les documenter avec API doc

1. Intégration d’un Middleware dans le routage : exemple pour l’authentification

