

Prompt chat gpt

- Je suis entrain de faire une api avec express

- donne moi le code http si l'user n'existe pas

- il n'y a pas de code http plus précis pour cette error ?

- quel est le code status pour une modification de donné ?

- j'utilise aussi swagger-express-ui

- voici mon model

```
const mongoose = require("mongoose")
const Schema = mongoose.Schema;
```

```
let userSchema = new Schema ({
  email: {
    type: String,
    required: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  }
});
```

```
module.exports = mongoose.model("User", userSchema)
```

- voici mes routes :

```
// const jwtMiddleware =
require("../middlewares/jwtMiddleware")
```

```
module.exports = (app) => {
  const userController = require("../controllers/userController")
  app.route("/users/register")
```

```

- .post(userController.userRegister);
- app.route("/users/login")
- .post(userController.userLogin);
- app.route("/users")
- .get(userController.listenAllUsers);
- app.route("/users/:user_id")
- // .all(jwtMiddleware.verifyToken)
- .get(userController.listenSingleUser)
- .put(userController.updateUser)
- .delete(userController.deleteUser);
-
- }

```

fait moi les commentaires swagger pour mes routes stp

```

voici mon fichier route : const jwtMiddleware =
require("../middlewares/jwtMiddleware")
/**
 * @swagger
 * components:
 *   schemas:
 *     User:
 *       type: object
 *       required:
 *         - email
 *         - password
 *       properties:
 *         email:
 *           type: string
 *           format: email
 *           description: Adresse e-mail de l'utilisateur (unique)
 *         password:
 *           type: string
 *           description: Mot de passe de l'utilisateur
 *         createdAt:
 *           type: Data
 *           description : Date de création
 */

```

```
module.exports = (app) => {  
  const userController = require("../controllers/userController")
```

```
  /**  
   * @swagger  
   * path:  
   *   /users/register:  
   *     post:  
   *       summary: Inscription d'un nouvel utilisateur  
   *       tags: [Users]  
   *       requestBody:  
   *         required: true  
   *         content:  
   *           application/json:  
   *             schema:  
   *               $ref: '#/components/schemas/User'  
   *       responses:  
   *         '200':  
   *           description: Utilisateur enregistré avec succès  
   *         '400':  
   *           description: Erreur de validation des données  
   */  
  app.route("/users/register")  
    .post(userController.userRegister);
```

```
  /**  
   * @swagger  
   * path:  
   *   /users/login:  
   *     post:  
   *       summary: Connexion d'un utilisateur existant  
   *       tags: [Users]  
   *       requestBody:  
   *         required: true  
   *         content:  
   *           application/json:  
   *             schema:  
   *               $ref: '#/components/schemas/User'  
   *       responses:  
   *         '200':  
   *           description: Connexion réussie  
   *         '401':
```

```
*      description: Identifiants invalides
*/
app.route("/users/login")
    .post(userController.userLogin);
```

```
/**
 * @swagger
 * path:
 *   /users:
 *     get:
 *       summary: Récupérer la liste de tous les utilisateurs
 *       tags: [Users]
 *       security:
 *         - BearerAuth: []
 *       responses:
 *         '200':
 *           description: Liste d'utilisateurs récupérée avec succès
 *         '401':
 *           description: Non autorisé (JWT invalide ou manquant)
 */
app.route("/users")
    .get(userController.listenAllUsers);
```

```
/**
 * @swagger
 * path:
 *   /users/{user_id}:
 *     get:
 *       summary: Récupérer les détails d'un utilisateur par ID
 *       tags: [Users]
 *       security:
 *         - BearerAuth: []
 *       parameters:
 *         - in: path
 *           name: user_id
 *           required: true
 *           schema:
 *             type: string
 *           description: ID de l'utilisateur à récupérer
 *       responses:
 *         '200':
 *           description: Détails de l'utilisateur récupérés avec succès
```

```

* '401':
*   description: Non autorisé (JWT invalide ou manquant)
* '404':
*   description: Utilisateur non trouvé
*/
app.route("/users/:user_id")
  .get(jwtMiddleware.verifyToken, userController.listenSingleUser);

```

```

/**
 * @swagger
 * path:
 *   /users/{user_id}:
 *     put:
 *       summary: Mettre à jour les détails d'un utilisateur par ID
 *       tags: [Users]
 *       security:
 *         - BearerAuth: []
 *       parameters:
 *         - in: path
 *           name: user_id
 *           required: true
 *           schema:
 *             type: string
 *       description: ID de l'utilisateur à mettre à jour
 *       requestBody:
 *         required: true
 *         content:
 *           application/json:
 *             schema:
 *               $ref: '#/components/schemas/User'
 *       responses:
 *         '200':
 *           description: Utilisateur mis à jour avec succès
 *         '401':
 *           description: Non autorisé (JWT invalide ou manquant)
 *         '404':
 *           description: Utilisateur non trouvé
 */
app.route("/users/:user_id")
  .put(jwtMiddleware.verifyToken, userController.updateUser);

```

```

/**

```

```

* @swagger
* path:
* /users/{user_id}:
* delete:
*   summary: Supprimer un utilisateur par ID
*   tags: [Users]
*   security:
*     - BearerAuth: []
*   parameters:
*     - in: path
*       name: user_id
*       required: true
*       schema:
*         type: string
*       description: ID de l'utilisateur à supprimer
*   responses:
*     '200':
*       description: Utilisateur supprimé avec succès
*     '401':
*       description: Non autorisé (JWT invalide ou manquant)
*     '404':
*       description: Utilisateur non trouvé
* /
app.route("/users/:user_id")
  .delete(jwtMiddleware.verifyToken, userController.deleteUser);

}

```

les routes ne s'affichent pas dans swagger ui

mon model s'affiche mais pas mes routes

code status pour un update stp

code status unauthorized

code status si l'user n'est pas un admin et qu'on veut un admin ?

je dois faire une liste des users, je veux que cela ça soit possible si j'en ai au moins 3
quel est le statut si j'en ai moins que 3 ?

je fais des invitations et je veux que si une personne a déjà répondu à l'invitation
cela me renvoie une error comme quooi elle a déjà répondu quel serait le code statu
pour cela ?

j'ai une propriété booléen qui peut être null je veux faire un if pour savoir si elle est
null comment faire ?

j'ai cette erreur : TypeError: Cannot read properties of null (reading 'accept')

je vais te donner mes fichiers models, controllers et route pour que tu m'écrive mes
routes

(envoi dans différents prompts l'ensemble de mes fichiers)

Resolver error at
paths./users/register.post.responses.200.content.application/json.schema.items.\$r
ef

Could not resolve reference: Could not resolve pointer: components/schema/User does not exist in document

voici mon archi:

```
app.js      models      package.json
controllers node_modules routes
middlewares package-lock.json swagger.json
```

j'ai certaine route qui sont protégé comment montré cela sur swagger

comment faire si mon request body n'est pas idendique à mon schema de mon components

comment faire si lors d'une erreur (400) par exemple je veux renvoyer un object json avec comme clé message: Email or password incorrect ? avec swagger

si jamais je peux avoir deux réponses différentes mais avec le même statut comment faire ?

dans mon swagger j'ai un format: email quels sont les différents type de format

si mon erreur est que mon token authentication n'est pas bon quel sera mon status (401 ou 403)?

piur le oneOf si je veux mélanger une ref et un exemple

je veux que ça renvoie un objet json avec un shemas de Membership et un string avec une clé token

ma réponse n'est pas soit quelque chose soit quelque chose d'autre mais un objet comprenant mes deux elements

si je veux que ça renvoie un tableau d'object Group

pour mettre la possibilité de undefined un champ on fait comment ?

avec swagger comment fair epour que cela me renvoi un objet avec en clé token et en retour une string