self-service-machine-api - STEP 14

Documentation Swagger

Il est important de documenter notre API REST afin d'aider les consommateurs de notre API REST.

Installation des dépendances

Nous allons commencer par installer 2 dépendances :

```
npm install swagger-jsdoc swagger-ui-express --save
```

Configuration

Ensuite voila le code pour la configuration :

```
// src/swagger.mjs

import swaggerJSDoc from "swagger-jsdoc";

const options = {
  definition: {
    openapi: "3.0.0",
    info: {
      title: "API self-service-machine",
      version: "1.0.0",
      description:
        "API REST permettant de gérer l'application self-service-machine",
    },
    servers: [
      {
        url: "http://localhost:3000",
      },
    ],
    components: {
      securitySchemes: {
        bearerAuth: {
          type: "http",
          scheme: "bearer",
          bearerFormat: "JWT",
        },
      },
      schemas: {
        Teacher: {
          type: "object",
          required: ["name", "price", "created"],
          properties: {
            id: {
              type: "integer",
              description: "L'identifiant unique du produit.",
            },
            name: {
              type: "string",
              description: "Le nom du produit.",
            },
            price: {
              type: "float",
              description: "Le prix du produit.",
            },
```

```
            created: {
              type: "string",
              format: "date-time",
              description: "La date et l'heure de l'ajout d'un produit.",
            },
          },
        },
        // Ajoutez d'autres schémas ici si nécessaire
      },
    },
    security: [
      {
        bearerAuth: [],
      },
    ],
  },
  apis: ["./src/routes/*.mjs"], // Chemins vers vos fichiers de route
};

const swaggerSpec = swaggerJSDoc(options);

export { swaggerSpec };
```

Utilisation de la configuration

Maintenant, il faut utiliser cette configuration dans `src/app.mjs`

```
...
import swaggerUi from "swagger-ui-express";
...

const port = 3000;

import { swaggerSpec } from "./swagger.mjs";

// Route pour accéder à la documentation Swagger
//const specs = swaggerJsdoc(options);
app.use(
  "/api-docs",
  swaggerUi.serve,
  swaggerUi.setup(swaggerSpec, { explorer: true })
);
```

Documentation de notre première route

Ensuite, nous pouvons commencer à définir la documentation d'une route, par exemple, la route GET /api/products/

```
...

/**
 * @swagger
 * /api/products/:
 *   get:
 *     tags: [Products]
 *     security:
 *       - bearerAuth: []
 *     summary: Retrieve all products.
 *     description: Retrieve all products. Can be used to populate a select HTML tag.
 *     responses:
 *       200:
 *         description: All products.
 *         content:
```

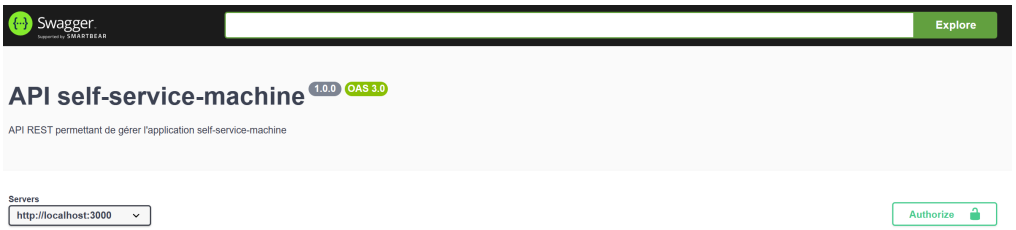
```
 *           application/json:
 *             schema:
 *               type: object
 *               properties:
 *                 data:
 *                   type: object
 *                   properties:
 *                     id:
 *                       type: integer
 *                       description: The product ID.
 *                       example: 1
 *                     name:
 *                       type: string
 *                       description: The product's name.
 *                       example: Big Mac
 *                     price:
 *                       type: number
 *                       description: The product's price.
 *                       example: 5.99
 *
 */
productsRouter.get("/", auth, (req, res) => {
...
```

Tester l'api en fournissant un token JWT

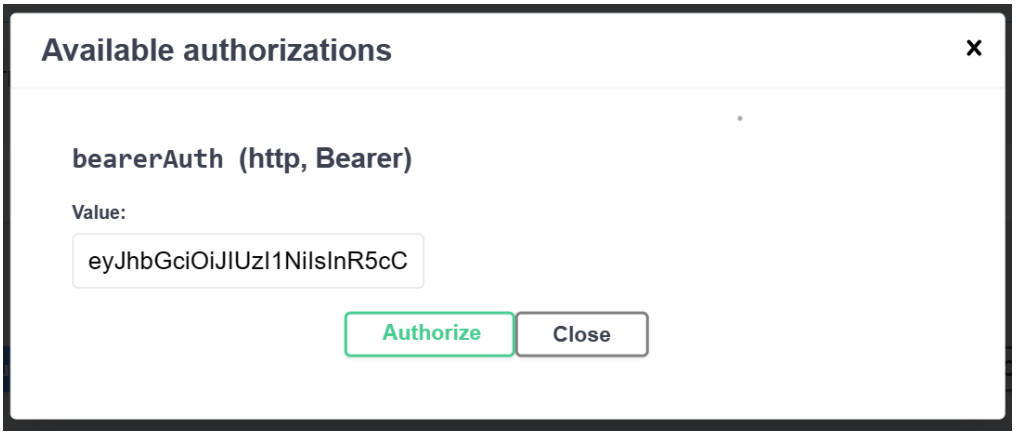
Notre API REST est protégé.

Donc seuls les consommateurs possédant un jeton JWT peuvent effectuer un appel HTTP via la documentation.

Pour cela, un consommateur doit commencer par cliquer sur le bouton `Authorize` (voir ci-dessous) :



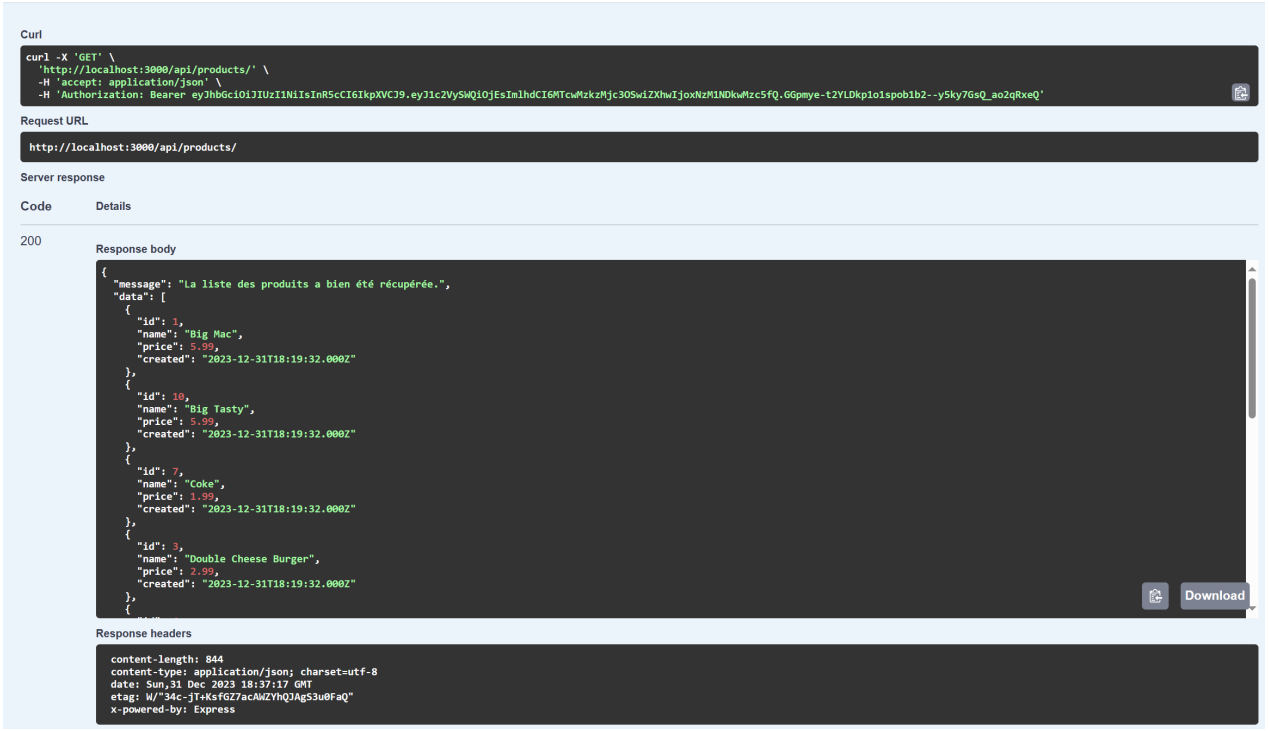
Puis copier / coller le jeton JWT :



Puis cliquer sur le bouton `Authorize`.

Et enfin le bouton `Close` :

Et maintenant l'appel HTTP à notre API REST depuis la documentation fonctionne



Voila ! Il ne reste plus qu'à documenter les autres routes !

Je vous laisse cela comme exercice :-)