




PIIQUANTE

PRESENTATION


PROJET 6 PIIQUANTE

OPENCLASSROOMS

Composition du site Piquante


**HOT TAKES**
THE WEB'S BEST HOT SAUCE REVIEWS

[SIGN UP](#) [LOGIN](#)

**HOT TAKES**
THE WEB'S BEST HOT SAUCE REVIEWS

[SIGN UP](#) [LOGIN](#)

[ALL SAUCES](#) [ADD SAUCE](#)


**HOT TAKES**
THE WEB'S BEST HOT SAUCE REVIEWS

[LOGOUT](#)


THE SAUCES



[ALL SAUCES](#) [ADD SAUCE](#)

**HOT TAKES**
THE WEB'S BEST HOT SAUCE REVIEWS

[LOGOUT](#)



Tabasco - McIlhenny


by Tabasco


Description

La pépite de Tabasco, une sauce unique et de qualité supérieure qui à l'origine était réservée à la famille McILHENNY

Ingrédients principaux

Vinaigre de vin blanc, piment, sel.

 0

 0

Heat

1

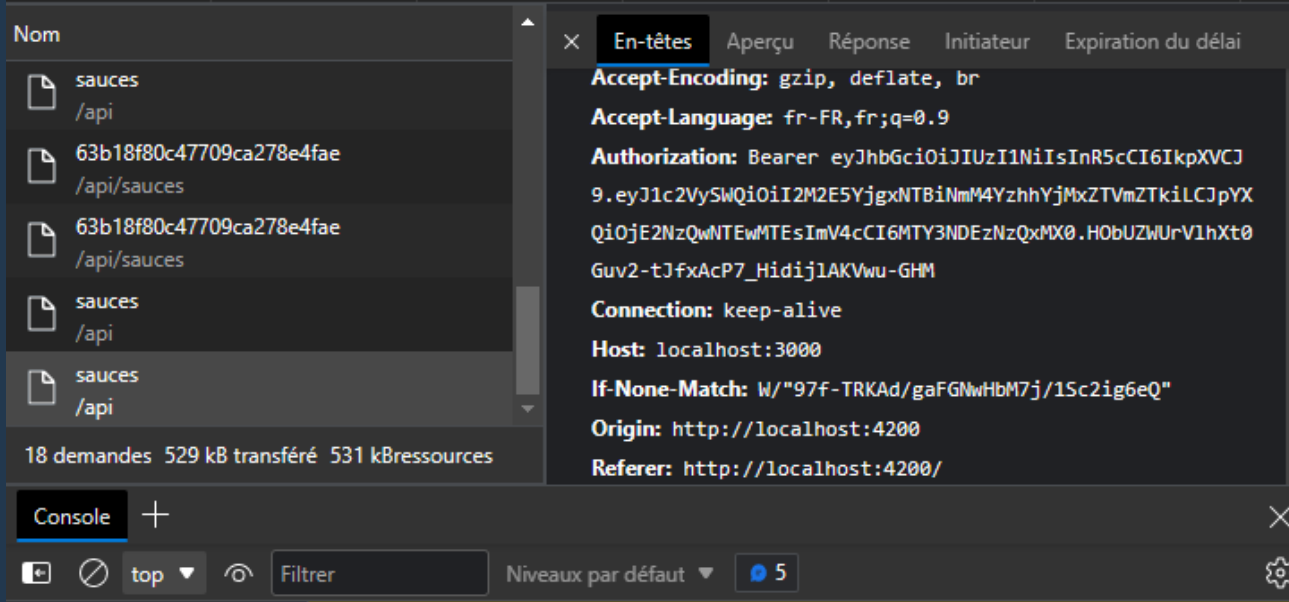
Technologies et outils utilisées

- Visual Studio Code
 - MongoDB Atlas
 - Postman
-
- Node (environnement d'exécution JavaScript open-source)
 - Bcrypt (algorithme de chiffrement)
 - Dotenv (module de Node.js)
 - Express (framework JavaScript pour Node.js)
 - Helmet (middleware pour Node.js)
 - jsonwebtoken (standard pour l'échange sécurisé de données)
 - Mongoose (module Node.js)
 - Mongoose-unique-validateur (plugin pour Mongoose)
 - multer (middleware pour Node.js)

Explication et fonctionnement

Le middleware d'authentification

```
16 | const jwt = require("jsonwebtoken");
17 | module.exports = (req, res, next) => {
18 |   try {
19 |     const token = req.headers.authorization.split(" ")[1];
20 |     const decodedToken = jwt.verify(token, "RANDOM_TOKEN_SECRET");
21 |     const userId = decodedToken.userId;
22 |     req.auth = { userId };
23 |     if (req.body.userId && req.body.userId !== userId) {
24 |       throw new Error("User ID non valable !");
25 |     } else {
26 |       console.log("Passage dans le middleware");
27 |       next();
28 |     }
29 |   } catch (error) {
```



Le middleware d'authentification va permettre de gérer et vérifier les informations d'identification d'un utilisateur.

Ils sera exécutés sur le serveur et inaccessibles pour l'utilisateur.

Jsonwebtoken est utilisé pour l'échange sécurisé de données en utilisant un jeton (Token) cryptographique.



HOT TAKES
THE WEB'S BEST HOT SAUCE REVIEWS

[SIGN UP](#) [LOGIN](#)

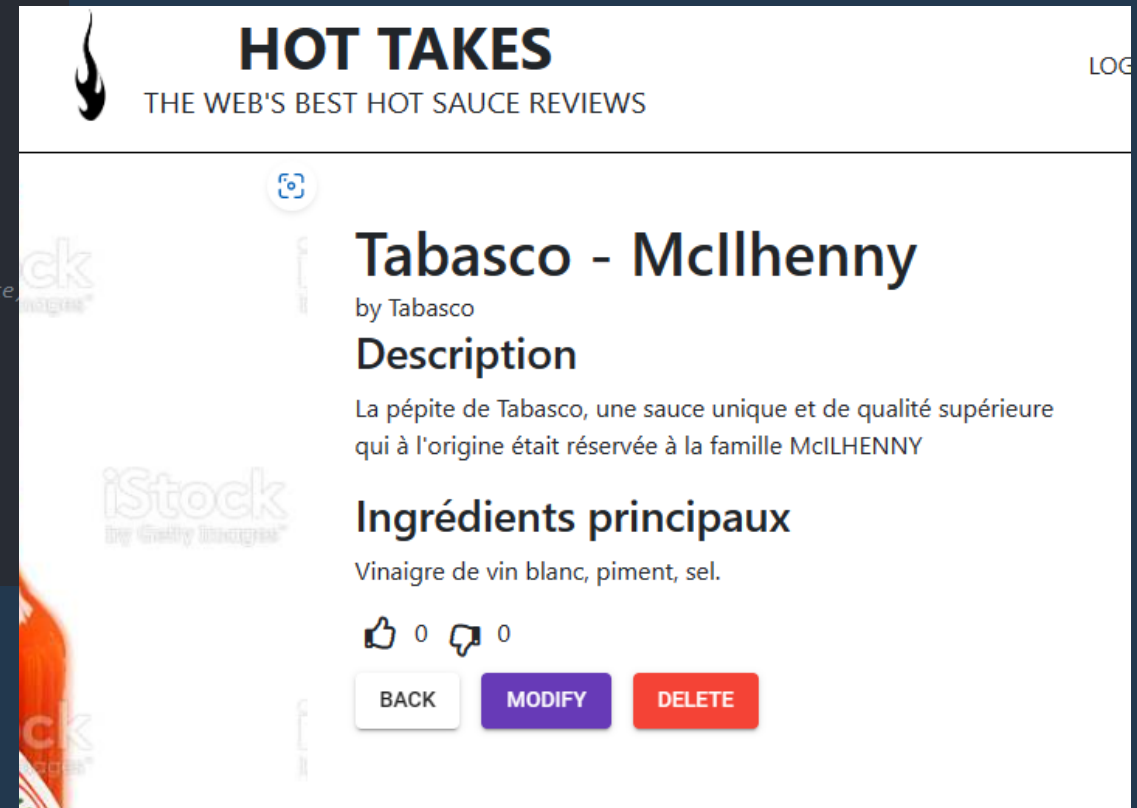
Email

Password

Les opérations CRUD

Les opérations CRUD (Create, Read, Update, Delete) sur les données stockées dans la base de données. Par exemple ici dans le dossier controllers nous avons une opérations CRUD, ce code va permettre de supprimer une sauce.

```
91 exports.deleteSauce = (req, res, next) => {
92   Sauce.findOne({ _id: req.params.id })
93     .then(sauce => {
94       if (sauce.userId !== req.auth.userId) {
95         res.status(401).json({ message: 'Not authorized' });
96       } else {
97         const filename = sauce.imageUrl.split('/images/')[1];
98         fs.unlink(`images/${filename}`, () => {
99           Sauce.deleteOne({ _id: req.params.id })
100             //Si suppression réussie, envoi d'un message (statut 200, indique la réussite d'une requête
101             .then(() => res.status(200).json({ message: 'Objet supprimé !' })))
102             //Sinon renvoi d'un message d'erreur 400 (Bad Request)
103             .catch(error => res.status(400).json({ error }));
104         });
105       }
106     })
107     .catch(error => res.status(500).json({ error }));
108 };
109
```



La gestion des sessions

Les informations de session et les jetons d'authentification sont généralement stockés sur le serveur.

```
15  const jwt = require('jsonwebtoken');
16
17  //Rappel de la fonction de hachage de bcrypt dans le mot de passe et "salage" du mot de passe
18  //10 fois (plus la valeur est élevée, plus l'exécution de la fonction sera longue,
19  //et plus le hachage sera sécurisé.
20  //C'est une fonction asynchrone qui renvoie une Promise dans laquelle nous recevons le hash généré dans
21  //le bloc then, Création d'un utilisateur et enregistrement dans la base de données,
22  //en renvoyant une réponse de réussite en cas de succès,
23  //et des erreurs avec le code d'erreur en cas d'échec.
24  exports.signup = (req, res, next) => {
25    bcrypt.hash(req.body.password, 10)
26      .then(hash => {
27        const user = new User({
28          email: req.body.email,
29          password: hash
30        });
31        user.save()
32          .then(() => res.status(201).json({ message: 'Utilisateur créé !'}))
33          .catch(error => res.status(400).json({ error }));
34      })
35      .catch(error => res.status(500).json({ error }));
36  };
```


La validation des données

La validation des données côté serveur est généralement exécutée sur le serveur avant de traiter les données.

```
2 //Importation de MONGOOSE
3 const mongoose = require('mongoose');
4
5 //Importation de mongoose-unique-validator
6 //plugin qui ajoute une validation de pré-enregistrement pour les champs uniques
7 //dans un schéma Mongoose, ce qui permet de ne valider q'un seul email par utilisateur
8 const uniqueValidator = require('mongoose-unique-validator');
```

The screenshot displays the MongoDB Atlas web interface. The top navigation bar includes the Atlas logo, a dropdown menu for 'Aykut's Org', and links for 'Gestionnaire d'accès' and 'Facturation'. The main navigation bar shows 'Services de données' as the active tab, with other options like 'Sélectionner un...', 'Services d'application', and 'Graphiques'. The left sidebar contains sections for 'DÉPLOIEMENT', 'SERVICES', and 'SÉCURITÉ'. The main content area is titled 'ClusterP6' and shows the following details:

- VERSION:** 5.0.14
- RÉGION:** AWS Paris (eu-west-3)
- NIVEAU DU CLUSTER:** M0 Sandbox (Général)

The 'Aperçu' tab is selected, showing a 'BAC À SABLE' (Sandbox) environment. The 'RÉGION' is 'Paris (eu-west-3)'. Below this, there are three nodes listed:

- AC... partition_00-00.f8... (SECONDAIRE)
- AC... partition_00-01.f8... (SECONDAIRE)
- AC... partition_00-02.f8... (PRIMAIRE)

A central message states: 'Il s'agit d'un cluster de niveaux partagés. Si vous avez besoin d'une base de données adaptée aux applications de production hautes performances, effectuez une mise à niveau vers un cluster dédié.' A 'Mise à niveau' (Upgrade) button is visible.

Performance metrics are shown in three panels:

- Opérations:** R:0 W:0 0,07/s. Graph showing operations over time (18/01/23 - 14:56).
- Taille logique:** 48.1 Ko. Graph showing logical size over time (31/12/22 - 01:21).
- Connexions:** 4. Graph showing connections over time (18/01/23 - 14:51).

Buttons for 'RELIER', 'CONFIGURATION', and a menu icon are located at the bottom right of the main content area.

Les traitements de fichiers

La gestion des téléchargements de fichiers, le stockage des fichiers sur le serveur et les traitements de fichiers.

```
1 // Importation de MULTER qui est un package de gestion de fichiers.
2 const multer = require('multer');
3
4 // Dictionnaires des types MIME des extension des fichiers
5 //qu'on peut trouver dans le frontend
6 const MIME_TYPES = {
7   'image/jpg': 'jpg',
8   'image/jpeg': 'jpg',
9   'image/png': 'png'
10 };
11
12
13 //Création d'une constante storage , à passer à multer comme configuration,
14 //qui contient la logique nécessaire pour indiquer à multer où enregistrer
15 //les fichiers entrants
16 //La méthode diskStorage() configure le chemin et le nom de fichier pour les fichiers entrants
17 const storage = multer.diskStorage({
18   // La fonction destination indique à multer d'enregistrer
19   //les fichiers dans le dossier images
20   destination: (req, file, callback) => {
21     callback(null, 'images')
22   },
23 });
```

The screenshot shows the 'HOT TAKES' web application interface. At the top, there's a navigation bar with 'ALL SAUCES' and 'ADD SAUCE' links, a logo with a flame, the title 'HOT TAKES', the tagline 'THE WEB'S BEST HOT SAUCE REVIEWS', and a 'LOGOUT' link. The main content area displays the 'ADD SAUCE' form. The form includes input fields for 'Name', 'Manufacturer', and 'Description'. Below the 'Description' field is a purple 'ADD IMAGE' button. There's also a 'Main Pepper Ingredient' input field. At the bottom of the form is a 'Heat' slider with a range from 0 to 10, currently set at 1, and a 'SUBMIT' button.

STRUCTURES

Les contrôleurs

Responsables de la gestion des interactions entre l'utilisateur et l'application. Ils reçoivent les requêtes de l'utilisateur, les traitent et renvoient les réponses appropriées.

Les routeurs

Responsables de l'acheminement des requêtes vers les contrôleurs appropriés. Ils examinent les informations de la requête, comme l'URL et les paramètres, pour déterminer quel contrôleur doit gérer la requête.

Les modèles

Responsables de la gestion des données de l'application. Ils effectuent des opérations CRUD (Create, Read, Update, Delete) sur les données stockées dans la base de données.

MÉTHODES POUR SÉCURISER LA BASE DE DONNÉES

Méthodes pour sécuriser la base de données selon le RGPD et l'OWASP.

Il existe plusieurs méthodes pour sécuriser une base de données selon les exigences du RGPD et de l'OWASP. Voici quelques exemples :

- Chiffrement des données sensibles : cela permet de protéger les données contre les accès non autorisés et les fuites de données.
- Authentification forte : Il est important d'utiliser des méthodes d'authentification robustes pour s'assurer que seuls les utilisateurs autorisés ont accès à la base de données.
- Mise à jour et maintenance régulières : Les mises à jour de sécurité et les correctifs doivent être installés régulièrement pour protéger la base de données contre les vulnérabilités connues.
- Contrôle d'accès : Il est important de configurer les autorisations d'accès de manière à ce que seuls les utilisateurs ayant besoin d'accéder à certaines données puissent le faire.
- Sauvegarde régulière : Il est important de sauvegarder régulièrement les données pour pouvoir les restaurer en cas de perte ou de corruption.
- Surveillance et détection des intrusions : Il est important de surveiller les accès à la base de données et d'être en mesure de détecter les tentatives d'intrusion pour pouvoir y réagir rapidement.

FIN DE LA PRESENTATION