

Soutenance

Projet 6

OpenClassrooms



Construisez une API sécurisée pour une application d'avis gastronomiques

Projet 6 - Piquante



01

Utilisation d'une API REST

02

Utilisation du CRUD

03

Modèle de données

04

Page Connexion / Inscription

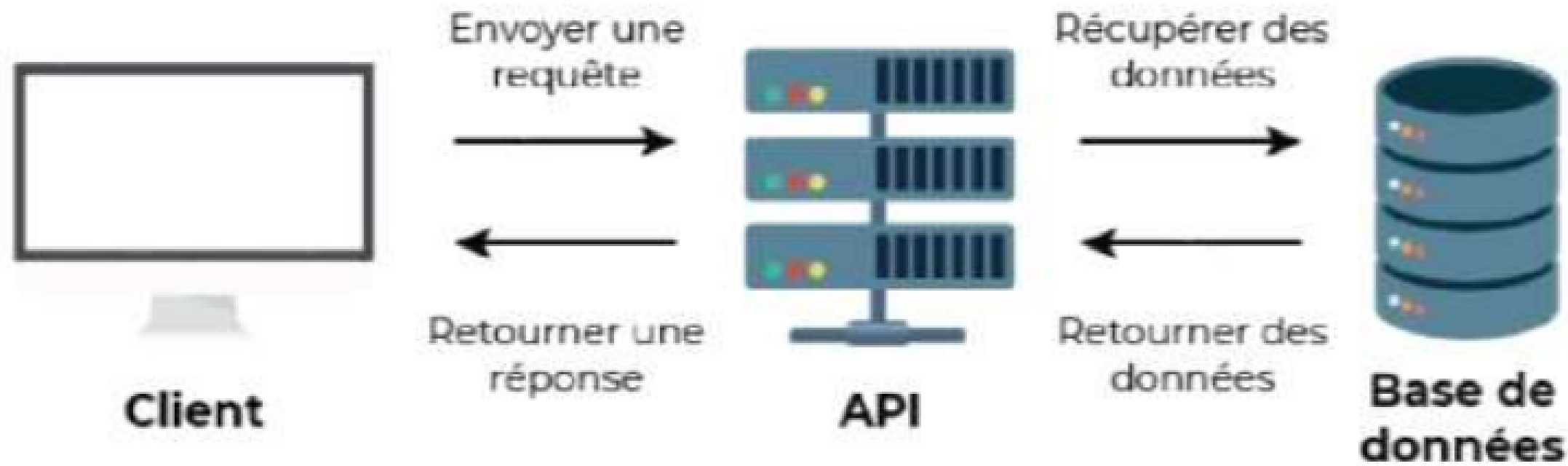
05

Page d'Accueil

Afin de valider le 6ème projet de ma formation chez Openclassrooms, j'ai réalisé le backend d'une application d'évaluation de sauces piquantes. Cette partie est « cachée », l'utilisateur ne la voit jamais. Elle permet pourtant la bonne fonctionnalité de l'application. L'application s'appelle "Piquante" et permet à ses utilisateurs d'ajouter leurs sauces gastronomiques. Ils peuvent 'liker' ou 'disliker' les différentes sauces. La partie backend de ce projet est développée en Node.js et express. Pour la base de données j'utilise MongoDB. Les enjeux du projet sont les suivants : •Ajuster le développement du backend pour qu'il fonctionne parfaitement avec le frontend, •Assurer la sécurité des données utilisateurs et de l'application (RGPD & OWAPS).



Utilisation d'une API REST



En créant la partie backend de ce projet on crée finalement notre propre API. Il s'agit d'une abréviation qui signifie Application Programming Interface (ou interface de programmation d'application, en français). En résumé, il s'agit d'un moyen de communication entre deux logiciels (soit différents composants d'une application, soit deux applications différentes). Voici les caractéristiques principales d'une API. Elle permet de :

- Communiquer des données.
- Communiquer entre les différents composants de l'application et entre l'application et d'autres développeurs. Ceci se fait grâce à l'utilisation de requêtes et de réponses.
- Donner un moyen d'accès aux données de façon réutilisable et standardisée. Voici leurs avantages principaux :
- Les développeurs peuvent utiliser des API publiques et privées.
- Les API publiques sont utilisables par tous sans restriction.
- Les API privées sont utilisables seulement par ceux qui ont un accès et y sont autorisés.

Utilisation du CRUD

Le CRUD est la liste des actions de base que vous pouvez effectuer sur une ressource. C'est un acronyme qui signifie Create (créer), Read (lire), Update (mettre à jour), et Delete (supprimer). Bien que le CRUD ne constitue pas vraiment un mécanisme technique en soi, chaque action CRUD est associée à un verbe

Action CRUD	Verbe HTTP associé
Create (Créer)	POST (Publier)
Read (Lire)	GET (Obtenir)
Update (Mettre à jour)	PUT (Mettre)
Delete (Supprimer)	DELETE (Supprimer)

Pour ce projet, je dois utiliser tous ces types de requêtes pour les sauces. Ainsi l'utilisateur sera en mesure d'ajouter/ supprimer / modifier et consulter sa sauce.

```
router.post('/', auth, multer, saucesCtrl.createSauce);  
router.put('/:id', auth, multer, saucesCtrl.modifySauce);  
router.delete('/:id', auth, saucesCtrl.deleteSauce);  
router.get('/:id', auth, saucesCtrl.getOneSauce);  
router.get('/', auth, saucesCtrl.sauces);  
router.post('/:id/like', auth, saucesCtrl.setStatut);
```

CRUD - routes pour les sauces

```
router.post('/signup', userCtrl.signup);  
router.post('/login', userCtrl.login);
```

Voici les routes nécessaires pour
l'inscription et la connexion à
l'application pour chaque utilisateur

Modèle de données

Voici les 2 modèles de données Mongoose à créer côté back

```
const mongoose = require( mongoose );

const sauceSchema = mongoose.Schema({
  //_id: { type: String, required: true }, // pas nécessaire puisque aut
  userId: { type: String, required: true },
  manufacturer: { type: String, required: true },
  name: { type: String, required: true },
  description: { type: String, required: true },
  mainPepper: { type: String, required: true },
  imageUrl: { type: String, required: true },
  heat: { type: Number, required: true },
  likes: { type: Number },
  dislikes: { type: Number },
  usersLiked: { type: [String] },
  usersDisliked: { type: [String] },
});

module.exports = mongoose.model('Sauce', sauceSchema);
```

Modèle Mongoose des sauces

Modèle de données

```
const mongoose = require('mongoose');
const uniqueValidator = require('mongoose-unique-validator'); // pour éviter les

const userSchema = mongoose.Schema({
  email: { type: String, required: true, unique: true }, // on met juste un
  password: { type: String, required: true },
});

userSchema.plugin(uniqueValidator); // on applique le module (adresse email doit
module.exports = mongoose.model('User', userSchema);
```

Modèle Mongoose des utilisateurs

Page Connexion Inscription

Pour la page de connexion / inscription voici les enjeux en terme de fonctionnalités :

- L'utilisateur qui a déjà un compte peut se connecter à l'application pour consulter les sauces.
- En développement web, on crée un controller côté Back. Il va chercher l'email de l'utilisateur dans la base de données et vérifier que le mot de passe associé correspond.
 - Le nouvel utilisateur peut créer un compte avec son adresse mail et son mot de passe.
- En développement web, l'utilisateur va être ajouté dans la base de données avec un mot de passe masqué grâce au plugin jsonwebtoken. • L'adresse email sera UNIQUE pour chaque utilisateur, 2 personnes ne peuvent pas avoir la même adresse email. • En développement web, on va utiliser un plugin d'unicité : mongoose-unique-validator



Email

rhani.data@gmail.com

Password

....

LOGIN

Page de Connexion

Page d'Accueil

Pour la page d'accueil voici les enjeux en terme de fonctionnalités

- Afficher les différentes sauces mises en ligne par les utilisateurs en récupérant ces informations dans la base de données via l'API,
- En développement web, ceci revient à récupérer les informations avec une requête HTTP de type GET.



THE SAUCES



SAUCE PIQUANTE
Heat: 7/10



SAUCE SOJA
Heat: 1/10



**SAUCE PIMENTÉE
SIRACHA COQ**
Heat: 8/10

Page d'accueil Piquante- site de sauces gastronomiques



Name

Manufacturer

Description

ADD IMAGE

Main Pepper Ingredient

Heat

1

SUBMIT

Page d'ajout d'un produit

Ici, nous pouvons y ajouter un nom, un manufacturé, une description, une image, une liste d'ingrédient et le niveau de force du piment

Page produits

Pour la page d'accueil voici les enjeux en terme de fonctionnalités

- Affichage de l'ensemble des informations d'une seule sauce sur cette page quand on clic dessus depuis la page d'accueil.
- En développement web, ceci revient à utiliser le paramètre de l'URL (dans lequel est présent l'id de la sauce en question et donc son numéro de référence). Le Back doit prévoir la réception d'une requête de type GET/:_id (correspondant à une sauce précise).
- Possibilité de modifier ou supprimer la sauce lorsque l'on est son créateur.
- En langage web, on va mettre en place une route de type PUT:id et DELETE:id côté API.
- Possibilité de liker ou disliker la sauce.
- En langage web ceci revient à ajouter, dans le modèle Mongoose, +1 dans le nombre de like ou dislike ET ajouter l'id de l'auteur de l'action dans le tableau des 'like' ou 'dislike'



Sauce Pimentée Sriracha Coq

by local

Description

La sauce Sriracha a seulement fait son apparition dans les années 80 mais fait parti aujourd'hui des sauces piquantes les plus consommées au Monde !



[BACK](#)

[MODIFY](#)

[DELETE](#)

Page Produits

Fin de la Soutenance Projet 6 OpenClassrooms



Construisez une API sécurisée pour une application d'avis gastronomiques