

# Test Technique

## 1. Introduction

Ce projet Laravel a été réalisé dans le cadre d'un test technique pour notre entretien d'embauche pour le poste de Développeur Backend confirmé en CDI. L'objectif est de développer une API permettant de gérer des profils et des commentaires, avec une authentification sécurisée pour les administrateurs.

## 2. Installation

### Prérequis :

- PHP >= 8.0
- Composer
- MySQL
- Node.js & npm (optionnel)

### Étapes d'installation :

1. Cloner le dépôt : `git clone https://github.com/TolgaGu/helloce`
2. Installer les dépendances : `composer install`
3. Configurer l'environnement : `cp .env.example .env` et modifier les paramètres.
4. Générer la clé d'application : `php artisan key:generate`
5. Migrer la base de données : `php artisan migrate`
6. Peupler la base de données : `php artisan db:seed`

## 3. Configuration

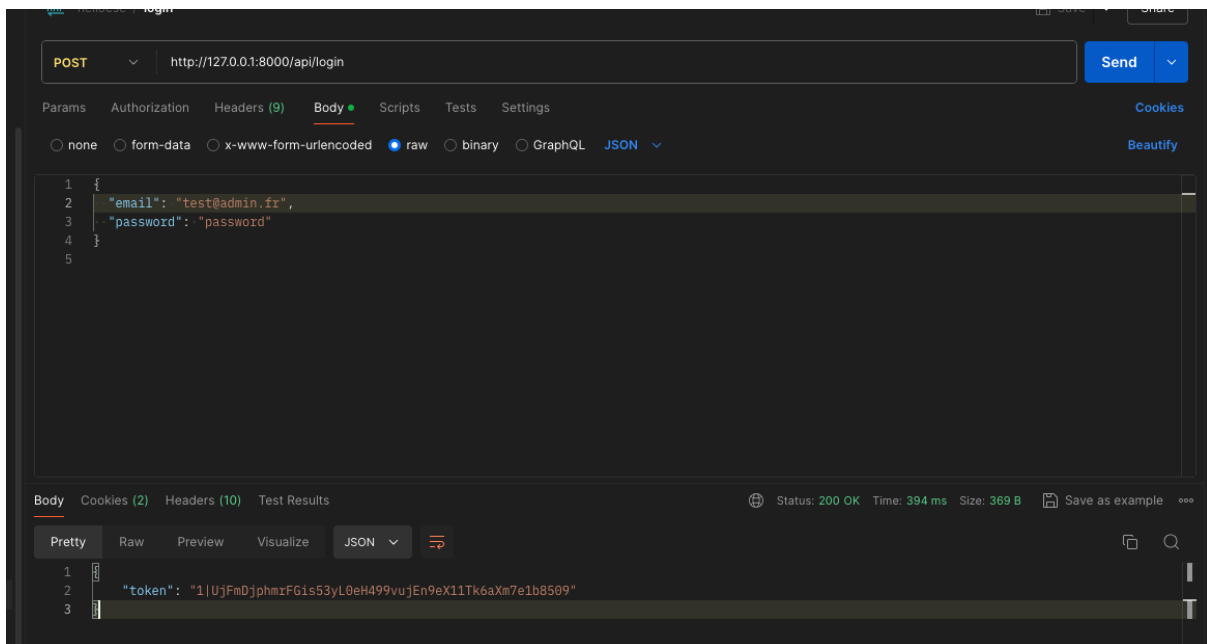
Le fichier `.env` doit être configuré pour inclure les paramètres de connexion à la base de données MySQL. Exécutez les migrations et les seeders pour préparer la base de données avec des données fictives.

## 4. Utilisation

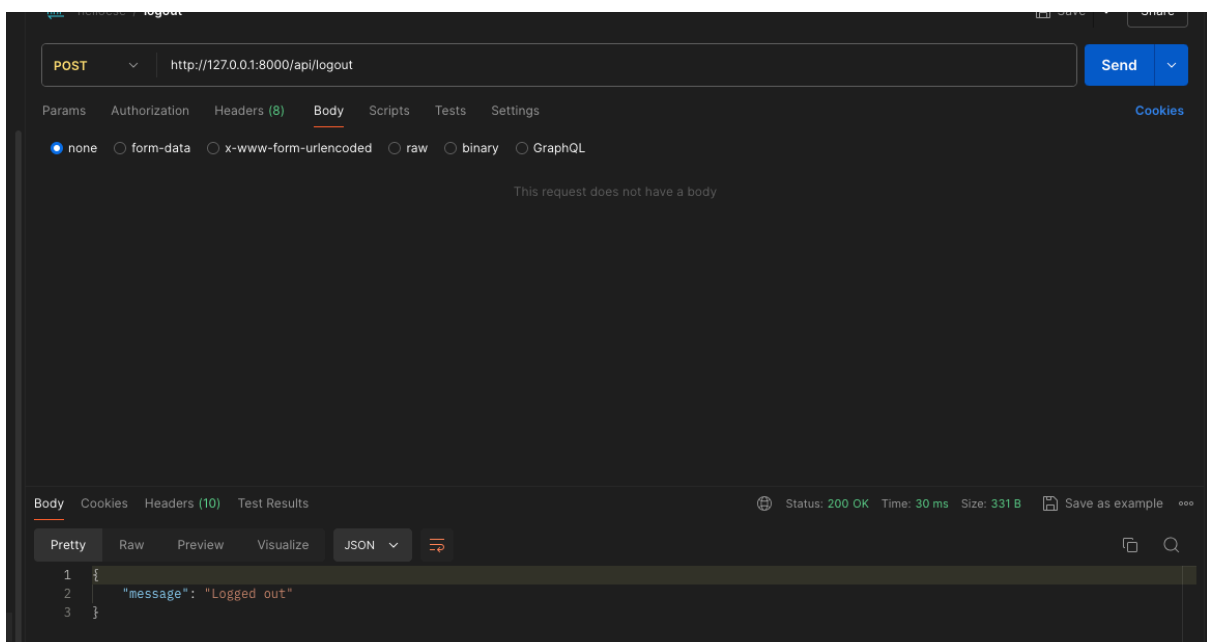
Démarrage du serveur : `php artisan serve`

Endpoints de l'API :

- **Authentification :**
  - `POST /api/login` : Permet aux administrateurs de se connecter.



- `POST /api/logout` : Permet aux administrateurs de se déconnecter.



- **Profils :**

- **GET /api/profils** : Récupère les profils actifs.

GET <http://127.0.0.1:8000/api/profils> Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies (2) Headers (10) Test Results Status: 200 OK Time: 22 ms Size: 3.13 KB Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   {
3     "id": 3,
4     "nom": "Hills",
5     "prenom": "Gladyce",
6     "image": "https://via.placeholder.com/640x480.png/084499?text=nemo",
7     "commentaires": []
8   },
9   {
10    "id": 5,
11    "nom": "Michel",
12    "prenom": "Michel",
13    "image": "https://fastly.picsum.photos/id/88/200/300.jpg?hmac=JmiMN7iyW4Saka8254HzDvb0jMSB2k9NwTN29MHwqa4",
14    "commentaires": [
15      {
16        "id": 6,
17        "administrateur_id": 1,
18        "profil_id": 5,
19        "contenu": "Ceci est un commentaire de test.",
20        "created_at": "2024-08-06T15:29:29.000000Z",
21        "updated_at": "2024-08-06T15:29:29.000000Z"
22      }
23    ]
24  },
25  {
26    "id": 6,
27    "nom": "Considine",
28    "prenom": "Coralie",
29    "image": "https://via.placeholder.com/640x480.png/08dd22?text=aspernatur",
30    "commentaires": []
31  },

```

- **GET /api/profils** : Récupère les profils actifs en étant authentifié (on affiche les profils tout statut confondu).

GET <http://127.0.0.1:8000/api/profils> Send

Params Authorization Headers (8) Body Scripts Tests Settings Cookies

Host ☒ User-Agent ☒ Accept ☒ Accept-Encoding ☒ Connection ☒ Authorization ☒

Key	Value	Description
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.41.0	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer 1 UjFmDjphmrFGis53yL0eH499vujEn9eX11Tk6aX...	

Body Cookies (2) Headers (10) Test Results Status: 200 OK Time: 27 ms Size: 8.26 KB Save as example

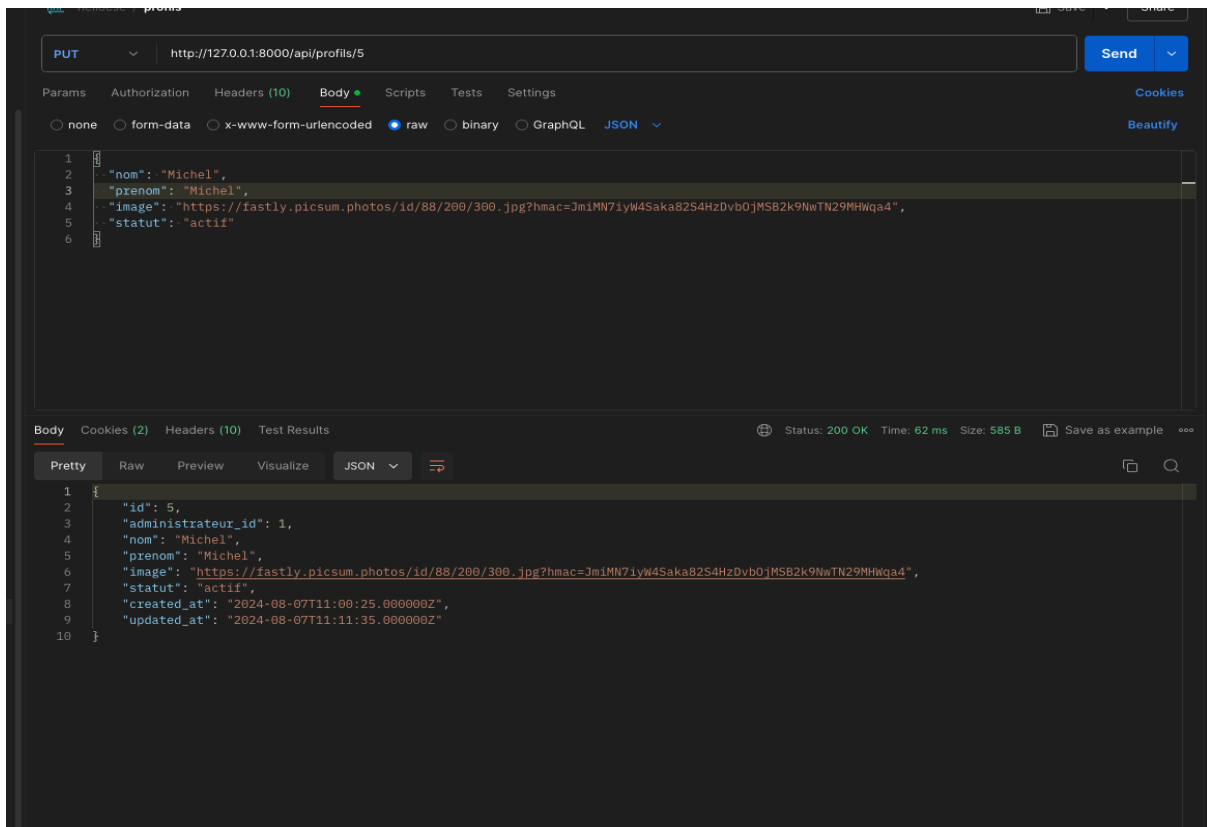
Pretty Raw Preview Visualize JSON

```

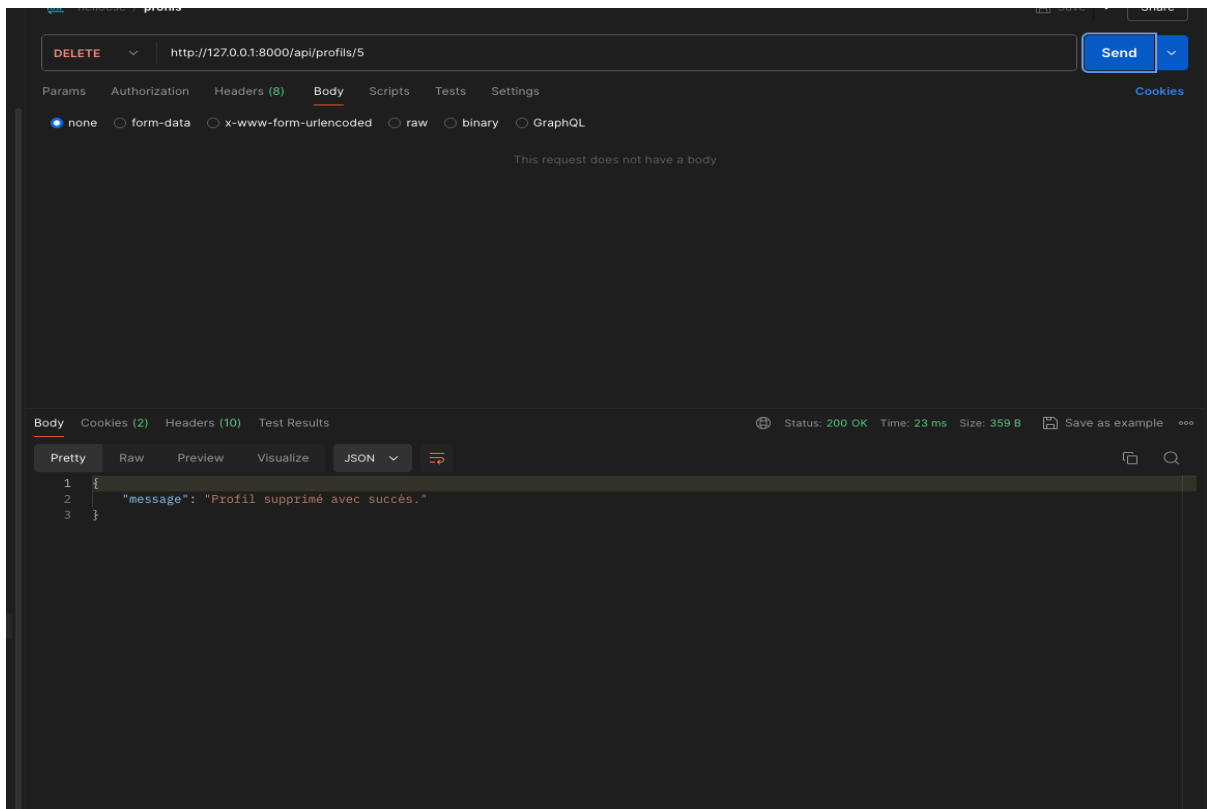
1 {
2   {
3     "id": 1,
4     "nom": "McDermott",
5     "prenom": "Florida",
6     "image": "https://via.placeholder.com/640x480.png/086608?text=veritatis",
7     "statut": "inactif",
8     "commentaires": []
9   },
10  {
11    "id": 2,
12    "nom": "Sanford",
13    "prenom": "Mara",
14    "image": "https://via.placeholder.com/640x480.png/0866ee?text=qui",
15    "statut": "actif",
16    "commentaires": []
17  },
18  {
19    "id": 3,
20    "nom": "Kuvalis",
21    "prenom": "Trent",
22    "image": "https://via.placeholder.com/640x480.png/08cc11?text=vero",
23    "statut": "actif",
24    "commentaires": []
25  },
26  {
27    "id": 4,
28    "nom": "Lowe",

```

- PUT /api/profils/{id} : Modifie un profil.

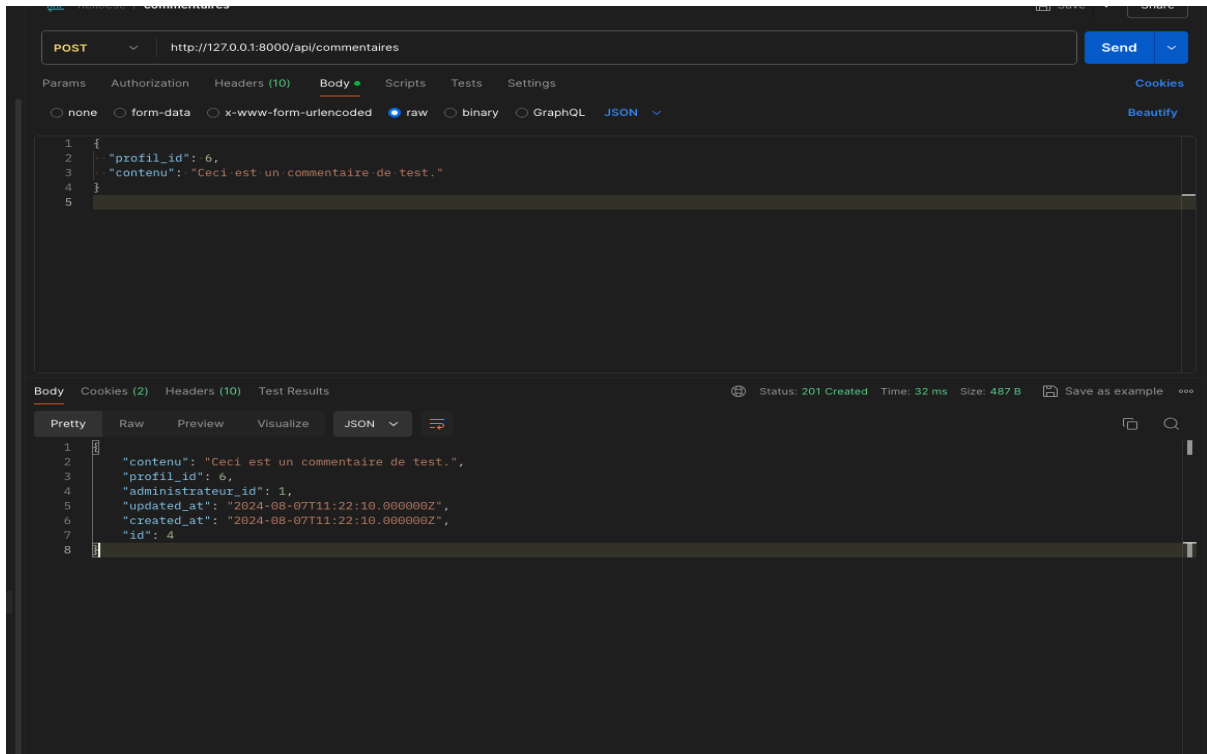


- DELETE /api/profils/{id} : Supprime un profil.

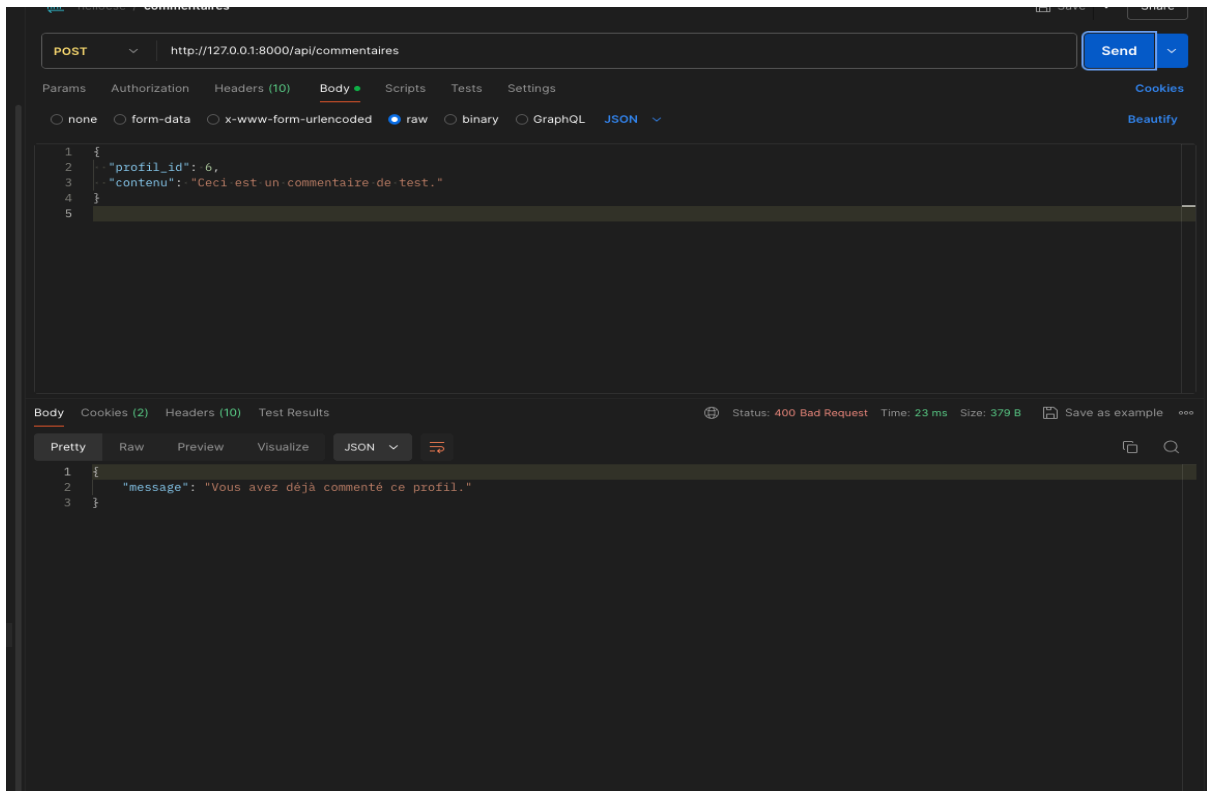


- **Commentaires :**

- **POST /api/commentaires** : Ajoute un commentaire à un profil (un administrateur ne peut commenter un profil qu'une seule fois).



- **DELETE /api/commentaires/{id}** : Supprime un commentaire.



## 5. Sécurité

L'authentification utilise Laravel Sanctum pour sécuriser les endpoints de l'API. Il est possible d'ajouter des mesures de sécurité supplémentaires incluant la protection CSRF et la gestion sécurisée des tokens (scopes, expirations, httponly, secure...).

## 6. Tests

Des tests unitaires ont été écrits pour valider les fonctionnalités principales du projet. Pour exécuter les tests, utilisez la commande : `php artisan test`.

## 7. Analyse du code

**PHPStan** : Utilisé pour l'analyse statique du code. Pour exécuter l'analyse :  
`vendor/bin/phpstan analyse`

**Laravel Pint** : Utilisé pour le formatage du code. Pour formater le code :  
`vendor/bin/pint`

## 8. Conclusion

Ce projet a permis de développer une API sécurisée pour la gestion de profils et de commentaires. Les principales fonctionnalités ont été implémentées et testées.