

# Documentation de l'API REST pour la Gestion d'Articles et d'Utilisateurs

## Introduction

Bienvenue dans l'API REST de gestion d'articles et d'utilisateurs. Cette API permet de réaliser des opérations CRUD (Create, Read, Update, Delete) sur les articles et les utilisateurs.

## Accès à l'API

L'API est accessible à l'adresse suivante : `http://localhost:8080/`

## Opérations

### Opérations sur les Articles

#### Récupérer tous les articles

- **Endpoint :** `/articles`
- **Méthode :** GET
- **Description :** Récupère la liste complète de tous les articles.

#### Récupérer un article spécifique

- **Endpoint :** `/articles/{id}`
- **Méthode :** GET
- **Description :** Récupère les détails d'un article en spécifiant son ID dans l'URL.

#### Ajouter un nouvel article

- **Endpoint :** `/articles`
- **Méthode :** POST
- **Description :** Ajoute un nouvel article. Les détails de l'article doivent être inclus dans le corps de la requête au format JSON.

#### Mettre à jour un article existant

- **Endpoint :** `/articles/{id}`
- **Méthode :** PUT
- **Description :** Met à jour les détails d'un article existant en spécifiant son ID dans l'URL. Les nouvelles informations doivent être incluses dans le corps de la requête au format JSON.

#### Supprimer un article

- **Endpoint :** `/articles/{id}`
- **Méthode :** DELETE
- **Description :** Supprime un article en spécifiant son ID dans l'URL.

### Opérations sur les Utilisateurs

#### Récupérer tous les utilisateurs

- **Endpoint :** `/users`
- **Méthode :** GET
- **Description :** Récupère la liste complète de tous les utilisateurs.

#### Récupérer un utilisateur spécifique

- **Endpoint :** `/users/{id}`
- **Méthode :** GET
- **Description :** Récupère les détails d'un utilisateur en spécifiant son ID dans l'URL.

#### Ajouter un nouvel utilisateur

- **Endpoint :** `/users`
- **Méthode :** POST
- **Description :** Ajoute un nouvel utilisateur. Les détails de l'utilisateur doivent être inclus dans le corps de la requête au format JSON.

#### Mettre à jour un utilisateur existant

- **Endpoint** : `/users/{id}`
- **Méthode** : PUT
- **Description** : Met à jour les détails d'un utilisateur existant en spécifiant son ID dans l'URL. Les nouvelles informations doivent être incluses dans le corps de la requête au format JSON.

### Supprimer un utilisateur

- **Endpoint** : `/users/{id}`
- **Méthode** : DELETE
- **Description** : Supprime un utilisateur en spécifiant son ID dans l'URL.

## Exemples

---

### Exemple de création d'un article

POST <http://localhost:8080/articles> Content-Type: application/json

```
{ "Title": "Nouvel Article", "Content": "Contenu de l'article" }
```

bash

### Réponse

json

```
{ "Id": 1, "Title": "Nouvel Article", "Content": "Contenu de l'article" }
```

### Exemple de récupération d'un utilisateur

bash

GET <http://localhost:8080/users/1> Réponse json

```
{ "Id": 1, "Username": "Utilisateur1", "Email": "utilisateur1@email.com", "Password": "motdepasse" }
```