

API REST

R4.01 Architecture Logicielle



Découvrir le compte rendu



Groupe C

Paul **BRESSOLLES**
Théo **WAZYDRAG**

SOMMAIRE

Introduction **1**

- a. Fonction principales
- b. Lien GitHub
- c. Conception de l'api

Méthodes **3**

- a. Moderator
- b. Publisher
- c. Non authentifié

Use Case **6**

- a. Moderator
- b. Publisher
- c. Visiteur

Introduction

L'objectif général de ce projet est de proposer une solution pour la gestion d'articles de blogs. Il s'agit en particulier de concevoir le backend de la solution, c'est-à-dire la partie cachée aux utilisateurs de l'application. La solution devra s'appuyer sur une architecture orientée ressources, c'est-à-dire sur une ou plusieurs API REST.

a. Fonction principales

Moderator

- Consulter n'importe quel article. Il accède à l'ensemble des informations décrivant un article : auteur, date de publication, contenu, liste des utilisateurs ayant liké l'article, nombre total de like, liste des utilisateurs ayant disliké l'article, nombre total de dislike.
- Supprimer n'importe quel article.

Publisher

- Poster un nouvel article.
- Consulter ses propres articles.
- Consulter les articles publiés par les autres utilisateurs. Il accède aux informations suivantes relatives à un article : auteur, date de publication, contenu, nombre total de like, nombre total de dislike.
- Modifier les articles dont il est l'auteur.
- Supprimer les articles dont il est l'auteur.
- Liker/disliker les articles publiés par les autres utilisateurs.

Non authentifié

- Consulter les articles existants. Seules les informations suivantes sont disponibles : auteur, date de publication, contenu.

b. Lien GitHub

Vous trouverez la globalité du projet en suivant le lien ci-dessous. Sur ce git, vous pouvez consulter le read-me pour avoir des précisions quant à l'utilisation de l'api.

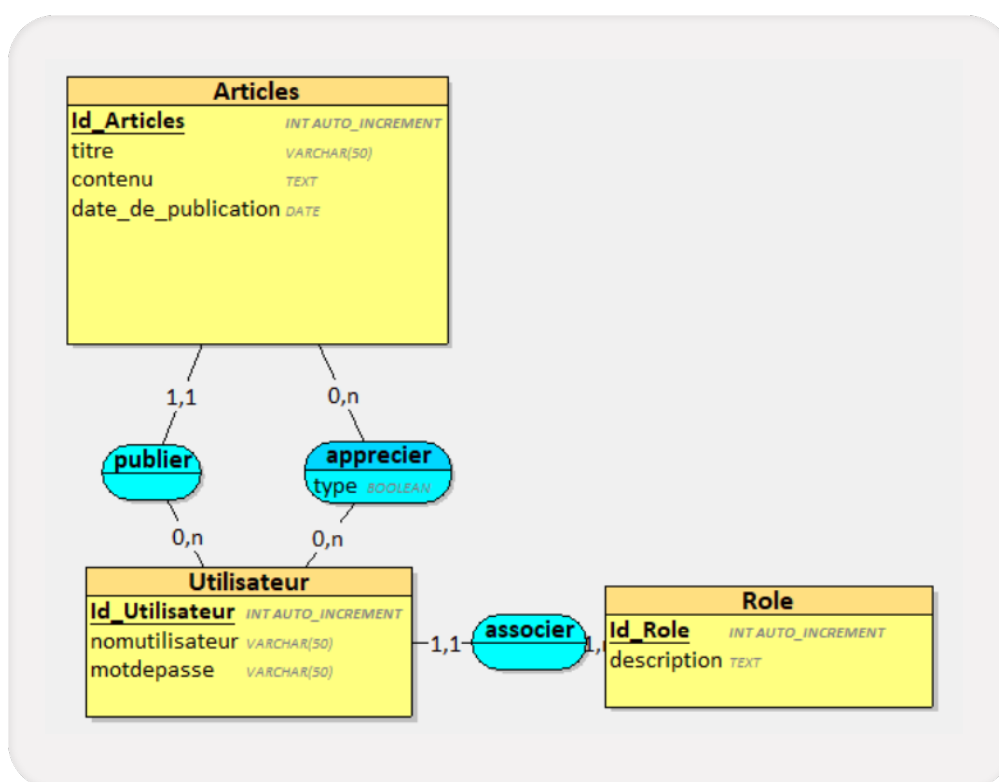
https://github.com/WTheo-dev/API_REST_GestionArticles

c. Conception de l'api

Les services de l'api sont disponible sur 3 urls.

- *localhost/ProjetR401/serveurToken*
- *localhost/ProjetR401/likeDislikeArticle*
- *localhost/ProjetR401/serveurApplication*

Pour identifier les différents rôles nous avons utilisé le système de JWTToken. Un utilisateur non authentifié génère lui aussi un token. Ce dernier sert à symboliser l'accès à l'api par un lien de la page connexion.



Méthodes

a. Moderator

GET

Affiches les informations sur tous les articles de la base de données.

Paramètre : **Null**

Body : **Null**

```
"status": 200,  
"status_message": "Succes ! Les donnees autorisees pour votre role sont accessibles",  
"data": [  
  {  
    "contenu": "PHP est un langage de programmation très populaire pour les sites web dynamiques. Voici ses avantages.",  
    "date": "2022-01-01",  
    "publisher": "Jacques",  
    "like": 2,  
    "listeLike": [  
      "Louis",  
      "Louise"  
    ],  
    "dislike": 1,  
    "listeDislike": [  
      "Elise"  
    ]  
  },  
]
```

RÉSULTAT

DELETE

Supprime n'importe quel article avec en entrée son id.

Paramètre : **id (article)**

Body : **Null**

ERREUR

```
$RETURN_CODE = 400;  
$STATUS_MESSAGE = "Erreur de syntaxe";
```

Si l'id ne correspond pas syntaxiquement à un valable, l'api retourne cette erreur. De même si l'id n'existe pas dans la bd

b. Publisher

GET

Affiches les informations sur tous les articles de la base de données.

Paramètre : **Null**

Body : **Null**

```

{
  "status": 200,
  "status_message": "Succes ! Les donnees autorisees pour votre role sont accessible",
  "data": [
    {
      "contenu": "PHP est un langage de programmation très populaire pour les sites web dynamiques. Voici ses avantages.",
      "date": "2022-01-01",
      "publisher": "Jacques",
      "like": 2,
      "dislike": 1
    }
  ]
}

```

RÉSULTAT**GET**

Affiches les informations sur tous les articles de la base de données rédigé par le publisher connecté.

Paramètre : **publisher=me**

Body : **Null**

```

{
  "status": 200,
  "status_message": "Succes ! Les donnees autorisees pour votre role sont accessible",
  "data": [
    {
      "contenu": "PHP est un langage de programmation très populaire pour les sites web dynamiques. Voici ses avantages.",
      "date": "2022-01-01",
      "publisher": "Jacques",
      "like": 2,
      "dislike": 1
    }
  ]
}

```

RÉSULTAT**POST**

Publie un article avec un titre et du contenu.

Paramètre : **Null**

Body : **titre : ' , contenu : ' '**

ERREUR

```

$RETURN_CODE = 400;
$STATUS_MESSAGE = "Erreur de syntaxe";

```

Si le titre ou le contenu n'est pas conforme, l'api retourne une erreur syntaxique.

DELETE

Supprime n'importe quel article avec en entrée son id si le publisher connecté est le propriétaire de l'article.

Paramètre : **id (article)**

Body : **Null**

ERREUR

```
$RETURN_CODE = 403;  
$STATUS_MESSAGE = "Cet article n'est pas le votre";
```

Si le publisher n'est pas le propriétaire il recevra une erreur.

PUT

Permet de modifier n'importe quel article avec son id si le publisher connecté est le propriétaire de l'article

Paramètre : **id (article)** Body : **titre : '', contenu : ''**

ERREUR

```
$RETURN_CODE = 403;  
$STATUS_MESSAGE = "Cet article n'est pas le votre";
```

Si le publisher n'est pas le propriétaire il recevra une erreur.

c. Non authentifié

GET

Affiches les informations sur tous les articles de la base de données.

Paramètre : **Null**

Body : **Null**

```
"status": 200,  
"status_message": "Succes ! Les donnees autorisees pour votre role sont accessible",  
"data": [  
  {  
    "contenu": "PHP est un langage de programmation très populaire pour les sites web dynamiques. Voici ses avantages.",  
    "date": "2022-01-01",  
    "publisher": "Jacques"  
  },  
  ]
```

RÉSULTAT

Use Case

a. Moderator

EN TANT QUE *moderator*

JE VEUX m'identifier

POUR récupérer le token me permettant de réaliser mes tâches sur l'api.

EN TANT QUE *moderator*

JE VEUX consulter toutes les informations disponibles sur l'ensemble des articles.

EN TANT QUE *moderator*

JE VEUX pouvoir supprimer n'importe quel article.

b. Publisher

EN TANT QUE *publisher*

JE VEUX m'identifier

POUR récupérer le token me permettant de réaliser mes tâches sur l'api.

EN TANT QUE *publisher*

JE VEUX consulter les informations disponibles sur l'ensemble des articles.

EN TANT QUE *publisher*

JE VEUX publier des articles avec un titre et un contenu

EN TANT QUE *publisher*

JE VEUX pouvoir modifier mes articles avec un titre et un contenu

EN TANT QUE *publisher*

JE VEUX pouvoir supprimer mes articles

EN TANT QUE *publisher*

JE VEUX réagir aux articles avec un like et un dislike

c. Visiteur

EN TANT QUE *visteur*

JE VEUX consulter les informations disponibles sur l'ensemble des articles.