

RAPPORT R4.01 API REST

JEHAN PHILIPON
BENOIT LECLERC

BUT informatique 2A
Groupe B

2

Sommaire

1. Lien GIT du projet	3
2. Modèle conceptuel de données.....	3
3. Serveur application	4
a. Méthode GET.....	4
b. Méthode POST	6
c. Méthode PUT.....	7
d. Méthode DELETE.....	8
1. Serveur d'authentification	9
a. Méthode POST	9
2. User stories	10

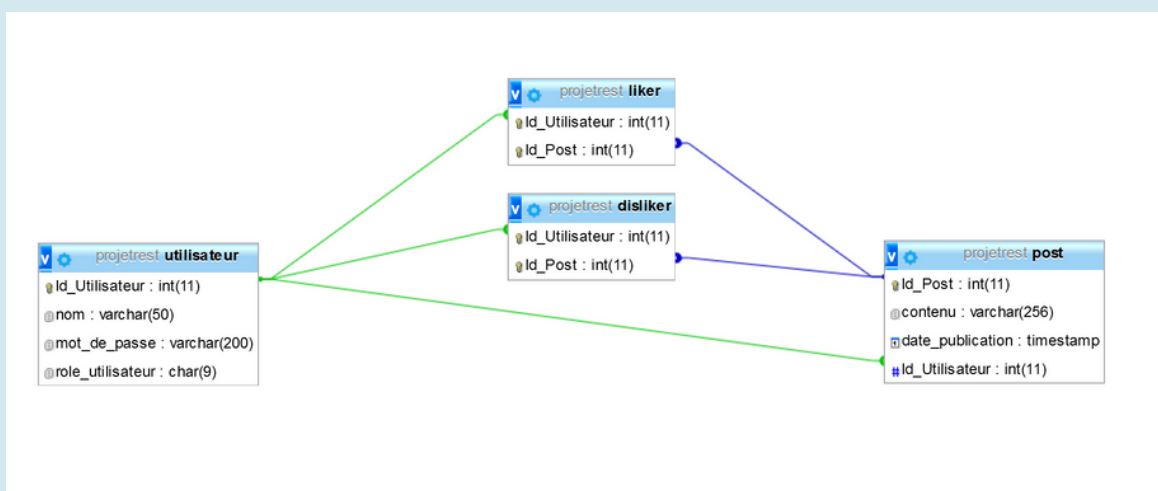
3



DEPOT GIT

<https://github.com/JehanPhlp/ProjetREST.git>

BASE DE DONNÉES : MCD



4

Serveur Application

Méthode GET :

Traitement :

- Si on met un "username" dans l'url (?username=xxx) le serveur renvoie les articles de cette utilisateur en vérifiant si l'utilisateur est moderator, publisher ou non connecté
- Si on met un "id" dans l'url (?id=xx) le serveur renvoie le poste avec cet id en vérifiant si l'utilisateur est moderator, publisher ou non connecté
- Si autre chose est mis, le serveur revoie une erreur "415 argument inattendu"
- S'il n'y a pas d'argument, le serveur renvoie tous les articles en vérifiant si l'utilisateur est moderator, publisher ou non connecté

Obtenir tous les articles :

GET <http://localhost/projetREST/serveurApplication.php>

Entrées

Nom	Type	Description
Rien	--	--

Sorties

Code	Message	Description
200	"affichage de posts"	

Retours

Nom	Type	Description
Id	Int	Id de l'article
Contenu	String	Contenu de l'article
date_publication	Date	Date de la publication
Id_utilisateur	Int	Id du publisher de l'article
Likes (publisher / moderator)	Int	Nombre de likes sur l'article
Dislikes (publisher / moderator)	Int	Nombre de dislikes sur l'article
Users_likes (moderator)	List<String>	Liste des utilisateurs qui ont liké l'article
Users_dislikes (moderator)	List<String>	Liste des utilisateurs qui ont disliked l'article

X le nombre d'articles

Obtenir les articles d'un publisher :

GET <http://localhost/projetREST/serveurApplication.php?username=xxx>

5

Entrées

Nom	Type	Description
Username	String	Username du publisher

Sorties

Code	Message	Description
200	"affichage de posts"	
415	"Argument inattendu: attendu id/username/rien"	Si l'argument est invalide

Retours

Nom	Type	Description
Id	Int	Id de l'article
Contenu	String	Contenu de l'article
date_publication	Date	Date de la publication
Id_utilisateur	Int	Id du publisher de l'article
Likes (publisher / moderator)	Int	Nombre de likes sur l'article
Dislikes (publisher / moderator)	Int	Nombre de dislikes sur l'article
Users_likes (moderator)	List<String>	Liste des utilisateurs qui ont liké l'article
Users_dislikes (moderator)	List<String>	Liste des utilisateurs qui ont disliké l'article

X le nombre d'articles

Obtenir un article à partir de son ID :

GET <http://localhost/projetREST/serveurApplication.php?id=12>

Entrées

Nom	Type	Description
Id	Int	Id de l'article

Sorties

Code	Message	Description
200	"affichage de posts"	
415	"Argument inattendu: attendu id/username/rien"	Si l'argument est invalide

Retours

Nom	Type	Description
Id	Int	Id de l'article
Contenu	String	Contenu de l'article
date_publication	Date	Date de la publication
Id_utilisateur	Int	Id du publisher de l'article
Likes (publisher / moderator)	Int	Nombre de likes sur l'article
Dislikes (publisher / moderator)	Int	Nombre de dislikes sur l'article
Users_likes (moderator)	List<String>	Liste des utilisateurs qui ont liké l'article
Users_dislikes (moderator)	List<String>	Liste des utilisateurs qui ont disliké l'article

6

Méthode POST:

Traitement :

- Le contenu doit être précisé dans le body en json (ex: {"contenu":"xxxx"})
- Le serveur vérifie que l'utilisateur a un token valide sinon envoie un 401, "Identification requise"
- Le serveur vérifie que l'utilisateur est un publisher sinon envoie 401, "Seuls les publishers peuvent publier"
- Le serveur vérifie que la requête contient un contenu sinon envoie 422, "missing parameter : contenu"

Créer un article :

POST http://localhost/projetREST/serveurApplication.php

Entrées

Nom	Type	Description	
Contenu	String	Contenu de l'article que vous voulez créer	Dans le body en json

Sorties

Code	Message	Description
200	"post crée"	
401	"Identification requise"	Si le token d'identification n'est pas valide
401	"Seul les publishers peuvent publier"	Si n'êtes pas connecté en tant que publisher

Retours

Nom	Type	Description
Rien	--	--

Méthode PUT:

Traitement :

- Le serveur vérifie que le token de l'utilisateur est valide sinon envoie un 401, "token invalide"
- Le serveur vérifie que l'id du post est renseigné dans le body en json (ex: {"id_post":"xxx"}) sinon envoie 422, "missing parameter : Id_Post"
- Le serveur vérifie que le post existe sinon 404, "Ce post n'existe pas"
- Le serveur vérifie ensuite que la rebrique like est renseigné dans le body en json (ex: {"like":1} ou {"like":-1}) ou que la rebrique contenu est renseigné dans le body en json (ex: {"contenu":"xxxx"}) sinon 422, "missing parameter : like or contenu"
- Si le like est renseigné le serveur vérifie que la valeur est bien 1 ou -1 (like ou dislike) sinon 422, "wrong parameter : like = (1:like / -1:dislike)"
- Le serveur vérifie que l'utilisateur est un publisher puis n'est pas le publisher du post sinon 401, "Vous n'êtes pas autorisé à liker ce post" ou 401, "vous ne pouvez pas liker votre propre post"

7

- Le serveur like ou dislike ensuite le post
- Le serveur vérifie que le post a pas déjà été like ou dislike par l'utilisateur sinon 405, "Vous avez déjà liker ou disliker ce post"
- Si il y a un contenu, le serveur vérifie que l'utilisateur est publier du post puis modifie le contenu sinon 401, "Vous n'êtes pas autorisé à modifier ce post"

Modifier un article :

PUT <http://localhost/projetREST/serveurApplication.php>

Entrées

Nom	Type	Description	
Id_Post	Int	Id le l'article que vous voulez modifier	Dans le body en json
Contenu	String	Nouveau contenu le l'article	

Sorties

Code	Message	Description
200	"Le post a bien été mis à jour"	
401	"token invalide"	Si le token est invalide
422	"missing parameter : Id_Post"	S'il manque le paramètre Id_Post
404	"Ce post n'existe pas"	Si le l'Id de l'article n'existe pas
401	"Vous n'êtes pas autorisé à modifier ce post"	Si vous n'êtes pas publier du post

Retours

Nom	Type	Description
Rien	--	--

Liker ou disliker un article :

PUT <http://localhost/projetREST/serveurApplication.php>

Entrées

Nom	Type	Description	
Id_Post	Int	Id le l'article que vous voulez modifier	Dans le body en json
Like	Int	1 pour liker l'article / -1 pour le disliker	

8

Sorties

Code	Message	Description
401	"token invalide"	Si le token est invalide
422	"missing parameter : Id_Post"	S'il manque le paramètre Id_Post
404	"Ce post n'existe pas"	Si le l'Id de l'article n'existe pas
422	"wrong parameter : like = (1:like / -1:dislike)"	Si like est différent de 1 ou -1
401	"Vous n'etes pas autorisé à liker ce post"	Si vous n'êtes pas connecté en tant que publisher
401	"vous ne pouvez pas liker votre propre post"	Si vous essayez de liker votre propre post
200	"post liké !"	
200	"post disliké !"	

Retours

Nom	Type	Description
Rien	--	--

Méthode DELETE:

Traitement :

- Le serveur vérifie que le token est valide sinon 401, "token invalide"
- Vérifie que l'id du post est renseigné dans le body en json (ex: {"id_post":"xxx"}) sinon 422, "missing parameter : Id_Post"
- Vérifie que le post existe sinon 404, "Ce post n'existe pas"
- Vérifie que l'utilisateur est modérateur ou publisher du post sinon 401, "vous n'êtes pas autorisé à supprimer ce post"
- Supprime le message

Supprimer un article :

POST <http://localhost/projetREST/serveurApplication.php>

Entrées

Nom	Type	Description
Id_Post	Int	Id le l'article que vous voulez supprimer

9

Sorties

Code	Message	Description
401	"token invalide"	Si le token est invalide
422	"missing parameter : Id_Post"	S'il manque le paramètre Id_Post
404	"Ce post n'existe pas"	Si le l'Id de l'article n'existe pas
422	"wrong parameter : like = (1:like / -1:dislike)"	Si like est différent de 1 ou -1
401	"Vous n'etes pas autorisé à liker ce post"	Si vous n'êtes pas connecté en tant que publisher
401	"vous ne pouvez pas liker votre propre post"	Si vous essayez de liker votre propre post
200	"post liké !"	
200	"post disliké !"	

Retours

Nom	Type	Description
Rien	--	--

Serveur Authentification

Méthode Post:

Traitement :

- Vérifie que l'username est dans le body en json (ex: {"username":"publisher_user"}) sinon il envoie 422, "missing parameter : username"
- Vérifie que le mot de pass est dans le bosdy en json (ex {"mot_de_passe":"publisher_mdp"}) sinon il envoie 422, "missing parameter : mot_de_passe"
- Fait la requête pour récupérer le rôle, si rien est renvoyé 404, "utilisateur introuvable dans la base de données"
- Ensuite envoie le jwt

Récupérer le token d'authentification :

POST <http://localhost/projetREST/serveurAuthentification.php>

Entrées

Nom	Type	Description
Username	String	Username de la personne voulant se connecter
Mot_de_passe	String	Mot de passe de la personne voulant se connecter

Dans le body en json

10

Sorties

Code	Message	Description
404	"utilisateur introuvable dans la base de données"	Si l'utilisateur n'est pas dans la base de données
200	"Authentification réussi !"	

Retours

Nom	Type	Description
Data	String	Token d'authentification à mettre dans le header de chaque requête sur le serveur Application

User Stories

En tant que Modérateur je souhaite pouvoir modifier un article que je juge inadéquat afin de modérer les contenus

En tant que publisher je souhaite pouvoir publier un article pour exprimer mes idées

En tant qu'utilisateur anonyme je souhaite pouvoir voir les articles de tous les publishers

En tant que publisher je souhaite pouvoir liker un article pour exprimer mon avis

En tant que publisher je souhaite pouvoir disliker un poste pour exprimer mon mécontentement