

Rapport API REST

Bonjour Bob

	Publisher	Liste des likes	Liste des dislikes	Nombre de likes
Langage singulier : Ils visent, colorent leur mules oubliées, phrases drôles, et deviennent des	Bob	Camille58		1
Une sœur dans un peu le courage de lui tard, elle l'a revu, et a réalisé l'amour de sa vie.	Camille58			0
l'espace, etant	Bob			0

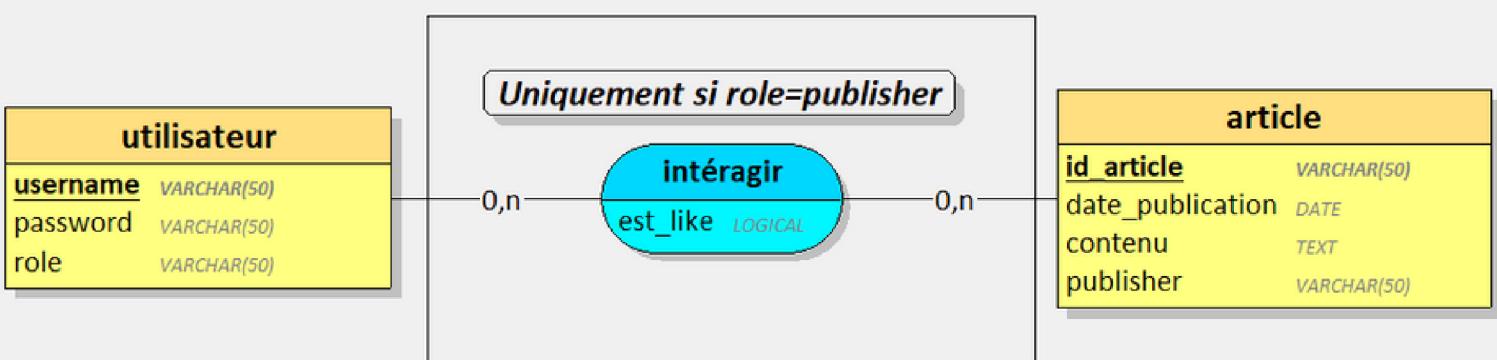
Ducourneau Gaïa

Sechi Chloé



[https://github.com/gaialou/
ArticleManagerAPI](https://github.com/gaialou/ArticleManagerAPI)

Notre MCD



Les spécifications

API authentification

- `POST` avec un json contenant `username=[valeur]` et `password=[texte]`

API articles

- `GET` : affiche tous les articles avec leur date et leur contenu (avec les rôles publisher et moderator, affiche également le publisher et le nombre de like et dislike)
- `GET` avec `?IdArticle=[IdArticle]` : Affiche l'article pour IdArticle correspondant s'il existe
- `POST` avec un json contenant `Id_article=[valeur]` et `Contenu=[texte]` : publier un nouvel article (disponible avec le rôle publisher uniquement)
- `PUT` avec `?IdArticle=[IdArticle]` avec un json contenant `Id_article=[valeur]` et `Contenu=[texte]` : Modifier un article
- `PATCH` avec `?IdArticle=[IdArticle]` avec un json contenant `est_like=[1(like) ou 0(dislike)]` : Permet de voter pour l'article pour IdArticle correspondant s'il existe (disponible avec le rôle publisher uniquement) Nous avons considéré qu'en dehors de la création le like ne change que de valeur, d'où l'utilisation du PATCH.
- `DELETE` avec `?IdArticle=[IdArticle]` : Supprimer un article (disponible avec les rôles publisher et moderator)

Spécification des API REST du projet avec les erreurs

API Authentication

```
Identification du type de méthode HTTP envoyée par le client
si la methode est{
    cas "POST" :
        Récupération des données envoyées par le Client en format JSON
        convertir en tableau PHP
        s'il y a des données dans username et password{
            Recherche de l'utilisateur dans la base de données
            si le usernmae et le password correspondent a ceux de la BD{
                créer le Header JWT = array('alg'=>'HS256', 'typ'=>'JWT');
                créer le payload JWT avec 'username', 'role' et temps d'expiration 1 heure
                Génération du JWT

                Si le JWT est valide{
                    Reponse 200, "Clé JWT générée avec succès"
                }Sinon{
                    Erreur 401, "Clé JWT échoué"
                }
            }Sinon{
                Erreur 401, "Erreur : Nom d'utilisateur ou mot de passe incorrect"
            }
        }Sinon{
            Erreur 400, "Nom d'utilisateur ou mot de passe manquant"
        }

        si JWT fourni par le client{
            Si JWT valide{
                Reponse 200, "Clé JWT valide"
            }Sinon{
                Erreur 401, "Clé JWT non valide"
            }
        }
    fin cas POST

    cas si une autre méthode est envoyer :
        erreur 405, "Méthode non autorisée"
    Fin
}
```

API Article

```
Recuperation du JWT
Extraction du role de l'utilisateur

Identification du type de méthode HTTP envoyée par le client
si la methode est{

    Cas "POST" :
        si le role est 'publisher' alors{
            Récupération des données envoyées par le Client
            Decodage du json
            si Id_article dans le lien n'est pas vide{
                $Date_publication = date du jour
                s'il n'y a pas Id_article dans le JSON{
                    Erreur 400, " JSON incomplet, Id_article manquant"
                }sinon{
                    s'il n'y a pas Contenu dans le JSON{
                        Erreur 400, " JSON incomplet, Contenu manquant"
                    }sinon{
                        extraire le publisher du JWT
                        Si id article n'est pas déjà attribuer alors{
                            inserer dans la base de données ce nouvel article avec toutes les informations
                            Reponse 200, "les données ont été ajoutées"
                        }sinon{
                            Erreur 409, "Identifiant déjà existant"
                        }
                    }
                }
            }
        }
    }
    Erreur 405, "Cette methode n'autorise pas de variable GET"
}
}sinon{
    Erreur 403, "Vous n'avez pas le role nécessaire pour cette méthode"
}
fin cas POST
```

```
Cas "PUT" :
    si le role est 'publisher' alors{
        Récupération des données envoyées par le Client
        Decodage du json
        extraire le publisher du JWT
        Si id article est attribuer alors{
            Si le publisher de l'article est le même que celui qui PUT alors{
                modifier dans la base de données cet article avec toutes les informations
                Reponse 200, "les données ont été mise à jour"
            }sinon{
                Erreur 403, "Vous n'êtes pas le publisher de l'article, vous ne pouvez pas le modifier"
            }
        }sinon{
            Erreur 404, "Pas d'article trouvé pour l'id donné"
        }
    }sinon{
        Erreur 403, "Vous n'avez pas le role nécessaire pour cette méthode"
    }
fin cas PUT
```

```

cas "DELETE" :
    si le role est 'publisher' ou 'moderator' alors {
        si l'id_article est bien dans le lien alors{
            si le role est 'moderator' alors{
                S'il existe un article pour l'id donnée alors{
                    Supprimer l'article
                    Reponse 200, "article supprimé"
                }sinon{
                    Erreur 404, "Pas d'article trouvé pour l'id donné"
                }
            }sinon{
                Extraire le publisher du JWT
                S'il existe un article pour l'id donnée alors{
                    Si le publisher de l'article est le même que celui qui DELETE alors{
                        Effacer l'article
                        Reponse 200, "OK : article supprimé"
                    }sinon{
                        Erreur 403, "Vous n'etes pas le pulisher de l'article, vous ne pouvez pas le supprimer"
                    }
                }sinon{
                    Erreur 404, "Pas d'article trouvé pour l'id donné"
                }
            }
        }sinon{
            Erreur 400, "DELETE ne fonctionne pas sans identifiant article"
        }
    }sinon{
        Erreur 403, "Vous n'avez pas le role nécessaire pour cette méthode"
    }
fin cas DELETE

```

```

case 'PATCH' :
    Si le role est publisher{
        Récupération des données envoyées par le Client
        Decode JSON
        s'il n'y a pas d'id_article dans le lien{
            extraire le publisher du JWT
            Si l'article existe bien alors{
                Si le publisher n'est pas celui qui a publier l'article{
                    Si le publisher a déjà interagis avec l'article alors{
                        Mettre à jour avec le nouveau vote
                        Reponse 200, "vote mis à jour"
                    }sinon{
                        Ajouter le vote dans la base de données
                        Reponse 200, "vote ajouté"
                    }
                }sinon{
                    Erreur 403, "Vous etes le pulisher de l'article, vous ne pouvez pas voter"
                }
            }sinon{
                Erreur 404, "Identifiant article introuvable"
            }
        }sinon{
            Erreur 400, "PATCH ne fonctionne pas sans identifiant article"
        }
    }sinon{
        Erreur 403, "Vous n'avez pas le role nécessaire pour cette méthode"
    }
Fin cas PATCH

```

```

Si ce n'est aucune des autres méthode alors c'est GET:
    Récupération des critères de recherche envoyés par le Client
    s'il y a id_article dans le lien alors{
        si le role est 'moderator' {
            Afficher pour l'article la liste des personnes qui ont like, ceux qui ont dislike, le nombre de like, de dislike, et tout les informations de article
        }sinon {
            Si le role est publisher{
                Afficher pour l'article le nombre de like, de dislike, et toute les informations de article
            }sinon{
                Afficher pour l'article la Date_publication, Contenu et le Publisher
            }
        }
    }
    Encode le resultat en JSON

}sinon{
    si le role est 'moderator' {
        Afficher pour chaque article la liste des personnes qui ont like, ceux qui ont dislike, le nombre de like, de dislike
    }sinon {
        Si le role est publisher{
            Afficher pour chaque article le nombre de like, de dislike, et toute les informations de article
        }sinon{
            Afficher pour chaque article la Date_publication, Contenu et le Publisher
        }
    }
    Encode le resultat en JSON

}
Envoi du resultat au Client
si le resultat n'est pas vide{
    Reponse 200, "données récupérées"
}sinon{
    Reponse 204, "Pas de résultat pour la requête demandée"
}
fin
}

```

Les Userstory

Modérateur

en tant que modérateur
je peux consulter les différent article je peux aussi les
supprimer peut importe le publisher à l'aide de la
méthode delete

Publisher

en tant que publisher
je peux utiliser la méthode Post pour rajouter un article
peux je like ou dislike un article pgrace à patch
je peux aussi modifier un article de ma creation a l'aide
de la methode put
et je ne peut supprimer que mes articles a l'aide la
methode delete

Anonyme

en tant que anonyme je peux seulement consulter les
articles publier et ne peut interagir avec eux

Interface utilisateur

Notre interface client permet au différent user à la limite de leur role attribué, de pouvoir utiliser de manière simple les différentes fonctionnalités de notre API tel que le like et le dislike et le post d'articles , elle permet aussi d'accéder au delete et enfin permet une visualisation permanente des articles à tout les user à l'aide du tableau qui utilise le get all de notre Api nous n'avons pas implémenter la modification d'article dans le client.

Bonjour Bob

Joli tableau de moderator ou publisher

Id_article	Date_publication	Contenu	Publisher	Liste des likes	Liste des dislikes	Nombre_likes	Nombre_dislikes	Vote
A0001	2022-11-16	Les comédiens ont un langage singulier : Ils se souviennent, improvisent, colorent leur mémoire, citent les formules oubliées, inventent même des phrases drôles, cruelles, qui se répètent et deviennent des citations.	Bob	Camille58		1	0	<button>Like</button> <button>Dislike</button> <button>Annuler mon vote</button>
A0002	2023-01-05	Elle a rencontré son âme sœur dans un café, mais n'a jamais eu le courage de lui parler. Des années plus tard, elle l'a revu, marié et avec des enfants, et a réalisé qu'elle avait laissé passer l'amour de sa vie.	Camille58			0	0	<button>Like</button> <button>Dislike</button> <button>Annuler mon vote</button>
A0003	2023-03-01	Après des mois de voyage dans l'espace, ils ont enfin atteint leur destination, seulement pour découvrir qu'ils étaient arrivés dans un univers parallèle où tout était exactement pareil, sauf eux-mêmes.	Bob			0	0	<button>Like</button> <button>Dislike</button> <button>Annuler mon vote</button>



Pour plus d'information
rendez vous sur notre github

lien : <https://github.com/gaialou/ArticleManagerAPI/>