

Groupomania

Documentation de l'API, au 29 août 2021

important :

- (*) authentification : lors de la réception d'une requête par le back-end, l'ID utilisateur est extrait du JWT (Authorization header "bearer")
- les dates sont en secondes et non en millièmes de seconde

les comptes utilisateurs				
description	verbe HTTP + URI	auth.*	body / form-data (exemple)	retour (exemple)
créer un compte	POST /api/auth/signup	NON	<pre>{ "firstname": "Lionel", "lastname": "Test", "position": "Dév Web", "email": "TEST@GMAIL.COM", "password": "Mot2Passe" }</pre>	<pre>{ "id": 4, "firstname": "Lionel", "lastname": "Test", "email": "test@gmail.com", "password": "", "position": "Dév Web", "role": "user", "createdate": 1630174423, "modifdate": 1630174423, "logindate": 1630174423, "accessToken": "...JWT...", "ok": "Création compte confirmée" }</pre>

s'authentifier	POST /api/auth/login	NON	{ "email": "TEST@GMAIL.COM", "password": "Mot2Passe" }	{ "id": 4, "firstname": "Lionel", "lastname": "Test", "position": "Dév Web", "email": "test@gmail.com", "password": "", "role": "user", "createdate": 1630174424, "modifdate": 1630174424, "logindate": 1630174424, "accessToken": "...JWT...", "ok": "Authentification confirmée" }
modifier son propre compte avec un mot de passe (on le change) ou = chaîne vide (on le conserve)	PUT /api/auth/:userId	OUI	{ "firstname": "Lionel", "lastname": "Test", "position": "Product Owner", "email": "TEST@GMAIL.COM", "password": "" } ou si le mdp change : { "firstname": "Lionel", "lastname": "Test", "position": "Product Owner", "email": "TEST@GMAIL.COM", "password": "nouveauM2P" }	{ "userId": "4", "firstname": "Lionel", "lastname": "Test", "email": "test@gmail.com", "position": "Product Owner", "role": "user", "modifdate": 1630177629, "password": "", "accessToken": "...JWT...", "ok": "Modification compte confirmée" }
supprimer son propre compte	DELETE /api/auth/:userId	OUI	NON	{ "ok": "Suppression compte confirmée" }

les articles publiés

description	verbe HTTP + URI	auth.*	body / form-data (exemple)	retour (exemple)
obtenir une liste d'articles selon des critères : <ul style="list-style-type: none"> - de recherche - de tri - de pagination 	POST /api/article/list	OUI	<pre>{ "payload": { "sortBy": "publicationDate", "invertSort": true, "pageQuantity": "1", "pageNumber": 99, "searchString": "" } }</pre>	<pre>{ "content": [{ "id": 23, "title": "Les églises rupestres de Lalibela", "message": "en Éthiopie", "image": "http://.../images/image_1629984036766.jpg", "userId": 2, "createdate": 1629984037, "modifdate": 1629984037, "email": "lionel.richie@gmail.com", "firstname": "Lionel", "lastname": "Richie", "totalLikes": 0, "totalComments": 0, "lastLike": null, "lastComment": null }], "page": 23 }</pre>
créer une publication	POST /api/article	OUI	form-data "title" = "Ma publication"; "message" = "mon message"; "image" = <i>fichier image</i>	<pre>{ "ok": "Article has been created", "newId": 24 }</pre>
supprimer une publication (la sienne, ou si l'on est admin ou moderator)	DELETE /api/article/:articleId	OUI	NON	<pre>{ "ok": "Article has been deleted" }</pre>

les appréciations (likes)				
description	verbe HTTP + URI	auth.*	body / form-data (exemple)	retour (exemple)
attribuer un like à un article (1 max. par compte)	POST /api/like/:articleId	OUI	NON	{ "ok": "article's likes found", "totalLikes": 1, "hasLiked": 1 }
obtenir les likes attribués à un article - total - lui ai-je attribué un like ?	GET /api/like/:articleId	OUI	NON	{ "ok": "article's likes found", "totalLikes": 0, "hasLiked": 0 }
retirer le like donné par un utilisateur à un article	DELETE /api/like/:articleId	OUI	NON	{ "ok": "article's likes found", "totalLikes": 0, "hasLiked": 0 }

les commentaires				
description	verbe HTTP + URI	auth.*	body / form-data (exemple)	retour (exemple)
ajouter un commentaire à un article	POST /api/comment/:articleId	OUI	{ "comment": "ceci est mon avis" }	{ "ok": "Comment has been added", "newId": 1 }
obtenir les commentaires rattachés à un article	GET /api/comment/:articleId	OUI	NON	{ "ok": "article's comments found", "comments": [{ "commentId": 1, "comment": "ceci est mon avis", "date": 1630185741, "userId": 8, "email": "test@mail.com", "firstname": "Lionel", "lastname": "Test" },], "hasCommented": 3 }
retirer un commentaire d'un utilisateur (le sien, ou si l'on est admin ou moderator)	DELETE /api/comment/:commentId	OUI	NON	{ "ok": "Comment has been removed" } même si le commentaire n'existe pas