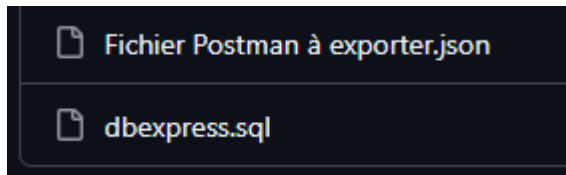


## Accès au GITHub

[VonPIP5/TP-ExpressJS](#)



← Fichier pour la collection Postman.

← Fichier pour importer la BDD.

## Accès au serveur

L'accès au serveur se fait via le port 5000, c'est ainsi que nous avons configuré notre serveur. Cela veut donc dire que pour voir le site après avoir lancé dans le cmd la commande `node server.js`, vous devez vous rendre sur le lien : <http://localhost:5000/>

## Les routes et leurs utilisations

### Routes de bases

Il existe deux routes de base :

Route	Méthode	Token	Admin	Inputs
<a href="#">/</a>	GET			
<a href="#">/test</a>	GET			

- La première à pour résultat d'envoyer le message : *Bienvenue sur l'API de Maël DESSUS et Julien LEFEBVRE*. Un message de bienvenue sur notre API.
- La deuxième à pour but de tester l'API avec l'envoi d'un message simple : *Hello*

## Enregistrer un utilisateur

Route	Méthode	Token	Admin	Inputs
/register	POST			<ul style="list-style-type: none"><li>• username</li><li>• email</li><li>• prenom</li><li>• nom</li><li>• password</li></ul>

Pour enregistrer un utilisateur sur la base de données il faut envoyer la requête a /register. Elle devra être en méthode POST et le contenu du body doit ressembler à cet exemple :

```
{
  "username": "user",
  "email": "user@example.com",
  "prenom": "Jack",
  "nom": "Lacroix",
  "password": "1234"
}
```

## Se connecter

Route	Méthode	Token	Admin	Inputs
/login	GET			<ul style="list-style-type: none"><li>• username</li><li>• password</li></ul>

Après s'être créé un compte on peut se connecter a l'aide de son nom utilisateur et mot de passe. En cas de concordance l'API nous retourne un token valable 1h. En cas d'échec, on obtient un message d'erreur. Voila à quoi doit ressembler le body :

```
{
  "username": "user",
  "password": "1234"
}
```

## Avoir les infos d'un utilisateur

Route	Méthode	Token	Admin	Inputs
/profil/{id}	get	x		<ul style="list-style-type: none"><li>• Token (authorization → bearer Token)</li></ul>

Pour obtenir les infos relatives à un utilisateur, on doit être connecté et disposer d'un token valide. Dans le lien de la route on ajoute l'id de l'utilisateur qu'on souhaite visualiser.

## Ajouter un fichier

Route	Méthode	Token	Admin	Inputs
/add-file	post	x		<ul style="list-style-type: none"><li>• Token (authorization → bearer Token)</li><li>• file (type file)</li></ul>

Pour ajouter un fichier, on doit être connecté et disposer d'un token valide. Dans le body de type form-data, on met le fichier dans un champ file.

## Supprimer un utilisateur

Route	Méthode	Token	Admin	Inputs
/users/rm/{id}	DELETE	x	x	<ul style="list-style-type: none"><li>• Token (authorization → bearer Token)</li></ul>

Pour supprimer un utilisateur, on doit être connecté et disposer d'un token valide ainsi qu'être administrateur. Dans le lien de la route on ajoute l'id de l'utilisateur qu'on souhaite supprimer.

## Bannir un utilisateur

Route	Méthode	Token	Admin	Inputs
/users/ban/{id}	PUT	x	x	<ul style="list-style-type: none"><li>Token (authorization → bearer Token)</li></ul>

Pour bannir un utilisateur, on doit être connecté et disposer d'un token valide ainsi qu'être administrateur. Dans le lien de la route on ajoute l'id de l'utilisateur qu'on souhaite bannir.

**Un utilisateur banni** n'aura plus la possibilité d'obtenir un token. S'il a déjà un token en activité, il n'aura pas la possibilité de faire d'autres commandes car une vérification de son droit d'accès est automatiquement effectuée à chaque commande demandée.

## Liste des utilisateurs

Route	Méthode	Token	Admin	Inputs
/users/list	GET	x	x	<ul style="list-style-type: none"><li>Token (authorization → bearer Token)</li></ul>

Pour avoir la liste des utilisateurs, on doit être connecté et disposer d'un token valide ainsi qu'être administrateur.

## Promouvoir un utilisateur admin

Route	Méthode	Token	Admin	Inputs
/users/up/{id}	PUT	x	x	<ul style="list-style-type: none"><li>Token (authorization → bearer Token)</li></ul>

Pour promouvoir un utilisateur admin, on doit être connecté et disposer d'un token valide ainsi qu'être administrateur. Dans le lien de la route on ajoute l'id de l'utilisateur qu'on souhaite promouvoir.

## Promouvoir un utilisateur admin




Route	Méthode	Token	Admin	Inputs
/users/down/{id}	PUT	x	x	<ul style="list-style-type: none"><li>• Token (authorization → bearer Token)</li><li>• id</li></ul>

Pour rétrograder un utilisateur, on doit être connecté et disposer d'un token valide ainsi qu'être administrateur. Dans le lien de la route on ajoute l'id de l'utilisateur qu'on souhaite rétrograder .

## Identifiant admin

username : admin

mot de passe : 1234

<div>↩ T ↪</div>										
	▼	id	username	email	nom	prenom	role	password	etatban	
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	5	admin	admi@admin.fr	admin	admin	1234	0