

SNAPCHAT API PAR MOUHOU_N

Les communications vers mon API passent par des requêtes HTTP(S) POST. L'API renvoi systématiquement un objet JSON de la forme :

```
{  
    error : , // false si pas d'erreur, sinon erreur écrite  
    data : // un objet contenant les données demandées  
    token : // pour la connexion seulement et les vérifications  
}
```

L'API peut recevoir les données

- via **AngularJS** :

Exemple : `$http.post('url', data) ;`

- via **javascript ou jQuery**

Exemple : `$.post("url", data);`

Inscription :

url = « <http://localhost:3000/signup> »

Envoyer un objet en post :

```
{  
    email : exemple@hotmail.fr,  
    password : *****  
}
```

Si l'email renseigné est valide et que le mot passe a bien était renseigné , la réponse que vous serait sous forme d'objet de type :

```
{  
    error : false,  
    data : object (avec l'email renseigné et le mot de passe hashé)  
    token : null  
}
```

Sinon pour n'importe quel erreurs, vous recevrez le meme type d'objet , avec l'erreur préciser tels que :

```
{  
    error : 'email invalide',  
    data : null  
    token : null  
}
```

les erreurs que vous rencontrerez seront très précise, vous pourrez donc les corriger très rapidement.

Connexion :

url = « <http://localhost:3000/login> »

Envoyer un objet en post :

```
{  
    email : exemple@hotmail.fr,  
    password : *****  
}
```

Si l'email renseigné est valide et que le mot passe a bien était renseigné , la réponse que vous serait sous forme d'objet de type :

```
{  
    error : false,  
    data : object (avec l'email , l'id et le token de la personne qui se connecte))  
    token : token généré pour la sécurité  
}
```

```
}
```

Sinon pour n'importe quel erreurs, vous recevrez le meme type d'objet , avec l'erreur préciser tels que :

```
{  
    error : 'email ou mot de passe incorrecte',  
    data : null  
    token : null  
}
```

Voire les membres inscrit :

url = « <http://localhost:3000/listuser> »

Envoyer un objet en post :

```
{  
    email : « de la personne connecter »  
    token : « enregistrer lors de la connection »  
}
```

Si l'email renseigné est valide et que le mot passe a bien était renseigné , la réponse que vous serait sous forme d'objet de type :

```
{  
    error : false,  
    data : object (avec la liste des personnes inscrite))  
    token : votre token généré  
}
```

Sinon si un paramètre est manquant, il sera précisé.

Voire la liste des snaps recu :

url = « <http://localhost:3000/listsnap> »

Envoyer un objet en post :

```
{  
    email : « de la personne connecter »  
    token : « enregistrer lors de la connection »  
}
```

Si l'email renseigné est valide et que le mot passe a bien était renseigné , la réponse sera sous forme d'objet de type :

```
{  
    error : false,  
    data : object (avec la liste des personnes inscrite))  
    token : votre token généré  
}
```

Sinon si un paramètre est manquant, il sera précisé.

Envoyer un snap

url = « <http://localhost:3000/sendpeax> »

Envoyer un file en post :

```
{  
    email : « de la personne connecter »  
    temps : durée du snap (INT),  
    id_receiver : « id du receveur (INT) »  
    token : « enregistrer lors de la connection »  
}
```

Si toutes les paramètre ont bien été renseigné , la réponse que vous sera sous forme d'objet de type :

```
{
    error : false,
    data : null,
    token : null
}
```

Sinon si un paramètre est manquant, il sera précisé.

Modification EMAIL

url = « <http://localhost:3000/editemail> »

Envoyer un objet en post :

```
{
    id: « id de la personne connecter »
    email : « de la personne connecter »
    newemail : « nouveau email »
    token : « enregistrer lors de la connection »
}
```

Si toutes les paramètre ont bien été renseigné , la réponse que vous sera sous forme d'objet de type :

```
{
    error : false,
    data : null,
    token : null
}
```

Sinon si un paramètre est manquant, il sera précisé.

Modification pass

url = « <http://localhost:3000/editpass> »

Envoyer un objet en post :

```
{
    id: « id de la personne connecter »
    email : « de la personne connecter »
    password : « nouveau mot de pass »
}
```

Si toutes les paramètre ont bien été renseigné , la réponse que vous sera sous forme d'objet de type :

```
{
    error : false,
    data : null,
    token : null
}
```

Sinon si un paramètre est manquant, il sera précisé.

Validation de vue

url = « <http://localhost:3000/vue> »

Envoyer un objet en post :

```
{
    email : « de la personne connecter »
    token : votre token
    id_snap : id du snap
}
```

Si toutes les paramètre ont bien été renseigné , la réponse que vous sera sous forme d'objet de type :

```
{  
    error : false,  
    data : null,  
    token : null  
}
```

Sinon si un paramètre est manquant, il sera précisé.