

DOCUMENTATION DE L'API

Note : Lorsqu'il y a par exemple :variable dans une url vous devez remplacer :variable par le paramètre en question. Lorsqu'il y a un Astérix « * » cela veut dire que le paramètre est obligatoire

Note2 : Dans les cas où on veut juste savoir si un appel a fonctionné ou non, la réponse sera de cette forme :

```
{
    success : TRUE | FALSE,
    message : MESSAGE
}
```

Utilisateurs

Création d'utilisateurs :

Url : /users

Méthode : POST

Paramètres :

- email *
- password *
- name
- address
- description
- city
- country
- picture (sous forme de fichier)

Remarques : La requête est en multipart/form-data et non urlencodé comme les autres.

Lister tous les utilisateurs :

Url : /users

Méthode : GET

Paramètres : aucun

Détail d'un utilisateur:

Url: /users/:idUser

Méthode : GET

Paramètres :

- :idUser *

Remarques : Les succès affichés lors d'un GET sur un utilisateur sont les succès que l'utilisateur a déjà réussi. Les succès qui ne sont pas affichés reviennent à dire qu'il ne les a pas complétés.

Supprimer un utilisateur :

Url : /users/:idUser

Méthode : DELETE

Paramètres :

- :idUser *

Remarques : Pour pouvoir supprimer un utilisateur vous devez avoir admin à true en étant connecté ou alors être connecté avec ce même compte que l'on veut supprimer.

Mettre à jour un utilisateur :

Url : /users/:idUser

Méthode : PUT

Paramètres :

- :idUser*
- password
- name

- address
- description
- city
- country

Remarques : Pour pouvoir mettre à jour un utilisateur vous devez avoir admin à true en étant connecté ou alors être connecté avec ce même compte que l'on veut mettre à jour.

Recevoir le token pour s'authentifier :

Url : /users/authenticate

Méthode : POST

Paramètres :

- email*
- password*

Playlists

Création de playlists :

Url : /playlists

Méthode : POST

Paramètres :

- name*

Remarques: le champs « created_by » sera automatiquement crée par l'ID de celui qui fais l'appel.

Lister toutes les playlists :

Url : /playlists

Méthode : GET

Paramètres : aucun

Détail d'une playlist:

Url: /playlists/:idPlaylist

Méthode : GET

Paramètres :

- :idPlaylist *

Remarques : dans l'index « songs » d'un GET sur une playlist vous pouvez récupérer toutes les musiques d'une playlist.

Supprimer une playlist :

Url : /playlists/:idPlaylist

Méthode : DELETE

Paramètres :

- :idPlaylist *

Remarques : Pour pouvoir supprimer une playlist vous devez avoir admin à true en étant connecté ou alors être connecté avec le compte qui a crée la playlist que l'on veut supprimer.

Mettre à jour le nom d'une playlist :

Url : /playlists/:idPlaylist

Méthode : PUT

Paramètres :

- :idPlaylist*
- name*

Remarques : Pour pouvoir mettre à jour une playlist vous devez avoir admin à true en étant connecté ou alors être connecté avec le compte qui a crée la playlist que l'on veut modifier.

Ajouter une musique à une playlist :

Url : /playlists/:idPlaylist

Méthode : POST

Paramètres :

- :idPlaylist*
- idSong*

Remarques : Pour pouvoir ajouter une musique à une playlist vous devez avoir admin à true en étant connecté ou alors être connecté avec le compte qui a créé la playlist que l'on veut modifier.

Supprimer une musique à une playlist :

Url : /playlists/:idPlaylist

Méthode : DELETE

Paramètres :

- :idPlaylist*
- idSong*

Remarques : Pour pouvoir supprimer une musique à une playlist vous devez avoir admin à true en étant connecté ou alors être connecté avec le compte qui a créé la playlist que l'on veut modifier.

Musiques

Création de musiques :

Url : /songs

Méthode : POST

Paramètres :

- name*
- artist*
- picture* (sous forme de fichier)
- price*
- file* (sous forme de fichier)
- difficulty*

Remarques: Une musique ne peut être créée que par l'administrateur (admin à « true ») La requête est en multipart/form-data et non urlencoded comme les autres.

Lister toutes les musiques :

Url : /songs

Méthode : GET

Paramètres : aucun

Détail d'une musique:

Url: /songs/:idSong

Méthode : GET

Paramètres :

- :idSong *

Supprimer une musique :

Url : /songs/:idSong

Méthode : DELETE

Paramètres :

- :idSong *

Remarques : Pour pouvoir supprimer une musique vous devez être administrateur.

Mettre à jour une musique :

Url : /songs/:idSong

Méthode : PUT

Paramètres :

- :idSong*
- name

- artist
- price
- difficulty
- downloaded

Remarques : Pour pouvoir mettre à jour une musique vous devez être admin.

Uploader ou mettre à jour une image pour la musique:

Url : /users/:idSong/picture

Méthode : POST

Paramètres :

- picture* (Le fichier à envoyer)
- :idSong*

Remarques: Pour récupérer l'image d'une musique l'url est la suivante : http://www.nomdedomaine.fr/uploads/avatar/ID_SONG/IMAGE_NAME (IMAGE_NAME contenue dans la variable picture de la musique qui est le nom du fichier uploadé) Autorisation administrateur nécessaire

Uploader ou mettre à jour la musique:

Url : /users/:idSong/music

Méthode : POST

Paramètres :

- music* (Le fichier à envoyer(format midi ou mid de préférence))
- :idSong*

Remarques: Pour récupérer une musique l'url est la suivante : http://www.nomdedomaine.fr/uploads/songs/ID_SONG/MUSIC_NAME (MUSIC_NAME contenue dans la variable file de la musique qui est le nom du fichier uploadé) Autorisation administrateur nécessaire

Uploader ou mettre à jour la preview d'une musique:

Url : /users/:idSong/preview

Méthode : POST

Paramètres :

- preview* (La preview a envoyer)
- :idSong*

Remarques: Pour récupérer une musique l'url est la suivante : www.nomdedomaine.fr/uploads/songs/ID_SONG/PREVIEW_NAME (PREVIEW_NAME contenue dans la variable preview de la musique qui est le nom du fichier uploadé) Autorisation administrateur nécessaire

Succès

Remarques : Nous ne pouvons que mettre la description globale de chaque succès et savoir qui les a validés mais concernant le fonctionnement de tel ou tel succès (par exemple atteindre 5 musiques achetées) ce sera défini au niveau du code de chaque applications. Les succès n'étant pas créés pour le moment, nous n'avons pas ajoutés les statistiques nécessaires aux modèles correspondants (par exemple nombre de musique achetées par utilisateurs)

Création d'un succès :

Url : /achievements

Méthode : POST

Paramètres :

- name*
- description*
- picture* (sous forme de fichier)

Remarques : Seul un administrateur peut créer un succès. La requête est en multipart/form-data et non urlencoded comme les autres.

Lister tous les succès :

Url : /achievements

Méthode : GET

Paramètres : aucun

Supprimer un succès :

Url : /achievements/:idAchievement

Méthode : DELETE

Paramètres :

- :idAchievement *

Remarques : Pour pouvoir supprimer un succès vous devez être administrateur.

Mettre à jour un succès :

Url : /achievements/:idAchievement

Méthode : PUT

Paramètres :

- :idAchievement*
- name
- description
- picture

Remarques : Pour pouvoir mettre à jour un succès vous devez être administrateur.

Compléter un succès à un utilisateur :

Url : /achievements/:idAchievement

Méthode : POST

Paramètres :

- :idAchievement*
- idUser*

Remarques : Cette requête pourra être refusée si l'utilisateur n'a pas les pré-requis pour compléter le succès.

News

Création de news :

Url : /news

Méthode : POST

Paramètres :

- name*
- description*
- picture* (sous forme de fichier)

Remarques: Une news ne peut être créée que par l'administrateur (admin à « true ») et le champ « author » sera automatiquement généré. La requête est en multipart/form-data et non urlencodée comme les autres.

Lister toutes les news :

Url : /news

Méthode : GET

Paramètres : aucun

Détail d'une news:

Url: /news/:idNews

Méthode : GET

Paramètres :

- :idNews *

Supprimer une news :

Url : /news/:idNews

Méthode : DELETE

Paramètres :

- :idNews *

Remarques : Pour pouvoir supprimer une news vous devez être administrateur.

Mettre à jour une news :

Url : /news/:idNews

Méthode : PUT

Paramètres :

- :idNews*
- name
- description
- picture

Remarques : Pour pouvoir mettre à jour une news vous devez être admin.