

DevWebRestApi

Quelques mots

Cette API permettra de récupérer, modifier, supprimer toutes les ressources stockées sur le serveur.

Les ressources de la base de données seront renvoyées au format JSON.

L'API est réalisée avec NodeJs ce qui lui permet d'être ultra légère est très rapide. Les modules utilisés sont : mysql, express, body-parser, url, cors.

Ressources

- Base de données : Toutes les données seront accessibles
- Ressources multimédias : Chaque point possède des ressources (images, sons, vidéos, etc.) et ces ressources seront stockées sur le serveur seront accessibles

Requêtes

Les ressources seront accessibles via des requêtes HTTP.

Verbes HTTP

- GET : Récupérer des ressources
- POST : Ajouter des ressources
- PUT : Modifier des ressources
- DELETE : Supprimer des ressources

EndPoints

Après avoir choisi quelle action vous souhaitez réaliser, vous devez renseigner quelle ressource vous souhaitez traiter.

chaque requête commencera par : `http://localhost:3000/api/`

Ensuite, vous devez compléter l'url par la table de la base de données qui vous intéresse (points, catégories, etc.), par exemple : `http://localhost/api/points`

Si vous vous arrêtez là vous recevrez toute la table que vous avez renseignée. Mais vous pouvez rajouter un second niveau pour rechercher des occurrences d'une table en fonction d'une autre table.

Par exemple :

```
http://localhost:3000/api/points/categories?nameCategorie='flore'
```


cette requête signifie : "je veux récupérer tous les points qui sont dans la catégorie flore".

Paramètres


À chaque endPoints vous pourrez/devrez (en fonction du contexte) compléter l'URL avec des paramètres. Ils vous permettront d'affiner votre requête. Vous devrez renseigner le nom de la colonne concernée et la valeur que vous recherchez :

Par exemple : `http://localhost:3000/api/points?idPoint=1` vous permettra de récupérer uniquement le point ayant pour id 1.

Vous pouvez y renseigner n'importe quelle colonne tant que celle-ci existe dans la base de données.

 Dans une requête GET, PUT ou DELETE, vous devez encadrer la valeur par des simples côtes si celle-ci est une valeur non numérique :

```
http://localhost:3000/api/points?namePoint='Colibris'
```

 Dans une requête POST, Quelle que soit le type de valeur, vous devez impérativement les encadrer avec des simples côtes