



Au cours du développement de notre application, nous aurons besoin d'une API. Une API est une interface permettant de relier un logiciel ou un service, à un autre. Ici, nous aurons besoin de ce genre d'outils dans le but de relier notre application codée en JavaScript, à une base de données. Nous avons choisi de créer une base de données MySQL. En effet, MySQL est un système de gestion de base de données relationnelles open source (1). De plus, MySQL est très performant (2), ce qui n'est pas forcément nécessaire étant donné que notre base de données sera plutôt petite, mais cela reste un point positif. Enfin, au vu des données que nous allons modifier au cours de notre application, il est plus simple et pratique d'utiliser ce genre de langage (3). Cela nous permettra par exemple de rajouter des lignes à notre base de données ou d'en modifier des existantes, tout cela depuis notre application.

Pour créer l'API dont nous avons besoin, nous utiliserons Node.js. Cela va être le plus pratique pour nous, car notre application est déjà codée dans ce langage, cela nous évitera d'avoir à utiliser encore un autre langage. Cette API va nous permettre de réaliser plusieurs opérations sur notre base de données, le tout depuis notre application et de manière automatisée (en tout cas nous l'espérons). Prenons un exemple simple : Un utilisateur lance une partie de snake.

Tout d'abord, nous avons besoin de connaître le nom d'utilisateur du joueur, information qui est stockée dans notre base de données. Pour le récupérer, nous allons créer une fonction GET, qui nous permettra d'aller chercher les informations demandées dans la base de données. Ensuite, à la fin de la partie, nous allons devoir réutiliser le GET pour connaître le nombre de BobCoins dont le joueur disposait avant sa partie.

Ayant fini de jouer, le joueur doit donc gagner de nouvelles pièces, et pour cela nous devons mettre à jour la table concernée. Nous avons donc besoin d'une fonction PUT, pour mettre à jour cette information en ajoutant les pièces gagnées avec la partie de snake.

Enfin, nous aurons peut être besoin d'ajouter une nouvelle ligne à notre base de données, pour par exemple, ajouter cette partie à l'historique des parties de ce joueur. Pour cela, nous utiliserons la fonction POST.

Nous n'aurons pas forcément besoin d'utiliser de suppression, mise à part pour de la maintenance. Nous avons donc quand même besoin de la fonction DELETE, qui nous permettra de supprimer des lignes dans notre base de données (si on veut bannir un joueur qui aurait triché par exemple).

(1) : <https://www.data-bird.co/sql/mysql>

(2) : <https://www.purestorage.com/fr/knowledge/what-is-mysql.html>

(3) : <https://blog.logrocket.com/build-rest-api-node-express-mysql/>