

ENSEIRB-MATMECA filière Réseaux et Systèmes d'Information

Réseaux et applications réparties

API Tennis

Any LAFAYE et Jordan BOUTEILLÉ
03/04/2020

Table des matières

I.	Présentation.....	2
II.	Outils.....	3
III.	Base de données d'API Tennis	4
IV.	Les requêtes HTTP	5
V.	Diagrammes de séquences.....	7

I. Présentation

API Tennis permet d'obtenir des données relatives au monde du tennis professionnel. Ces informations concernent les joueurs professionnels, les matchs disputés en France, les résultats, et les statistiques de ces joueurs.

Cette API est l'intermédiaire entre le client et le serveur. La base de données Tennis a été créée pour ce projet. Elle contient les données concernant les joueurs, les matchs, et les résultats (cf. Base de données d'API Tennis). Elle n'est pas complète.

API Tennis est une API REST. Un client envoie une requête HTTP à API Tennis qui, selon la demande de l'utilisateur, communique avec deux autres API pour répondre au client (voir le schéma ci-dessous - Communication entre le client et le serveur).

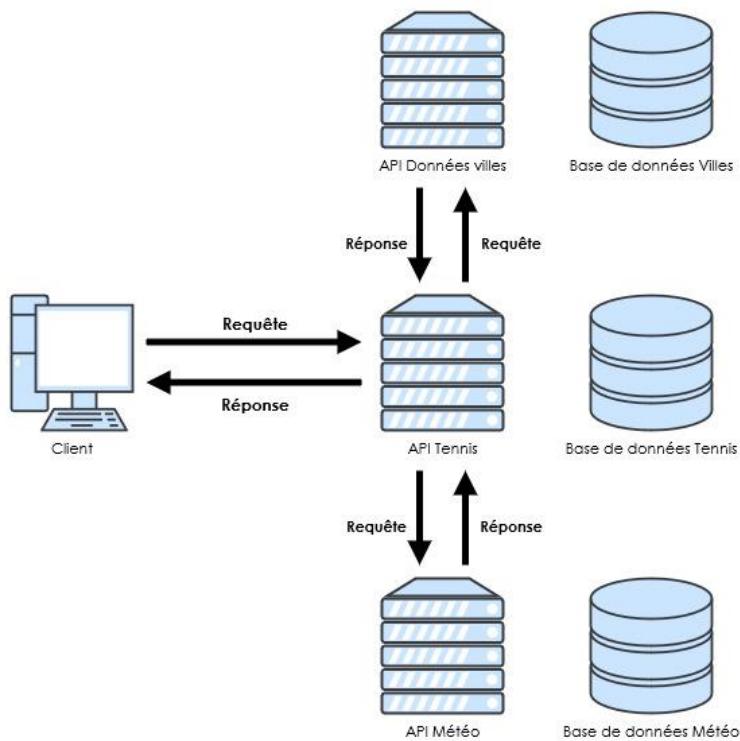


Figure 1 - Communication entre le client et le serveur

L'API Données Villes permet d'obtenir les données des villes françaises contenues dans la base de données Villes. L'API Météo quant à elle, permet de récupérer grâce à ces données, les informations météo de ces villes qui sont stockées dans la base de données Météo.

II. Outils

Pour réaliser ce projet, le langage de programmation préféré est Node.js. Il est de plus en plus utilisé parce qu'il est "puissant" et "rapide". Il possède en effet, une architecture asynchrone qui permet au programme de ne pas exécuter les lignes du code dans l'ordre où elles sont écrites.

Au début de ce projet, ce langage était inconnu de notre équipe. Mais nous avons fait ce choix pour en apprendre davantage sur ce dernier.

Quant au système de gestion de base de données, nous avons préféré MySQL qui est plus adapté à l'utilisation d'API Tennis.

III. Base de données d'API Tennis

Comme expliqué dans la partie Présentation, une base de données a été conçue. Ci-dessous, le modèle conceptuel de données (MCD) :

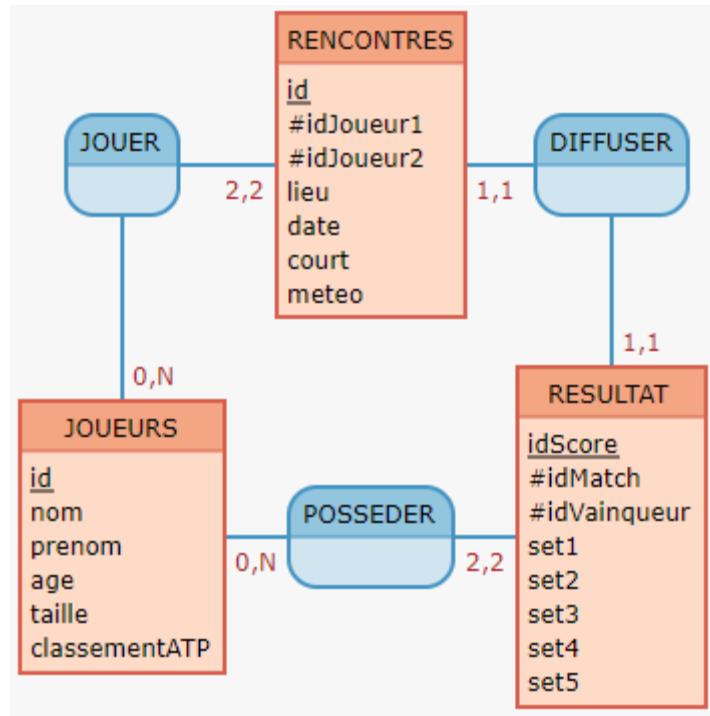


Figure 2 - MCD de la base de données

La base de données MySQL est composée de trois tables :

- JOUEURS intégrant les données des joueurs de tennis.
- RENCONTRES possédant les matchs qui ont été et vont être joués. On peut connaître le lieu, la météo prévue, etc. Cette table contient deux clés étrangères faisant référence au champ id de la table JOUEURS.
- RESULTAT qui pour chaque match joué, stocke le vainqueur du match et le score. Le champ idVainqueur est une clé étrangère de JOUEURS et idMatch de la table RENCONTRES.

Comme nous allons le voir par la suite, des requêtes SQL seront envoyées afin de récupérer les données utiles en fonction des requêtes HTTP.

IV. Les requêtes HTTP

Les requêtes HTTP de l'API Tennis

API Tennis permet à un client d'envoyer des requêtes HTTP de type GET pour obtenir/lire des informations stockées dans la base de données Tennis, et elle lui répond au format JSON.

Nous présenterons dans un premier temps, les requêtes HTTP de l'API ne sollicitant que la base de données Tennis, puis nous parlerons d'une autre requête HTTP faisant appel à des API externes.

API et Base de données Tennis

Les différentes URL accessibles via API Tennis sont :

- [http://\[adresse_serveur\]/joueurs](http://[adresse_serveur]/joueurs)

Cette requête affiche tous les joueurs de tennis contenus dans la base de données.

- [http://\[adresse_serveur\]/joueur/{NOM}](http://[adresse_serveur]/joueur/{NOM})

L'API recherche le joueur en question grâce au paramètre « NOM » dans la requête et envoie les informations relatives à ce tennismen.

- [http://\[adresse_serveur\]/resultat/{Nb}](http://[adresse_serveur]/resultat/{Nb})

Nous pouvons obtenir les résultats des « Nb » derniers matchs.

- [http://\[adresse_serveur\]/stats/{NOM}/{Nb}](http://[adresse_serveur]/stats/{NOM}/{Nb})

Il est utile de connaître les statistiques d'un joueur (nombre de victoires). Ainsi, l'API obtiendra les « Nb » derniers matchs du joueur identifié par « NOM » et il calculera ses statistiques.

- [http://\[adresse_serveur\]/prochainsMatchs/{Nb}](http://[adresse_serveur]/prochainsMatchs/{Nb})

Enfin, l'API REST permet de fournir « Nb » prochains matchs (Nb = nombre de matchs) qui se dérouleront en France. Ainsi, pour une ligne (une rencontre) nous aurons : la date, les joueurs qui s'affrontent, le lieu, le court ainsi que la météo. Mais pour obtenir cette météo, notre API fait appel à des API externes.

Appels vers des APIs externes

Pour récupérer la météo, API Tennis fait appel à l'API [météo-concept](#). Cette dernière fournit la météo sur plusieurs jours grâce au code INSEE d'une commune.

La requête HTTP de type GET à envoyer pour obtenir la météo se fait à partir de l'URL suivante :

<https://api.meteo-concept.com/api/forecast/daily/{Jour}?token={token}&insee={insee}>

Le paramètre « Jour » correspond à la journée à laquelle nous souhaitons obtenir la météo (allant de 0 pour aujourd'hui et 13 pour J+13). La météo peut aller jusqu'à 14 jours maximum. Puis, « token » est une clé API afin de récupérer des données de l'API météo. Pour finir, « INSEE » est le code INSEE associé à une ville.

Ici, nous avons besoin du code INSEE des communes pour acquérir la météo. Donc une [seconde API](#) est appelée via l'URL suivante :

<https://api-adresse.data.gouv.fr/search/?q={Ville}>

Le seul paramètre sera {ville} étant le lieu d'un match.

Les données au format JSON de ces requêtes seront traitées par API Tennis pour mettre à jour la base de données.

Ainsi, pour mettre la météo à jour, l'URL à utiliser vers l'API est la suivante :
[http://\[adresse_serveur\]/MAJMeteo](http://[adresse_serveur]/MAJMeteo)

Cette requête permettra de modifier la table RENCONTRES en ajoutant la météo pour les matchs à venir.

V. Diagrammes de séquences

Pour conclure, deux diagrammes de séquences décrivant les différentes interactions liées à API Tennis.

A. Du client vers la base de données Tennis

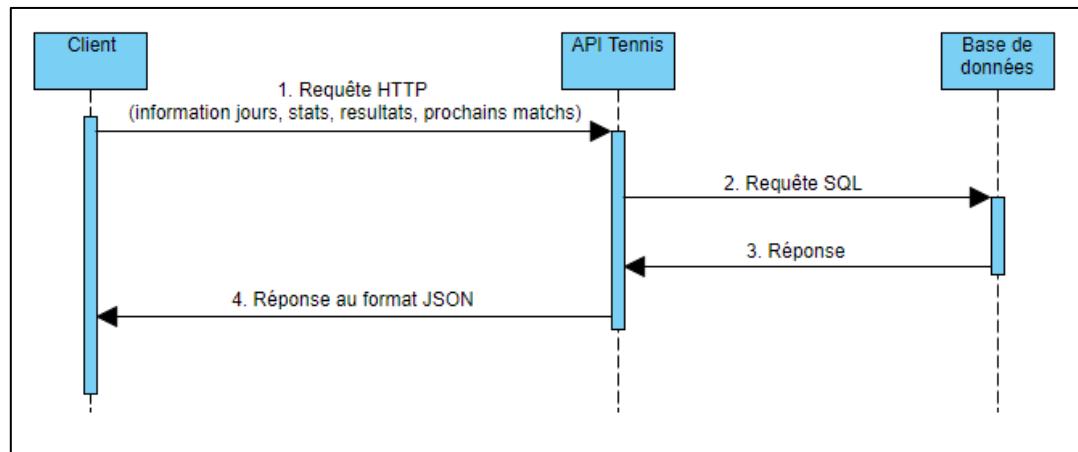


Figure 3 - Diagramme de séquence n°1

B. Appels vers les deux API externes

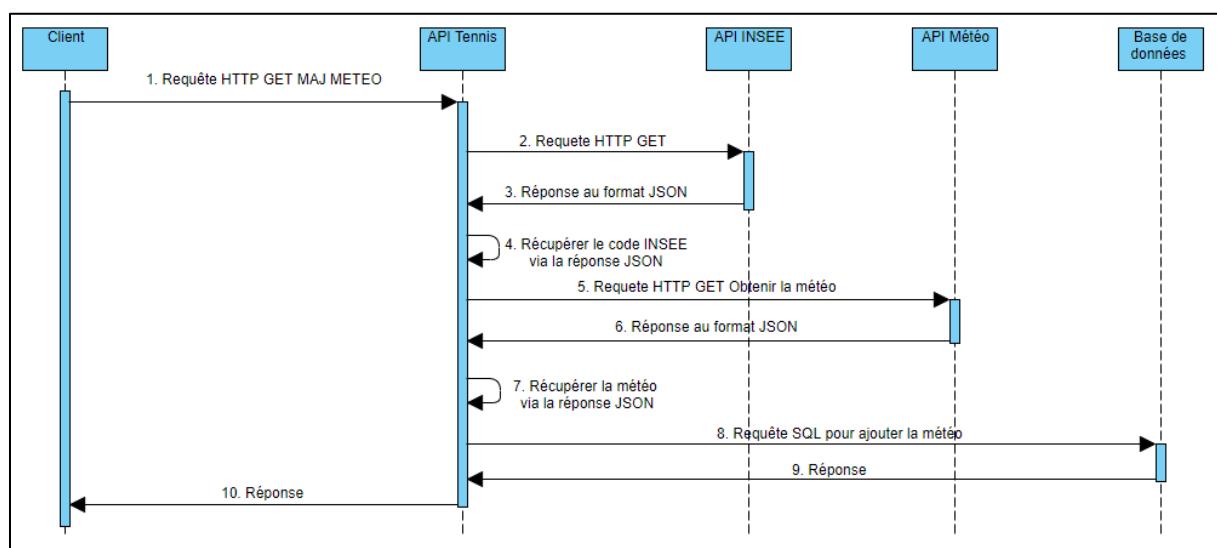


Figure 4 - Diagramme de séquence n°2