

MAC Projet

Bot Telegram de recette de
cuisine

Bouyiatotis Stéphane – Gomes Da Costa Joshua
– Valvona Guillaume



MAC : Projet

Bot Telegram de recette de cuisine

Le but de ce projet était de réaliser un bot Telegram de recettes de cuisine permettant à des utilisateurs d'interagir entre eux et permettant de proposer des recettes, de les partager et d'en discuter. Les utilisateurs peuvent utiliser différents types de recherches afin de trouver une recette. Les recettes sont notables et un système de recommandation a été mis en place.

Fonctionnalité

Le bot en lui-même possède plusieurs fonctionnalités :

- Recherche de recette : l'utilisateur pourra rechercher une recette par son nom. Une liste lui sera alors proposée et la personne n'aura plus qu'à sélectionner la recette pour la faire apparaître dans le chat.
- Recherche par ingrédient : le même principe que ci-dessus sera proposé, pour la recherche, mais avec des ingrédients ou le temps de préparation.
- L'utilisateur pourra sélectionner son régime, végétarien ou non, et les plats recherchés s'adapteront.
- Il pourra aussi demander au bot d'afficher les recettes les plus populaires, les recettes qu'il a aimées ou demander au bot de choisir une recette pour lui.

Base de données

Nous avons deux systèmes de stockage. Le premier étant MongoDB, qui est une base de données NoSQL qui sert principalement à stocker nos données de recette de cuisine et qui permet leurs recherches.

Le deuxième est Neo4j, qui lui va stocker sous forme de graphe NoSQL les différentes interactions entre les utilisateurs, les ingrédients et les recettes.

MongoDB

Pour remplir notre BDD nous utilisons un fichier .csv contenant nos recettes et les inscrivant dans notre BDD au lancement.

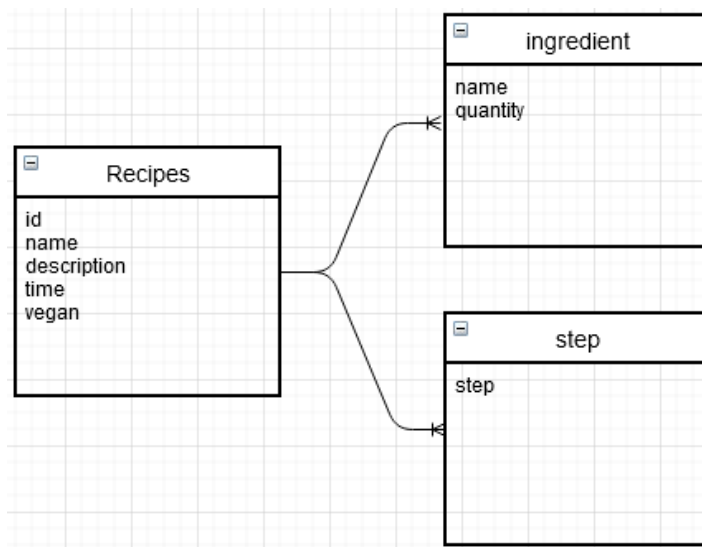


Figure 1 : Modèle de stockage de donnée

Pour les recettes, chacune possède :

- Un id sous forme de nombre
- Un nom sous forme de texte
- Une description sous forme de texte
- Le temps de préparation en nombre, représente les minutes.
- Si le plat est pour les végétariens sous forme de booléen

Les recettes possèdent aussi une liste d'ingrédient avec :

- Leurs noms sous forme de texte
- Et leurs quantités sous forme de texte

Les recettes comprennent une marche à suivre séparée en plusieurs étapes et sont affichées par step qui contient la marche à suivre d'une étape.

Neo4j

Le second système de stockage, Neo4j, lui va s'occuper de garder les liens entre les personnes et les recettes.

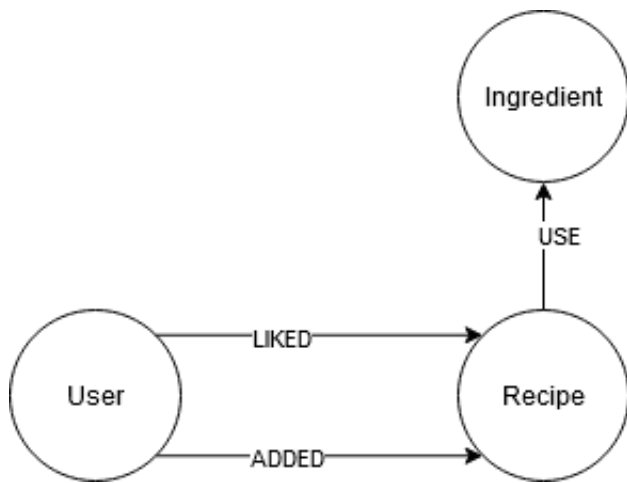


Figure 2 Modèle de stockage du graphe

Dans le graphe, nous considérons les recettes, les ingrédients et les utilisateurs comme des nœuds et l'existence des relations suivantes :

- Un utilisateur note une recette, la note a pour attribut :
 - Un rang sous forme de nombre allant de 1 à 5
 - Une date sous forme de DateTime de Neo4j
- Un utilisateur ajoute une recette, l'ajout a une date sous forme de DateTime de Neo4j
- Une recette utilise des ingrédients, l'utilisation n'a pas d'attribut

Les attributs des nœuds quant à eux sont les suivant :

Pour les recettes, on a :

- Un id sous forme de nombre
- L'Id dans à la base MongoDB sous forme de texte
- Un nom sous forme de texte
- Si le plat est pour les végétariens sous forme de booléen

Les ingrédients ont :

- Un id sous forme de nombre
- Un nom sous forme de texte

Enfin les utilisateurs ont :

- Un id sous forme de nombre
- L'id correspondant au compte Telegram
- Un booléen pour savoir s'il s'agit d'un bot
- Un booléen pour savoir s'il est végétarien
- Un nom sous forme de texte

Requêtes

Requête à la bd Neo4j

On effectue une recherche en fonction d'un ingrédient.

```
$ MATCH (:Ingredient{name: $ingredientName})-[:USE]-(r:Recipe) RETURN r LIMIT $nb
```

On retourne les recettes les plus appréciées par l'ensemble des utilisateurs.

```
1 MATCH (:User)-[l:LIKED]-(r:Recipe)
2 RETURN r, count(l.rank) AS vote, avg(l.rank) AS avg
3 ORDER BY avg DESC LIMIT $topSize
```

On retourne les recettes les plus appréciées par l'utilisateur.

```
1 MATCH (:User{id: $userId})-[l:LIKED]-(r:Recipe)
2 RETURN r,l ORDER BY l.rank DESC LIMIT $nb
```

On demande à Neo4j de nous retourner un utilisateur selon son id, on va utiliser cet utilisateur pour tout ce qui touche au régime végétarien par exemple.

```
$ MATCH (u:User{ id: $userId }) RETURN u
```

La requête compte le nombre d'ingrédients en commun entre chaque recette non notée et les recettes bien notées. Cette requête est utilisée afin de recommander une liste de recette à un utilisateur.

```
1 MATCH (:User{id: $userId})-[l:LIKED]→(rl:Recipe)-[:USE]→(i:Ingredient)
2 MATCH (r:Recipe)-[:USE]→(i:Ingredient)
3 WHERE r < rl AND l.rank > 3
4 RETURN DISTINCT r.name AS recipe, count(rl) AS nbIngredientShared
5 ORDER BY nbIngredientShared DESC
6 LIMIT $nb
```

Requête à la bd MongoDB

Ici on voit la fonction qui cherche les recettes via une expression régulière. Elle est utilisée dans la requête en ligne.

```
getRecipes(search, isVege, nb) {  
  return this.collection.find({ 'name': new RegExp(search) }).limit(nb).toArray();  
}
```

La fonction suivante permet de chercher les recettes qui entre dans un certain laps de temps.

```
getRecipeByDuration( min = 0, max, nb) {  
  return this.collection.find().sort({duration:1}).toArray().then((result) => {  
    return result.filter((it) => (it.duration >= min && ( max == null || it.duration <= max)));  
  });  
}
```

Le client

Pour le client, nous avons utilisé un bot Telegram que nous avons nommée « duchemin_bot » et que nous invoquons via « @duchemin_bot » pour faire appel à lui dans une discussion. Ce bot est ensuite utilisable avec une liste de commandes.

Commandes

Il est possible d'interagir avec le bot via les commandes suivantes :

- **/help** : Récapitule les commandes disponibles et comment les utiliser
- **/start** : Indique si le bot est lancé
- **/searchByIngredient** : Permet de rechercher une recette en fonction d'ingrédients
- **/searchRecipeShot** : Réalise une recherche sur tous les plats demandant un temps de préparation court (<=20 min)
- **/searchRecipeMedium** : Réalise une recherche sur tous les plats demandant un temps de préparation intermédiaire (21 à 60min)
- **/searchRecipeLong** : Réalise une recherche sur tous les plats demandant un temps de préparation long (>60 min)
- **/searchLikedRecipe** : Recherche les plats que vous avez aimés.
- **/toggleVegan** : Passe à true ou à false si vous voulez filtrer les recettes non végétariennes.
- **/recommendercipes** : Le bot va vous recommander une recette en fonction des recettes que vous avez aimées.

Sous chaque recette, il est également possible de la noter de 1 à 5 avec des étoiles.