“Temps d’Or” : un site de Larnak inc.

Mise en situation:

L’entreprise Larnak inc. vous contacte pour un contrat juteux. Son ambition est de créer un site rivalisant avec AirBnB pour les particuliers en Belgique. Pour ce site, elle a besoin d’un back end robuste et structuré. Celui-ci permettra à ses utilisateurs de proposer des chambres à louer ou/et de louer des chambres d’autres utilisateurs.

Votre job sera de fournir à Larnak inc. une API Rest sur base de leur cahier des charges à l’aide de vos connaissances en gestion de bases de données, en Java et en le framework Spring.

Cahier des charges:

Pour son API de “Temps d’or”, l’entreprise vous donne les contraintes suivantes :

- La base de données :

Pour la base de données, Larnak inc. a pris les devant pour l’analyse et vous fournit un schéma Entité-Relation (annexe 1).

Comme vous pouvez le voir sur ce schéma, les utilisateurs auront la possibilité de proposer des chambres à louer et pourront indiquer les différentes activités à proximité de celle-ci. Les utilisateurs pourront aussi louer des chambres louées par d’autres utilisateurs.

Les nom des activités et les username des utilisateurs devront tous deux être unique. La description des activités sera nullable, toutes autres informations sont requises et donc non nullables.

- L’API :

L’API sera protégé par une sécurité de type basic auth dont les credentials pourront se trouver en mémoire de l’application. Toutes les requêtes nécessiteront une identification afin d’obtenir une ressource avec succès. Toute sauf:

- une requête de login
- une requête de logout (voir *https://stackoverflow.com/questions/46003881/how-to-log-out-with-spring-security-basic-auth*)

Pour son back end, Larnak inc. demande à ce que les requêtes suivantes puissent être réalisées:

Pour les Utilisateurs:

On entend pour "*informations pertinentes des utilisateurs*" les informations suivantes:

id, username, nom, prénom, email, la liste des id de ses réservations et la liste des id des chambres qu'il gère.

- Récupérer tous les utilisateurs :

Cette requête permettra de récupérer un JSON reprenant les informations les plus pertinentes concernant les utilisateurs.

- Récupérer un utilisateur :

A partir de son id, il devra être possible de récupérer les infos pertinentes concernant l'utilisateur concerné.

- Ajouter un utilisateur :

Grâce aux informations capitales pour garantir l'insertion de l'utilisateur en DB, cette requête permettra d'y inscrire un nouveau membre. Faire en sorte que le password ne soit pas écrit en clair dans la DB/le body de la requête serait un plus.

- Supprimer un utilisateur :

A partir de son id, cette requête permettra de supprimer l'utilisateur concerné.

- Modifier un utilisateur :

Il devrait être possible de modifier l'adresse email et le password d'un utilisateur. Faire en sorte que le password ne soit pas écrit en clair dans la DB/le body de la requête serait un plus.

Pour les chambres à louer:

On entend pour "*informations pertinentes des chambres*" les informations suivantes:

id, adresse, ville, nombres de places, id du gérant, nombre de réservations

- Récupérer les infos de toutes le chambres:

Cette requête permettra de récupérer les infos pertinentes sur toutes les chambres.

- Récupérer les infos d'une chambre:

Cette requête permettra de récupérer les infos pertinentes d'une chambre identifiée par son id.

- Récupérer les chambres ayant une certaines activité à proximité:

Cette requête permettra de récupérer les infos pertinentes sur toutes les chambres ayant une certaine activité à proximité

- Ajouter une chambre:

Cette requête permettra d'ajouter une chambre gérée par un utilisateur en fournissant en plus des infos pertinentes, les différentes activités à proximité.

- Supprimer une chambre:

Cette requête permettra de supprimer une chambre identifiée via un id.

- Vérifier la disponibilité d'une chambre pour une tranche de date :

Cette requête permettra d'indiquer s'il est possible de faire une réservation pour une chambre identifiée par son id selon un date de début et de fin de séjour.

- Changer le gérant d'une chambre:

Cette requête permettra de changer le gérant d'une chambre pour un autre utilisateur.

Pour les activités :

- Récupérer toutes les activités :

Récupérer toutes les infos sur toutes les activités.

- Ajouter une activité :

Permet d'ajouter une activité en lui donnant un nom et possiblement une description.

- Supprimer une activité :

Permet de supprimer une activité identifiée par un id.

Les contraintes sur les données entrantes :

Les données entrantes devront être soumises à une validation.

- Pour les utilisateurs :

- username: non vide et avec une taille comprise entre 4 et 20 caractères
- nom : nom vide et de taille inférieure ou égale à 50 caractères
- prenom : nom vide et de taille inférieure ou égale à 50 caractères
- email : non vide et doit ressembler à un email
- password : non vide et de taille supérieure ou égale à 6 caractères

- Pour les chambres :

- place : au minimum une et requis
- adresse : non null et de taille entre 10 et 100 caractères
- ville : non null et de taille entre 2 et 50 caractères

- Pour les réservation :

- nombre de personne: inférieure au nombre de place disponible pour la chambre
- date d'arrivé : dans le futur
- date de départ : après la date d'arrivée

- Pour les activités :

- nom : non vide et de taille inférieure ou égale à 20 caractères
- description : de taille inférieure ou égale à 255 caractères