

Les questions 3 et 4

Les données collectées peuvent être enregistrées dans un schéma flexible qui permet aux objets d'être stockés de deux manières différentes dans une base de données NoSQL comme RavenDB.

Collection de véhicules

- Champ : "_id" (numéro d'identification unique de la voiture, créé automatiquement ou à l'aide d'une clé spéciale)
- Champs : "name" (nom du véhicule) et "image" (URL de l'image de la voiture).
- Champ "Characteristics" (objet fournissant des détails spécifiques aux voitures)
- Champ "A_credit" (objet fournissant des détails spécifiques aux voitures) : "A_credit" (objet contenant des informations sur le financement automobile)
- Champ : "En_location" (objet contenant des détails sur l'emplacement de la voiture) "En_location" (objet contenant des détails sur la location de la voiture)

En créant un nouveau document pour chaque objet automobile extrait des différents sites, nous pouvons utiliser notre API pour importer les données dans la base de données. Par exemple, nous pouvons produire un document JSON contenant les éléments de schéma pertinents pour chaque automobile et l'insérer dans la collection "Cars" de la base de données.

En utilisant les bibliothèques et les outils offerts par RavenDB, nous pouvons mener des activités CRUD (créer, lire, mettre à jour, supprimer) sur la base de données RavenDB afin de fournir les données par l'intermédiaire de notre API. Afin d'obtenir les données et de les fournir aux clients, nous pouvons configurer des points d'extrémité d'API pour se connecter à la base de données et effectuer les activités nécessaires.

Nous pouvons, par exemple, développer des points de terminaison tels que :

- GET /cars : renvoie une liste de tous les véhicules de la collection "Cars".
- GET /cars/nom : Renvoie une voiture spécifique après l'avoir récupérée à l'aide de son nom.
- POST /cars : Insère un nouveau document avec les données spécifiées dans la collection "Cars" pour créer une nouvelle voiture.
- PUT /cars/id : Met à jour une voiture existante avec les nouvelles informations fournies et l'identification unique de la voiture.
- DELETE /cars/{id} : Supprime une voiture spécifique à l'aide de son identifiant unique.

Dans notre cas, nous créons simplement des fonctions GET et utilisons les données stockées dans un fichier JSON.