GUIDE DU DEVELOPPEUR

Créer des applications web de classe mondiale en utilisant Node.js et Couchbase

Eric Bishard,
porte-parole des développeurs

Couchbase

Sommaire

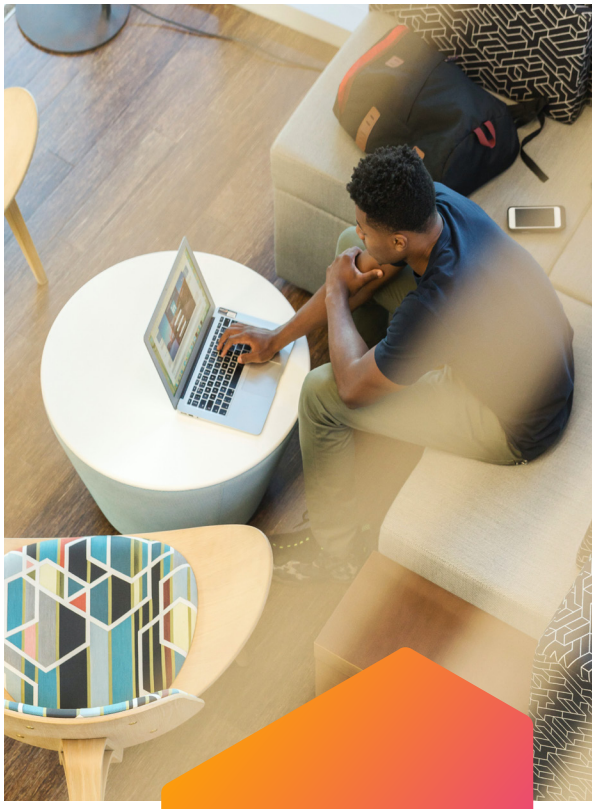
Introduction

En tant que développeur Node.js, la dernière chose dont vous voulez avoir à vous soucier est la base de données de votre application. Après tout, il y a des questions plus importantes qui demandent votre temps et votre attention.

Heureusement, les bases de données NoSQL telles que Couchbase sont naturellement adaptées aux applications JavaScript en raison de leur utilisation native des documents JSON. Avec JavaScript, du début à la fin, le développement est intégré, ce qui vous laisse libre de relever des défis plus intéressants.

Si vous cherchez à construire une application web de classe mondiale en utilisant Node.js, Couchbase est la solution idéale pour votre base de données. L'écosystème Couchbase offre un ensemble d'outils de premier ordre et l'IDE idéal pour les développeurs Node.js,de sorte que la seule limite à ce que vous pouvez construire est votre imagination.

Dans ce guide du développeur, nous allons examiner de plus près comment développer des applications web performantes à l'aide de Node.js et de Couchbase. Commençons.



Pourquoi Couchbase ?

Couchbase est une base de données distribuée de documents JSON avec toutes les capacités d'un magasin de valeurs clés, d'un moteur de recherche et d'un RDBMS, comprenant du SQL, des transactions ACID distribuées, et plus encore.

D'un côté, Couchbase est conçue pour les microservices et l'informatique sans serveur basée sur la consommation dans le cloud ; de l'autre côté, elle est conçue pour l'informatique périphérique sur des appareils mobiles/IoT connectés occasionnellement et localement. En résumé : Couchbase est parfait pour les déploiements multiclouds.

Parce que Couchbase gère les documents JSON, elle élimine le besoin d'un schéma hard-coded dans la base de données, bien que Couchbase inclue une structure logique pour rendre l'organisation de ces documents aussi intuitive que les schémas de la vieille école. La définition de l'objet de l'application, disponible en JSON, est le schéma éditable que vous pouvez contrôler en tant que développeur.
Vous n'écrivez qu'une seule fois le JSON dans la base de données et vous pouvez ensuite lui appliquer de multiples capacités de traitement des données, notamment :

- Programmabilité SQL complète, y compris les transactions ACID
- Vitesse de traitement de la mise en cache en mémoire
- Capacités de stockage clé-valeur
- Recherche en texte intégral (recherche d'informations)
- Analyse des données (interrogation ad hoc)
- Programmation pilotée par les événements (réactive) et saisie des données de changement

Ainsi, Couchbase sert de système d'enregistrement fiable, tout en gérant simultanément des opérations à valeur-clé d'une latence de l'ordre de la microseconde, des requêtes SQL et des recherches de texte en quelques millisecondes, ainsi que des requêtes analytiques ad-hoc s'étendant sur des dizaines de secondes, sans qu'aucune requête ne fasse obstacle à une autre.

Les bases de données traditionnelles n'offrent tout simplement pas ces possibilités.

Lorsque vous utilisez Couchbase, vous réduisez la prolifération des données et des bases de données, vous améliorez la sécurité, vous diminuez l'administration, et vous réduisez le coût global. Mais surtout, Couchbase vous permet de créer des applications rapidement et de les déployer à grande échelle.

Nouveau sur Couchbase ? Commencez par consulter notre documentation pour les développeurs
Ou bien, essayez Couchbase Cloud gratuitement en vous inscrivant aujourd'hui.

Utiliser Node.js avec Couchbase

Il existe trois façons principales d'utiliser Node.js avec Couchbase Server :

- Le SDK Node.js
- Le modélisateur de données objet (ODM) du Node Ottoman
- La RAGE Stack avec Couchbase

Examinons de plus près chacune d'entre elles.

Node.js SDK

Le SDK Couchbase Node.js est le choix par défaut des développeurs JavaScript qui créent des applications avec Node.js.

La version 3.x du SDK est une réécriture complète de l'API 2.x, offrant une surface plus simple et ajoutant la prise en charge des futures fonctionnalités de Couchbase Server comme les Collections et les Scopes.

Le SDK 3.x promet également de réduire la complexité du JavaScript asynchrone dans les applications clientes, d'étendre les API de gestion et d'offrir au développeur de meilleures options de débogage et de journalisation.

Pour plus d'informations sur la façon d'utiliser le SDK Node.js, lisez Installer et commencer à utiliser le SDK Node.js avec Couchbase Server ou visitez notre portail des développeurs Node.js.

Node Ottoman ODM

Ottoman est un modélisateur de données d'objets (ODM) pour le SDK Node.js de Couchbase qui fournit un schéma JSON et une validation pour NoSQL. Il est conçu pour éliminer la plupart des codes passe-partout nécessaires à la création d'applications Node.js avec Couchbase, afin que vous puissiez créer des systèmes faciles à concevoir, à maintenir et à faire évoluer.

Avec Ottoman, vous déclarez le schéma dans votre code. Bien que Couchbase n'impose pas de schéma pour vos documents, la plupart des applications ont besoin d'un autour de la création certain niveau de schéma, même en NoSQL. Bien qu'Ottoman crée une abstraction par rapport au SDK de Couchbase, les avantages l'emportent sur les inconvénients. Un développeur crée beaucoup de logique et de la mise à jour des documents, de l'écriture avant et après le cycle de vie, du travail avec les structures de données et de la validation.

Pour plus d'informations sur l'utilisation de l'ODM Ottoman avec Couchbase, lisez notre guide de démarrage rapide pour Ottoman destiné aux développeurs ou faites un tour d'horizon d'Ottoman dans cet article de blog.

RAGE Stack avec Couchbase

Vous avez également la possibilité d'utiliser la RAGE stack (React, Apollo Client, GraphQL et Express) aux côtés de Couchbase Server pour développer des applications JavaScript complètes.

Vous pouvez utiliser React, Apollo et GraphQL du côté client, et Express et Express GraphQL du côté serveur. L'application Express utilise également le SDK Couchbase Node.js, qui vous permet de vous connecter à votre base de données Couchbase, d'effectuer des requêtes et de renvoyer des données à l'API.

Pour une démonstration complète du développement full-stack avec RAGE et Couchbase, regardez cette présentation Couchbase Connect à la demande (avec le repo de code qui l'accompagne).

Exemples de projets utilisant Couchbase et Node.js

Maintenant que vous savez comment utiliser Node.js et Couchbase ensemble, il est temps de vous plonger dans des exemples de cas d'utilisation et de projets.
Le cas le plus populaire d'utilisation de Couchbase et Node.js pour les profils d'utilisateurs et les magasins de session.

Profil de l'utilisateur et stockage des sessions

L'un des cas d'utilisation les plus courants d'une base de données NoSQL est la création d'un profil utilisateur et d'une banque de sessions. Pourquoi NoSQL spécifiquement ? Parce que les profils doivent souvent être flexibles et accepter des changements de données fréquents. Si les modifications sont possibles avec un RDBMS traditionnel, les changements réguliers avec une base de données relationnelle nécessitent plus de travail et imposent des pénalités de performance plus importantes.

Alors, comment créer un profil d'utilisateur en utilisant Node.js et Couchbase Server ? Voici un aperçu de haut niveau. Pour une présentation détaillée, veuillez consulter cette vidéo : Développer un profil d'utilisateur et une banque de session avec Node.js.

Lorsqu'il s'agit de gérer les données des utilisateurs, vous devez pouvoir créer une banque de profils d'utilisateur et une banque de sessions, puis leur associer d'autres documents JSON. Voici quelques principes importants à retenir lors de la création d'une banque de profils d'utilisateurs :

- Stockez les données du compte, comme les noms d'utilisateur et les mots de passe, dans un document de profil.
- Transmettre les données sensibles de l'utilisateur avec chaque demande d'action de l'utilisateur.
- Utilisez une session qui expire après une durée déterminée.
- Stocker les documents de session avec une limite d'expiration.

Article de blog vidéo : Développer une banque de profils d'utilisateurs et une banque de sessions avec Node.js - Vidéo
Ancien article de blog : Créer une banque de profils d'utilisateurs avec Node.js et une base de données NoSQL

ÉTAPE 1 : CREER L'API AVEC NODE ET EXPRESS

Tout d'abord, créez un répertoire de projet pour votre application Node.js et installez toutes les dépendances.

Cela crée un répertoire de travail pour votre projet et initialise un nouveau projet Node. Vos dépendances incluent le SDK Node.js pour Couchbase et Express Framework et d'autres bibliothèques utilitaires comme body-parser pour accepter les données JSON via les requêtes POST, uuid pour générer des clés uniques et bcryptjs pour hacher vos mots de passe afin de dissuader les utilisateurs malveillants.

Ensuite, démarrez votre application avec un fichier server.js. Vous devez également créer un index dans Couchbase Server car vous utiliserez le langage de requête N1QL pour l'un de vos points de terminaison. Les index primaires ne sont pas recommandés pour le code de production.

ÉTAPE 2 : ENREGISTREMENT D'UN NOUVEL UTILISATEUR DANS LE MAGASIN DE PROFILS

Un profil utilisateur contient toutes les informations décrivant un utilisateur, telles que son adresse, son numéro de téléphone, ses profils de médias sociaux, etc. Ce n'est jamais une bonne idée de stocker les informations d'identification d'un compte dans le même document que les informations de base du profil d'un utilisateur. Vous aurez besoin d'un minimum de deux documents pour chaque utilisateur, alors regardons comment vous pouvez structurer ces documents.

Votre document de profil utilisateur doit comporter une clé que vous utiliserez pour vous référer à des documents connexes. Le plus souvent, cette clé est un UUID généré automatiquement.

Votre document de profil aura une valeur JSON qui comprendra deux propriétés : l'adresse électronique et une propriété de type. La propriété type est un indicateur important qui décrit notre document de manière similaire à la façon dont une table organise les enregistrements dans une base de données relationnelle. Il s'agit d'une convention standard dans une base de données de documents.

Le document de compte associé à votre profil d'utilisateur aura une clé qui est égale à l'adresse électronique d'un utilisateur donné. Votre document de compte devrait également avoir un type, ainsi qu'un pid se référant à la clé de votre document de profil ainsi qu'une adresse électronique et un mot de passe haché.

Vous disposez maintenant d'un modèle établi pour chaque document et d'une stratégie pour relier ces documents sans avoir créé de contraintes de base de données.

ÉTAPE 3 : UN POINT D'ACCES POUR LA CREATION DE COMPTES

L'étape suivante consiste à créer un point de terminaison pour la création de compte dans votre fichier server.js. Assurez-vous que tous les éléments suivants sont inclus.

Tout d'abord, vérifiez que la demande comporte à la fois une adresse électronique et un mot de passe.

Ensuite, votre point de terminaison doit créer un objet de compte et un objet de profil sur la base des données envoyées dans la demande. Le pid que vous enregistrez dans l'objet de compte doit être une clé unique et doit être défini comme la clé de document pour votre objet de profil.

Le document du compte doit utiliser l'adresse électronique comme clé. À l'avenir, si d'autres détails du compte sont nécessaires (comme une autre adresse électronique, une connexion sociale, etc.), vous pourrez associer d'autres documents au même profil.

Plutôt que de sauvegarder le mot de passe dans l'objet de compte en texte brut, vous devriez le hacher en utilisant Bcrypt. Le mot de passe doit être supprimé de l'objet profil pour des raisons de sécurité. (Pour plus d'informations sur le hachage du mot de passe, consultez ce tutoriel).

Les données étant prêtes, il est temps de les insérer dans Couchbase. L'objectif de cette sauvegarde doit être tout ou rien. Vous voulez que les documents relatifs au compte et au profil soient tous deux créés avec succès, sinon l'ensemble de la sauvegarde doit être annulée. En fonction du succès de la sauvegarde, vous retournerez certaines informations au client.
Vous auriez également pu utiliser des requêtes N1QL pour insérer les données, mais il est plus facile d'utiliser des opérations CRUD sans perte de performance.

ÉTAPE 4 : UTILISATION DE JETONS DE SESSION POUR LES DONNEES SENSIBLES

Une fois le profil et le compte de l'utilisateur créés, vous voulez que l'utilisateur se connecte et établisse une session qui sera stockée dans la base de données en faisant référence à son profil d'utilisateur. Ce document de session doit être configuré pour expirer et être supprimé de la base de données.

Le modèle de données de session, comme les autres documents, devrait avoir un type différent. Comme pour le document de compte, il doit avoir une propriété pid qui fait référence au profil de l'utilisateur cible.

Le code qui rend cela possible doit se trouver dans votre point de connexion.

Ce point de terminaison doit valider les données entrantes, puis exécuter une recherche de compte par adresse électronique. Si des données reviennent pour l'adresse électronique, vous pouvez comparer le mot de passe entrant avec le mot de passe haché renvoyé par la recherche de compte. Si cela réussit, votre point final crée une nouvelle session pour l'utilisateur.

Contrairement à l'opération d'insertion précédente, vous devez définir une expiration du document d'une heure (3600 s). Si l'expiration n'est pas actualisée, le document doit disparaître. C'est une bonne chose car cela oblige l'utilisateur à se reconnecter et à créer une nouvelle session. Ce jeton de session est ensuite transmis à chaque demande future à la place du mot de passe.

ÉTAPE 5 : GESTION D'UNE SESSION UTILISATEUR AVEC DES JETONS

Éventuellement, vous voudrez obtenir des informations sur le profil d'un utilisateur et associer de nouvelles informations à ce profil. Pour cela, vous devez confirmer l'autorité par le biais de la session.

Vous pouvez confirmer la validité de la session à l'aide d'un middleware. Une fonction Middleware peut être ajoutée à n'importe quel endpoint Express. Cette validation doit être une fonction simple qui a accès à la requête HTTP de votre endpoint.

Votre code doit vérifier la présence d'un en-tête d'autorisation dans la requête. Si vous avez un jeton porteur valide avec un identifiant de session (sid), vous pouvez effectuer une recherche. Votre document de session doit contenir l'identifiant de profil. Si la recherche de session est réussie, enregistrez l'identifiant de profil (pid) dans la demande.

Ensuite, vous devrez rafraîchir l'expiration de la session et passer par le middleware pour revenir au point de terminaison. Si la session n'existe pas, aucun identifiant de profil ne sera transmis et la demande échouera.

Après cela, vous pouvez utiliser votre middleware pour obtenir des informations sur le profil dans votre endpoint de compte. Remarque : la validation doit avoir lieu avant le reste de la demande.

REFLEXIONS FINALES

Ce sont les étapes de haut niveau pour créer un profil d'utilisateur et un magasin de session en utilisant Node.js et Couchbase.

Là encore, pour une présentation détaillée, veuillez consulter cette vidéo : Développer un profil utilisateur et une banque de session avec Node.js. Et pour une lecture plus avancée, veuillez consulter cet article : User Profile Store : Advanced Data Modeling Part 1. Tout le code mentionné dans les articles ci-dessus peut être trouvé dans le repo couchbaselabs/couchbase-nodejs-blog-api sur GitHub.

Autres exemples de projets

Bien que les magasins de profils d'utilisateurs et de sessions soient le cas d'utilisation le plus populaire, Couchbase et Node.js peuvent également être utilisés pour une variété d'autres applications et projets, notamment :

- Fonctionnalité de recherche en texte intégral (autonome ou faisant partie d'un projet plus vaste)
- Applications Chatbot
- Points d'intérêt et applications de voyage
- Applications Bitcoin et autres crypto-monnaies
- Analyse d'ensembles massifs de données

Conclusion

Vous ne devriez pas avoir à réfléchir à deux fois quant au choix de la base de données à utiliser pour votre prochaine application Node.js. En tant que base de données documentaire NoSQL, Couchbase est parfaitement adaptée.

Dans ce guide du développeur, vous avez appris :

- Pourquoi Couchbase est un choix naturel quand on construit avec Node.js
- Comment utiliser Node.js avec Couchbase, y compris le SDK, l'ODM Ottoman et la RAGE stack.
- Quels projets communs utilisent à la fois Node.js et Couchbase pour relever les défis actuels en matière de développement ?

Pour plus d'informations sur la manière d'utiliser Node.js avec Couchbase, consultez notre portail pour les développeurs Node.js.

Désormais, lorsqu'il s'agit de créer une nouvelle application web, vos possibilités sont infinies. La seule question qui se pose est la suivante : que construirez-vous ?

Couchbase

About Couchbase

At Couchbase, we believe data is at the heart of the enterprise. We empower developers and architects to build, deploy, and run their mission-critical applications. Couchbase delivers a high-performance, flexible and scalable modern database that runs across the data center and any cloud. Many of the world's largest enterprises rely on Couchbase to power the core applications their businesses depend on.

For more information, visit www.couchbase.com.