

INF4375 - Paradigmes des échanges Internet

MongoDB

Jacques Berger

Objectifs

Introduire les bases de données «documents»

Présenter MongoDB

Prérequis

Aucun

NoSQL

Terme fréquemment utilisé depuis 2009

Désigne une caractéristique de certains systèmes de bases de données

NoSQL

Le terme NoSQL peut s'appliquer à tous types de base de données qui n'est pas relationnelle

C'est donc en opposition aux SGBDR, qui sont dominants sur le marché

NoSQL

Les bases de données NoSQL sont souvent des bases de données de type «document», c'est-à-dire qu'on y stocke des documents entiers

Par exemple, des documents XML, des documents JSON, etc.

MongoDB

MongoDB est une base de données NoSQL très flexible et très légère

Elle n'a pas de notion d'intégrité référentielle, contrairement aux SGBDR

- Pas de structure de tables

- Pas de clés étrangères

- Pas de contraintes d'unicité

- Pas de gestion de transactions

MongoDB

Optimisée pour stocker des ensembles de clé-valeur

Elle stocke des documents JSON

Version actuelle : 3.4

MongoDB

MongoDB est donc plus flexible et plus souple
mais permet moins d'assurer l'intégrité
référentielle des données

Elle est plus rapide qu'un SGBDR, mais rien
n'assure la cohérence des données

NoSQL

Les bases de données NoSQL sont de plus en plus utilisées lorsqu'on doit manipuler de grands volumes de données et dans les environnements en temps réel

Utilisation

MongoDB est conçue pour être distribuée

On communique avec MongoDB au travers de son interface http

Utilisation

Pour utiliser MongoDB, il faut démarrer un service au préalable

```
mongod
```

Ensuite, on peut utiliser n'importe quel client `mongodb` pour interagir avec

Concepts

Base de données

MongoDB permet d'avoir plusieurs bases de données sur le même serveur

Concepts

Collection

On a plusieurs collections dans une base de données

On stocke les données dans les collections

Concepts

Les collections sont l'équivalent des tables dans un SGBDR

Les collections n'imposent pas de structure pour ses enregistrements, contrairement à une table d'un SGBDR

Concepts

Tous les objets dans une collection pourraient être différents, mais on tente de conserver une structure similaire pour l'ensemble des objets d'une collection

Concepts

Objet

Un objet est un enregistrement dans une collection

C'est l'équivalent d'une rangée dans un SGBDR

C'est un objet Javascript

Clés

Chaque objet dans une collection possède un champ `_id` servant de clé primaire

Cette clé est composée de 24 caractères hexadécimaux

Elle est gérée par mongodb

Clés

La clé est typée avec une fonction ObjectId

On se sert de cette clé pour les opérations de modification, de suppression et de recherche

C'est un principe similaire à une clé primaire auto-incrémentale d'un SGBDR

Console

L'exécutable mongo permet d'ouvrir une console et de faire des requêtes et des opérations sur les données

Très utile pour naviguer dans les données et expérimenter

Console

On choisit la base de données avec la commande
use

```
use inf4375
```

Console

La commande show permet d'afficher quelques données utiles

Pour afficher la liste des bases de données :

```
show dbs
```

Console

Lorsqu'on a choisit une base de données, la commande `show` nous permet également d'avoir la liste des collections

```
show collections
```

Opérations

La plupart des opérations s'effectuent directement sur la collection

Le format d'une requête calque un appel de fonction

```
db.collection.operation(paramètres)
```


Insertion

La méthode `insert` permet d'ajouter un objet ou un ensemble d'objets

```
db.disco.insert(parametre)
```

Le paramètre peut être un objet ou un array de plusieurs objets

Insertion

Si la collection n'existe pas, elle sera créée automatiquement

La clé primaire, si elle n'est pas définie dans l'objet, sera générée automatiquement

```
db.disco.insert({artist:"Iced Earth", title:"The Dark Saga", year:1996})
```

Requêtes

La méthode `find` permet de faire des recherches dans les objets de la collection

```
db.disco.find()
```

La méthode `find` sans paramètre retourne un curseur permettant l'itération sur tous les objets de la collection

Requêtes

Il est possible de faire des requêtes aussi sélectives qu'avec un SGBDR en passant un objet requête en paramètre

```
db.disco.find({year: 2007})
```

Requêtes

La recherche supporte également les expressions régulières

```
db.disco.find({artist: new RegExp("maid", "i")})
```

Requêtes

On peut rechercher en utilisant plusieurs types d'opérateurs, comme \$gt

```
db.disco.find({year: {$gt: 2000}})
```

Requêtes

On peut également rechercher en spécifiant plusieurs critères et il existe beaucoup d'opérateurs pour faire des recherches avancées

Consultez la documentation

<http://docs.mongodb.org/manual/reference/operator/>

Requêtes

Toutes les requêtes vues à date retournent l'objet au complet

Dans certains cas, nous ne voulons que certains champs de l'objet

Requêtes

En passant un deuxième objet en paramètre à la méthode `find`, on peut lui spécifier les champs que l'on veut avoir, c'est l'équivalent d'une projection avec SQL

`_id` est toujours retourné, sauf si explicitement caché

```
db.disco.find({year: 2007}, {title: true})
```

Requêtes

Si le champ `_id` est un `ObjectId`, il faut utiliser la fonction `ObjectId` pour l'utiliser dans une requête

```
db.disco.find({_id: ObjectId("51c91ef2fdcab488f34af4f9")})
```

Suppression

La méthode `remove` permet de supprimer les éléments d'une collection

Utilisée sans paramètre, `remove` supprime tous les objets de la collection

```
db.disco.remove()
```

Suppression

Le premier paramètre de `remove` est un objet requête permettant de choisir les critères à appliquer pour déterminer les objets qui seront détruits

L'objet requête est le même que pour la méthode `find`

```
db.disco.remove({artist: "Iron Maiden", year: {$lt: 1990}})
```

Suppression

Le deuxième paramètre est un booléen qu'on doit mettre à true si on ne désire que supprimer un seul objet

Ce paramètre est à false par défaut

```
db.disco.remove({year: {$gt: 2010}}, true)
```

Modification

La méthode update permet de modifier l'état d'un objet

Elle prend 3 paramètres

Consultez la référence

<http://docs.mongodb.org/manual/reference/method/db.collection.update/>

Modification

Le premier paramètre est le critère de sélection, l'objet requête, identique à la méthode find

Le deuxième paramètre est un objet décrivant les modifications à apporter, ce paramètre utilise des opérateurs particuliers (référence)

Modification

Les opérateurs les plus courants sont `$set` et `$addToSet`

`$set` permet de modifier la valeur d'une propriété de l'objet

`$addToSet` permet d'ajouter un élément dans un array

Modification

```
// Ajoute une propriété à l'album
db.disco.update({title: "No Prayer For The Dying"}, {$set:
{favorite: true}})

// Change le titre d'un album
db.disco.update({_id: ObjectId("51c91e26fdcab488f34af4f6")},
{$set: {title: "Belive"}})
```

Modification

Par défaut, la modification ne sera appliquée qu'à un seul objet, pour modifier plusieurs objets en même temps, il faut utiliser l'option multi

C'est le dernier paramètre

```
db.disco.update({}, {$set: {good_music: true}}, {multi: true})
```

```
db.disco.update({artist: "Air Supply"}, {$set: {good_music: false}}, {multi: true})
```

Supprimer une collection

On peut supprimer une collection complète avec la méthode drop

L'équivalent d'un DROP TABLE en SQL

```
db.disco.drop()
```

Documentation

Allez consulter la documentation

Beaucoup de fonctionnalités sont disponibles

mongodump

L'exécutable mongodump permet de produire un backup des bases de données du serveur

Le backup sera placé dans un répertoire dump

mongorestore

L'exécutable mongorestore permet de remettre les données d'un backup fait avec mongodump

Attention : si les collections existent déjà sur le serveur, les collections du backup sont ignorées

L'idéal est de vider la BD avant un mongorestore

Vider la BD

Il existe une fonction pour supprimer une base de données du serveur

```
db.dropDatabase();
```

Plus loin...

MongoDB

<http://www.mongodb.org/>