

TP7. à la découverte des bases NoSQL

Organisation

Les objectifs de ce travail pratique sont les suivants :

- Comprendre les bénéfices et les limites des bases nosql
- Comprendre les différents types de bases nosql
- Réaliser une application en utilisant une API comme Morphia proche de JPA en se plaçant dans un cadre classique de développement au dessus d'une base orientée document comme Mongo.
- Comparer avec une base de données *clé valeur* haute performance comme Redis

Sujet

Création d'une application simple qui utilise une base de données MongoDB

Organisation.

Vous trouverez dans ce /share/m1miage/SIR/nosql :

- Un moteur de base de données mongodb.

Étape 1. Installation de la base de données MongoDB.

from [<http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Synopsis

This tutorial provides a method for installing and running the MongoDB server (i.e. "**mongod.exe**") on the Microsoft Windows platform through the *Command Prompt*. Operating MongoDB with Windows is similar to MongoDB on other platforms. Most components share the same operational patterns.

Procédure

Unzip mongoDB in w:\mongodb

Set up the Environment

Start the *Command Prompt* by selecting the *Start Menu*, then *All Programs*, then *Accessories*, then right click *Command Prompt*, and select *Run as Administrator* from the popup menu. In the *Command Prompt*, issue the following commands:

```
cd \  
move w:\mongodb-win32-* w:\mongodb
```

Note

MongoDB is self-contained and does not have any other system dependencies. You can run MongoDB from any folder you choose. You may install MongoDB in any directory (e.g. `w:\test\mongodb`)

MongoDB requires a *data folder* to store its files. The default location for the MongoDB data directory is `w:\data\db`. Create this folder using the *Command Prompt*. Issue the following command sequence:

```
md w:\mongodb\data  
md w:\mongodb\data\db
```

Note: You may specify an alternate path for `\data\db` with the `dbpath` setting for `mongod.exe`, as in the following example:

```
w:\mongodb\bin\mongod.exe --dbpath w:\mongodb\data\db
```

Start MongoDB

To start MongoDB, execute from the *Command Prompt*:

```
w:\mongodb\bin\mongod.exe --dbpath w:\mongodb\data\db
```

This will start the main MongoDB database process. The **waiting for connections** message in the console output indicates that the `mongod.exe` process is running successfully.

Note

Depending on the security level of your system, Windows will issue a *Security Alert* dialog box about blocking “some features” of `C:\mongodb\bin\mongod.exe` from communicating on networks. All users should select **Private Networks, such as my home or work network** and click **Allow access**. For additional information on security and MongoDB, please read the *Security and Authentication* wiki page.

Warning

Do not allow `mongod.exe` to be accessible to public networks without running in “Secure Mode” (i.e. **auth.**) MongoDB is designed to be run in “trusted environments” and the

database does not enable authentication or “Secure Mode” by default.

Connect to MongoDB using the `mongo.exe` shell. Open another *Command Prompt* and issue the following command:

```
w:\mongodb\bin\mongo.exe
```

Note:

Executing the command `start w:\mongodb\bin\mongo.exe` will automatically start the `mongo.exe` shell in a separate *Command Prompt* window.

The `mongo.exe` shell will connect to `mongod.exe` running on the localhost interface and port 27017 by default. At the `mongo.exe` prompt, issue the following two commands to insert a record in the `test` *collection* of the default `test` database and then retrieve that record:

```
> db.test.save( { a: 1 } )  
> db.test.find()
```

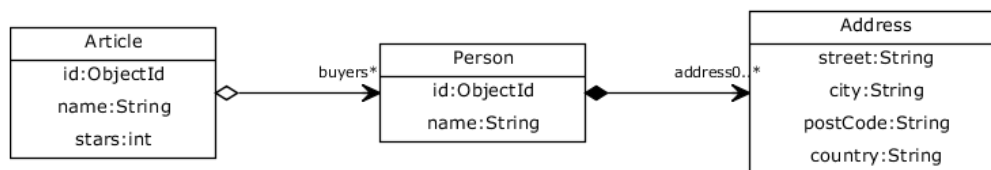
See also

“`mongo`” and “*mongo Shell JavaScript Quick Reference.*”

Votre base est prête.

Étape 2. Utilisation de morphia pour la connexion à la base de données.

Faites le schéma de données suivant à l’aide de classes Java



Modèle de données

Créez un projet Maven avec l’artefact `maven.archetype.quickstart`.

Ajoutez la dépendance suivante à votre projet.

```
<dependency>  
  <groupId>com.google.code.morphia</groupId>  
  <artifactId>morphia</artifactId>  
  <version>0.102</version>  
</dependency>  
<dependency>  
  <groupId>cglib</groupId>  
  <artifactId>cglib-nodep</artifactId>
```

```

    <version>[2.1_3,)</version>
    <type>jar</type>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>com.thoughtworks.proxytoys</groupId>
    <artifactId>proxytoys</artifactId>
    <version>1.0</version>
    <type>jar</type>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>2.11.3</version>
</dependency>

```

.....

à partir de la doc suivante.

<http://code.google.com/p/morphia/wiki/GettingStarted>

<https://github.com/mongodb/morphia>

annotez votre modèle de données pour le mapping en utilisant les annotations Morphia.

Créez des objets en base et testez des requêtes.

par exemple.

```

public static void main( String[] args ) throws UnknownHostException
{
    Morphia morphia = new Morphia();
    Mongo mongo = new Mongo();
    morphia.map(Person.class).map(Address.class);
    Datastore ds = morphia.createDatastore(mongo, "my_database");

    Person p = new Person();
    p.setName("Tintin");

    Address address = new Address();
    address.setStreet("123 Some street");
    address.setCity("Some city");
    address.setPostCode("123 456");
    address.setCountry("Some country");
    //set address
    p.setAddress(address);
}

```

```

// Save the POJO
ds.save(p);
for (Person e : ds.find(Person.class))
    System.err.println(e);
}

```

Exécutez votre programme.

Maintenant créez plusieurs objets Article, Person et Address, et stockez-les dans la base.

Récupérez vos objets et imprimez-les sur la console.

Questions: Quelles sont les limites d'une base de données orientées document ? (à répondre dans le compte rendu de TP (le readme))

Étape 3. A la découverte de Redis

[D'après wikipedia]

Redis (de l'anglais *REmote DIctionary Server* qui peut-être traduit par « serveur de dictionnaire distant » et jeu de mot avec *Redistribut*e¹) est un système de gestion de base de données clef-valeur scalable, très hautes performances, écrit avec le langage de programmation C ANSI et distribué sous licence BSD. Il fait partie de la mouvance NoSQL et vise à fournir les performances les plus élevées possibles.

Données manipulées

Redis permet de manipuler des types de données simples : chaînes de caractères, tableaux associatifs, listes, ensembles et ensembles ordonnés.

Stockage en RAM

Une des principales caractéristiques de Redis est de conserver l'intégralité des données en RAM. Cela permet d'obtenir d'excellentes performances en évitant les accès disques, particulièrement coûteux.

Lorsque la taille des données est trop importante pour tenir en mémoire, Redis peut également utiliser de la mémoire virtuelle.

Afin d'assurer la conservation des données en cas d'incident — la mémoire vive étant volatile — Redis offre la possibilité de « capturer » l'état de la base dans un fichier. Cette technique ne permettant pas de garantir la conservation des manipulations effectuées entre deux captures, il est également possible de conserver une trace de toutes ces manipulations. En cas d'incident, la base peut être restaurée en les ré-appliquant dans l'ordre.

Déploiement

Redis supporte la réplication via un modèle maître-esclave à des fins de résistance aux pannes et de répartition de la charge. Toutes les écritures doivent se faire via l'instance maîtresse, mais il est possible de faire des lectures sur les instances esclaves — si toutefois il n'y a pas de besoin critique d'accéder avec certitude à des données mises à jour.

Plus de documentation ici.

<http://www.barreverte.fr/une-courte-introduction-a-redis>

Lancez le serveur Redis.

redis-server.exe depuis une ligne de commande windows.

Créez un projet Maven avec l'artefact maven.archetype.quickstart.

Ajoutez la dépendance suivante à votre projet.

```
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
  <version>2.0.0</version>
  <type>jar</type>
  <scope>compile</scope>
</dependency>
```

Dans votre projet testez la connexion à Redis. Parcourez l'API pour regarder les types d'objets stockable et les types de requêtes disponibles.

Exemple 1.

```
public class App
{
public static void main( String[] args )
{
    Jedis jedis = new Jedis("localhost");
    jedis.set("foo", "bar");
    String value = jedis.get("foo");

    System.err.println(value);
}
}
```

Exemple 2.

```
public static void main(String[] args) {
    Jedis jedis = new Jedis("localhost");
    System.out.println(jedis.get("counter"));
    jedis.incr("counter");
    System.out.println(jedis.get("counter"));
}
```

Exemple 3.

```
public static void main(String[] args) throws InterruptedException {
    String cacheKey = "cachekey";
    Jedis jedis = new Jedis("localhost");
    // adding a new key
    jedis.set(cacheKey, "cached value");
    // setting the TTL in seconds
    jedis.expire(cacheKey, 15);
}
```

```

// Getting the remaining ttl
System.out.println("TTL:" + jedis.ttl(cacheKey));
Thread.sleep(1000);
System.out.println("TTL:" + jedis.ttl(cacheKey));
// Getting the cache value
System.out.println("Cached Value:" + jedis.get(cacheKey));

// Wait for the TTL finishes
Thread.sleep(15000);

// trying to get the expired key
System.out.println("Expired Key:" + jedis.get(cacheKey));
}

```

Exemple 4.

```

public static void main(String[] args) {
    String cacheKey = "languages";
    Jedis jedis = new Jedis("localhost");
    // Adding a set as value

    jedis.sadd(cacheKey, "Java");
    jedis.sadd(cacheKey, "C#");
    jedis.sadd(cacheKey, "Python");// SADD

    // Getting all values in the set: SMEMBERS
    System.out.println("Languages: " + jedis.smembers(cacheKey));
    // Adding new values
    jedis.sadd(cacheKey, "Java");
    jedis.sadd(cacheKey, "Ruby");
    // Getting the values... it doesn't allow duplicates
    System.out.println("Languages: " + jedis.smembers(cacheKey));
}

```

Questions: Quelles sont les types de données stockés dans Redis, que peut on faire comme types de requêtes ? (à répondre dans le compte rendu de TP (le readme)

Bon TP

Olivier

