

## 用 MongoDB 和 Spring Data 创建 Java 应用

【IT168 专稿】当前，NOSQL 时代其实已经来临，在 NOSQL 的大家族中，MongoDB 以其可扩展性强，性能高以及开源占据了重要的地位。在 MongoDB 中，存储数据的方式不再是关系型数据库，而是以象 JSON 格式那样进行存储，所以可以称为文档型的 NOSQL。本文将以一个实际的例子，介绍如何运用 Spring 开源框架下的 Spring Data 数据存取框架结合 MongoDB，创建一个简单的 CRUD 应用(增删改查)。本文要求读者有一定的 MongoDB 和 Spring 的基础知识。

### 关于 Spring Data For MongoDB

大家对 Spring 框架应该是相当熟悉了，而 Spring data 则是 spring 新推出的一套方便开发者对关系数据库以及 NOSQL 进行存取开发的基于 spring 的 API 框架。而其中的一个分支 Spring Data for MongoDB 则是专门为 MongoDB 度身订造的用于方便开发者对 MongoDB 进行操作的一组 API。如果开发者熟悉 Spring 开发的话，则可以很容易地使用 Spring Data for MongoDB 进行开发。在 Spring Data For MongoDB 中，比如封装了 MongoTemplate，这个模板工具类有点象 JdbcTemplate 工具类，可以很方便地进行 MongoDB 的常用操作。它其中包括能在文档对象和 POJO 中做对应的关联映射，还有异常的封装都跟 spring 是一个体系的，而且对对象的查询更新等，都可以依旧使用开发者熟悉的比如 Query, Criteria 等进行数据库的查询等操作，此外，还可以使用 JPA 对 MongoDB 进行操作。开发者可以在如下的地址下载 Spring for MongoDB:<http://www.springsource.org/download/community>

### 安装 MongoDB

下面我们快速安装 MongoDB, 如果用户已经熟悉如何安装 MongoDB, 则可以跳过本小节, 直接进入下一小节。

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

1) 下载最新版本的 MongoDB, 解压缩到某个目录, 比如 c:\mongodb

2) mongodb 需要一个目录去存放数据, mongodb 默认的数据存放目录为 c:\data\db, 但我们可以随意指定目录去存放数据, 这里我们指定 c:\mongodb\data\db 去存放数据。

3) 接下来是启动的工作, 到命令行方式下输入如下指令:

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

要注意的是, 如果路径中含有空格, 则需要使用双引号包括起来, 如:

```
c:\mongodb\bin\mongod.exe -dbpath "c:\mongodb\data\db store place"
```

如果要以服务的形式在 windows 中启动, 请参考下面地址中的介绍:

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

### 使用 Spring Data For MongoDB 创建应用

接下来, 我们开始创建一个简单的 CRUD 功能的 JAVA 应用, 首先要在工程的 lib 目录下, 添加上 Spring Data For MongoDB 中相关的 lib 包。我们将工程命名为 NatureStore,

#### 步骤 1 创建 POJO

代码如下:

```
package com.angeslate.naturestore.domain;
```

```
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Document
public class Tree {

    @Id
    private String id;

    private String name;

    private String category;

    private int age;

    public Tree(String id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCategory() {
        return category;
    }
}
```

```
public void setCategory(String category) {
    this.category = category;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

@Override
public String toString() {
    return "Person [id=" + id + ", name=" + name + ", age=" + age
        + ", category=" + category + "]";
}
}
```

这里创建了一个 Tree 的 POJO，并且充分使用了 Spring 的注解功能，可以看到，@Document 注解，表示这个 POJO 最终要持久化为 MongoDB 中的 document，@id 指出了需要持久化的@id。