



服务端开发-Spring Data Mongodb、Redis

陶召胜

Spring Data项目

- Spring Data JDBC
- Spring Data JPA
- Spring Data MongoDB
- Spring Data Neo4j
- Spring Data Redis
- Spring Data Cassandra

内容

1、 MongoDB

- ✓ MongoDB 概念
- ✓ MongoDB Shell 命令使用
- ✓ 基于客户端访问
- ✓ 基于Spring data
- ✓ 两种方式插入数据性能对比

2、 Redis

- ✓ Redis数据类型
- ✓ Redis命令
- ✓ Spring data RedisTemplate
- ✓ 对象序列化

NoSQL

- NoSQL(Not Only SQL) , 指的是非关系型的数据库
- 没有声明性查询语言
- 没有预定义的模式
- 键-值对存储、列存储、文档存储、图形数据库

MongoDB

- MongoDB 是由C++语言编写的，是一个基于分布式文件存储的开源数据库系统。
- 文档存储一般用类似json的格式存储，存储的内容是文档型的。这样也就有机会对某些字段建立索引，实现关系数据库的某些功能

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

MongoDB Shell

- mongosh
- MongoDB Shell是MongoDB自带的交互式Javascript shell,用来对MongoDB进行操作和管理的交互式环境

MongoDB 概念

SQL术语/概念	MongoDB术语/概念	解释/说明
database	database	数据库
table	collection	数据库表/集合
row	document	数据记录行/文档
column	field	数据字段/域

id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas



```
{
  "_id": ObjectId("5146bb52d8524270060001f3"),
  "age": 25,
  "city": "Los Angeles",
  "email": "mark@abc.com",
  "user_name": "Mark Hanks"
}
{
  "_id": ObjectId("5146bb52d8524270060001f2"),
  "age": 31,
  "city": "Dallas",
  "email": "richard@abc.com",
  "user_name": "Richard Peter"
}
```

Spring data mongodb

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>  
    spring-boot-starter-data-mongodb  
  </artifactId>  
</dependency>
```


基于客户端访问

- `com.mongodb.client.MongoClients`

基于Spring data

- 数据库连接配置
 - ✓ `spring.data.mongodb.uri=mongodb://localhost:27017/test`
- 接口定义
- 领域类注解
 - ✓ `org.springframework.data.mongodb.core.mapping.Document`

两种方式插入数据性能对比

■ 基于客户端访问

```
{
  _id: ObjectId("65276ca0e9e6143e634c48ae"),
  name: 'Daren Berge',
  address: 'Apt. 462 9093 Schultz Center, Wanettamouth, RI 42491',
  birthday: ISODate("1965-02-25T17:15:23.316Z"),
  favouriteColor: [ 'maroon', 'orange', 'cyan', 'lime', 'olive' ],
  age: 58,
  amount: Decimal128("29"),
  phones: [
    { type: 'home', number: '(811) 788-8911' },
    { type: 'work', number: '1-969-966-6531' },
    { type: 'cell', number: '1-025-120-0416 x5188' }
  ]
}
```

■ 基于Spring data

```
{
  _id: ObjectId("65276e197a63e2239d0bd737"),
  name: 'Dr. Jamie Littell',
  address: '29440 Mayer Extension, West Marvel, ME 83429-5178',
  birthday: ISODate("1972-07-22T21:23:07.187Z"),
  age: 51,
  balance: '70',
  favouriteColor: [ 'silver', 'plum', 'lavender', 'mint green', 'grey' ],
  phones: [
    { type: 'home', number: '528.872.2388 x22549' },
    { type: 'work', number: '291-234-9370' },
    { type: 'cell', number: '773.946.5191 x0422' }
  ],
  _class: 'tacos.perf.Person'
}
```

工具java-faker

- <https://github.com/DiUS/java-faker>

<dependency>

<groupId>com.github.javafaker</groupId>

<artifactId>javafaker</artifactId>

<version>1.0.2</version>

</dependency>

内容

1、 MongoDB

- ✓ MongoDB 概念
- ✓ MongoDB Shell 命令使用
- ✓ 基于客户端访问
- ✓ 基于Spring data
- ✓ 两种方式插入数据性能对比

2、 Redis

- ✓ Redis数据类型
- ✓ Redis命令
- ✓ Spring data RedisTemplate
- ✓ 对象序列化

Redis

- 下载: <https://redis.io/download>
- key-value的Hash表结构, value是某数据结构
- 内存数据库 (缓存)
- 集群
- 主从 (master/slave) 复制
- 数据持久化
- 注意key、value区分大小写

Redis 命令

- redis-server
 - ✓ Redis配置, redis.conf
 - ✓ 默认端口号6379
- redis-cli, 客户端程序, `redis-cli -h host -p port -a password`
 - ✓ client list命令, 用于查看当前所有连接到服务器的客户端信息, 其中包括当前客户端所使用的数据库ID
 - ✓ info命令, 可以查看当前数据库的信息, 其中包括所使用的内存、键值对数量、客户端连接数等
 - ✓ config get命令, 可以查看Redis服务器的配置参数, 如: config get maxclients、config get databases (获取默认的数据库个数)
 - ✓ select 0-15, 选择数据库
 - ✓ flushdb, 删除当前数据库中的所有key
 - ✓ flushall, 删除所有数据库中的key
- Redis 命令参考: <http://doc.redisfans.com/>

Redis数据类型

- 指的是value类型
- String
- List
- Hash
- Set

Jedis和Lettuce

- Jedis 和Lettuce都是连接Redis Server的客户端程序
- RedisConnectionFactory接口, JedisConnectionFactory
- SpringBoot2.x 后默认使用的不再是Jedis而是lettuce, 所以spring-boot-starter-data-redis 依赖了:
lettuce-core、spring-data-redis、spring-boot-starter
- RedisConnectionFactory接口实现自动生成在上下文中, 可直接注入使用

spring data redis

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-redis</artifactId>  
</dependency>
```

使用RedisTemplate

- RedisTemplate<String, Product>
- StringRedisTemplate

RedisTemplate的子API

- 使用简单的值
 - ✓ opsForValue()----ValueOperations
 - ✓ set、get
- 使用List类型的值
 - ✓ opsForList()----ListOperations
 - ✓ rightPush、leftPop、range
- 在Set上执行操作
 - ✓ opsForSet()----SetOperations
 - ✓ add、difference、union、intersect
- 绑定到某个key上
 - ✓ boundListOps("cart")

指定序列化器

- 默认处理: JdkSerializationRedisSerializer, 对象需要实现Serializable接口
- Jackson2JsonRedisSerializer
- StringRedisSerializer

作业

- 更换上节课的作业，使用Spring Data mongodb做数据库持久化
- 要求连自己的mongodb数据库，不要使用内嵌数据库 (de.flapdoodle.embed.mongo)

Spring Boot Contacts

First Name:

Last Name:

Phone #:

Email:

- san zhang : 13312341234, zhangsan@163.com
- si li : 13345674567, lisi@163.com

提交

- 所有源代码，压缩成一个文件。不包含编译后的class文件（删除target目录）
- 浏览器访问应用，录入通讯录，然后访问MongoDB Shell，运行以下两个命令的截图：
 - ✓ `show collections`
 - ✓ `db.contact.find()`

```
test> show collections
contact
test> db.contact.find()
[
  {
    _id: ObjectId("65201117bb68071e994dddc2"),
    firstName: 'san',
    lastName: 'zhang',
    phoneNumber: '13312341234',
    emailAddress: 'zhangsan@163.com',
    _class: 'contacts.Contact'
  },
  {
    _id: ObjectId("65201162bb68071e994dddc3"),
    firstName: 'si',
    lastName: 'li',
    phoneNumber: '13345674567',
    emailAddress: 'lisi@163.com',
    _class: 'contacts.Contact'
  }
]
test>
```

下次课

- 时间：周一晚
- 地点：逸夫馆 馆3报告厅（在逸夫馆 馆3 一楼）

谢谢观看！

