

# Redis的两个框架：Jedis与Redisson

Jedis api 在线网址

<http://tool.oschina.net/uploads/apidocs/redis/clients/jedis/Jedis.html>

redisson 官网<https://redisson.org/>

redisson git项目地址<https://github.com/redisson/redisson>

## 1 概述

### 1.1. 主要内容

本文的主要内容为对比Redis的两个框架：Jedis与Redisson，分析各自的优势与缺点，为项目中Java缓存方案中的Redis编程模型的选择提供参考。

## 2. Jedis与Redisson对比

### 2.1. 概况对比

Jedis是Redis的Java实现的客户端，其API提供了比较全面的Redis命令的支持；Redisson实现了分布式和可扩展的Java数据结构，和Jedis相比，功能较为简单，不支持字符串操作，不支持排序、事务、管道、分区等Redis特性。Redisson的宗旨是促进使用者对Redis的关注分离，从而让使用者能够将精力更集中地放在处理业务逻辑上。

### 2.2. 编程模型

Jedis中的方法调用是比较底层的暴露的Redis的API，也即Jedis中的Java方法基本和Redis的API保持着一致，了解Redis的API，也就能熟练的使用Jedis。而Redisson中的方法则是进行比较高的抽象，每个方法调用可能进行了一个或多个Redis方法调用。

如下分别为Jedis和Redisson操作的简单示例：

Jedis设置key-value与set操作：

```
Jedis jedis = ...;
```

```
jedis.set("key", "value");
```

```
List values = jedis.mget("key", "key2", "key3");
```

Redisson操作map：

```
Redisson redisson = ...
```

```
RMap map = redisson.getMap("my-map"); // implement java.util.Map
```

```
map.put("key", "value");
```

```
map.containsKey("key");
```

```
map.get("key");
```

### 2.3. 可伸缩性

Jedis使用阻塞的I/O，且其方法调用都是同步的，程序流需要等到sockets处理完I/O才能执行，不支持异步。Jedis客户端实例不是线程安全的，所以需要通过连接池来使用Jedis。

Redisson使用非阻塞的I/O和基于Netty框架的事件驱动的通信层，其方法调用是异步的。Redisson的API是线程安全的，所以可以操作单个Redisson连接来完成各种操作。

#### 2.4. 数据结构

Jedis仅支持基本的数据类型如：String、Hash、List、Set、Sorted Set。

Redisson不仅提供了一系列的分布式Java常用对象，基本可以与Java的基本数据结构通用，还提供了许多分布式服务，其中包括（BitSet, Set, Multimap, SortedSet, Map, List, Queue, BlockingQueue, Deque, BlockingDeque, Semaphore, Lock, AtomicLong, CountdownLatch, Publish / Subscribe, Bloom filter, Remote service, Spring cache, Executor service, Live Object service, Scheduler service）。

在分布式开发中，Redisson可提供更便捷的方法。

#### 2.5. 第三方框架整合

- 1 Redisson提供了和Spring框架的各项特性类似的，以Spring XML的命名空间的方式配置RedissonClient实例和它所支持的所有对象和服务；
- 2 Redisson完整的实现了Spring框架里的缓存机制；
- 3 Redisson在Redis的基础上实现了Java缓存标准规范；
- 4 Redisson为Apache Tomcat集群提供了基于Redis的非黏性会话管理功能。该功能支持Apache Tomcat的6、7和8版。
- 5 Redisson还提供了Spring Session会话管理器的实现。