java 中通过 Lettuce 来操作 Redis

作者：flowaters

原文链接：https://ld246.com/article/1534668275802

来源网站：链滴

前言

Redis初体验中记录了Redis的基本操作。

本文学习一下如何通过Java操作Redis。

Java操作Redis的库有两个，Jedis和Lettuce，目前SpringBoot 2.x中已经将Jedis换成了Lettuce。

本文直接从Lettuce来学习。

基本使用

第一个例子

依赖

pom.xml

```
    <dependency>
        <groupId>io.lettuce</groupId>
        <artifactId>lettuce-core</artifactId>
        <version>5.0.4.RELEASE</version>
    </dependency>
```

代码

BasicUsage.java

```
package testRedis;

import io.lettuce.core.RedisClient;
import io.lettuce.core.api.StatefulRedisConnection;
import io.lettuce.core.api.sync.RedisStringCommands;

public class BasicUsage {
    public static void main(String[] args) {
        // client
        RedisClient client = RedisClient.create("redis://localhost");

        // connection, 线程安全的长连接，连接丢失时会自动重连，直到调用close关闭连接。
        StatefulRedisConnection<String, String> connection = client.connect();

        // sync, 默认超时时间为60s.
        RedisStringCommands<String, String> sync = connection.sync();
        sync.set("host", "note.abeffect.com");
        String value = sync.get("host");
        System.out.println(value);

        // close connection
        connection.close();
        // shutdown
        client.shutdown();
    }
}
```

输出

```
[DEBUG] (main) Using Console logging
[DEBUG] (main) Starting UnsafeSupport init in Java 1.8
[TRACE] (main) sun.misc.Unsafe.theUnsafe ok
[TRACE] (main) sun.misc.Unsafe.copyMemory ok
[TRACE] (main) java.nio.Buffer.address ok
[DEBUG] (main) Unsafe is available
Aug 19, 2018 11:29:50 AM io.lettuce.core.EpollProvider <clinit>
信息: Starting without optional epoll library
Aug 19, 2018 11:29:50 AM io.lettuce.core.KqueueProvider <clinit>
信息: Starting without optional kqueue library
note.abeffect.com
```

其它同步使用方式

设定超时时间为20s

```
        RedisClient client = RedisClient.create(RedisURI.create("localhost", 6379));
        client.setDefaultTimeout(Duration.ofSeconds(20));
```

使用RedisURI

```
        RedisURI redisUri = RedisURI.Builder.redis("localhost").withPassword("authentication").wi
hDatabase(2).build();
        RedisClient client = RedisClient.create(redisUri);
```

或者

```
        RedisURI redisUri = RedisURI.create("redis://authentication@localhost/2");
        RedisClient client = RedisClient.create(redisUri);
```

异步使用例子

异步调用，可以避免将CPU浪费在等待网络IO和磁盘IO时上，实现提高资源使用率。

基本例子

```
package testRedis;

import java.util.concurrent.ExecutionException;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.TimeoutException;
import io.lettuce.core.RedisClient;
import io.lettuce.core.RedisFuture;
import io.lettuce.core.api.StatefulRedisConnection;
import io.lettuce.core.api.async.RedisStringAsyncCommands;

public class AsynchronousAPI {
    public static void main(String[] args) {
        // client
        RedisClient client = RedisClient.create("redis://localhost");

        // connect
        StatefulRedisConnection<String, String> connection = client.connect();

        // async
        RedisStringAsyncCommands<String, String> async = connection.async();

        RedisFuture<String> future = async.get("host");

        try {
            String value = future.get(60, TimeUnit.SECONDS);
            System.out.println(value);
        } catch (InterruptedException | ExecutionException | TimeoutException e) {
            e.printStackTrace();
        }

        connection.close();
        client.shutdown();
    }
}
```

结果

```
[DEBUG] (main) Using Console logging
[DEBUG] (main) Starting UnsafeSupport init in Java 1.8
[TRACE] (main) sun.misc.Unsafe.theUnsafe ok
[TRACE] (main) sun.misc.Unsafe.copyMemory ok
[TRACE] (main) java.nio.Buffer.address ok
[DEBUG] (main) Unsafe is available
Aug 19, 2018 3:36:01 PM io.lettuce.core.EpollProvider <clinit>
信息: Starting without optional epoll library
Aug 19, 2018 3:36:01 PM io.lettuce.core.KqueueProvider <clinit>
信息: Starting without optional kqueue library
note.abeffect.com
```

使用Consumer监听器

```
        RedisFuture<String> future = async.get("host");

        future.thenAccept(new Consumer<String>() {
            @Override
            public void accept(String value) {
                System.out.println(value);
            }
```

```
    });
```

使用Lambda表达式

```
    RedisFuture<String> future = async.get("host");
    future.thenAccept(System.out::println);
```

使用独立的线程池

为了防止阻塞默认的线程池，可以在单独的线程池中执行异步请求。

```
    Executor sharedExecutor = Executors.newFixedThreadPool(1);
    RedisFuture<String> future = async.get("host");
    future.thenAcceptAsync(System.out::println, sharedExecutor);
```

更多的例子见Asynchronous-API 官方文档

Reactive调用

```
package testRedis;

import io.lettuce.core.RedisClient;
import io.lettuce.core.api.StatefulRedisConnection;
import io.lettuce.core.api.reactive.RedisStringReactiveCommands;

public class ReactiveAPI {
  public static void main(String[] args) {
    // client
    RedisClient client = RedisClient.create("redis://localhost");

    // connect
    StatefulRedisConnection<String, String> connection = client.connect();

    // reactive
    RedisStringReactiveCommands<String, String> reactive = connection.reactive();

    reactive.get("host").subscribe(System.out::println);

    try {
      Thread.sleep(1000L);
    } catch (InterruptedException e) {
      e.printStackTrace();
    }

    connection.close();
    client.shutdown();
  }
}
```

结果

```
[DEBUG] (main) Using Console logging
```

```
[DEBUG] (main) Starting UnsafeSupport init in Java 1.8
[TRACE] (main) sun.misc.Unsafe.theUnsafe ok
[TRACE] (main) sun.misc.Unsafe.copyMemory ok
[TRACE] (main) java.nio.Buffer.address ok
[DEBUG] (main) Unsafe is available
Aug 19, 2018 4:00:09 PM io.lettuce.core.EpollProvider <clinit>
信息: Starting without optional epoll library
Aug 19, 2018 4:00:09 PM io.lettuce.core.KqueueProvider <clinit>
信息: Starting without optional kqueue library
note.abeffect.com
```

更多查看Reactive API官方文档，或者Reactive相关资料。

Pub/Sub

```
package testRedis;

import io.lettuce.core.RedisClient;
import io.lettuce.core.pubsub.RedisPubSubListener;
import io.lettuce.core.pubsub.StatefulRedisPubSubConnection;
import io.lettuce.core.pubsub.api.sync.RedisPubSubCommands;

public class PubSubApi {
    public static void main(String[] args) {
        RedisClient client = RedisClient.create("redis://localhost");

        // connection
        RedisPubSubListener<String, String> listener = new RedisPubSubListener<String, String
() {
            @Override
            public void message(String pattern, String channel) {
                System.out.println("message: " + pattern + ", " + channel);
            }

            @Override
            public void message(String pattern, String channel, String message) {
                System.out.println("message: " + pattern + ", " + channel + ", " + message);
            }

            @Override
            public void psubscribed(String pattern, long count) {
                System.out.println("psub: " + pattern + ", " + count);
            }

            @Override
            public void punsubscribed(String pattern, long count) {
                System.out.println("punsub: " + pattern + ", " + count);
            }

            @Override
            public void subscribed(String channel, long count) {
                System.out.println("sub: " + channel + ", " + count);
            }
            @Override
            public void unsubscribed(String channel, long count) {
                System.out.println("ubsub: " + channel + ", " + count);
            }
        };

        StatefulRedisPubSubConnection<String, String> pubSubConnection = client.connectPubSub();

        pubSubConnection.addListener(listener);

        RedisPubSubCommands<String, String> connection = pubSubConnection.sync();
        connection.subscribe("channel");

        try {
            Thread.sleep(100000L);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        pubSubConnection.close();
        client.shutdown();
    }
}
```

启动中，在redis中执行：

```
127.0.0.1:6379> PUBLISH channel 1
(integer) 1
127.0.0.1:6379> PUBLISH channel 2
(integer) 1
127.0.0.1:6379> PUBLISH channel 3
(integer) 1
```

输出结果

```
[DEBUG] (main) Using Console logging
[DEBUG] (main) Starting UnsafeSupport init in Java 1.8
[TRACE] (main) sun.misc.Unsafe.theUnsafe ok
[TRACE] (main) sun.misc.Unsafe.copyMemory ok
[TRACE] (main) java.nio.Buffer.address ok
[DEBUG] (main) Unsafe is available
Aug 19, 2018 4:41:34 PM io.lettuce.core.EpollProvider <clinit>
信息: Starting without optional epoll library
Aug 19, 2018 4:41:34 PM io.lettuce.core.KqueueProvider <clinit>
信息: Starting without optional kqueue library
sub: channel, 1
message: channel, 1
message: channel, 2
message: channel, 3
```

参考

原文链接：java 中通过 Lettuce 来操作 Redis

- lettuce - Advanced Java Redis client

原文链接：java 中通过 Lettuce 来操作 Redis