

Dubbo源码解析（二十九）远程调用——redis协议

[dubbo](#)  [java](#) 阅读约 19 分钟

远程调用——redis协议

目标：介绍redis协议的设计和实现，介绍dubbo-rpc-redis的源码。

前言

dubbo支持的redis协议是基于Redis的，[Redis](#) 是一个高效的 KV 存储服务器，跟memcached协议实现差不多，在dubbo中也没有涉及到关于redis协议的服务暴露，只有服务引用，因为在访问服务器时，Redis客户端可以在服务器上存储也可以获取。

源码分析

（一）RedisProtocol

该类继承了AbstractProtocol类，是redis协议实现的核心。

1.属性

```
/**
 * 默认端口号
 */
public static final int DEFAULT_PORT = 6379;
```

2.export

```
@Override
public <T> Exporter<T> export(final Invoker<T> invoker) throws RpcException {
    // 不支持redis协议的服务暴露，抛出异常
    throw new UnsupportedOperationException("Unsupported export redis service. url: " + invoker.getUrl());
}
```

可以看到不支持服务暴露。

3.refer

```
@Override
public <T> Invoker<T> refer(final Class<T> type, final URL url) throws RpcException {
    try {
        // 实例化对象池
        GenericObjectPoolConfig config = new GenericObjectPoolConfig();
        // 如果 testOnBorrow 被设置, pool 会在 borrowObject 返回对象之前使用 PoolableObjectFactory的 validateObject
        // 来验证这个对象是否有效
        // 要是对象没通过验证, 这个对象会被丢弃, 然后重新选择一个新的对象。
        config.setTestOnBorrow(url.getParameter("test.on.borrow", true));
        // 如果 testOnReturn 被设置, pool 会在 returnObject 的时候通过 PoolableObjectFactory 的validateObject 方法
        // 验证对象
        // 如果对象没通过验证, 对象会被丢弃, 不会被放到池中。
        config.setTestOnReturn(url.getParameter("test.on.return", false));
        // 指定空闲对象是否应该使用 PoolableObjectFactory 的 validateObject 校验, 如果校验失败, 这个对象会从对象池中被清
        // 除。
        // 这个设置仅在 timeBetweenEvictionRunsMillis 被设置成正值 ( >0 ) 的时候才会生效。
        config.setTestWhileIdle(url.getParameter("test.while.idle", false));
        if (url.getParameter("max.idle", 0) > 0)
            // 控制一个pool最多有多少个状态为空闲的jedis实例。
            config.setMaxIdle(url.getParameter("max.idle", 0));
        if (url.getParameter("min.idle", 0) > 0)
            // 控制一个pool最少有多少个状态为空闲的jedis实例。
            config.setMinIdle(url.getParameter("min.idle", 0));
        if (url.getParameter("max.active", 0) > 0)
            // 控制一个pool最多有多少个jedis实例。
            config.setMaxTotal(url.getParameter("max.active", 0));
```

可以看到首先是对连接池的配置赋值，然后创建连接池后，根据redis的get、set、delete命令来进行相关操作。

后记

该部分相关的源码解析地址：<https://github.com/CrazyHZM/i...>

该文章讲解了远程调用中关于redis协议实现的部分，逻辑比较简单。接下来我将开始对rpc模块关于rest协议部分进行讲解。

阅读 422 • 更新于 11月8日

👍 赞 1

🔖 收藏 1

💰 赞赏

🔗 分享

本作品系 原创 ， 作者保留所有权利，未经作者允许，禁止转载和演绎



crazyhzm

🔖 265 🔄

关注作者

0 条评论

得票 • 时间



撰写评论 ...

提交评论

推荐阅读

Dubbo 源码分析 - SPI 机制

SPI全称为ServiceProviderInterface，是Java提供了一种服务发现机制。SPI的本质是将接口实现类的全限定名配置在文件中，并由服...

[coolblog](#) • 阅读 250