

Please visit website: <http://cxyroad.com>

简单实现的场景，Redisson 是一个很好的选择。

然而，如果你的系统需要复杂的消息传递保证或持久性，可能需要考虑其他消息队列系统，如 RabbitMQ、Apache Kafka 等。

****工具类和示例**:**

使用 Redisson 实现发布/订阅模式可以通过创建一个服务类来封装 Redisson 的发布/订阅功能，并使用 Spring 的注解来定义单频道和多频道的订阅。

下面是一个实现的例子，包括了工具类的创建和单频道订阅、多频道订阅的示例。

Redisson 配置首先，需要配置 Redisson 客户端作为一个 Spring Bean，以便在整个应用程序中使用。

```
...
import org.redisson.api.RedissonClient;
import org.redisson.config.Config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class RedissonConfig {

    @Bean(destroyMethod = "shutdown")
    public RedissonClient redissonClient() {
        Config config = new Config();
        config.useSingleServer().setAddress("redis://127.0.0.1:6379");
        return Redisson.create(config);
    }
}
...
```

RedisPubSubService 工具类:创建一个服务类来封装 Redisson 的发布/订阅逻辑。

```
...
import org.redisson.api.RedissonClient;
import org.redisson.api.listener.MessageListener;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
@Service
public class RedisPubSubService {

    private final RedissonClient redissonClient;

    @Autowired
    public RedisPubSubService(RedissonClient redissonClient) {
        this.redissonClient = redissonClient;
    }
}
```

...

发布消息示例:创建一个 REST 控制器，用于发布消息到指定的频道。

...

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/pubsub")
public class PubSubController {

    private final RedisPubSubService redisPubSubService;

    @Autowired
    public PubSubController(RedisPubSubService redisPubSubService) {
        this.redisPubSubService = redisPubSubService;
    }

    @PostMapping("/publish/{topic}")
    public String publishMessage(@PathVariable String topic,
        @RequestBody String message) {
        redisPubSubService.publishToTopic(topic, message);
        return "Message published to " + topic;
    }
}
```

...

在这个控制器中，publishMessage 方法允许客户端通过发送 POST 请求到 `/pubsub/publish/{topic}` 来发布消息到指定的频道。

这些示例展示了如何在 Spring 应用程序中使用 Redisson 实现发布/订阅模式。RedisPubSubService 服务类封装了 Redisson 的发布/订阅操作，

而 SingleTopicSubscriber 和 MultiTopicSubscriber 组件分别用于订阅单个频道和多个频道。PubSubController 提供了一个 RESTful 接口来发布消息。

![扫码_搜索联合传播样式-白色版.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/24b8cda84da6431e84f713a1efa50937~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1816&h=624&s=4532874&e=bmp&b=fe fefe)

原文链接: <https://juejin.cn/post/7354019135992823848>