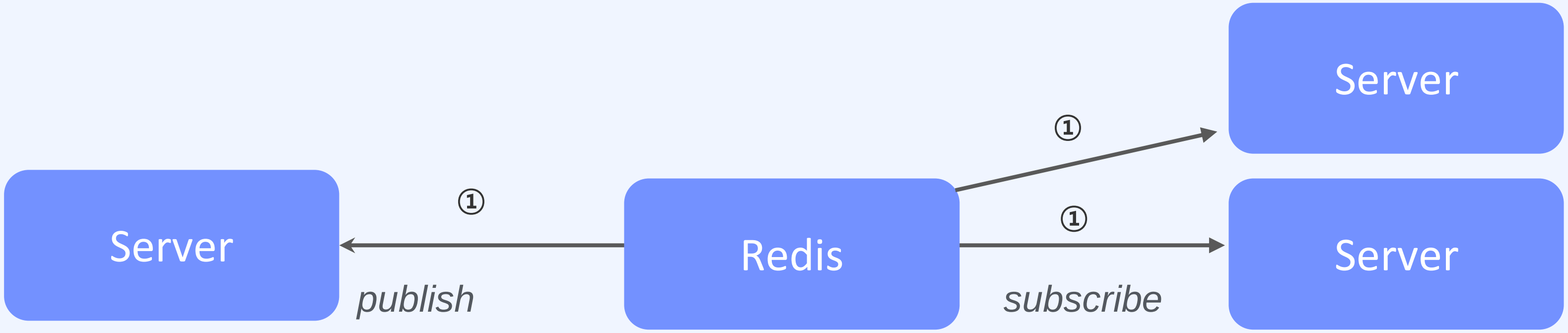
Redis

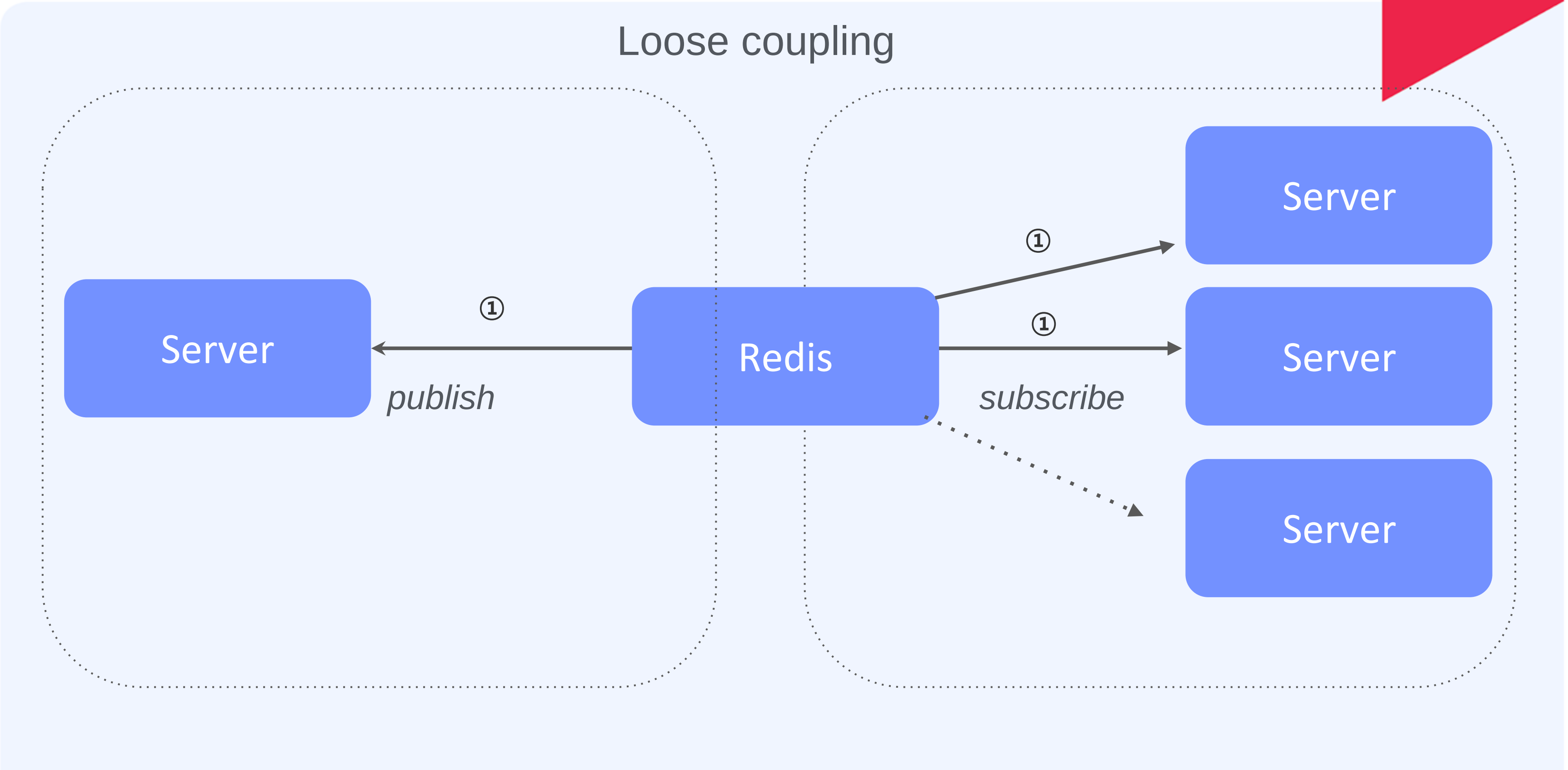
18 Spring Boot Pub/Sub

Pub/Sub

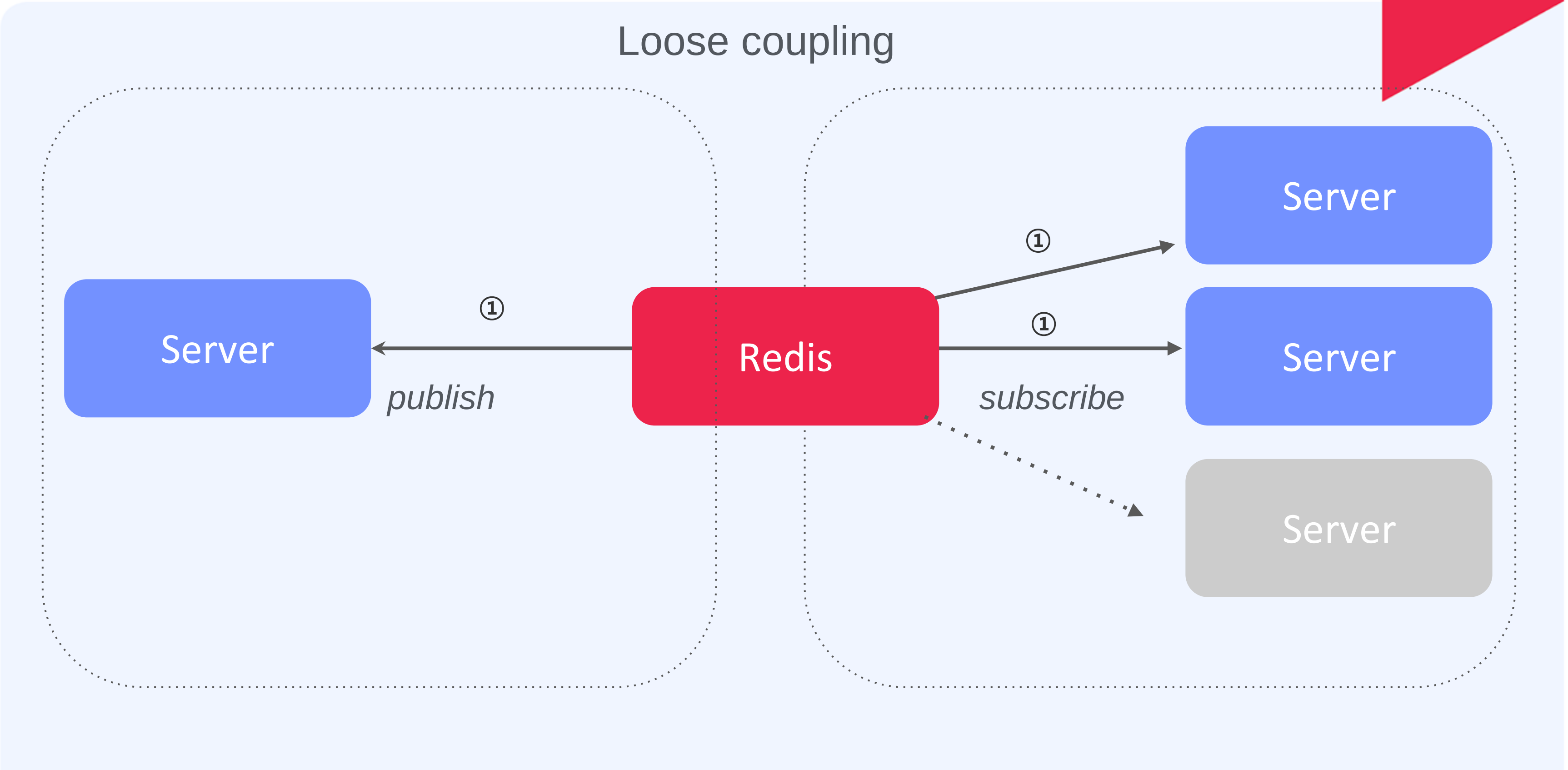


Publish/Subscribe

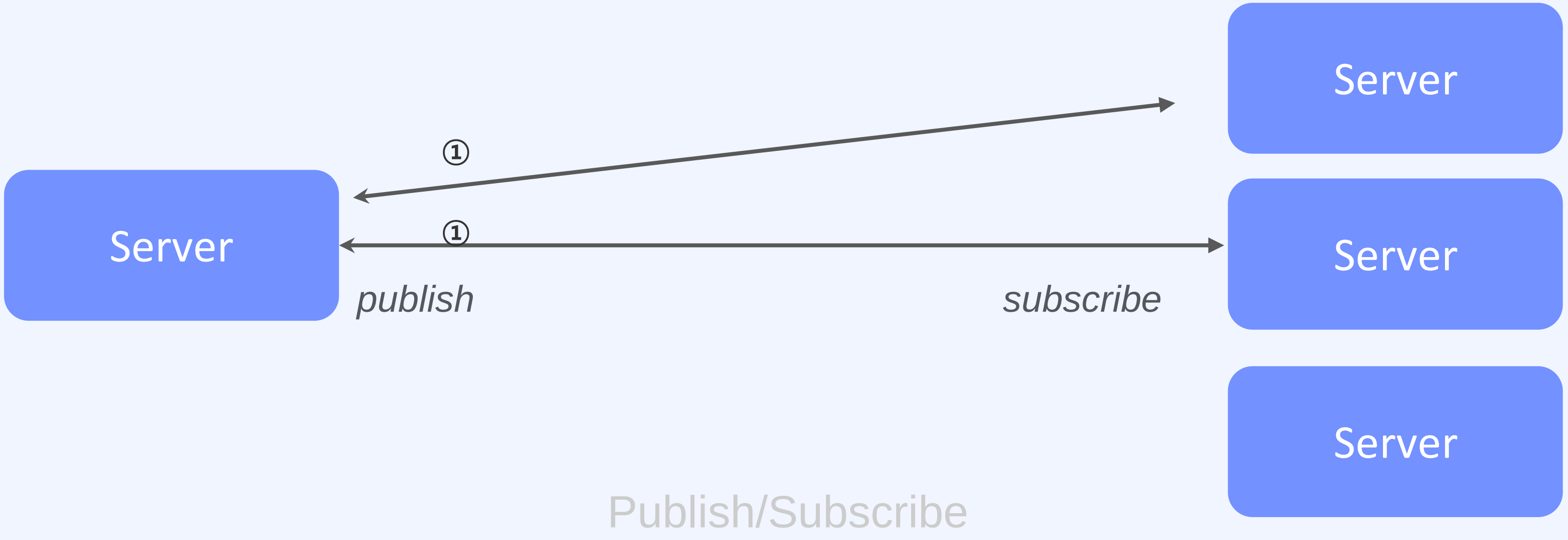
Pub/Sub


Loose coupling
Server
①
publish
Redis
①
①
subscribe
Server
Server
Server

Pub/Sub


Loose coupling
Server
①
publish
Redis
①
①
subscribe
Server
Server
Server

Pub/Sub

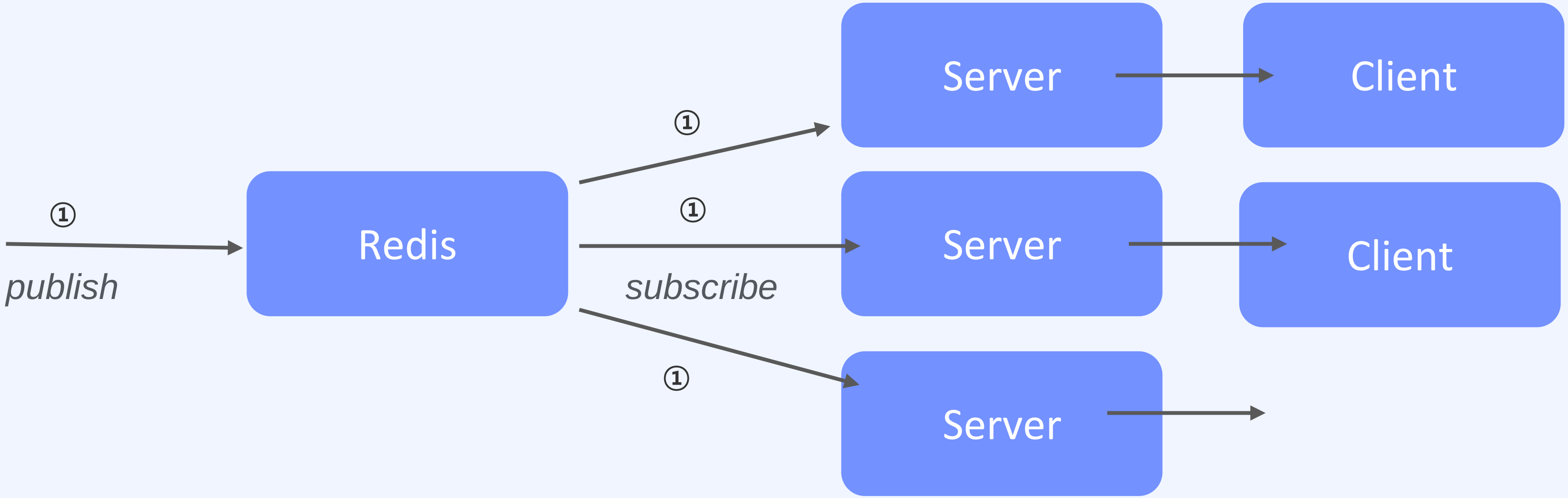

Server
Server
Server
Server
①
①
publish
subscribe
Publish/Subscribe

Pub/Sub

Pub/Sub

1. Event
2. Notification
3. Realtime message
4. …

Pub/Sub


①
publish
Redis
①
①
subscribe
①
Server
Server
Server
Client
Client

Redis pub/sub

Pub/Sub

Redis

```
127.0.0.1:6379> SUBSCRIBE users:unregister

Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "users:unregister"
3) (integer) 1
```

Pub/Sub

Redis

```
127.0.0.1:6379> PUBLISH users:unregister 100
(integer) 1
```

```
127.0.0.1:6379> SUBSCRIBE users:unregister
...
1) "message"
2) "users:unregister"
3) "100"
```

Pub/Sub

Redis

```
127.0.0.1:6379> PUBSUB channels
1) "users:unregister"
```

```
127.0.0.1:6379> PUBSUB numsub users:unregister
1) "users:unregister"
2) (integer) 1
```

Pub/Sub

Redis

```
127.0.0.1:6379> PSUBSCRIBE users:*

Reading messages... (press Ctrl-C to quit)
1) "psubscribe"
2) "users:*"
3) (integer) 1
```

Pub/Sub

Redis

```
127.0.0.1:6379> PUBLISH users:1234 adsf
(integer) 1
127.0.0.1:6379> PUBLISH users:5555 adsf
(integer) 1
```

↓

```
127.0.0.1:6379> PSUBSCRIBE users:*
...
1) "pmessage"
2) "users:*"
3) "users:5555"
4) "adsf"
```

Pub/Sub

실습

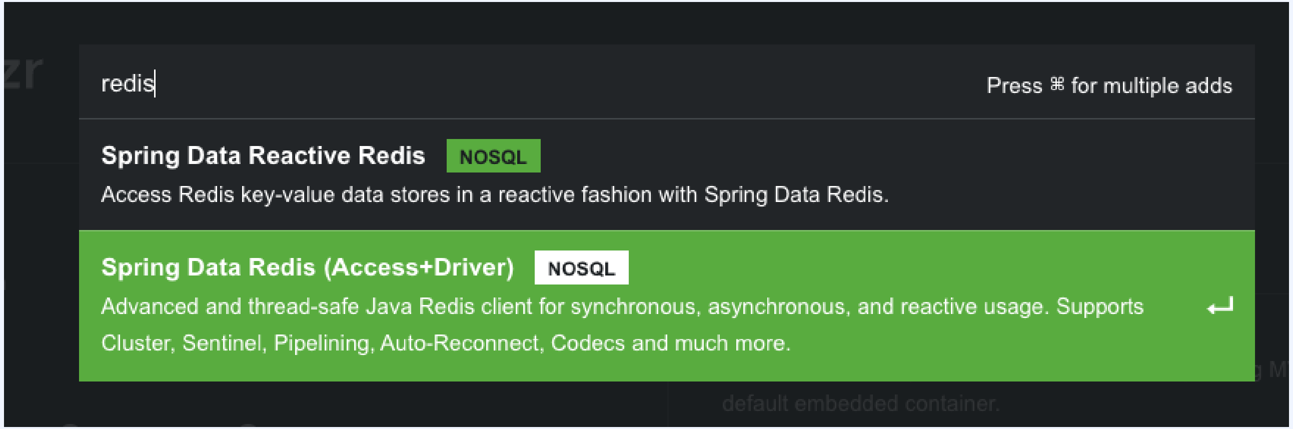
Pub/Sub

Spring boot

Spring boot

Pub/Sub

Spring boot



redis

Press ⌘ for multiple adds

Spring Data Reactive Redis NOSQL

Access Redis key-value data stores in a reactive fashion with Spring Data Redis.

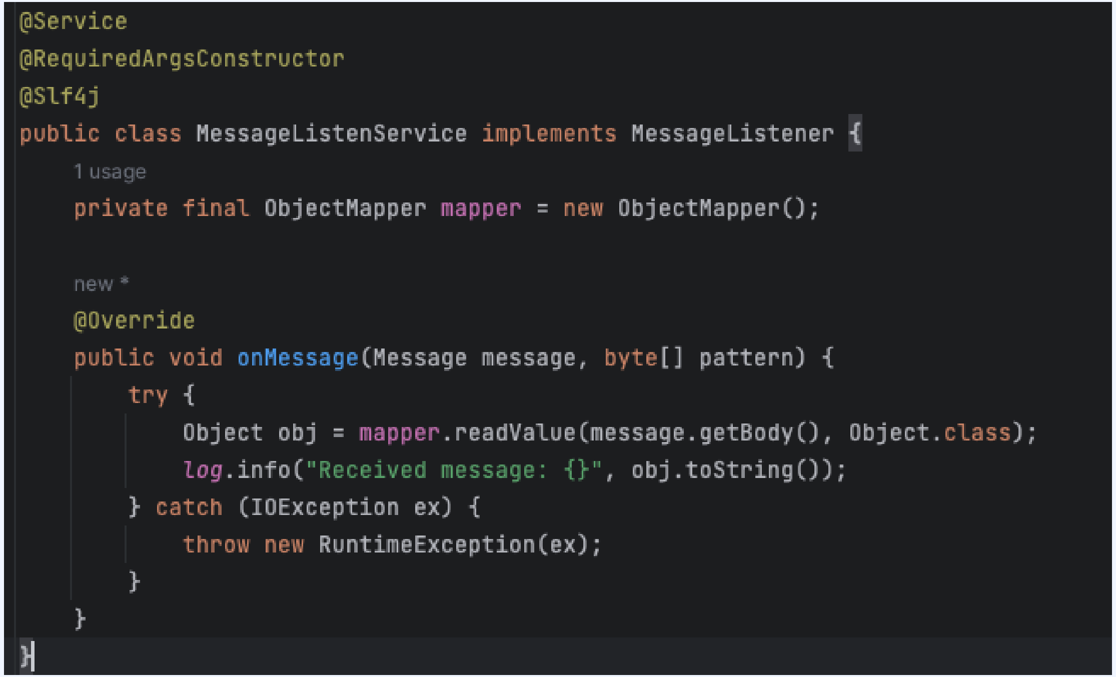
Spring Data Redis (Access+Driver) NOSQL

Advanced and thread-safe Java Redis client for synchronous, asynchronous, and reactive usage. Supports Cluster, Sentinel, Pipelining, Auto-Reconnect, Codecs and much more.

https://start.spring.io/

Pub/Sub

Spring boot



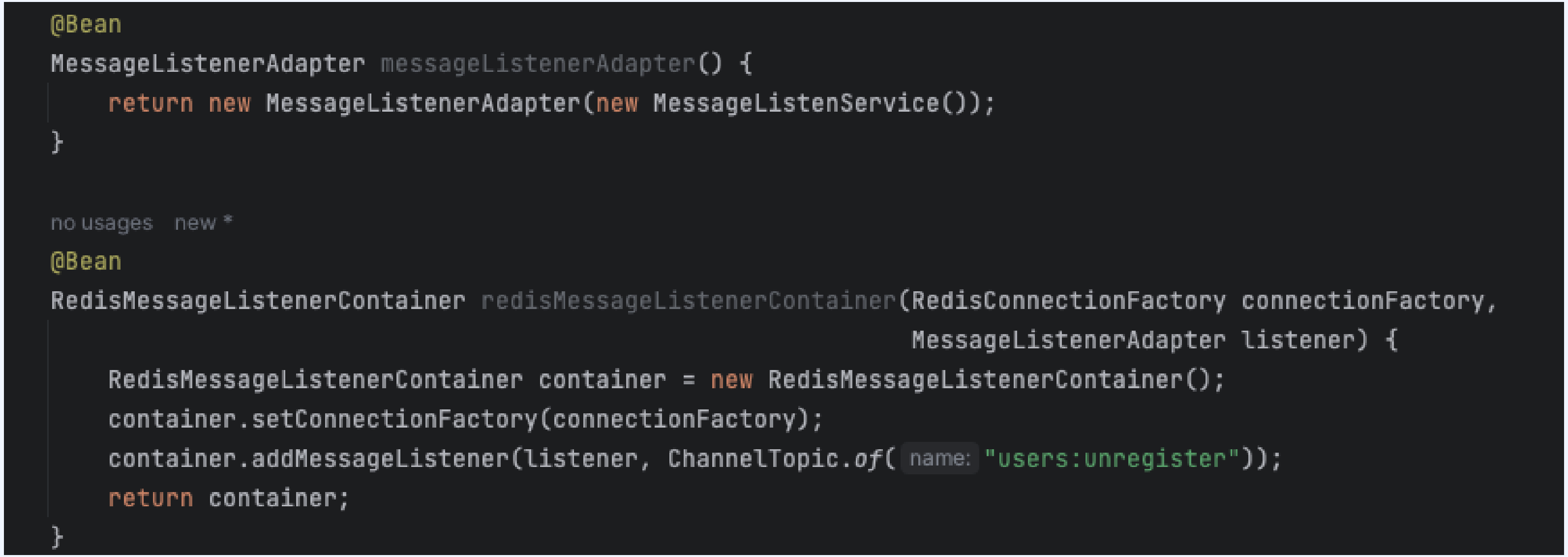
```
@Service
@RequiredArgsConstructor
@Slf4j
public class MessageListenService implements MessageListener {
    1 usage
    private final ObjectMapper mapper = new ObjectMapper();

    new *
    @Override
    public void onMessage(Message message, byte[] pattern) {
        try {
            Object obj = mapper.readValue(message.getBody(), Object.class);
            log.info("Received message: {}", obj.toString());
        } catch (IOException ex) {
            throw new RuntimeException(ex);
        }
    }
}
```

Pub/Sub

Spring boot

2. Redis

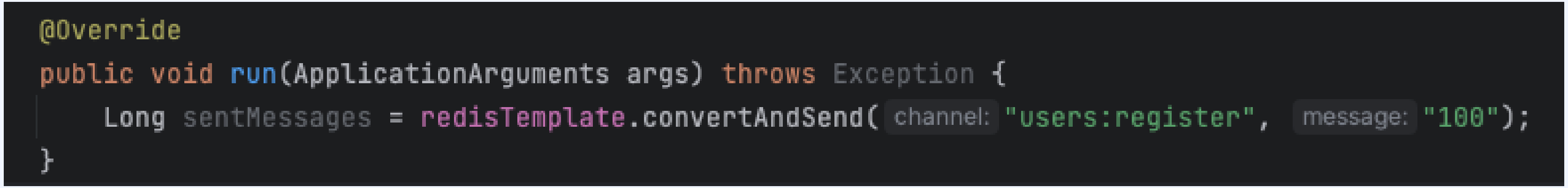


```
@Bean
MessageListenerAdapter messageListenerAdapter() {
    return new MessageListenerAdapter(new MessageListenService());
}

no usages  new *
@Bean
RedisMessageListenerContainer redisMessageListenerContainer(RedisConnectionFactory connectionFactory,
                                                            MessageListenerAdapter listener) {
    RedisMessageListenerContainer container = new RedisMessageListenerContainer();
    container.setConnectionFactory(connectionFactory);
    container.addMessageListener(listener, ChannelTopic.of( name: "users:unregister"));
    return container;
}
```

Pub/Sub

Spring boot



```
@Override
public void run(ApplicationArguments args) throws Exception {
    Long sentMessages = redisTemplate.convertAndSend( channel: "users:register",  message: "100");
}
```

Pub/Sub

실습