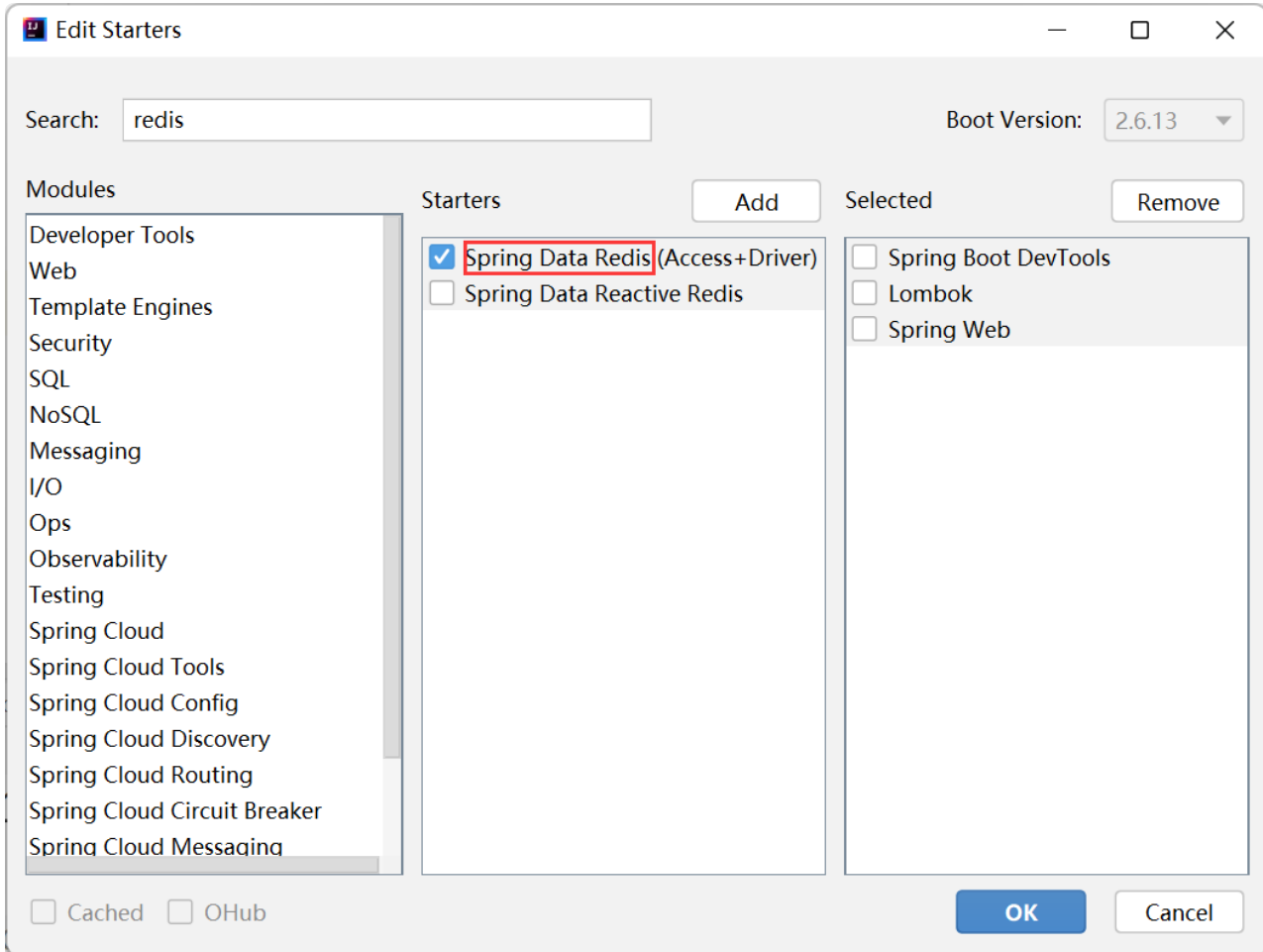


SpringBoot 集成 Redis

1. 添加 redis 依赖



或将以下配置添加到 pom.xml 中：

```
XML | 复制代码
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-data-redis</artifactId>
4 </dependency>
```

2. 配置 redis

```
1 spring.redis.database=0
2 spring.redis.port=6379
3 spring.redis.host=82.157.146.10
4 # 可省略
5 spring.redis.lettuce.pool.min-idle=5
6 spring.redis.lettuce.pool.max-idle=10
7 spring.redis.lettuce.pool.max-active=8
8 spring.redis.lettuce.pool.max-wait=1ms
9 spring.redis.lettuce.shutdown-timeout=100ms
```

3.手动操作 redis

```
1 import org.springframework.beans.factory.annotation.Autowired;
2 import org.springframework.data.redis.core.StringRedisTemplate;
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 import java.util.concurrent.TimeUnit;
7
8 @RestController
9 public class RedisController {
10
11     @Autowired
12     private StringRedisTemplate stringRedisTemplate;
13     // 在 redis 存储数据
14     @RequestMapping("/setrs")
15     public String setRedis(String name, String value) {
16         stringRedisTemplate.opsForValue().set(name, value,
17             30, TimeUnit.SECONDS);
18         return "Set redis success.";
19     }
20     // 读取 redis 中的数据
21     @RequestMapping("/getrs")
22     public String getRedis(String name) {
23         Object valObj = stringRedisTemplate.opsForValue().get(name);
24         if (valObj != null) return valObj.toString();
25         return "Null";
26     }
27 }
```

4.注解操作 redis

4.1 开启缓存

▼

Java | 复制代码

```
1  @SpringBootApplication
2  @EnableCaching
```

4.2 注解操作 redis

```
1  import org.springframework.cache.annotation.CacheEvict;
2  import org.springframework.cache.annotation.CachePut;
3  import org.springframework.cache.annotation.Cacheable;
4  import org.springframework.util.StringUtils;
5  import org.springframework.web.bind.annotation.RequestMapping;
6  import org.springframework.web.bind.annotation.RestController;
7
8  @RestController
9  public class RedisAnnotationController {
10
11      @Cacheable(value = "spring:cache", key = "#name")
12      @RequestMapping("/setrs")
13      public String setRedis(String name, String value) {
14          if (!StringUtils.hasLength(name) || !StringUtils.hasLength(value))
15          {
16              return "请先输入 name 和 value";
17          }
18          return value;
19      }
20
21      @RequestMapping("/uprs")
22      @CachePut(value = "spring:cache", key = "#name")
23      public String updateRedis(String name, String value) {
24          if (!StringUtils.hasLength(name) || !StringUtils.hasLength(value))
25          {
26              return "请先输入 name 和 value";
27          }
28          return value;
29      }
30
31      @RequestMapping("/delrs")
32      @CacheEvict(value = "spring:cache", key = "#name")
33      public String delUser(String name) {
34          return "delete success";
35      }
36  }
```