《Spring Boot 基础教程》

第 17 节

使用 NoSQL 数据库-redis

redis windows 版本下载： https://github.com/MSOpenTech/redis/releases

一、添加依赖

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

二、配置文件：

```
#redis
spring.redis.host=localhost
spring.redis.port=6379
#spring.redis.password=123456
#spring.redis.database=0
#spring.redis.pool.max-active=8
#spring.redis.pool.max-idle=8
#spring.redis.pool.max-wait=-1
#spring.redis.pool.min-idle=0
#spring.redis.timeout=0
```

三、测试

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.redis.core.StringRedisTemplate;
import org.springframework.data.redis.core.ValueOperations;
import org.springframework.stereotype.Component;

/**
 * @author wujing
 */
@Component
public class RoncooRedisComponent {

 @Autowired
 private StringRedisTemplate stringRedisTemplate;

 public void set(String key, String value) {
     ValueOperations<String, String> ops = this.stringRedisTemplate.opsForValue();
     if (!this.stringRedisTemplate.hasKey(key)) {
         ops.set(key, value);
         System.out.println("set key success");
```

```
            } else {
                // 存在则打印之前的 value 值
                System.out.println("this key = " + ops.get(key));
            }
        }

        public String get(String key) {
            return this.stringRedisTemplate.opsForValue().get(key);
        }

        public void del(String key) {
            this.stringRedisTemplate.delete(key);
        }
    }
```

```
@Autowired
private RoncooRedisComponent roncooRedisComponent;

@Test
public void set() {
    roncooRedisComponent.set("roncoo", "hello world");
}

@Test
public void get() {
    System.out.println(roncooRedisComponent.get("roncoo"));
}

@Test
public void del() {
    roncooRedisComponent.del("roncoo");
}
```

注：生产环境下，如果外网可以访问，一定要设置密码！