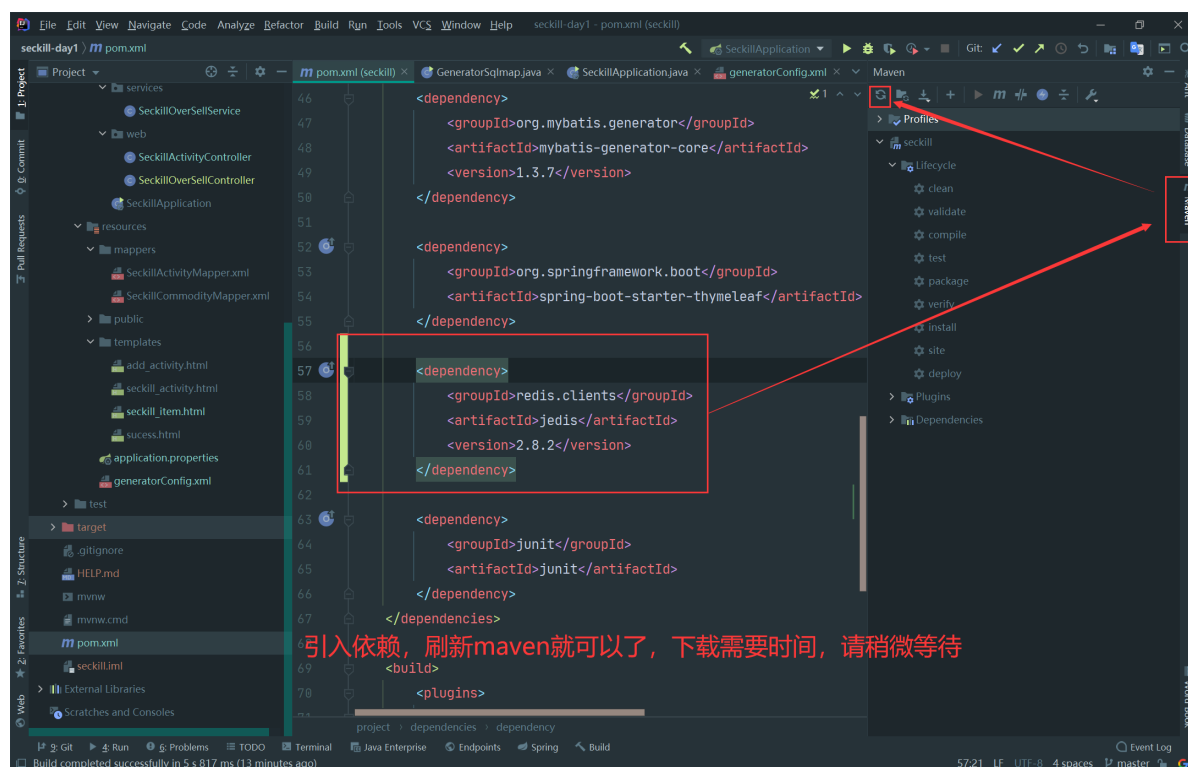


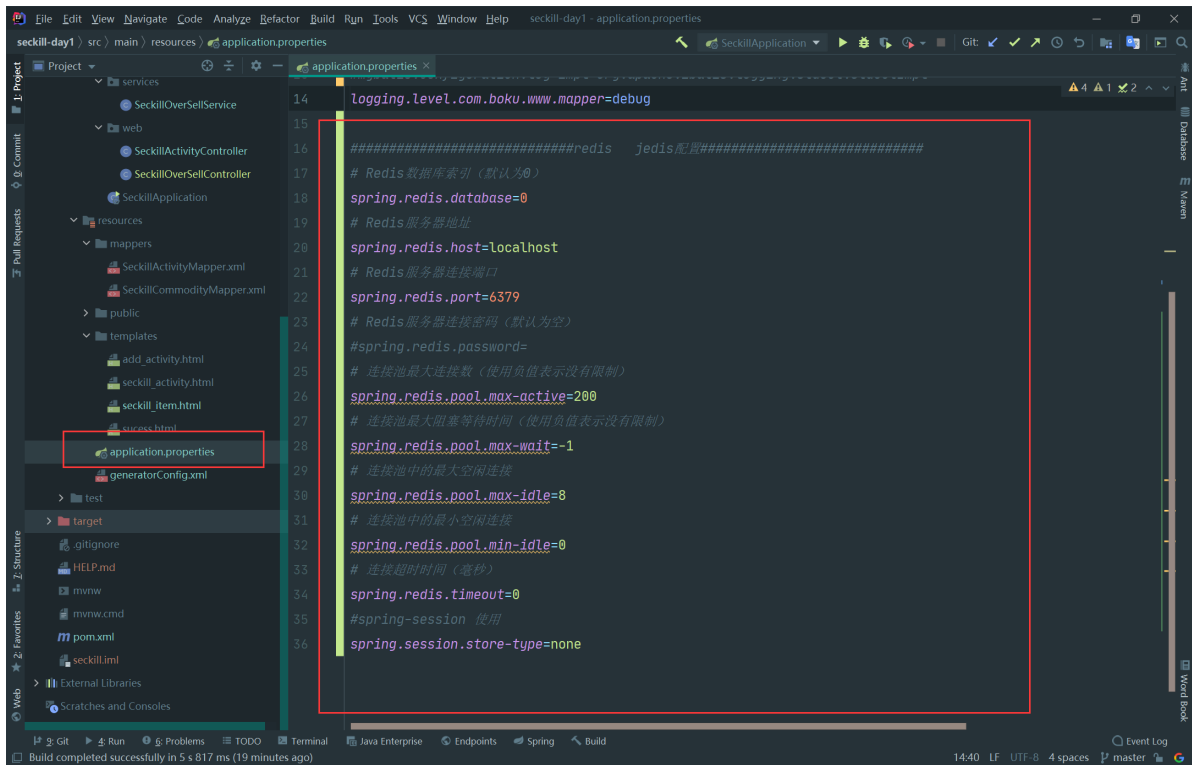
# 《保姆级教程》第五集 Jedis 的引入 & Redis 工具类的开发

## 1. 引入 Jedis 依赖，Jedis —— java 操作 Redis 的工具



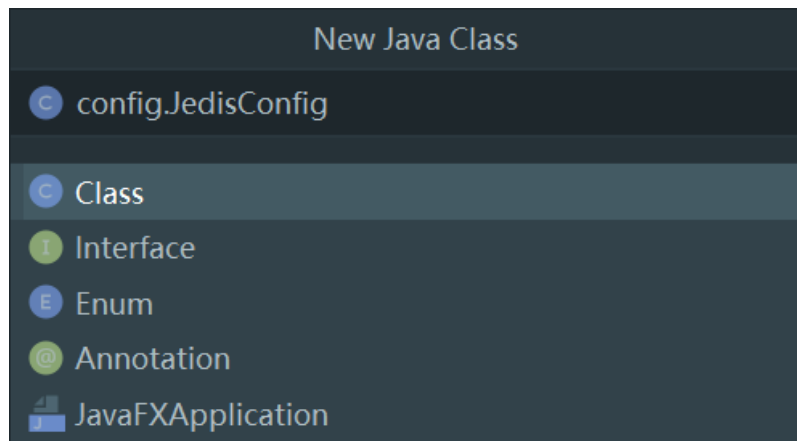
```
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
  <version>2.8.2</version>
</dependency>
```

## 2. application.properties 配置文件中配置 Redis

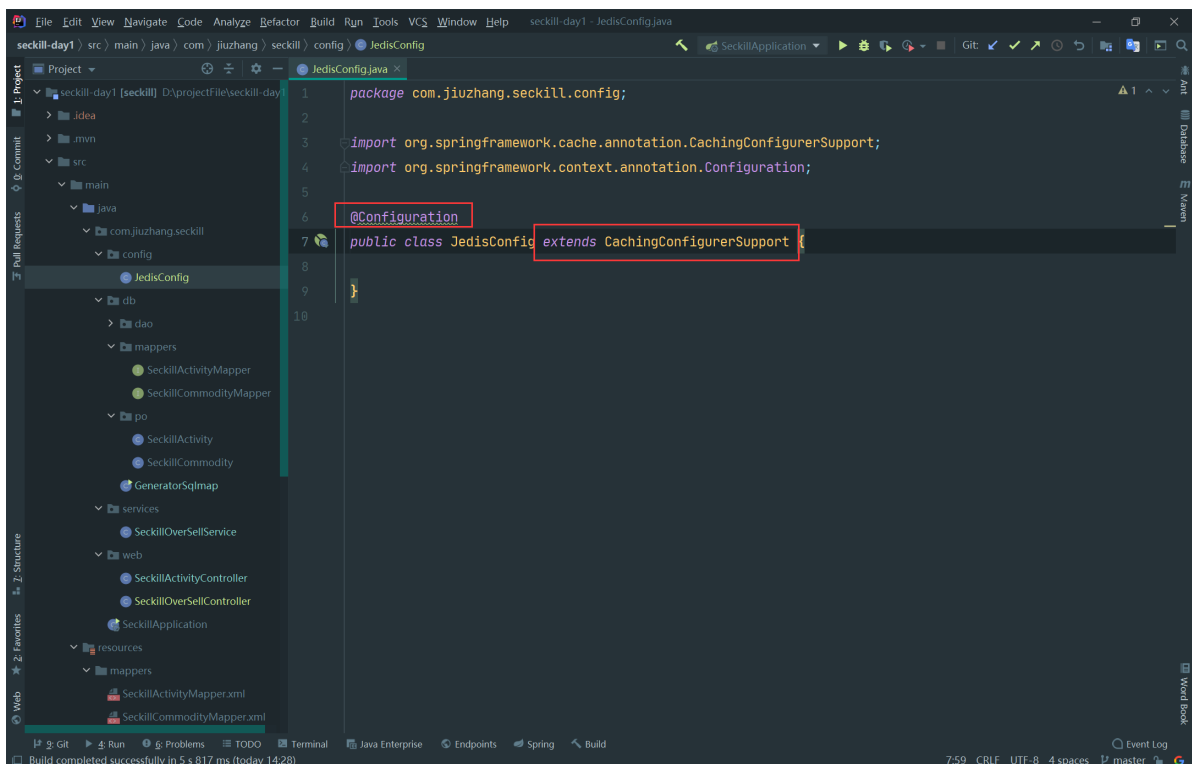


```
#####redis    jedis配置#####  
# Redis数据库索引（默认为0）  
spring.redis.database=0  
# Redis服务器地址  
spring.redis.host=localhost  
# Redis服务器连接端口  
spring.redis.port=6379  
# Redis服务器连接密码（默认为空）  
#spring.redis.password=  
# 连接池最大连接数（使用负值表示没有限制）  
spring.redis.pool.max-active=200  
# 连接池最大阻塞等待时间（使用负值表示没有限制）  
spring.redis.pool.max-wait=-1  
# 连接池中的最大空闲连接  
spring.redis.pool.max-idle=8  
# 连接池中的最小空闲连接  
spring.redis.pool.min-idle=0  
# 连接超时时间（毫秒）  
spring.redis.timeout=0  
#spring-session 使用  
spring.session.store-type=none
```

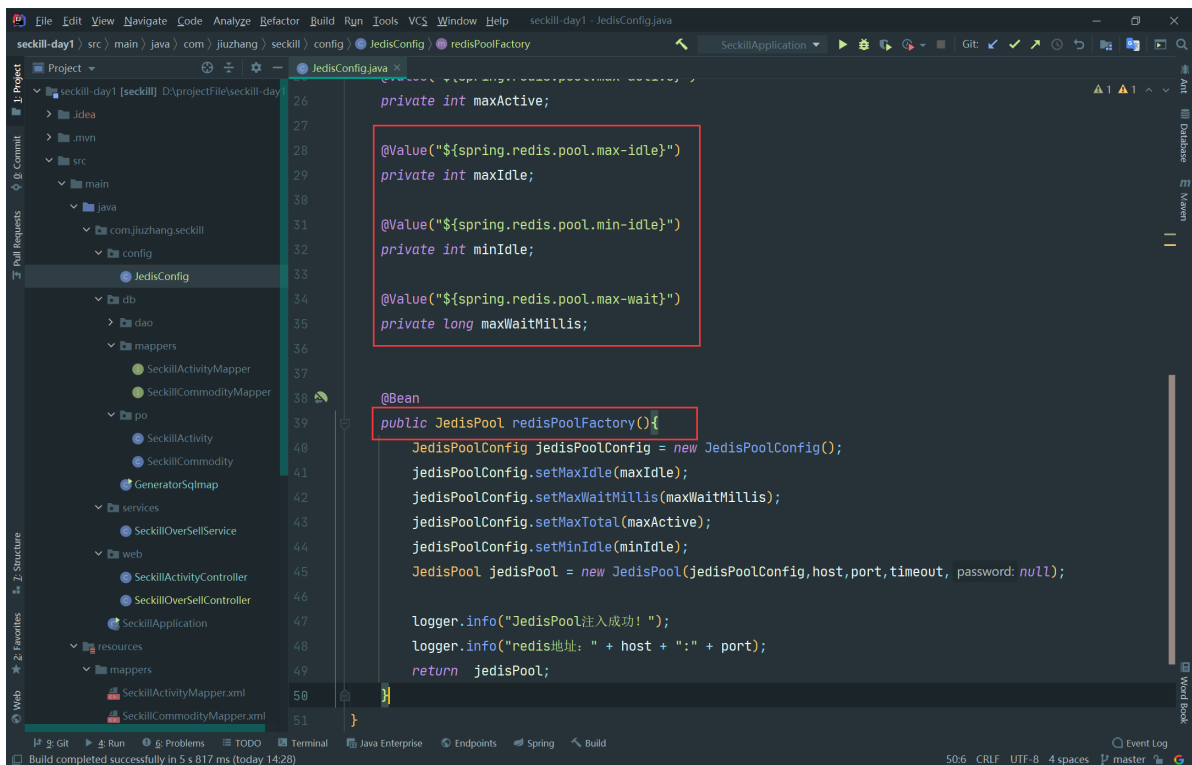
### 3. 创建 JedisConfig



## 4. 加上 @Configuration 注解 & 继承 CachingConfigurerSupport 类



## 5. 完成属性注入，向 Spring 容器注入 JedisPool



```
private Logger logger = LoggerFactory.getLogger(JedisConfig.class);

@Value("${spring.redis.host}")
private String host;

@Value("${spring.redis.port}")
private int port;

@Value("${spring.redis.timeout}")
private int timeout;

@Value("${spring.redis.pool.max-active}")
private int maxActive;

@Value("${spring.redis.pool.max-idle}")
private int maxIdle;

@Value("${spring.redis.pool.min-idle}")
private int minIdle;

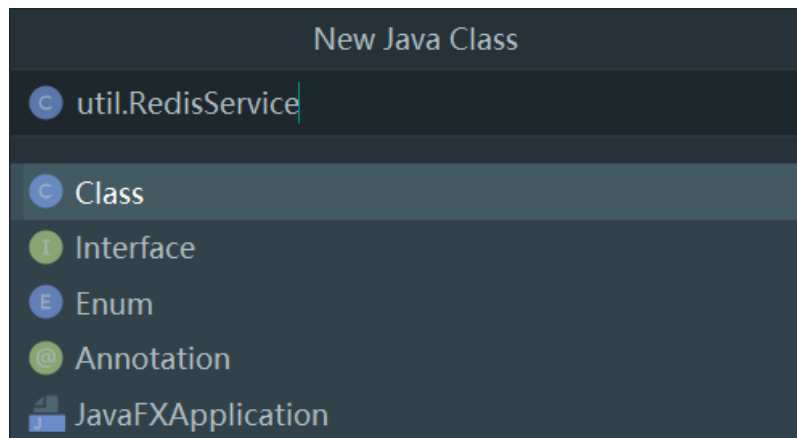
@Value("${spring.redis.pool.max-wait}")
private long maxWaitMillis;

@Bean
public JedisPool redisPoolFactory(){
    JedisPoolConfig jedisPoolConfig = new JedisPoolConfig();
    jedisPoolConfig.setMaxIdle(maxIdle);
    jedisPoolConfig.setMaxWaitMillis(maxWaitMillis);
    jedisPoolConfig.setMaxTotal(maxActive);
    jedisPoolConfig.setMinIdle(minIdle);
    JedisPool jedisPool = new JedisPool(jedisPoolConfig, host, port, timeout, null);

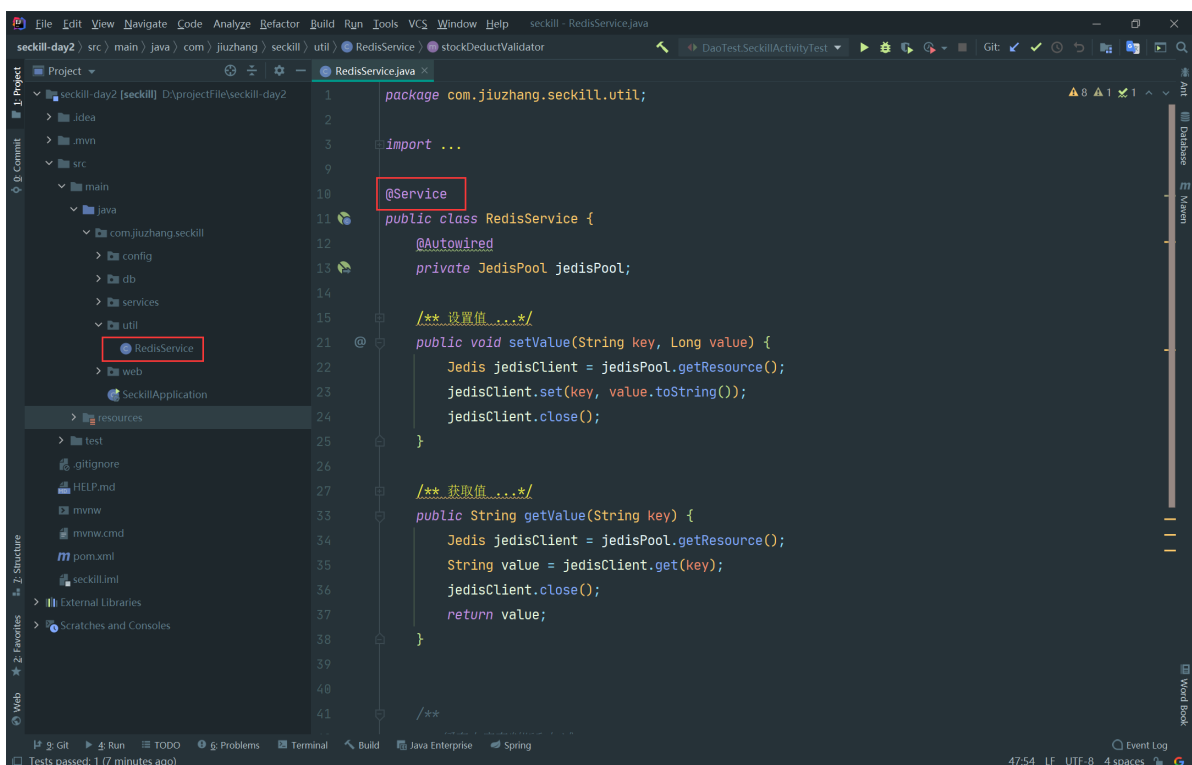
    logger.info("JedisPool注入成功! ");
    logger.info("redis地址: " + host + ":" + port);
}
```

```
    return jedisPool;
}
```

## 6. 创建 Redis 工具类 RedisService



## 7. 添加 @Service 注解 & 编写 添加值和获取值的方法



```
@Autowired
private JedisPool jedisPool;

/**
 * 设置值
 *
 * @param key
 * @param value
 */
public void setValue(String key, Long value) {
    Jedis jedisClient = jedisPool.getResource();
```

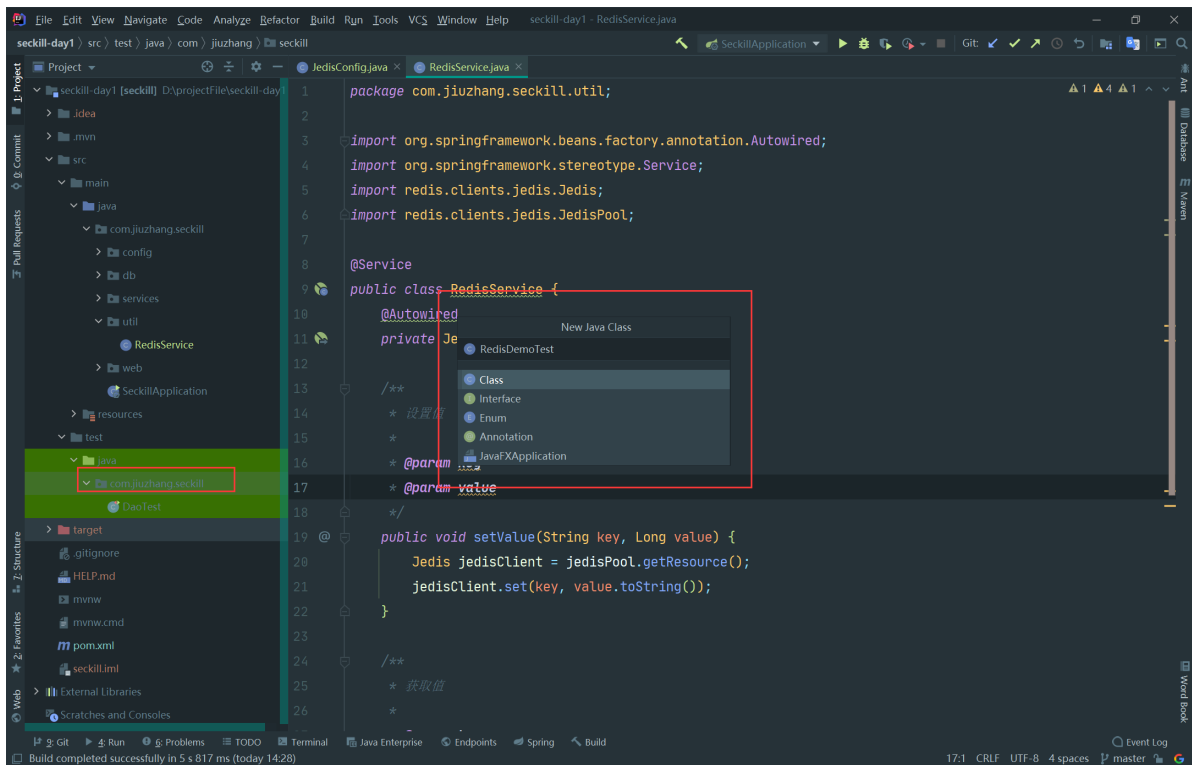
```

        jedisClient.set(key, value.toString());
        jedisClient.close();
    }

    /**
     * 获取值
     *
     * @param key
     * @return
     */
    public String getValue(String key) {
        Jedis jedisClient = jedisPool.getResource();
        String value = jedisClient.get(key);
        jedisClient.close();
        return value;
    }

```

## 8. 创建 Redis 测试类



## 9. 编写测试方法

```

@Resource
private RedisService redisService;

@Test
public void stockTest(){
    redisService.setValue("stock:19",10L);
}

@Test
public void getStockTest(){
    String stock = redisService.getValue("stock:19");
    System.out.println(stock);
}

```

}

## 10. 测试

