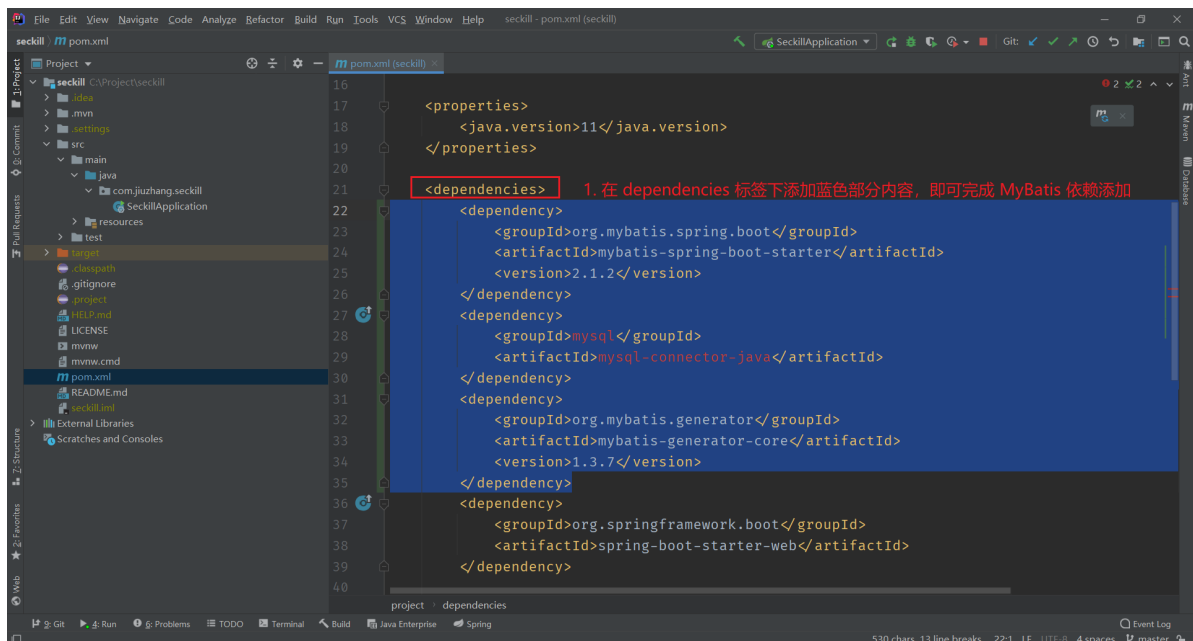


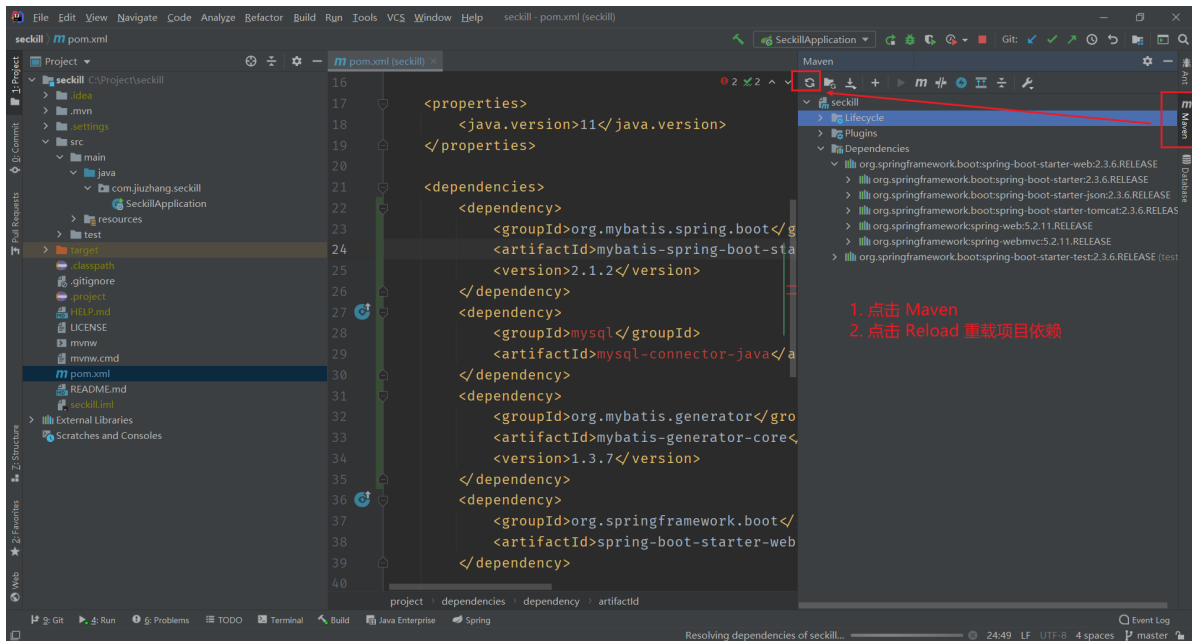
# 《保姆级教程》第二集：整合 MyBatis 及逆向生成

## 2.1 添加 MyBatis 相关依赖

修改 pom.xml 文件，在 `dependencies` 标签下添加这些内容：

```
<dependency>
  <groupId>org.mybatis.spring.boot</groupId>
  <artifactId>mybatis-spring-boot-starter</artifactId>
  <version>2.1.2</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
</dependency>
<dependency>
  <groupId>org.mybatis.generator</groupId>
  <artifactId>mybatis-generator-core</artifactId>
  <version>1.3.7</version>
</dependency>
```





## 2.2 搞定 MyBatis 逆向生成

首先复制两个文件到项目里：

### 2.2.1 复制 generatorConfig.xml

这个是 generatorConfig.xml 文件，复制到 resources 文件夹下，和 application.properties 同级。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN" "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">

<generatorConfiguration>
    <context id="testTables" targetRuntime="MyBatis3">
        <commentGenerator>
            <property name="suppressAllComments" value="true" />
        </commentGenerator>
        <jdbcConnection>
            <driverClass>com.mysql.cj.jdbc.Driver</driverClass>
            <connectionURL>jdbc:mysql://localhost:3306/seckill_jiuzhang?serverTimezone=GMT</connectionURL>
            <userId>root</userId>
            <password>root</password>
        </jdbcConnection>
        <javaTypeResolver>
            <property name="forceBigDecimals" value="false" />
        </javaTypeResolver>

        <!-- targetProject:生成PO类的位置 -->
        <javaModelGenerator targetPackage="com.jiuzhang.seckill.db.po"
            targetProject="./src/main/java">
            <!-- enableSubPackages:是否让schema作为包的后缀 -->
            <property name="enableSubPackages" value="true" />
            <!-- 从数据库返回的值被清理前后的空格 -->
            <property name="trimStrings" value="true" />
        </javaModelGenerator>
    </context>
</generatorConfiguration>
```

```

</javaModelGenerator>

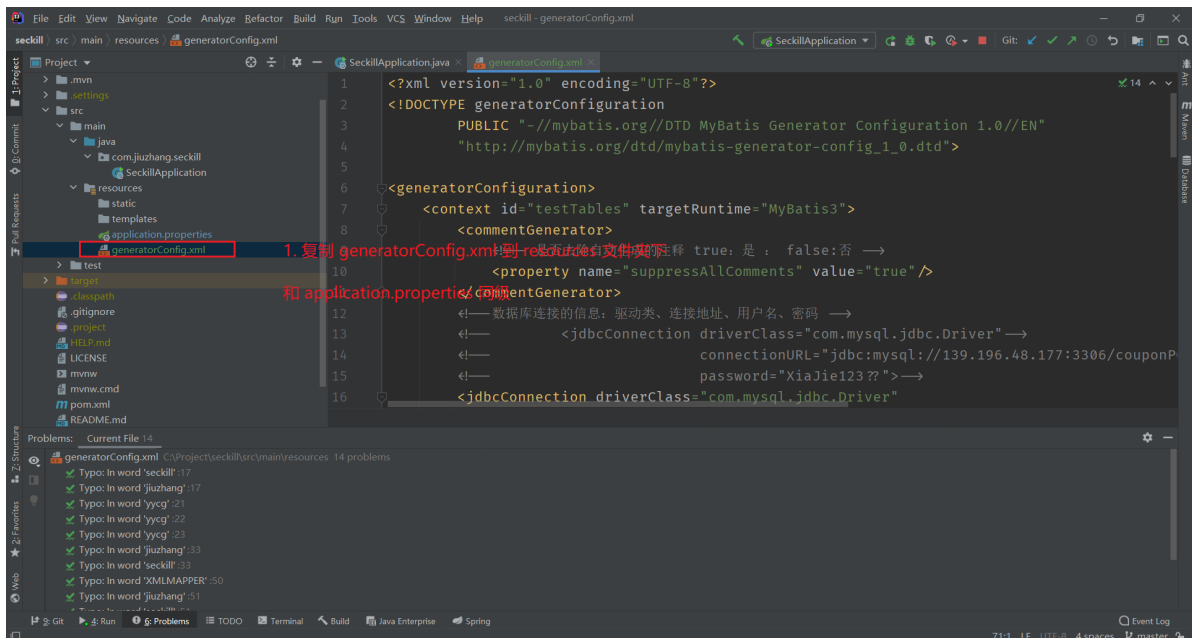
<!-- targetProject: mapper映射文件生成的位置 -->
<sqlMapGenerator targetPackage="mappers"
targetProject="./src/main/resources">
    <!-- enableSubPackages: 是否让schema作为包的后缀 -->
    <property name="enableSubPackages" value="true" />
</sqlMapGenerator>

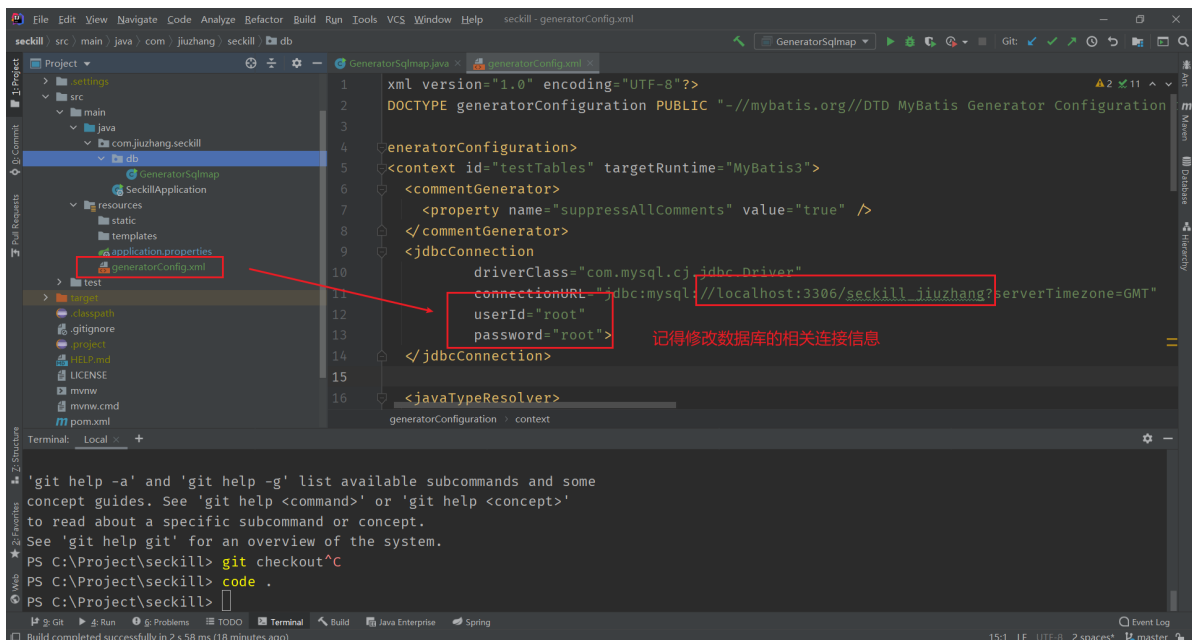
<!-- targetPackage: mapper接口生成的位置 -->
<javaClientGenerator type="XMLMAPPER"
targetPackage="com.jiuzhang.seckill.db.mappers" targetProject="./src/main/java">
    <!-- enableSubPackages: 是否让schema作为包的后缀 -->
    <property name="enableSubPackages" value="true" />
</javaClientGenerator>

<table schema="" tableName="seckill_activity"
domainObjectName="SeckillActivity" enableCountByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
enableUpdateByExample="false" selectByExampleQueryId="false"></table>
<table schema="" tableName="seckill_commodity"
domainObjectName="SeckillCommodity" enableCountByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
enableUpdateByExample="false" selectByExampleQueryId="false"></table>

</context>
</generatorConfiguration>

```





## 2.2.2 复制 GeneratorSqlmap.java

复制 GeneratorSqlmap.java 到 db/GeneratorSqlmap.java

```
package com.jiuzhang.seckill.db;

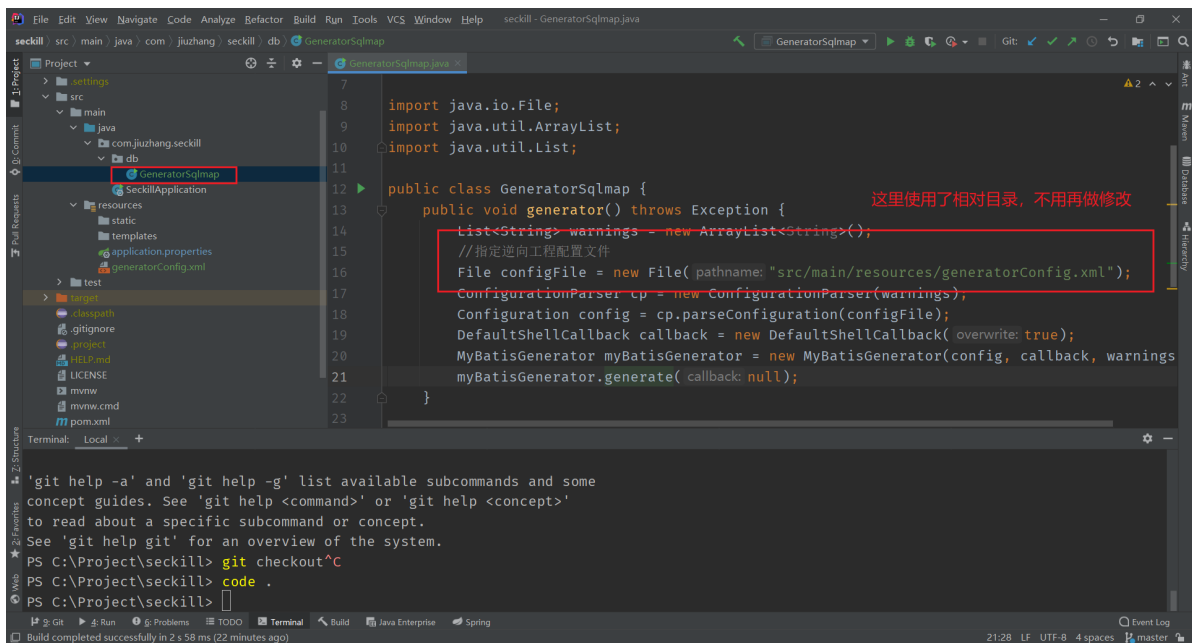
import org.mybatis.generator.api.MyBatisGenerator;
import org.mybatis.generator.config.Configuration;
import org.mybatis.generator.config.xml.ConfigurationParser;
import org.mybatis.generator.internal.DefaultShellCallback;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

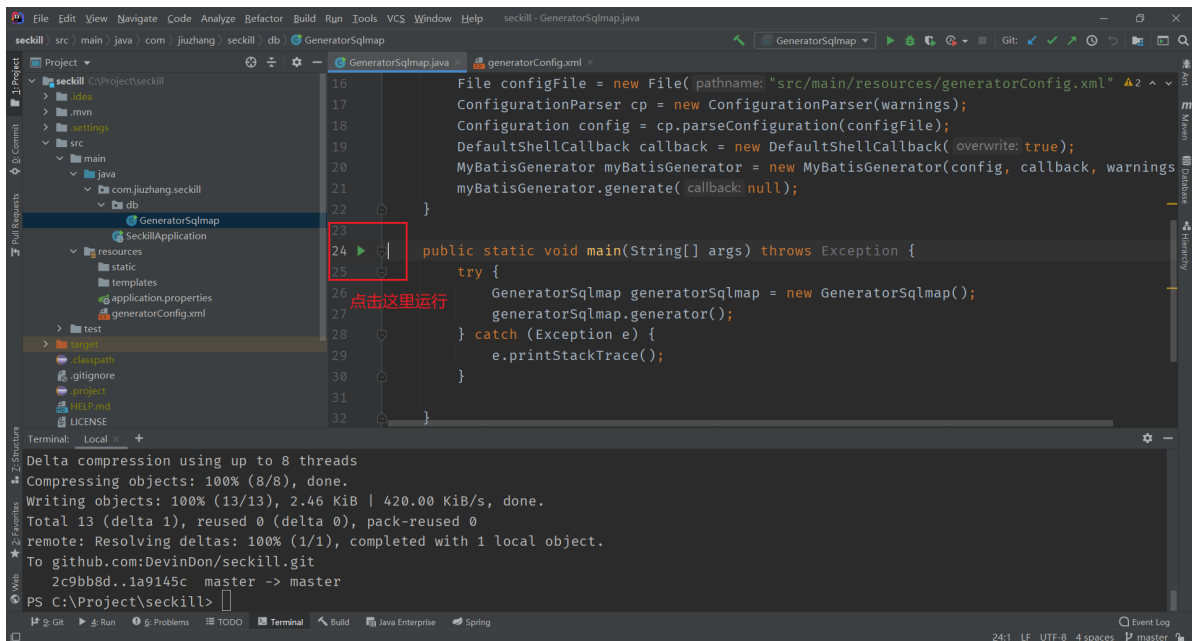
public class GeneratorSqlmap {

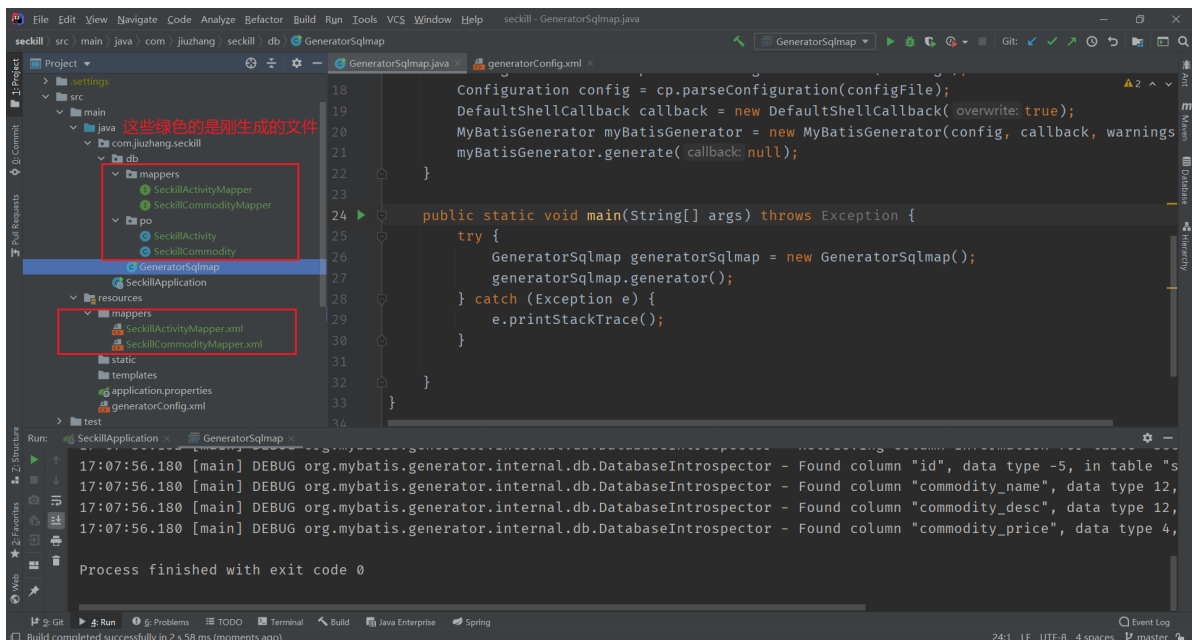
    public void generator() throws Exception {
        List<String> warnings = new ArrayList<String>();
        //指定逆向工程配置文件
        File configFile = new File("src/main/resources/generatorConfig.xml");
        ConfigurationParser cp = new ConfigurationParser(warnings);
        Configuration config = cp.parseConfiguration(configFile);
        DefaultShellCallback callback = new DefaultShellCallback(true);
        MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,
        callback, warnings);
        myBatisGenerator.generate(null);
    }

    public static void main(String[] args) throws Exception {
        try {
            GeneratorSqlmap generatorSqlmap = new GeneratorSqlmap();
            generatorSqlmap.generator();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



## 2.2.3 执行生成





## 2.2.4 配置启动类扫描

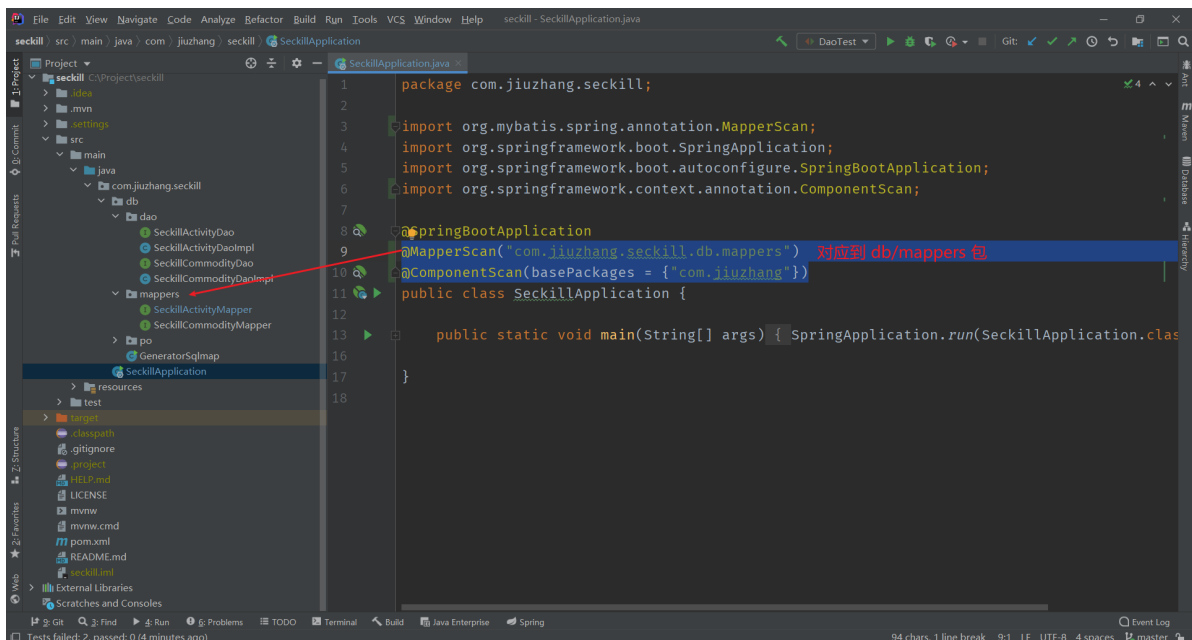
```
package com.jiuzhang.seckill;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@MapperScan("com.jiuzhang.seckill.db.mappers")
@ComponentScan(basePackages = {"com.jiuzhang"})
public class SeckillApplication {

    public static void main(String[] args) {
        SpringApplication.run(SeckillApplication.class, args);
    }

}
```



## 2.3 配置 application.properties

# Web 服务端口, 访问 `http://localhost:8080`

`server.port=8080`

# 数据源, 这里配置为我们的 MySQL 连接

`spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver`

`spring.datasource.url=jdbc:mysql://localhost:3306/seckill_jiuzhang?`

`serverTimezone=GMT&useUnicode=true&characterEncoding=utf8&useSSL=true`

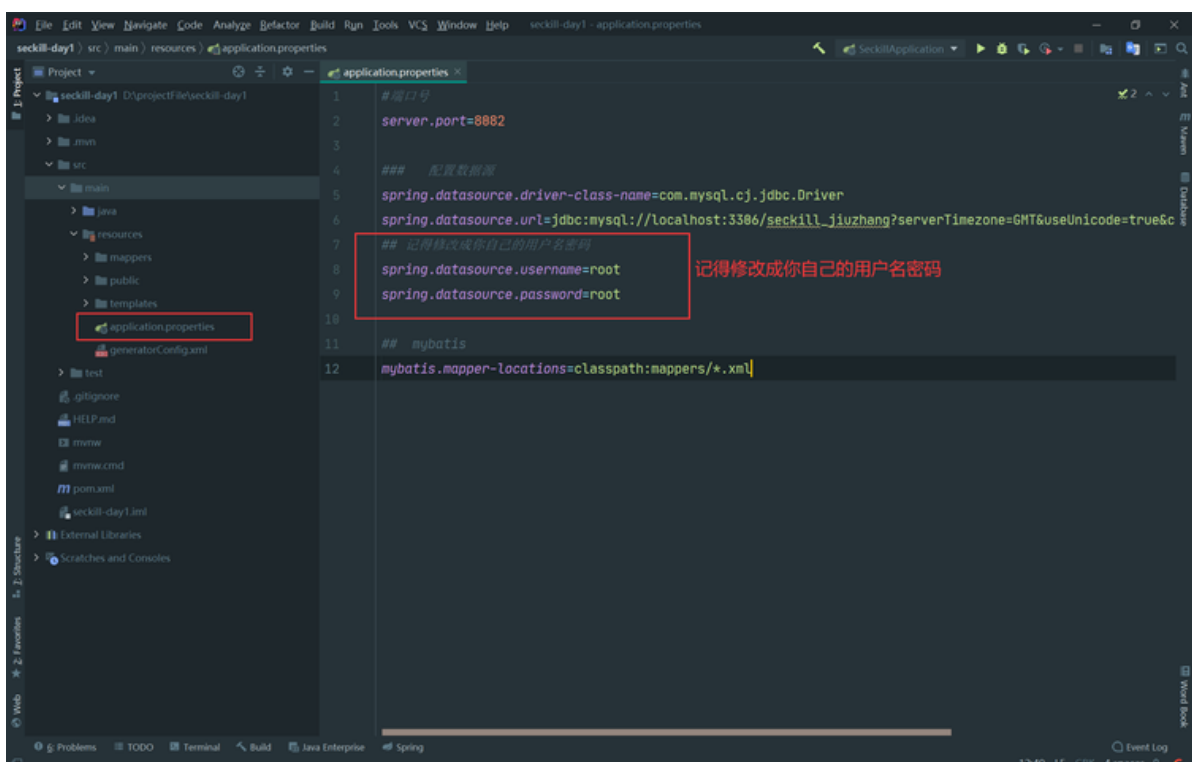
# 记得更改用户名和密码为本地所使用的

`spring.datasource.username=root`

`spring.datasource.password=root`

# MyBatis Mapper Config

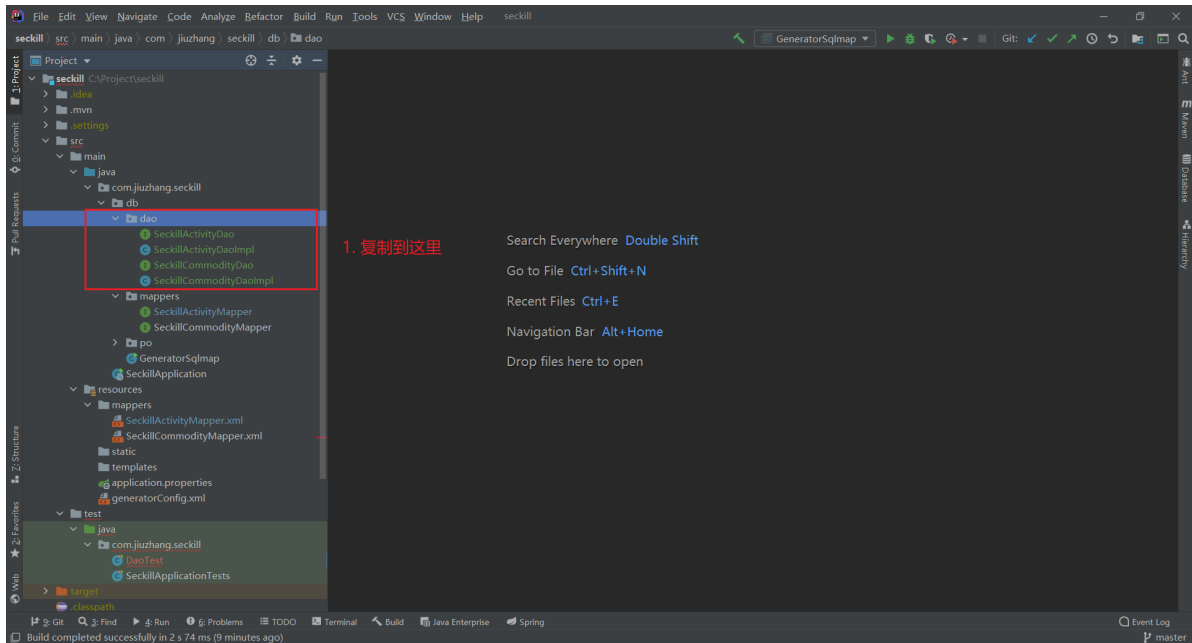
`mybatis.mapper-locations=classpath:mappers/*.xml`



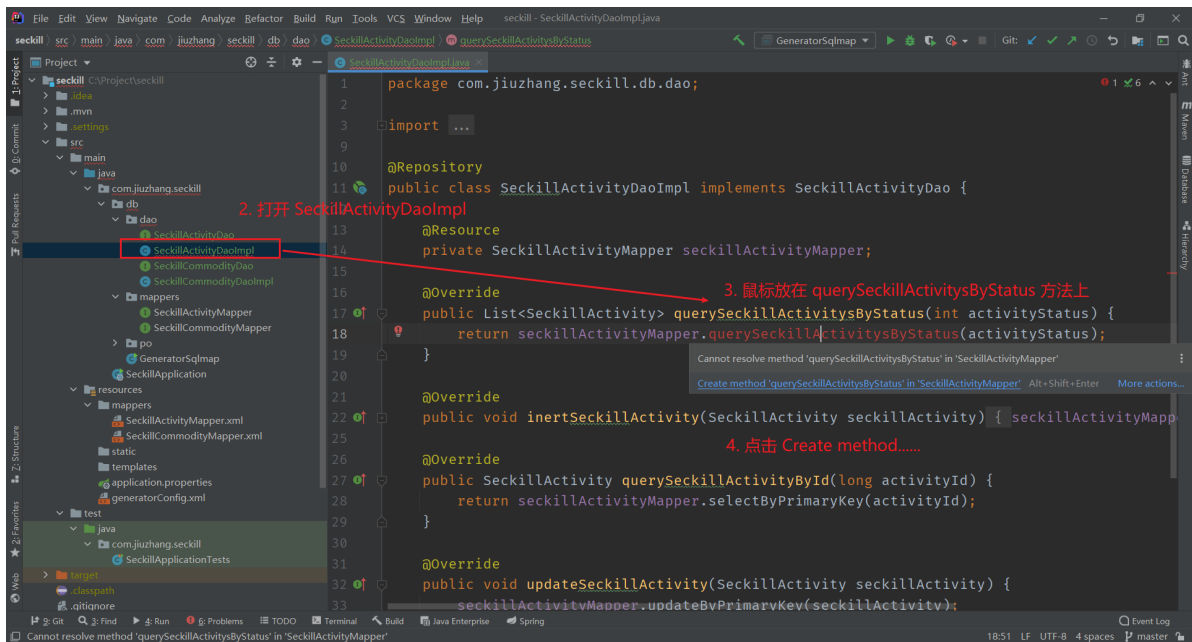
## 2.4 配置 DAO 数据访问对象

### 2.4.1 复制 db/dao 到项目里的 db/dao

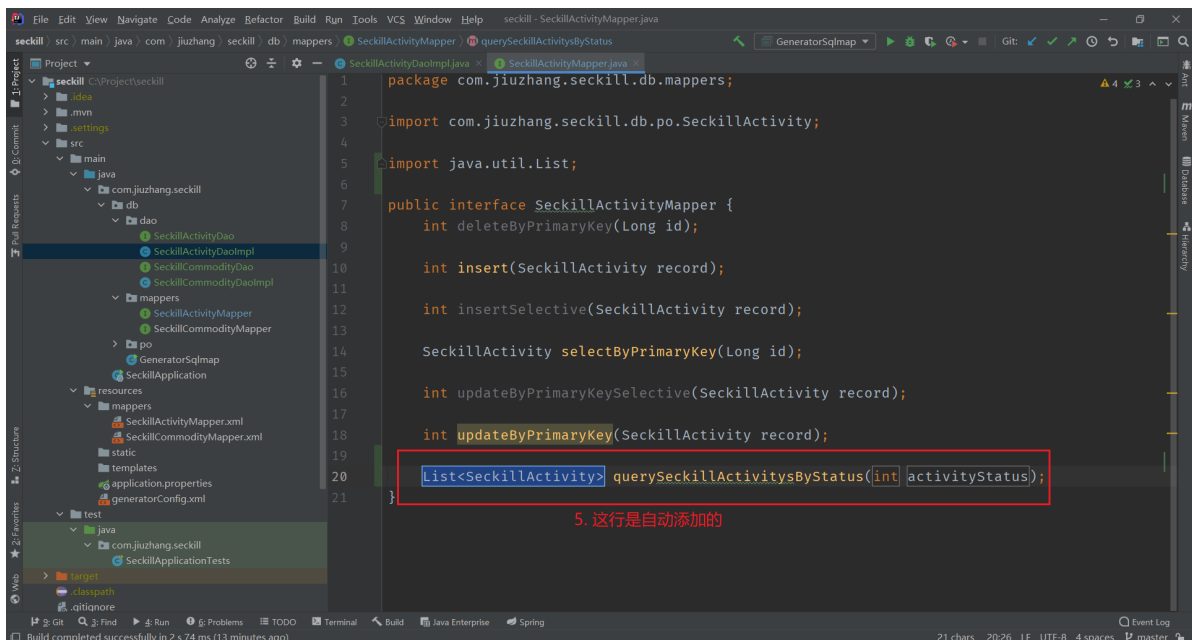
[点击这里下载 dao 文件夹](#)，解压后复制过去。



### 2.4.2 实现方法

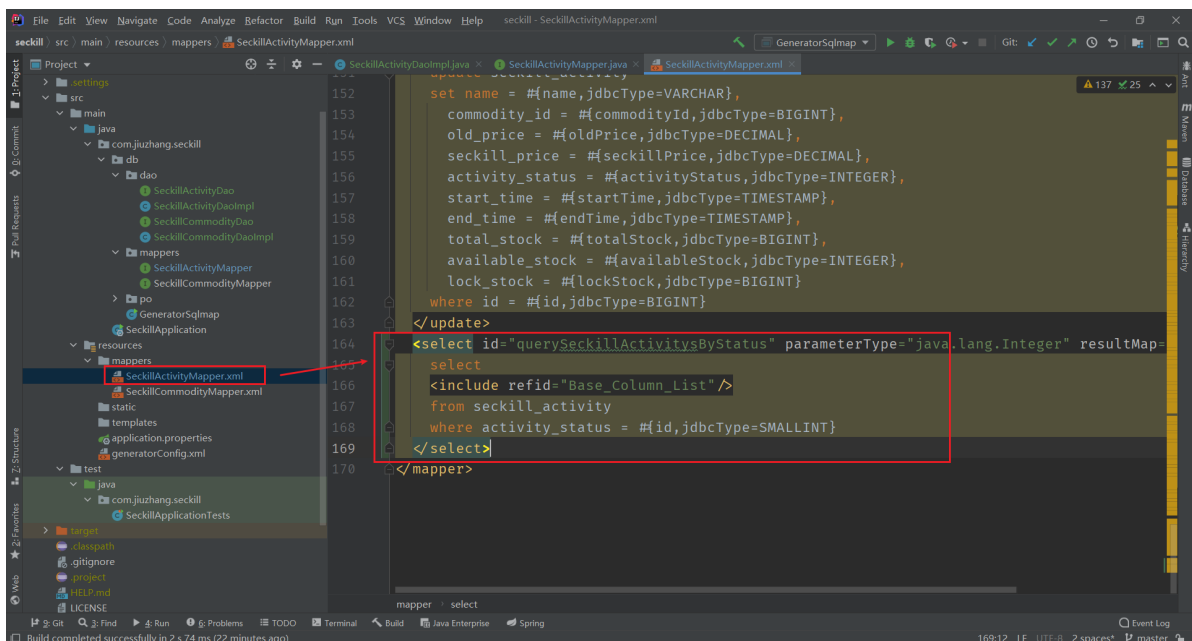






```
<select id="querySeckillActivitysByStatus" parameterType="java.lang.Integer"
resultMap="BaseResultMap">
    select
    <include refid="Base_Column_List"/>
    from seckill_activity
    where activity_status = #{id,jdbcType=SMALLINT}
</select>
```

复制上面的代码到 `resources/mappers/SeckillActivityMapper.xml` 的 `<Mapper>` 标签内。



## 2.5 添加测试类

添加 `DaoTest.java` 到 `test` 文件夹下。

```
package com.jiuzhang.seckill;

import com.jiuzhang.seckill.db.dao.SeckillActivityDao;
import com.jiuzhang.seckill.db.mappers.SeckillActivityMapper;
import com.jiuzhang.seckill.db.po.SeckillActivity;
```

```

import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import javax.annotation.Resource;
import java.math.BigDecimal;
import java.util.List;

@SpringBootTest
public class DaoTest {

    @Resource
    private SeckillActivityMapper seckillActivityMapper;

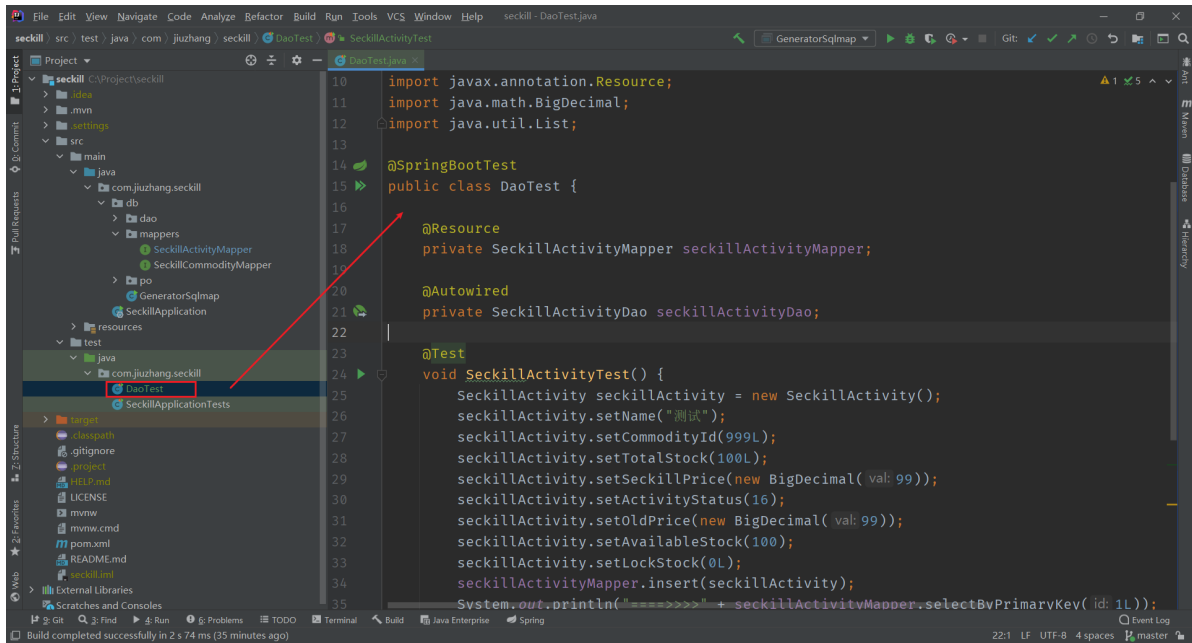
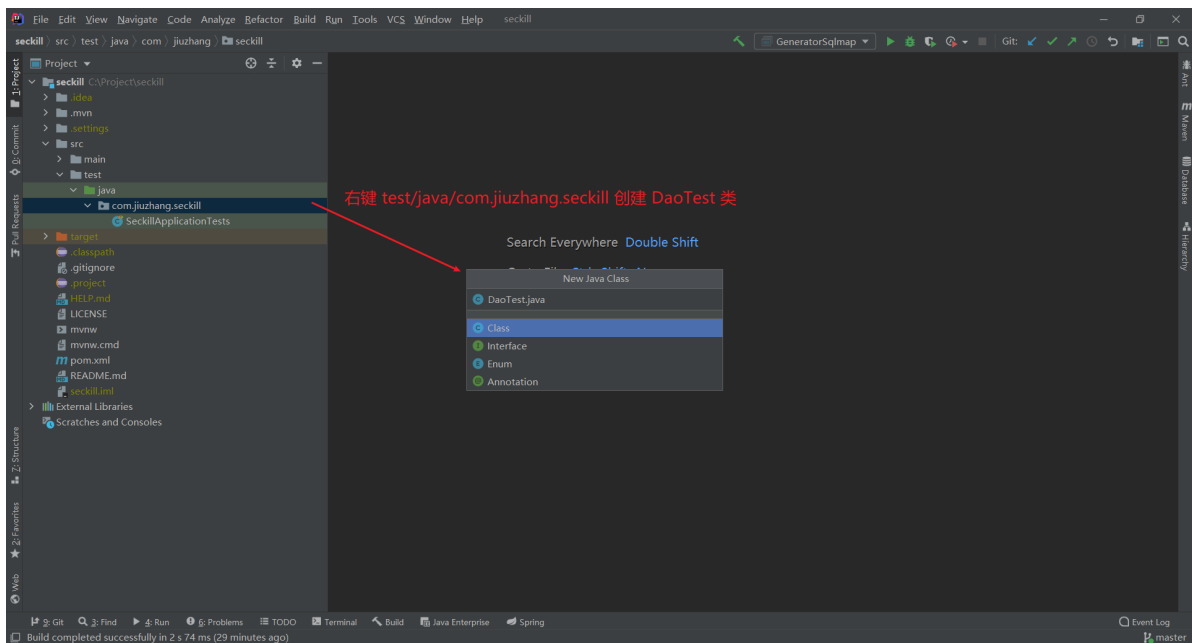
    @Autowired
    private SeckillActivityDao seckillActivityDao;

    @Test
    void seckillActivityTest() {
        SeckillActivity seckillActivity = new SeckillActivity();
        seckillActivity.setName("测试");
        seckillActivity.setCommodityId(999L);
        seckillActivity.setTotalStock(100L);
        seckillActivity.setSeckillPrice(new BigDecimal(99));
        seckillActivity.setActivityStatus(16);
        seckillActivity.setOldPrice(new BigDecimal(99));
        seckillActivity.setAvailablestock(100);
        seckillActivity.setLockStock(0L);
        seckillActivityMapper.insert(seckillActivity);
        System.out.println("====>>>>" +
seckillActivityMapper.selectByPrimaryKey(1L));
    }

    @Test
    void setSeckillActivityQuery(){
        List<SeckillActivity> seckillActivitys =
seckillActivityDao.querySeckillActivitysByStatus(0);
        System.out.println(seckillActivitys.size());
        seckillActivitys.stream().forEach(seckillActivity ->
System.out.println(seckillActivity.toString()));
    }

}

```



## 2.6 测试结果

