

一、MyBatis 代码自动生成

1、配置：

数据源 `jdbcConnection`

Mapper 文件路径:`sqlMapGenerator`

Mapper 接口文件路径:`javaClientGenerator`

实体类路径: `javaModelGenerator`

配置数据库表名: `table`

```
<?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="MysqlContext" targetRuntime="MyBatis3Simple"
defaultModelType="flat">
        <property name="beginningDelimiter" value="`"/>
        <property name="endingDelimiter" value="`"/>

        <!-- 通用 mapper 所在目录 -->
        <plugin type="tk.mybatis.mapper.generator.MapperPlugin">
            <property name="mappers" value="club.banyuan.my.mapper.MyMapper"/>
        </plugin>

        <jdbcConnection driverClass="com.mysql.cj.jdbc.Driver"

connectionURL="jdbc:mysql://localhost:3306/shoppingstreet?useSSL=false&server
Timezone=UTC"

                        userId="root"
                        password="rootroot">
        </jdbcConnection>

        <!-- 对应生成的 pojo 所在包 (实体类) -->
        <javaModelGenerator targetPackage="club.banyuan.entity"
targetProject="src/main/java"/>

        <!-- 对应生成的 mapper 所在目录 (xml) -->
```

```

        <sqlMapGenerator targetPackage="club.banyuan.mapper"
targetProject="src/main/resources"/>

        <!-- 配置 mapper 对应的 java 映射(接口) -->
        <javaClientGenerator targetPackage="club.banyuan.mapper"
targetProject="src/main/java" type="XMLMAPPER"/>

        <!-- 数据库表 -->
        <table tableName="News"></table>
        <!--         <table tableName="user"></table>-->
    </context>
</generatorConfiguration>

```

运行：GeneratorDisplay 中的 main 方法

2、应用：

复制实体类，mapper 接口、sqlMapper 以及所有 Mapper 的父接口 MyMapper 到新建工程中（注意：路径）

在新建工程引入：

```

<!--mapper-->
<dependency>
    <groupId>tk.mybatis</groupId>
    <artifactId>mapper-spring-boot-starter</artifactId>
    <version>1.2.4</version>
</dependency>

```

二、Mapper 接口使用

在 pom 文件中导入 SpringBoot 和 MyBatis 所需的 jar

导入 tk 的 MyBatis

导入测试用的 MyBatis-spring-boot-test 和 junit

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

```

```
<groupId>club.banyuan</groupId>
<artifactId>GenerateBatisTest</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>

<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.3.1.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
<properties>
  <java.version>1.8</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    <exclusions>
      <exclusion>
        <groupId>org.junit.vintage</groupId>
        <artifactId>junit-vintage-engine</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
```

```
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.20</version>
</dependency>
<!--mybatis-->
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>1.3.1</version>
</dependency>
<!--mapper-->
<dependency>
    <groupId>tk.mybatis</groupId>
    <artifactId>mapper-spring-boot-starter</artifactId>
    <version>1.2.4</version>
</dependency>
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter-test</artifactId>
    <version>1.3.2</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13</version>
    <scope>test</scope>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
```

```
</plugins>

</build>

</project>
```

配置 SpringBoot 和 MyBatis 所需的信息 resources/application.yml

```
mybatis:
  type-aliases-package: club.banyuan.entity
  mapper-locations: classpath:club/banyuan/mapper/*Mapper.xml
spring:
  datasource:
    driverClassName: com.mysql.cj.jdbc.Driver
    url:
jdbc:mysql://localhost:3306/shoppingstreet?&useSSL=false&serverTimezone=UTC
    username: root
    password: rootroot
```

在 test 路径下新建测试类，注意：测试类与启动类在同级目录下

```
package club.banyuan;

import club.banyuan.entity.News;
import club.banyuan.mapper.NewsMapper;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mybatis.spring.boot.test.autoconfigure.MybatisTest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.autoconfigure.jdbc.AutoConfigureTestDatabase;
import org.springframework.test.annotation.Rollback;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import java.util.List;

@RunWith(SpringJUnit4ClassRunner.class)
@MybatisTest
@AutoConfigureTestDatabase(replace = AutoConfigureTestDatabase.Replace.NONE)
//这个是启用自己配置的数据元，不加则采用虚拟数据源
```

```

@Rollback(false)    //这个是默认是回滚，不会 commit 入数据库，改成 false 则 commit
public class TestGenerateBatis {

    @Autowired
    private NewsMapper newsMapper;

    @Test
    public void testGetAll(){
        List<News> newsList = newsMapper.selectAll();
        for (News news : newsList) {
            System.out.println(news);
        }
    }
}

```

在启动类中，配置 mapper 接口的路径，注意使用的是 tk.mybatis 中的
@MapperScanner 注解

```

package club.banyuan;

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

@SpringBootApplication
@MapperScan(basePackages = "club.banyuan.mapper")
public class SpringBootApplication {

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

```