

# 代码生成插件

## 插件目的

日常开发过程中，有很多重复且无用的工作，往往会影响开发进度。例如，一个含有几十个字段的表，要为其创建JavaBean类就是一样非常繁琐的工作；编写Mybatis的Mapper文件要将字段名与类的属性一一对应，会看的眼花缭乱。本插件的目的就是将这些繁琐的工作自动化，从而减轻程序员的工作，让大家能够更加专注于业务逻辑的实现。

## 插件支持的Eclipse版本和项目结构

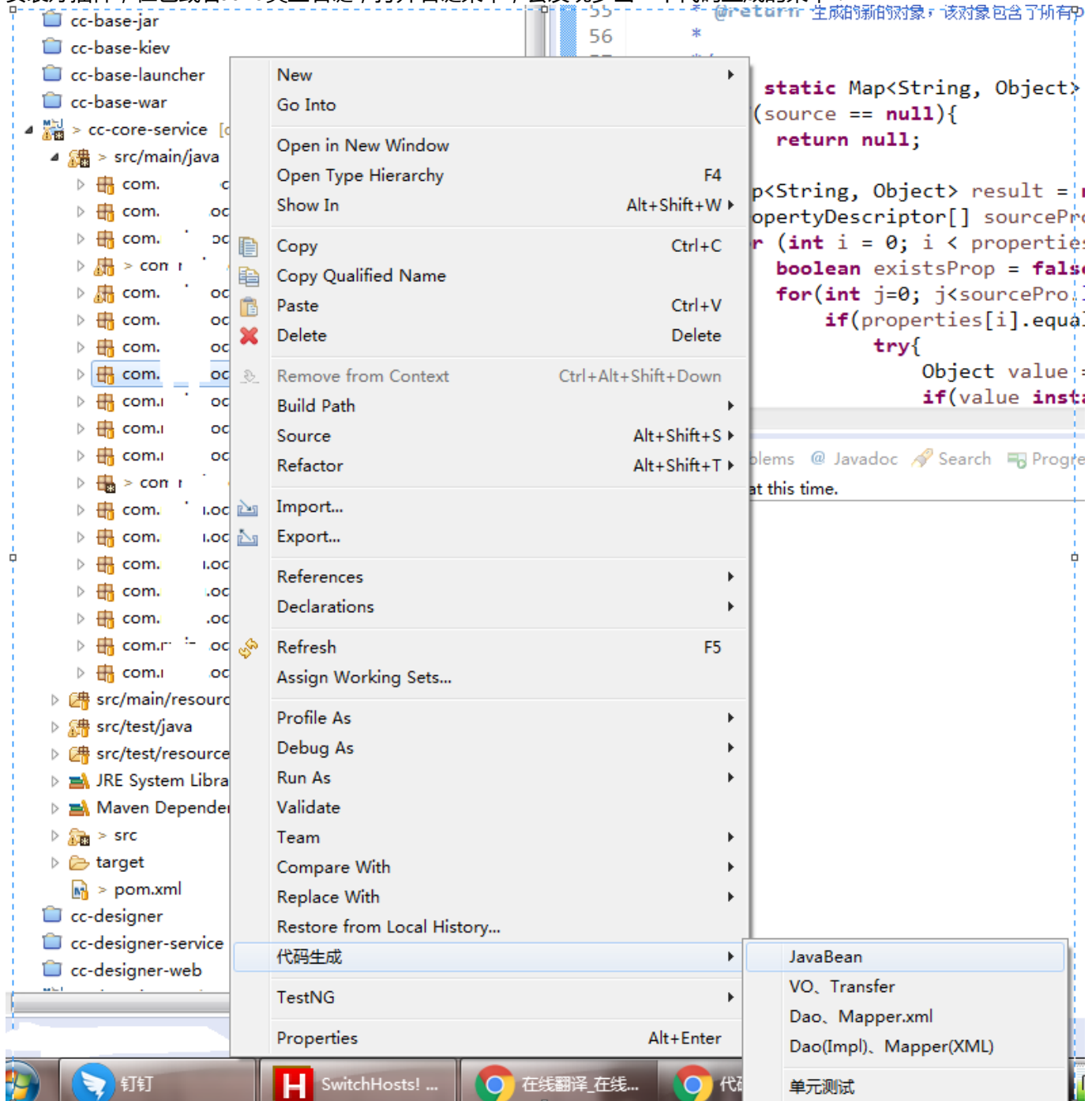
目前主流的Eclipse版本为Juno和Luna，插件能很好支持这两个Eclipse版本。在其他Eclipse版本中如果出现启动问题，可以尝试重新启动。插件适用于Spring MVC + mybatis为基本架构的项目。

## 插件的下载及安装

- 安装，将zip文件解压，并将解压获得的文件夹com.thirdparty.plugin.code.generate放到eclipse的plugins目录下，重启eclipse

## 插件的功能

- 安装好插件，在包或者Java类上右键，打开右键菜单，会发现多出一个代码生成的菜单：



- 支持生成JavaBean
- 支持生成Dao、DaoImpl、Mapper.java、Mapper.xml文件
- 支持生成单元测试文件
- 文件生成之后，会自动打开

## 插件的使用

### 准备工作

- 所有的业务代码（JavaBean、Dao、DaoImpl、Mapper.java、Mapper.xml）都是基于数据库表的，所以，首先数据表必须准备好。
- 配置好test4j.properties。这个文件大家都很熟悉了，为DaoImpl做单元测试的必备文件。该文件的位置在\${project\_dir}/src/test/resources/test4j.properties。一个示例如下：

```
stats.database.type=mysql
stats.database.url=jdbc:mysql://172.16.10.201:3306/MM_STORE_0909?characterEncoding=utf-8&cacheServerConfiguration=true&useLocalSession
stats.database.userName=mysqluser
stats.database.password=mysqluser
stats.database.schemaNames=MM_STORE_0909
```

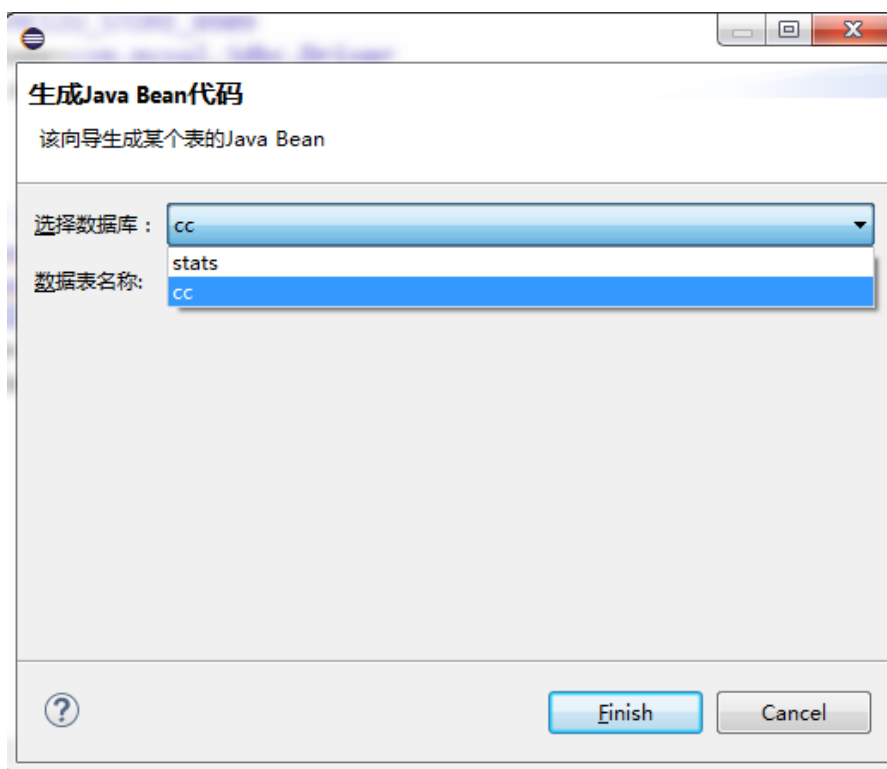
```
stats.database.driverClassName=com.mysql.jdbc.Driver
stats.database.only.testdb.allowing=false

cc.database.type=mysql
cc.database.url=jdbc:mysql://172.16.114.9:3306/MM_CC?characterEncoding=utf-8&cacheServerConfiguration=true&useLocalSessionState=true
cc.database.userName=mysqluser
cc.database.password=mysqluser
cc.database.schemaNames=MM_CC
cc.database.driverClassName=com.mysql.jdbc.Driver
cc.database.only.testdb.allowing=false
```

- test4j.properties中的数据库前缀，例如上面的stats、cc，将作为插件中选择数据库的标识。

## 生成JavaBean

- 选择要将JavaBean生成到哪个包下。
- 右键菜单中，选择【代码生成】-【JavaBean】，会弹出如下菜单



- 选择数据库，并填写数据表的名字，单击【确定】，那么对应的Java Bean会生成到当前包，并自动打开。
- 一个简单的生成实例如下（该例子只展示了几个字段，事实上该表有30多个字段）

```
package com.mm.ocean.cc.activity.dao;

import java.util.Date;

public class Activity {

    /**
     * 主键ID
     */
    private Integer id;

    /**
     * 活动名称
     */
    private String name;

    /**
     * 活动类型，九宫格（SUDOKU），刮刮乐（SCRATCH）
     */
    private String type;

    /**
     * MZOS版本
     */
}
```

```

private String mzos;

/**
 * 支持的客户端版本,
 */
private String supportVersions;

/**
 * 活动开始时间
 */
private Date startTime;

/**
 * 活动结束时间
 */
private Date endTime;

public Integer getId() {
    return id;
}

public void setId( Integer id ) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName( String name ) {
    this.name = name;
}

public String getType() {
    return type;
}

public void setType( String type ) {
    this.type = type;
}

public String getMzos() {
    return mzos;
}

public void setMzos( String mzos ) {
    this.mzos = mzos;
}

public String getSupportVersions() {
    return supportVersions;
}

public void setSupportVersions( String supportVersions ) {
    this.supportVersions = supportVersions;
}

public Date getStartTime() {
    return startTime;
}

public void setStartTime( Date startTime ) {
    this.startTime = startTime;
}

public Date getEndTime() {
    return endTime;
}

public void setEndTime( Date endTime ) {
    this.endTime = endTime;
}
}

```

- 在包下，同名文件已存在，如何处理

因为Java Bean往往与数据表一一对应，因此，如果有同名存在，会直接覆盖

- 选择的包错了，或者应该放在其他项目怎么办

可以直接拖动到正确的项目，或者包下即可。因为有的domain是单独的项目，而只存储domain的项目，是不会有单元测试的test4j.properties文件的。因此，可以在service项目下生成，然后手动拖到domain项目中。

## 生成VO、Transfer两个文件

生成VO与JavaBean类一致，Transfer是将VO与JavaBean进行转换。

## 生成Dao、Mapper.xml两个文件

- 有的项目组使用的是Dao + Mapper.xml的两个文件的形式，而不是Dao + DaoImpl + Mapper + Mapper.xml的形式。生成方式与后面的四个文件的形式相似，统一在下面叙述。

## 生成Dao(Impl)、Mapper(xml)四个文件

- 为什么要与JavaBean的生成分成两个菜单

因为JavaBean的包与这四个文件之间并不存在某种固定的规则。而这四个文件都需要引用到JavaBean的全名，因此，需要首先确定JavaBean的全路径。

- JavaBean由于拖动，其所在的包与原始生成的包不同，生成这四个文件时能自动引用吗

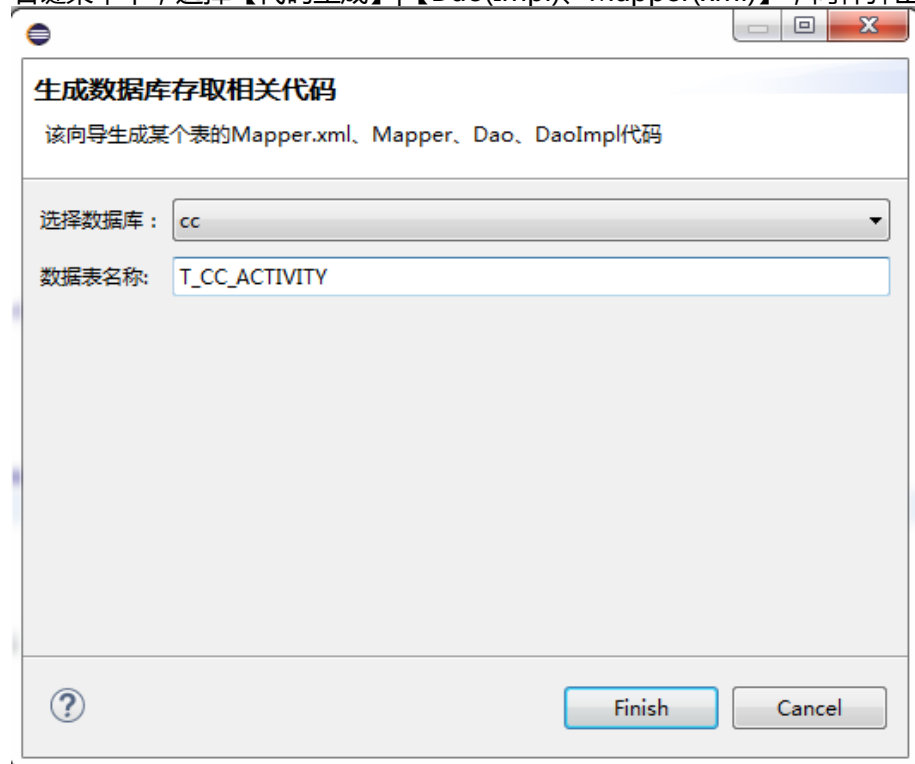
可以，只要JavaBean在当前项目，或者当前项目所引用的源码中。例如JavaBean原本在service项目中生成，将其拖动到domain项目中，service项目依赖了domain，那么生成这四个文件时，依然可以自动获取。但是如果JavaBean中的属性增加或减少，与原始数据表字段数不一致，就无法自动识别

- 选择代码存放的包

这个时候就要慎重了，选好代码存放的包。可以是dao、dao.impl、dao.impl.mysql中的任何一个，例如com.mm.ocean.cc.activity.dao、com.mm.ocean.cc.activity.dao.impl、com.mm.ocean.cc.activity.dao.impl.mysql；还可以是无dao路径，例如com.mm.ocean.cc.activity，此时，生成的文件均会放在com.mm.ocean.cc.activity.dao、com.mm.ocean.cc.activity.dao.impl、com.mm.ocean.cc.activity.dao.impl.mysql，同时Mapper.xml会放在src/main/resources对应的目录下。

- 选择数据库与数据表

右键菜单中，选择【代码生成】|【Dao(Impl)、Mapper(xml)】，同样弹出选择数据库和数据表对话框



- 生成的Dao代码示例

```
package com.mm.cc.activity.dao;

import com.mm.cc.domain.Activity;

public interface ActivityDao {
```

```

public Activity getById( Integer id );

public int addActivity( Activity activity );

public int updateById( Activity activity );

public int deleteById( Integer id );
}

```

## ■ 生成的DaoImpl代码示例

```

package com.mm.cc.activity.dao.impl;

import org.springframework.stereotype.Repository;
import org.springframework.beans.factory.annotation.Autowired;
import com.mm.cc.domain.Activity;
import com.mm.cc.activity.dao.ActivityDao;
import com.mm.cc.activity.dao.impl.mysql.ActivityMapper;

@Repository
public class ActivityDaoImpl implements ActivityDao {

    @Autowired
    private ActivityMapper activityMapper;

    @Override
    public Activity getById( Integer id ) {
        return activityMapper.getById( id );
    }

    @Override
    public int addActivity( Activity activity ) {
        return activityMapper.addActivity( activity );
    }

    @Override
    public int updateById( Activity activity ) {
        return activityMapper.updateById( activity );
    }

    @Override
    public int deleteById( Integer id ) {
        return activityMapper.deleteById( id );
    }
}

```

## ■ 生成的Mapper代码示例

```

package com.mm.cc.activity.dao.impl.mysql;

import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;
import com.mm.cc.domain.Activity;

@Repository
public interface ActivityMapper {

    public Activity getById( @Param( "id" ) Integer id );

    public int addActivity( @Param( "activity" ) Activity activity );

    public int updateById( @Param( "activity" ) Activity activity );

    public int deleteById( @Param( "id" ) Integer id );
}

```

## ■ 生成的Mapper.xml代码示例（注意，select语句会包含所有字段，insert语句包含除自增字段外的所有字段，update语句除主键外的所有字段）

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.mm.cc.activity.dao.impl.mysql.ActivityMapper">

    <select id="getById" resultType="com.mm.cc.domain.Activity">

```

```

SELECT A.FID id,
       A.FNAME name,
       A.FTYPE type,
       A.FMZOS mzos,
       A.FSUPPORT_VERSIONS supportVersions,
       A.FSTART_TIME startTime,
       A.FEND_TIME endTime,
       A.FICON icon,
       A.FMORE_CHANCE moreChance,
       A.FDECLARATION declaration,
       A.FBG_COLOR bgColor,
       A.FTOP_BG topBg,
       A.FTOP_COLOR topColor,
       A.FTITLE_VERTICAL_LINE titleVerticalLine,
       A.FPRIMARY_TITLE_COLOR primaryTitleColor,
       A.FSECONDARY_TITLE_COLOR secondaryTitleColor,
       A.FPARAGRAPH_LINE_COLOR paragraphLineColor,
       A.FSPLIT_LINE_COLOR splitLineColor,
       A.FSTAR_COLOR starColor,
       A.FSTAR_BG starBg,
       A.FDOWN_BUTTON_COLOR downButtonColor,
       A.FDOWN_PROGRESS_COLOR downProgressColor,
       A.FTEXT_COLOR textColor,
       A.FBG_MUSIC bgMusic,
       A.FACTION_MUSIC actionMusic,
       A.FEXTRA_CSS extraCss,
       A.FEARNED_POINTS_PER_DOWN earnedPointsPerDown,
       A.FEXPEND_POINTS_PER_LOTTERY expendPointsPerLottery,
       A.FINDEX_URL indexUrl,
       A.FCREATE_TIME createTime,
       A.FUPDATE_TIME updateTime,
       A.FRETURN_LOTTERY returnLottery,
       A.FFONT_BG fontBg,
       A.FTOP_ICON_COLOR topIconColor
FROM T_CC_ACTIVITY A
WHERE A.FID=#{id}
</select>

<insert id="addActivity" useGeneratedKeys="true" keyProperty="activity.id">
    INSERT INTO T_CC_ACTIVITY(
        FNAME, FTYPE, FMZOS, FSUPPORT_VERSIONS, FSTART_TIME,
        FEND_TIME, FICON, FMORE_CHANCE, FDECLARATION, FBG_COLOR,
        FTOP_BG, FTOP_COLOR, FTITLE_VERTICAL_LINE, FPRIMARY_TITLE_COLOR, FSECONDARY_TITLE_COLOR,
        FPARAGRAPH_LINE_COLOR, FSPLIT_LINE_COLOR, FSTAR_COLOR, FSTAR_BG, FDOWN_BUTTON_COLOR,
        FDOWN_PROGRESS_COLOR, FTEXT_COLOR, FBG_MUSIC, FACTION_MUSIC, FEXTRA_CSS,
        FEARNED_POINTS_PER_DOWN, FEXPEND_POINTS_PER_LOTTERY, FINDEX_URL, FCREATE_TIME, FUPDATE_TIME,
        FRETURN_LOTTERY, FFONT_BG, FTOP_ICON_COLOR)
    VALUES #{activity.name}, #{activity.type}, #{activity.mzos}, #{activity.supportVersions}, #{activity.startTime},
            #{activity.endTime}, #{activity.icon}, #{activity.moreChance}, #{activity.declaration}, #{activity.bgColor},
            #{activity.topBg}, #{activity.topColor}, #{activity.titleVerticalLine}, #{activity.primaryTitleColor}, #{activity.secondaryTitleColor},
            #{activity.paragraphLineColor}, #{activity.splitLineColor}, #{activity.starColor}, #{activity.starBg}, #{activity.downButtonColor},
            #{activity.downProgressColor}, #{activity.textColor}, #{activity.bgMusic}, #{activity.actionMusic}, #{activity.extraCss},
            #{activity.earnedPointsPerDown}, #{activity.expendPointsPerLottery}, #{activity.indexUrl}, #{activity.createTime}, #{activity.updateTime},
            #{activity.returnLottery}, #{activity.fontBg}, #{activity.topIconColor}
</insert>

<update id="updateById">
    UPDATE T_CC_ACTIVITY SET FNAME=#{activity.name}, FTYPE=#{activity.type}, FMZOS=#{activity.mzos}, FSUPPORT_VERSIONS=#{activity.supportVersions},
        FEND_TIME=#{activity.endTime}, FICON=#{activity.icon}, FMORE_CHANCE=#{activity.moreChance}, FDECLARATION=#{activity.declaration},
        FTOP_BG=#{activity.topBg}, FTOP_COLOR=#{activity.topColor}, FTITLE_VERTICAL_LINE=#{activity.titleVerticalLine}, FPRIMARY_TITLE_COLOR=#{activity.primaryTitleColor},
        FSECONDARY_TITLE_COLOR=#{activity.secondaryTitleColor}, FPARAGRAPH_LINE_COLOR=#{activity.paragraphLineColor}, FSPLIT_LINE_COLOR=#{activity.splitLineColor},
        FSTAR_COLOR=#{activity.starColor}, FSTAR_BG=#{activity.starBg}, FDOWN_BUTTON_COLOR=#{activity.downButtonColor},
        FDOWN_PROGRESS_COLOR=#{activity.downProgressColor}, FTEXT_COLOR=#{activity.textColor}, FBG_MUSIC=#{activity.bgMusic}, FACTION_MUSIC=#{activity.actionMusic},
        FEXTRA_CSS=#{activity.extraCss}, FEARNED_POINTS_PER_DOWN=#{activity.earnedPointsPerDown}, FEXPEND_POINTS_PER_LOTTERY=#{activity.expendPointsPerLottery},
        FINDEX_URL=#{activity.indexUrl}, FCREATE_TIME=#{activity.createTime}, FUPDATE_TIME=#{activity.updateTime}, FRETURN_LOTTERY=#{activity.returnLottery},
        FFONT_BG=#{activity.fontBg}, FTOP_ICON_COLOR=#{activity.topIconColor}
    WHERE FID=#{activity.id}
</update>

<delete id="deleteById">
    DELETE FROM T_CC_ACTIVITY WHERE FID=#{id}
</delete>
</mapper>

```

## ■ 同名文件的处理

如果在同一目录下已经生成过四个文件中的任意一个，再次生成时，将会以xxxDao1.java、xxxDaoImpl1.java、xxxMapper1.java、xxxMapper1.xml为名的文件，而不会覆盖原有文件

## 生成单元测试

### ■ 支持生成DaoImpl、Service、ServiceImpl为后缀的java类

- 选择要生成单元测试的Java类，右键菜单中选择【代码生成】|【单元测试】将会生成对应的单元测试文件，并自动放至test/src/java下的同包名下
- 同名文件与Dao、DaoImpl的处理相同，不会覆盖
- 生成的单元测试文件，并非最终结果，可能还需要手动调整
- 其中Service的单元测试是按mock方式进行，而DaoImpl的单元测试，是真实连接数据库进行。如果有未导入类错误，可以alt+shift+o手动导入一下