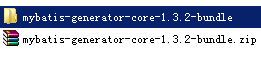
# mybatis逆向工程（MyBatis Generator）

## 什么是mybatis的逆向工程

 mybatis官方为了提高开发效率，提高自动对单表生成sql，包括 ：mapper.xml、mapper.java、表名.java(po类)



在企业开发中通常是在设计阶段对表进行设计 、创建。

在开发阶段根据表结构创建对应的po类。

mybatis逆向工程的方向：由数据库表----》java代码

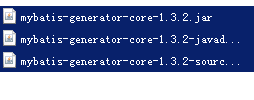
## 逆向工程 使用配置

运行逆向工程 方法：

本教程使用通过java程序运行逆向工程。

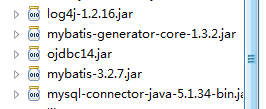


逆向工程 运行所需要的jar包：



数据库驱动包

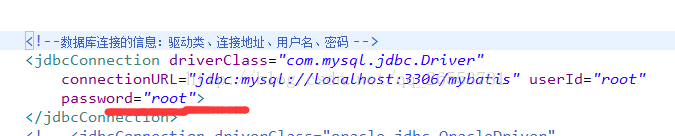
 总之：



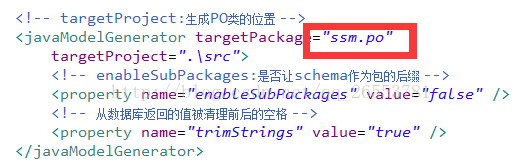
### generatorConfig.xml配置

需要使用用配置的地方：

1、连接数据库的地址和驱动



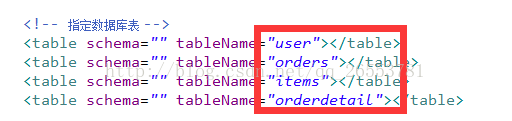
2、需要配置po类的包路径



3、需要配置mapper包的路径



4、指定数据表



### java程序

通过java程序生成mapper类、po类。。

public classGeneratorSqlmap {

public void generator() throws Exception{

List<String>warnings = newArrayList<String>();

boolean overwrite = true;

//指定逆向工程配置文件

FileconfigFile = newFile("generatorConfig.xml");

ConfigurationParsercp = newConfigurationParser(warnings);

Configurationconfig = cp.parseConfiguration(configFile);

DefaultShellCallbackcallback = newDefaultShellCallback(overwrite);

MyBatisGeneratormyBatisGenerator = new MyBatisGenerator(config,

callback,warnings);

myBatisGenerator.generate(null);

}

public static void main(String[] args) throws Exception {

try {

GeneratorSqlmapgeneratorSqlmap = new GeneratorSqlmap();

generatorSqlmap.generator();

}catch(Exception e) {

e.printStackTrace();

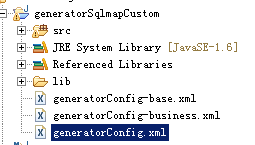
}

}

}

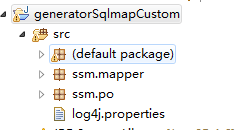
## 使用逆向工程生成的代码

### 第一步配置generatorConfig.xml

参考：

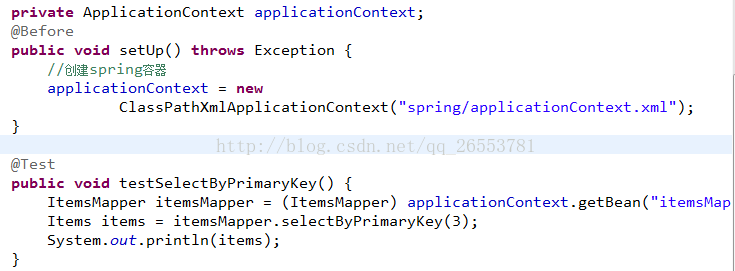
### 第二步配置执行java程序

执行java程序后，所生成的代码已经生成到 工程中。



### 第三步 将生成的代码拷贝到工程 中

和Spring整合后，看上一文章



# generatorConfig.xml

<?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>

<!-- 是否去除自动生成的注释 true：是 ： false:否 -->

<property name="suppressAllComments" value="true" />

</commentGenerator>

<!--数据库连接的信息：驱动类、连接地址、用户名、密码 -->

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

connectionURL="jdbc:mysql://localhost:3306/e3mall-32" userId="root"

password="root">

</jdbcConnection>

<!-- 默认false，把JDBC DECIMAL 和 NUMERIC 类型解析为 Integer，为 true时把JDBC DECIMAL 和

NUMERIC 类型解析为java.math.BigDecimal -->

<javaTypeResolver>

<property name="forceBigDecimals" value="false" />

</javaTypeResolver>

<!-- targetProject:生成PO类的位置 -->

<javaModelGenerator targetPackage="cn.e3mall.pojo"

targetProject=".\src">

<!-- enableSubPackages:是否让schema作为包的后缀 -->

<property name="enableSubPackages" value="false" />

<!-- 从数据库返回的值被清理前后的空格 -->

<property name="trimStrings" value="true" />

</javaModelGenerator>

<!-- targetProject:mapper映射文件生成的位置 -->

<sqlMapGenerator targetPackage="cn.e3mall.mapper"

targetProject=".\src">

<!-- enableSubPackages:是否让schema作为包的后缀 -->

<property name="enableSubPackages" value="false" />

</sqlMapGenerator>

<!-- targetPackage：mapper接口生成的位置 -->

<javaClientGenerator type="XMLMAPPER"

targetPackage="cn.e3mall.mapper"

targetProject=".\src">

<!-- enableSubPackages:是否让schema作为包的后缀 -->

<property name="enableSubPackages" value="false" />

</javaClientGenerator>

<!-- 指定数据库表 -->

<table schema="" tableName="tb\_content"></table>

<table schema="" tableName="tb\_content\_category"></table>

<table schema="" tableName="tb\_item"></table>

<table schema="" tableName="tb\_item\_cat"></table>

<table schema="" tableName="tb\_item\_desc"></table>

<table schema="" tableName="tb\_item\_param"></table>

<table schema="" tableName="tb\_item\_param\_item"></table>

<table schema="" tableName="tb\_order"></table>

<table schema="" tableName="tb\_order\_item"></table>

<table schema="" tableName="tb\_order\_shipping"></table>

<table schema="" tableName="tb\_user"></table>

</context>

</generatorConfiguration>

逆向工程文件：http://download.csdn.net/detail/qq\_26553781/9765707

# 介绍

## mapper接口中的方法解析



## example实例解析

mybatis的逆向工程中会生成实例及实例对应的example，example用于添加条件，相当where后面的部分

xxxExample example = new xxxExample();

Criteria criteria = new Example().createCriteria();



## 应用举例

### 查询

1. selectByPrimaryKey()

User user = XxxMapper.selectByPrimaryKey(100); //相当于select \* from user where id = 100

1. selectByExample() 和 selectByExampleWithBLOGs()

UserExample example = new UserExample();

Criteria criteria = example.createCriteria();

criteria.andUsernameEqualTo("wyw");

criteria.andUsernameIsNull();

example.setOrderByClause("username asc,email desc");

List<?>list = XxxMapper.selectByExample(example);

//相当于：select \* from user where username = 'wyw' and username is null order by username asc,email desc

注：在iBator逆向工程生成的文件XxxExample.java中包含一个static的内部类Criteria，Criteria中的方法是定义SQL 语句where后的查询条件。

### 插入数据

insert()

User user = new User();

user.setId("dsfgsdfgdsfgds");

user.setUsername("admin");

user.setPassword("admin")

user.setEmail("wyw@163.com");

XxxMapper.insert(user);

//相当于：insert into user(ID,username,password,email) values ('dsfgsdfgdsfgds','admin','admin','wyw@126.com');

### 更新数据

1. updateByPrimaryKey()

User user =new User();

user.setId("dsfgsdfgdsfgds");

user.setUsername("wyw");

user.setPassword("wyw");

user.setEmail("wyw@163.com");

XxxMapper.updateByPrimaryKey(user);

//相当于：update user set username='wyw', password='wyw', email='wyw@163.com' where id='dsfgsdfgdsfgds'

1. updateByPrimaryKeySelective()

User user = new User();

user.setId("dsfgsdfgdsfgds");

user.setPassword("wyw");

XxxMapper.updateByPrimaryKey(user);

//相当于：update user set password='wyw' where id='dsfgsdfgdsfgds'

1. updateByExample() 和 updateByExampleSelective()

UserExample example = new UserExample();

Criteria criteria = example.createCriteria();

criteria.andUsernameEqualTo("admin");

User user = new User();

user.setPassword("wyw");

XxxMapper.updateByPrimaryKeySelective(user,example);

//相当于：update user set password='wyw' where username='admin'

updateByExample()更新所有的字段，包括字段为null的也更新，建议使用 updateByExampleSelective()更新想更新的字段

### 删除数据

1. deleteByPrimaryKey()

XxxMapper.deleteByPrimaryKey(1); //相当于：delete from user where id=1

1. deleteByExample()

UserExample example = new UserExample();

Criteria criteria = example.createCriteria();

criteria.andUsernameEqualTo("admin");

XxxMapper.deleteByExample(example);

//相当于：delete from user where username='admin'

### 查询数据数量

countByExample()

UserExample example = new UserExample();

Criteria criteria = example.createCriteria();

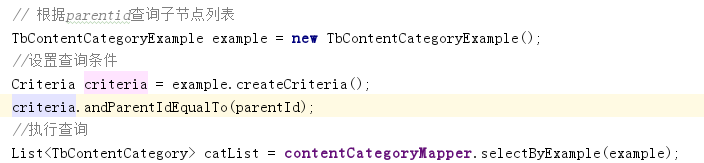
criteria.andUsernameEqualTo("wyw");

int count = XxxMapper.countByExample(example);

//相当于：select count(\*) from user where username='wyw'

# 实例

## 条件查询



## 插入

去当前事务最后生成的主键

