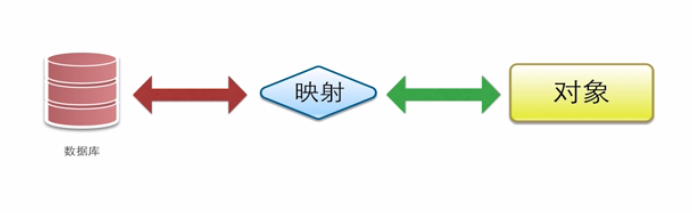
基于Mybatis实现Dao

1. Mybatis用来做什么？



提供数据库和实体之间的映射。

1. Mybatis特点

Mybatis和Hibernate的一个显著区别就是，需要自己写sql

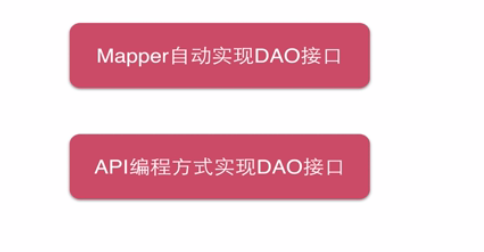


1. Mybatis怎么用
2. sql写在哪里？



推荐这种方式

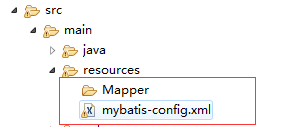
1. 如何实现Dao接口



这种方式不需要写实现类（推荐）

基于Mybatis实现DAO编程

1. 创建一个mybatis-config.xml和一个Mapper目录



1. 配置mybatis-config.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE configuration  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">    <configuration>  <!-- 配置全局属性 -->  <settings>  <!-- 使用jdbc的getGeneratedKeys获取数据库自增主键值 -->  <setting name=*"useGeneratedKeys"* value=*"true"*/>  <!-- 使用列别名替换列名 默认为true  select name as title from table  -->  <setting name=*"useColumnLabel"* value=*"true"*/>  <!-- 开启驼峰命名转换：Table(create\_time) ->Entity(createTime) -->  <setting name=*"mapUnderscoreToCamelCase"* value=*"true"*/>  </settings>  </configuration> |

其中，数据库的连接配置，没有在这里配置，一般交给Spring配置

1. 配置有关mapper

|  |
| --- |
| <?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=*"edu.hubu.seckill.dao.SeckillDao"*>  <!-- 目的：为dao接口方法提供sql语句配置 -->    <update id=*"reduceNumber"*>  <!-- 具体的sql语句 -->  update  seckill  set  number = number -1  where seckill\_id = #{seckillId}  and start\_time <![CDATA[ <= ]]> #{killTime}  and end\_time <![CDATA[ >= ]]> #{killTime}  and number > 0;  </update>    <select id=*"queryById"* parameterType=*"long"* resultType=*"Seckill"*>  select seckill\_id,name,start\_time,end\_time,create\_time  from seckill  where seckill\_id = #{seckillId}  </select>    <select id=*"queryAll"* resultType=*"Seckill"*>  select seckill\_id,name,start\_time,end\_time,create\_time  from seckill  order by cereat\_time desc  limit #{offest},#{limit}  </select>  </mapper> |

这里 resultMap标签没有配，这个标签实际上就是将数据库中的字段与实体的属性相对应，但是，因为在前面的mybatis-config.xml中已经配置好了别名已经驼峰命名。这里就没有配置的必要了，如果有需要也可以配置这个标签

示例：

|  |
| --- |
| <resultMap type="UserAlias" id="UserResult">  <id column="id" jdbcType="INTEGER" property="id"/>  <result column="username" jdbcType="VARCHAR" property="username"/>  <result column="password" jdbcType="VARCHAR" property="password.encrypted"/>  <result column="administrator" jdbcType="BOOLEAN" property="administrator"/>  </resultMap> |

|  |
| --- |
| <?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=*"edu.hubu.seckill.dao.SeckillSuccessDao"*>    <insert id=*"insertSuccessKill"*>  <!-- 主键冲突，报错,处理方法（insert ignore into） -->  insert ignore into success\_killed(seckill\_id,user\_phone,state)  values (#{seckillId},#{userPhone},0)  </insert>    <select id=*"queryByIdWithSeckill"* resultType=*"SeckillSuccess"*>  <!-- 根据id查询SuccessKilled并携带Seckill实体 -->  <!-- 如何告诉Mybatis把结果映射到SeckillSuccess同时映射seckill属性 -->  <!-- 可以自由控制SQL -->  select  sk.seckill\_id,  sk.user\_phone,  sk.create\_time,  sk.state,  s.seckill\_id "seckill.seckill\_id",  s.name "seckill.name",  s.number "seckill.number",  s.start\_time "seckill.start\_time",  s.end\_time "seckill.end\_time",  s.create\_time "seckill.create\_time"  from success\_killed sk  inner join seckill s on sk.seckill\_id = s.seckill\_id  where sk.seckill\_id = #{seckillId}  and user\_phone = #{userPhone}    </select>  </mapper> |

这里，在select方法中，因为，查询映射到实体的时候，在实体属性中包含了另外一个 实体。采用的方法是，使用inner join方法，先将两个表中的数据都查询出来，然后通

过在字段名后面的“ ”内容中告诉mybatis这是实体中某个实体属性中的属性(例如：s.end\_time "seckill.end\_time")