### 一、项目搭建

关于项目搭建，小宝鸽以前写过一篇[Spirng+SpringMVC+Maven+Mybatis+MySQL项目搭建](http://blog.csdn.net/u013142781/article/details/50380920)，这篇文章提供了详细的搭建过程，而且提供了源码下载，接下来的将在这个源码的基础上继续开发。所以建议各位猿友可以把猿友下载一下。

### 二、分页插件的介绍

博主采用的插件是PageHelper这个插件，使用起来十分方便。该插件支持以下数据库：

Oracle

Mysql

MariaDB

SQLite

Hsqldb

PostgreSQL

DB2

SqlServer(2005+)

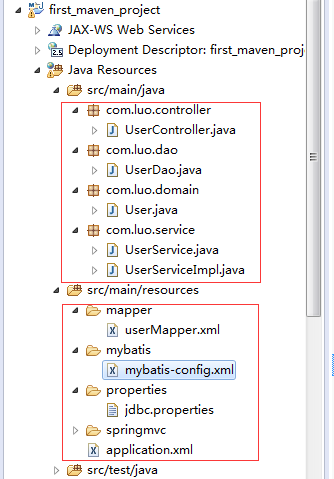
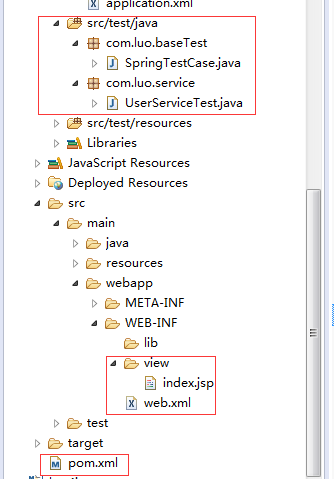
Informix

H2

对应于github的项目地址：   
<https://github.com/pagehelper/Mybatis-PageHelper>。   
关于使用建议猿友们可以详细阅读下这里的介绍：   
<http://git.oschina.net/free/Mybatis_PageHelper/blob/master/wikis/HowToUse.markdown>   
读一遍之后可能还不能完全懂怎么使用，可以先看完博主的这篇文章，然后慢慢再消化，但是上面的那篇文章还是需要从头到尾看一遍。

### 三、PageHelper的使用

在”一、项目搭建“里下载好了源码，下载的源码导入eclipse后的结构里面，有如下的文件：

接下来我们并不需要新增文件，直接在这些文件上修改就好了。

**3.1、引入依赖**

在pom.xml文件添加如下依赖：

<!-- mybatis分页插件依赖 -->

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

<version>4.0.0</version>

</dependency>

**3.2、在Mybatis配置xml中配置拦截器插件**

在mybatis-config.xml文件中添加插件后的内容如下：

<?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>

<plugins>

<!-- com.github.pagehelper为PageHelper类所在包名 -->

<plugin interceptor="com.github.pagehelper.PageHelper">

<property name="dialect" value="mysql"/>

<!-- 该参数默认为false -->

<!-- 设置为true时，会将RowBounds第一个参数offset当成pageNum页码使用 -->

<!-- 和startPage中的pageNum效果一样-->

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

<!-- 该参数默认为false -->

<!-- 设置为true时，使用RowBounds分页会进行count查询 -->

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

<!-- 设置为true时，如果pageSize=0或者RowBounds.limit = 0就会查询出全部的结果 -->

<!-- （相当于没有执行分页查询，但是返回结果仍然是Page类型）-->

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

<!-- 3.3.0版本可用 - 分页参数合理化，默认false禁用 -->

<!-- 启用合理化时，如果pageNum<1会查询第一页，如果pageNum>pages会查询最后一页 -->

<!-- 禁用合理化时，如果pageNum<1或pageNum>pages会返回空数据 -->

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

<!-- 3.5.0版本可用 - 为了支持startPage(Object params)方法 -->

<!-- 增加了一个`params`参数来配置参数映射，用于从Map或ServletRequest中取值 -->

<!-- 可以配置pageNum,pageSize,count,pageSizeZero,reasonable,不配置映射的用默认值 -->

<!-- 不理解该含义的前提下，不要随便复制该配置 -->

<property name="params" value="pageNum=start;pageSize=limit;"/>

<!-- always总是返回PageInfo类型,check检查返回类型是否为PageInfo,none返回Page -->

<property name="returnPageInfo" value="check"/>

</plugin>

</plugins>

</configuration>

**3.3、mapper文件添加sql**

在userMapper.xml文件里面添加sql：

<!-- 根据用户名查询 -->

<select id="selectUserByUserName" parameterType="java.lang.String" resultMap="BaseResultMap">

SELECT \*

FROM t\_user

WHERE 1 = 1

<if test="userName != null and userName !=''">

AND USER\_NAME = #{userName,jdbcType=VARCHAR}

</if>

ORDER BY USER\_ID

</select>

**3.4、dao添加对用方法**

UserDao.java添加如下方法：

List<User> selectUserByUserName(@Param("userName") String userName);

记得@Param必须有

**3.5、service添加分页查询方法**

UserService.java文件添加方法

PageInfo<User> queryByPage(String userName,Integer pageNo,Integer pageSize);

**3.6、实现类添加对应实现**

UserServiceImpl.java添加实现方法

@Override

public PageInfo<User> queryByPage(String userName, Integer pageNo,Integer pageSize) {

pageNo = pageNo == null?1:pageNo;

pageSize = pageSize == null?10:pageSize;

PageHelper.startPage(pageNo, pageSize);

List<User> list = userDao.selectUserByUserName(userName);

//用PageInfo对结果进行包装

PageInfo<User> page = new PageInfo<User>(list);

//测试PageInfo全部属性

System.out.println(page.getPageNum());

System.out.println(page.getPageSize());

System.out.println(page.getStartRow());

System.out.println(page.getEndRow());

System.out.println(page.getTotal());

System.out.println(page.getPages());

System.out.println(page.getFirstPage());

System.out.println(page.getLastPage());

System.out.println(page.isHasPreviousPage());

System.out.println(page.isHasNextPage());

return page;

}

**3.7、单元测试类添加分页查询测试方法**

UserServiceTest.java文件添加方法：

@Test

public void queryByPageTest(){

PageInfo<User> page = userService.queryByPage(null, 1, 1);

System.out.println(page);

}

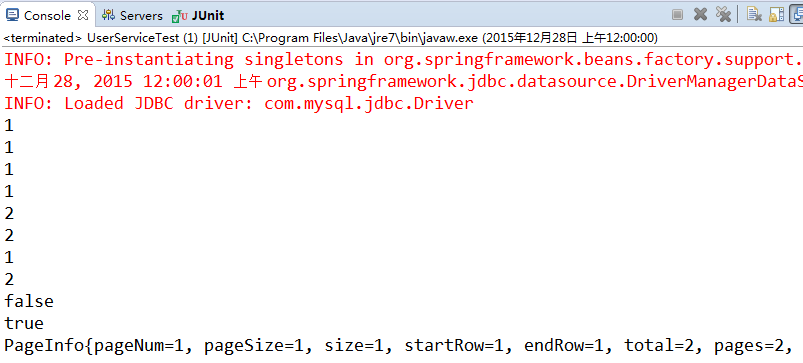
**3.8、运行**

**在小宝鸽进行了上面的修改之后，第一运行报错了！！！！最后是pom.xml文件注释了下面两个包的依赖才运行成功了，注释的依赖如下：**



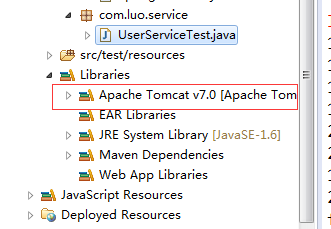
具体原因现在小宝鸽也不知道，应该是包有冲突的原因吧，如果猿友们知道原因麻烦私信我或评论解释一下。

然后运行单元测试类，结果如下：



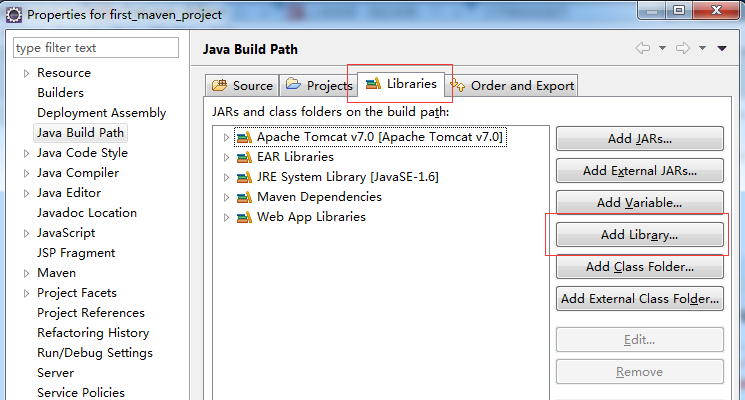
**3.9、补充说明**

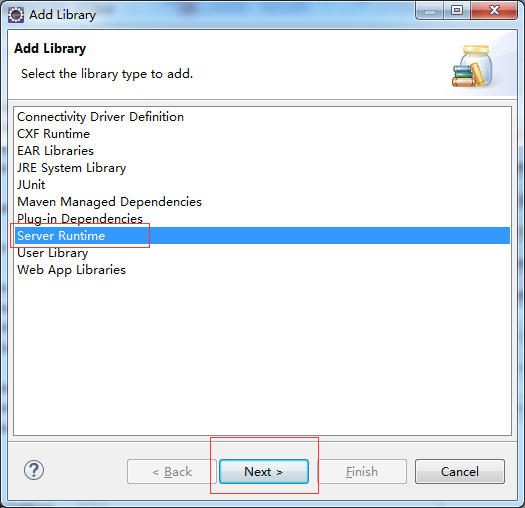
在3.8中，我注释了那两个包依赖是web项目相关的依赖，如果仍需要进行前端开发的猿友可以引入tomcat就好了，引入结果如下：

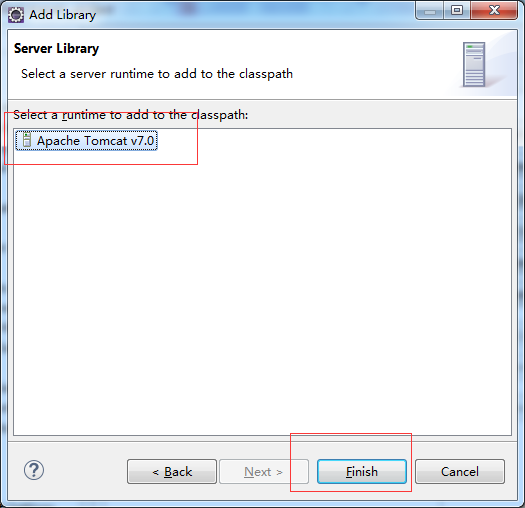


引入过程：

选中项目右键–>build path，然后如下操作：







### 4、源码下载

<http://download.csdn.net/detail/u013142781/9379685>