1. 项目名称：

会议管理系统

1. 项目说明：

以Java为开发语言，利用SSM框架|Spring+Mybatis+SpringMvc，数据库oracle对项目进行编写，此项目包含用户界面登录、会议安排通知、预定会议查询、用户注册、部门管理（管理员）、注册审批（管理员）、搜索员工（管理员）、添加会议室（管理员）、查询会议室、预定会议、搜索会议功能。

1. 项目的思路来源：

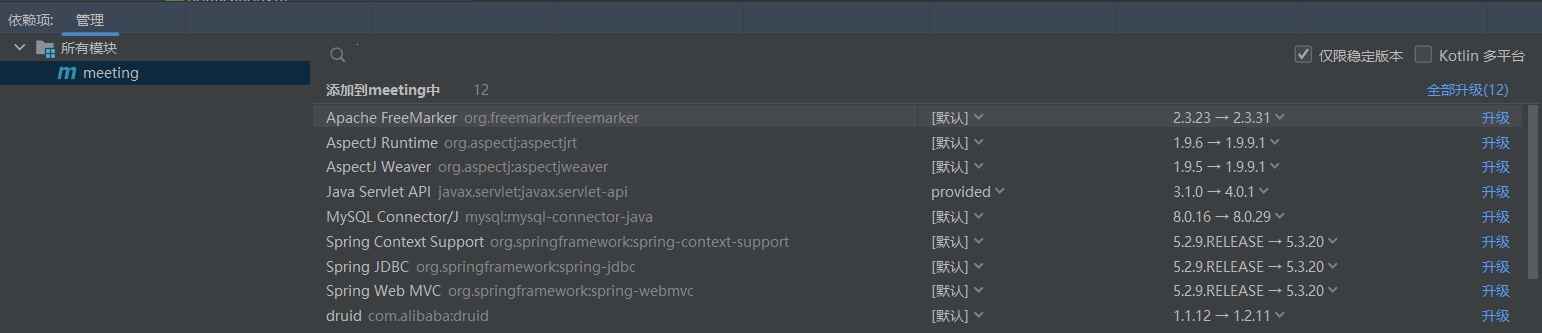
江南一点雨：[SSM实战项目之会议管理系统（2020版）\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1ep4y1S7s6?spm_id_from=333.337.search-card.all.click)

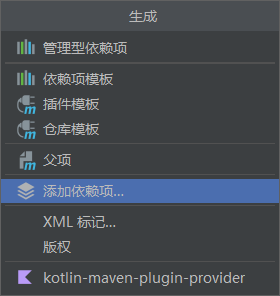
1. 项目开发的环境：

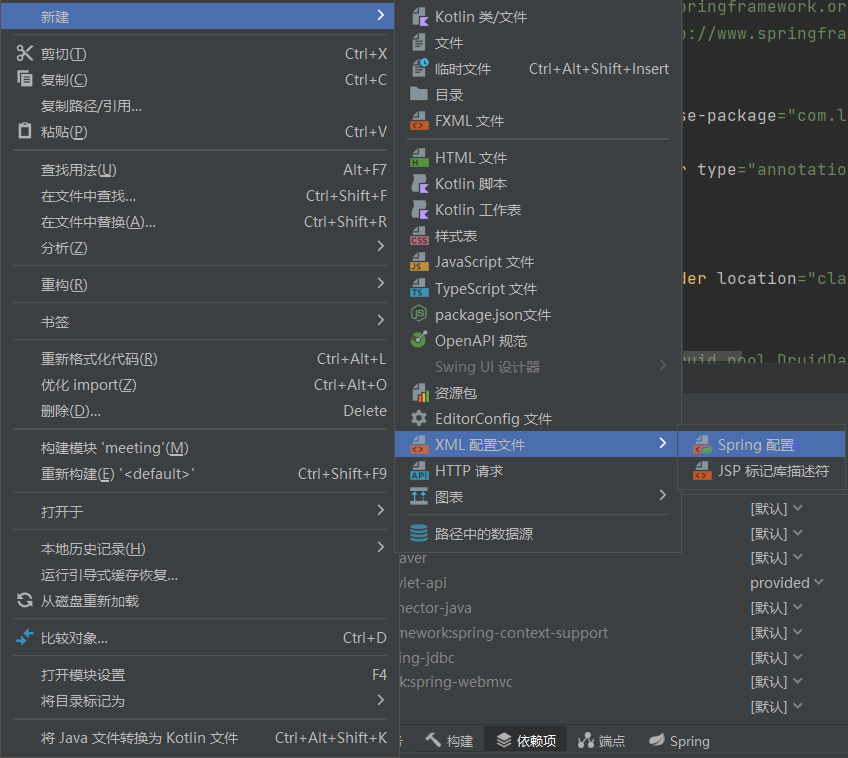
Java 1.8，Oracle 19c，MySQL，idea2022，tomcat 9.0.64

五、项目的框架与代码分析：

1. 对pom.xml文件进行修改，添加项目所需要的依赖：
2. org.springframework.spring-webmvc
3. Java.servlet-api
4. org.springframework.spring-jdbc
5. com.alibaba
6. mysql-connecter-java
7. mybaits
8. mybaits-spring
9. aspectjweaver
10. aspectjrt
11. spring-context-support
12. freemarker
13. jackson-databind

 添加依赖方法：在<dependencies></dependencies>下alt+ins打开添加依赖项搜索并进行添加，默认使用各项依赖的最新版本。



2.在resource文件夹加中加入applicationContext.xml，这里的xml文件要设置成spring配置：

建立好文件后对组件扫描，mybaits，druid连接池，事务进行配置：

1. 组件扫描：利用<context:componet-scan> </context:componet-scan>进行配置，扫描时要排除controller “expression= ’org.springframework.stereotype.Controller’ ”
2. Mybaits：<context:property-placeholder location="classpath:db.properties"/>
3. druid连接池：此处对下文的db.properties进行配置即可
4. 事务：引入org.springframework.jdbc.datasource.DataSourceTransactionManager类添加方法add，insert，updata，delete。
5. 在resource夹中添加db.properties:

此文件作为项目与数据库之间的桥梁，对其配置时一定要比对好自己的url链接是否正确，可以先利用idea连接数据库，在数据库页面中可以寻找到自己的url链接，在其后接:

name(数据库的名称)?serverTimezone=GMT%2B8

例如：

db.url = jdbc:mysql://localhost:3306/meetdemo?serverTimezone=GMT%2B8

添加用户名与密码：

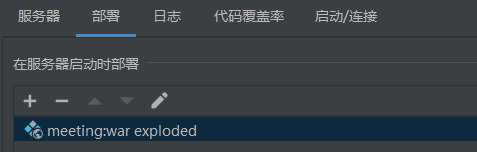
db.username =

db.password =

1. 这时可以测试一下依赖的引用是否正确，所以对tomcat进行配置：



对其进行配置时先进行添加部署工件：



在应用程序上下文处只留一个/:



在src.main.java.com.lhz.meeting.controller下建立java类文件，编写一个简单的代码测试，

public class HelloController {

@GetMapping("/hello")

public String hello(Model model) {

model.addAttribute("hello");

return "hello";

}

}

六、心得体会

对于本此项目的实现，时间相对于Java小白来说有点短，只能利用c与python的语言理解去尝试跟着教程编写代码以及bug的修复，对于刚开始的数据库环境配置也有许多不理解的地方，在网上搜索教程明白了监听服务的建立与打开方式，知道了与项目进行连接的url的编写，学会了对数据库的建立以及表的建立，ssm项目的基本框架和各项功能的实现方法，本项目不是一个前后端分离的项目，在日后的学习过程中补充java知识，学习如何搭建一个前后端分离的项目，并向更复杂更实际的项目冲刺！！！！！