**作业一 自己总结的SSM知识点**，github仓库地址：

https://github.com/zh1101/zh1101.git **一、Spring**

spring IOC创建对象的两种方式

1.创建容器对象

2.在当前包的所在类下找有关bean的配置文件 测试方便

**二、spring主配置文件创建对象的方式**

1.调用无参构造器

2.调用带参构造器

3.用工厂模式创建 工厂里面分为静态方法和非静态方法

**三、spring给对象属性赋值依赖注入的方式**

1.构造函数赋值

2.set方法注入值

分为 普通字段赋值和集合属性赋值(list map property)

**四、di 处理依赖关系**

四种方式

最常用的在主配置文件中通过set方法注入,在service接收dao注入值,在controller接收service注入值

内部bean

p名称空间,给属性注入值

自动装配 配置简单,但维护麻烦

**五、配置方式和注解方式的区别**

配置便于维护 但配置太多很繁琐

注解开发方便简化配置,但不利于后期维护,如果修改对象创建,关系处理等需要改代码

**六、Java中的代理**

静态代理

    1.目标对象必须实现接口

    2.代理对象要实现和目标对象一样的接口

动态代理(jdk代理)

    目标对象一定要实现接口,代理对象不用实现接口

Cglib代理(spring默认支持) 也叫子类代理

    jdk动态代理有一个限制,使用动态代理的对象必须实现接口

    如果向代理没有实现接口类,使用cglib

    1.目标对象可以不实现接口

    2.目标类不能final 如果final报错

    3.方法是final/static,不会被代理拦截,会直接执行目标对象方法

**代理总结**

    1.目标对象有实现接口,使用jdk动态代理

    2.没有实现接口,使用cglib代理

    3.没有实现接口且为final,不能进行aop编程,不能生成代理会报错

**七、spring生成代理对象的过程**

    1.创建容器对象时,根据切入点表达式拦截的类,生成代理对象

    2.目标对象有实现接口,使用jdk动态代理

    3.没有实现接口,使用cglib代理

    4.从容器获取代理后的对象

    5.执行代理对象方法,运行时动态植入切面类中的通知

**八、AOP(面向切面编程)是什么**

    实现原理就是Java动态代理,但是动态代理必须实现接口,所以spring的aop是用cglib实现的

**九、事务**

    一组操作的执行单元,相对于数据库操作来讲事务管理的是一组SQL指令(增删改查)

    事务的一致性要求操作必须全部执行成功,出现任何一条SQL语句没有执行成功,这一组

    操作会全部回滚

spring声明式事务管理,就是粗粒度的事务控制

spring创建SessionFactory的方式

所有配置都在spring中完成

spring对dao操作的支持

spring提供了JDBCTemple模版的工具类,对原始的jdbc操作进行了简化

**mybatis**

mybatis操作dao层 直接写SQL语句 更加灵活

查询操作:返回单条记录时,表字段和对应实体属性命名一致时可以不使用resultMap属性配置,直接使用resultType=”返回的全类名”

查询操作:返回多条记录时,返回结果配置的不是集合类型,而是集合元素的类型,参数通过Map等方式进行封装

**mybatis两种查询方式**

association 用于多对一或一对一

collection 用于一对多或多对多

Mybatis和hibernate一样，也使用缓存；缓存分为一级缓存和二级缓存，一级缓存指在SqlSession内；二级缓存能被所有的SqlSession共享

mybatis简化配置:扫描式加载SQL映射文件

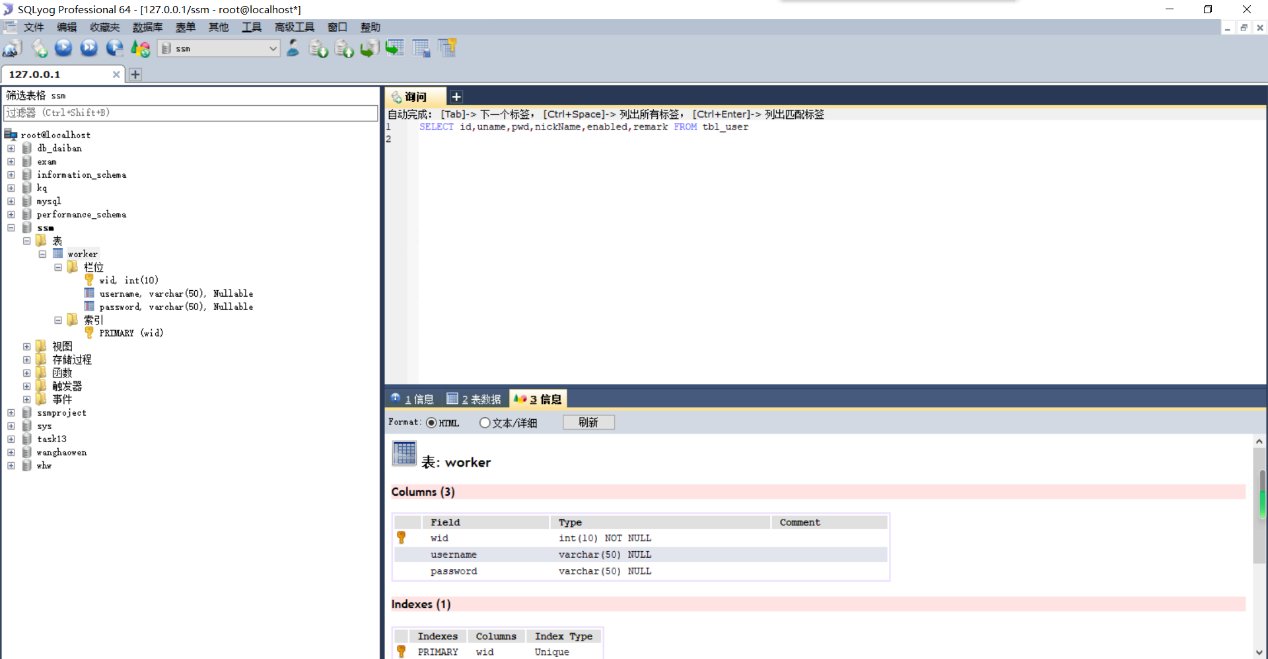
**springmvc**

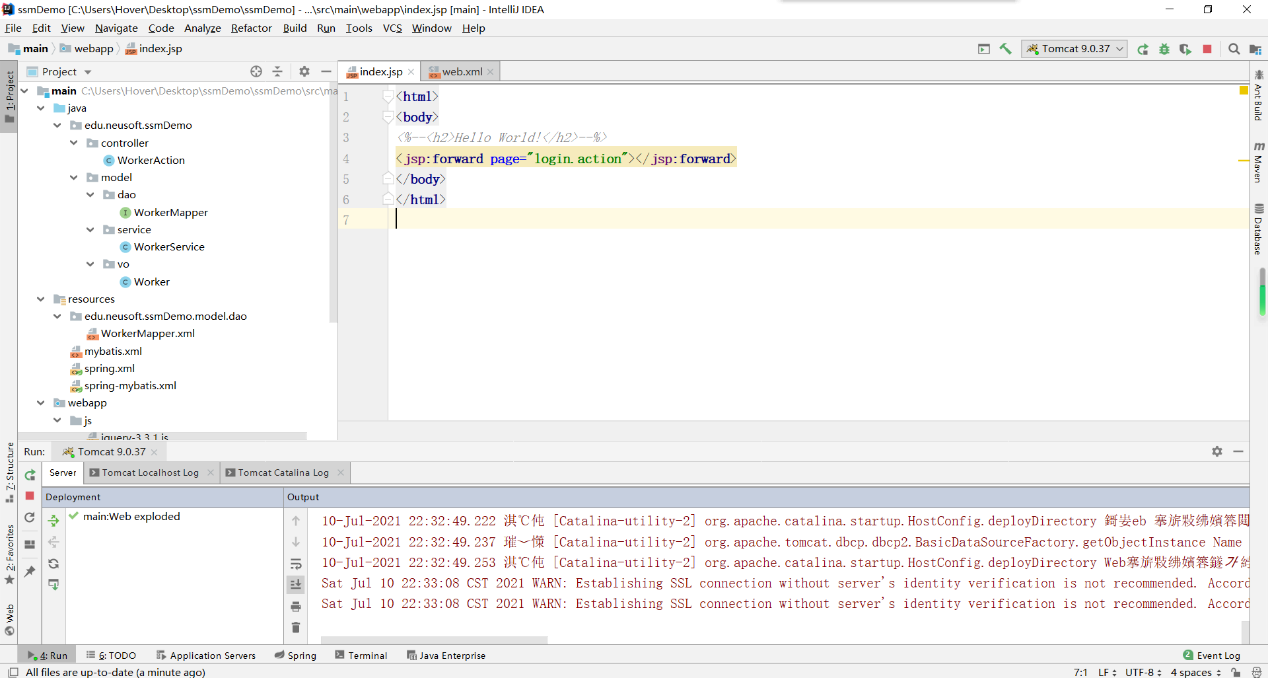
配置文件都需要配置什么

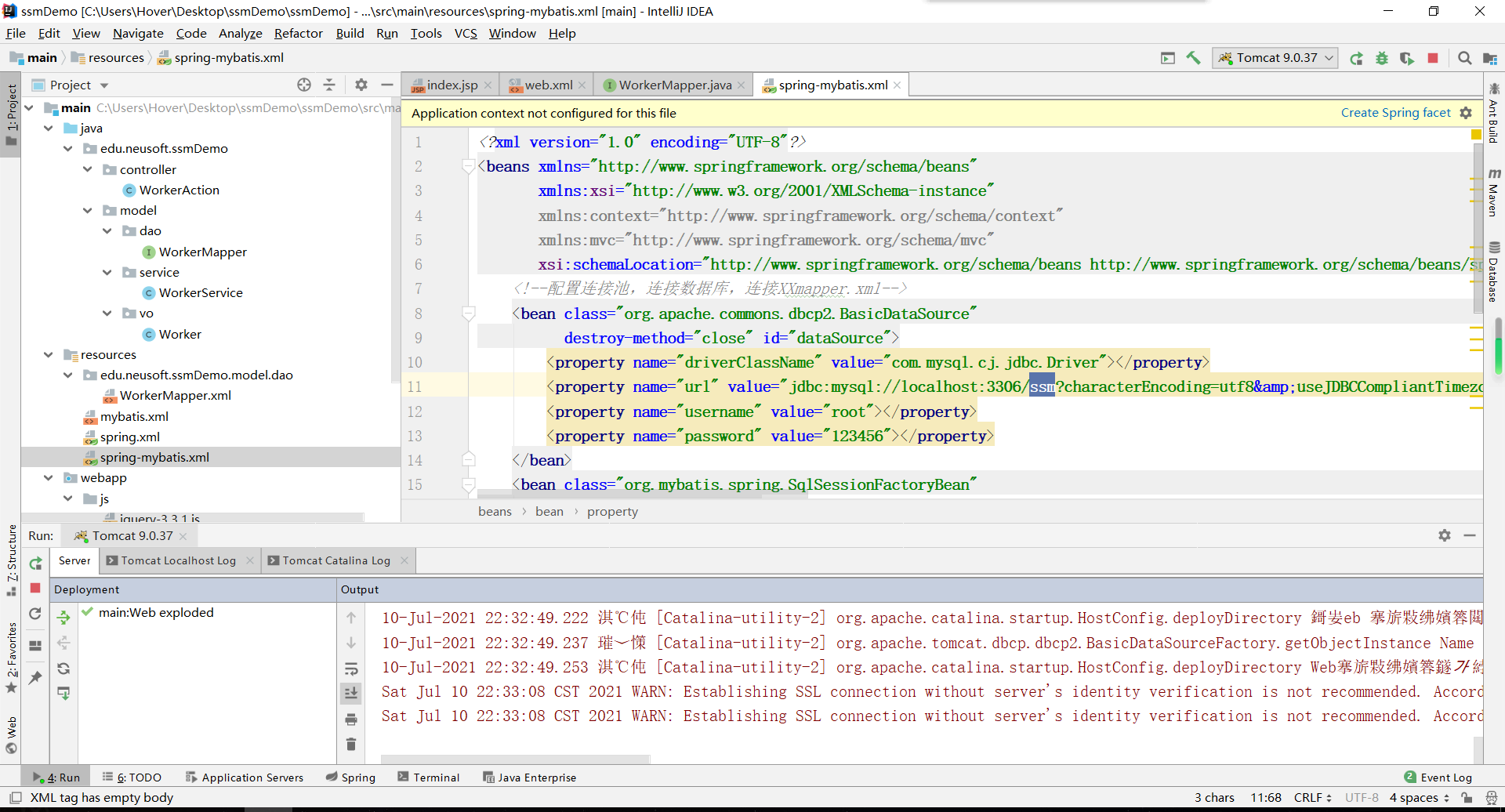
前端过滤器,控制器,视图解析器,中文乱码处理

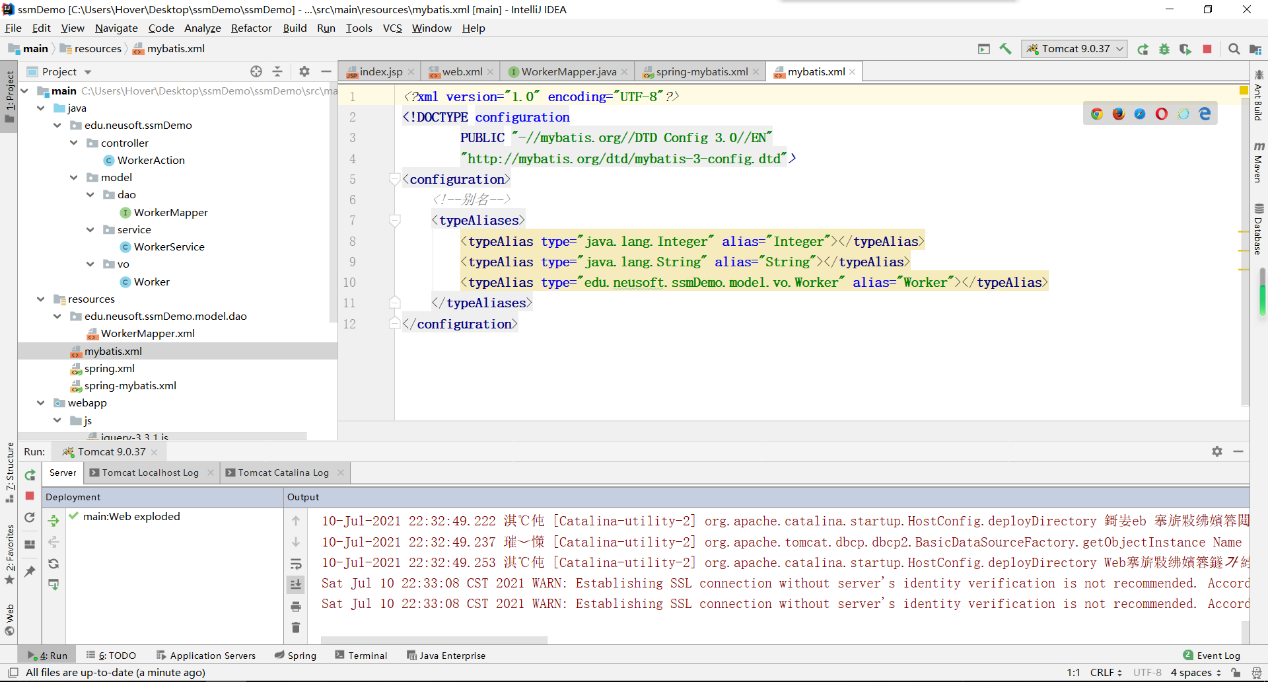
**作业二 报告**

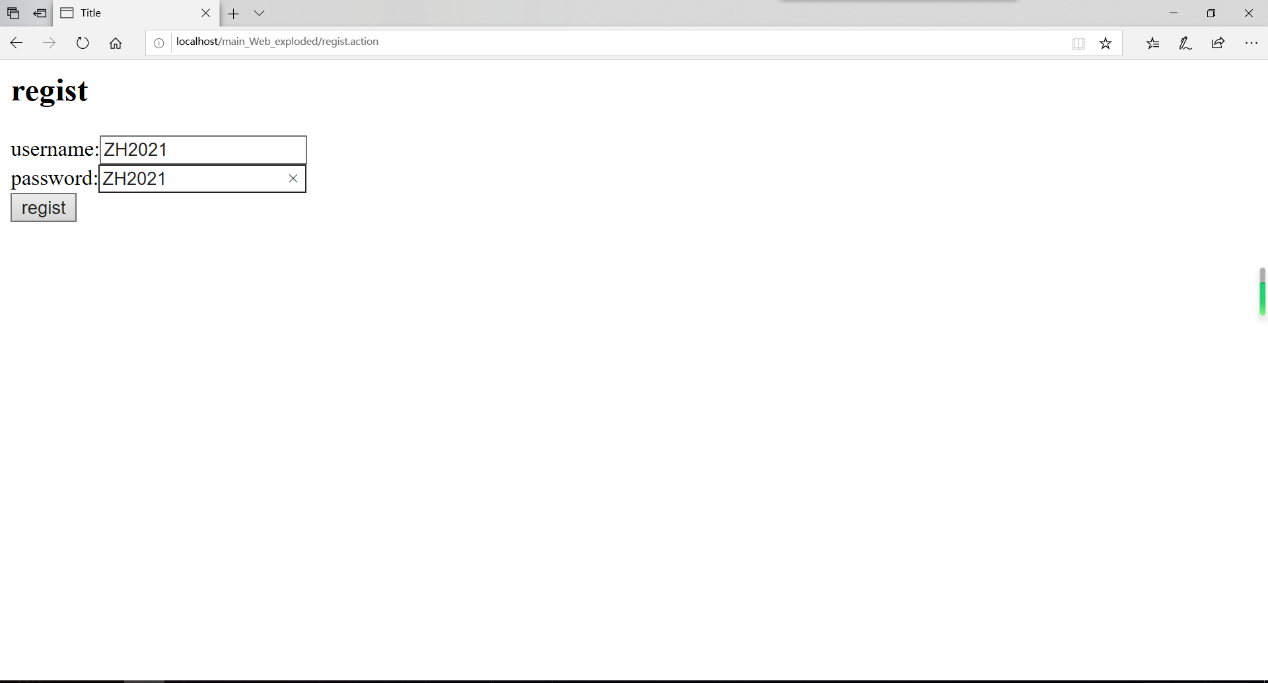
**SSM用户登录报告**

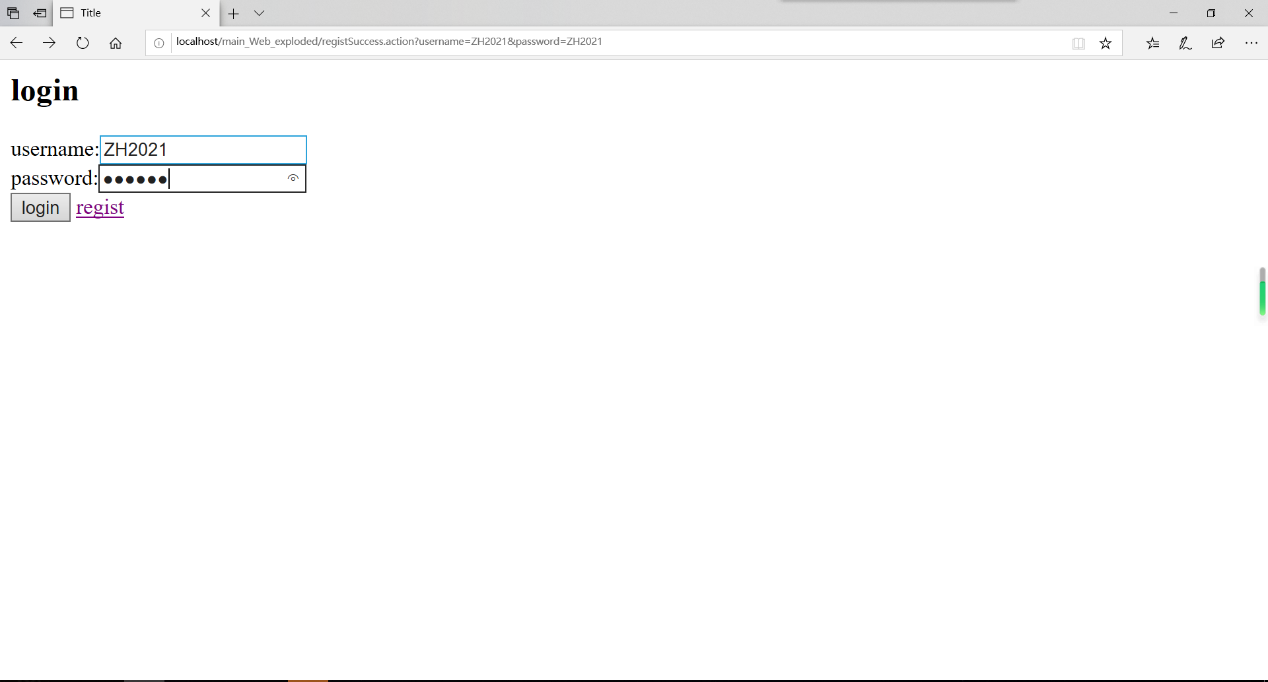












简介：通过SSM框架，对数据库进行增删改查，可以登陆注册并且显示所有用户信息

**需求：输入用户名和密码进行登录验证**

ssm具体实现步骤：

1. 首先建立数据库

2. 先在pojo中定义实体类，对应数据库中的各个列，定义对象属性

3. 写mapper.xml，其中写sql语句，对应数据库中的操作，定义需要实现的功能，如insert、select、update等

4. 写mapper.java接口，将mapper.xml中的操作按照id映射成java函数

5. 写service.java接口，为控制层提供服务，并写serviceImpl.java在其中自动装配mapper，完成相应功能并返回给控制层

6. 写controller.java，连接页面请求和服务层，获取页面请求的参数，通过service的自动装配，映射不同的URL到相应到不同的地方处理函数，并获取参数，对参数处理返回给服务层。

7. 写jsp静态页面，返回参数和获取的数据

整个过程为：数据库—>实体—>mapper.xml—>mapper.java(interface)—>service.java(interface) —>service.java—>controller.java—>jsp