**指定maven环境的jdk版本和字符集**

**添加在pom.xml中**

<build>

<plugins>

<!-- 设置编译版本为1.8 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

<encoding>UTF-8</encoding>

</configuration>

</plugin>

</plugins>

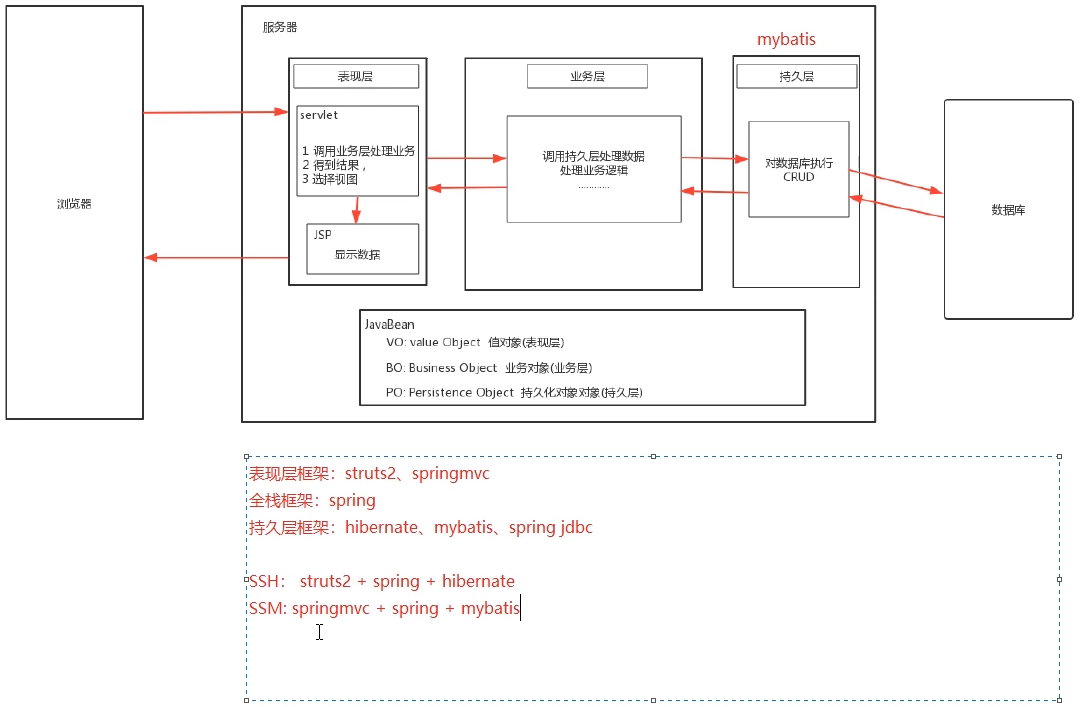
</build>

<scope>test</scope>

决定Jar包的作用范围

SSM=Spring MVC + Spring + Mybatis

框架就是一套规范



Hibernate是全自动持久层框架，可以全自动生成SQL，无法自行优化

Mybatis是半自动框架，需要手动编写SQL，可以自行优化，更加方便灵活

**<property name="url" value="jdbc:mysql:///mybatis\_db?useUnicode=true&amp;characterEncoding=UTF-8"/>**

加上encoding的设置解决中文乱码问题

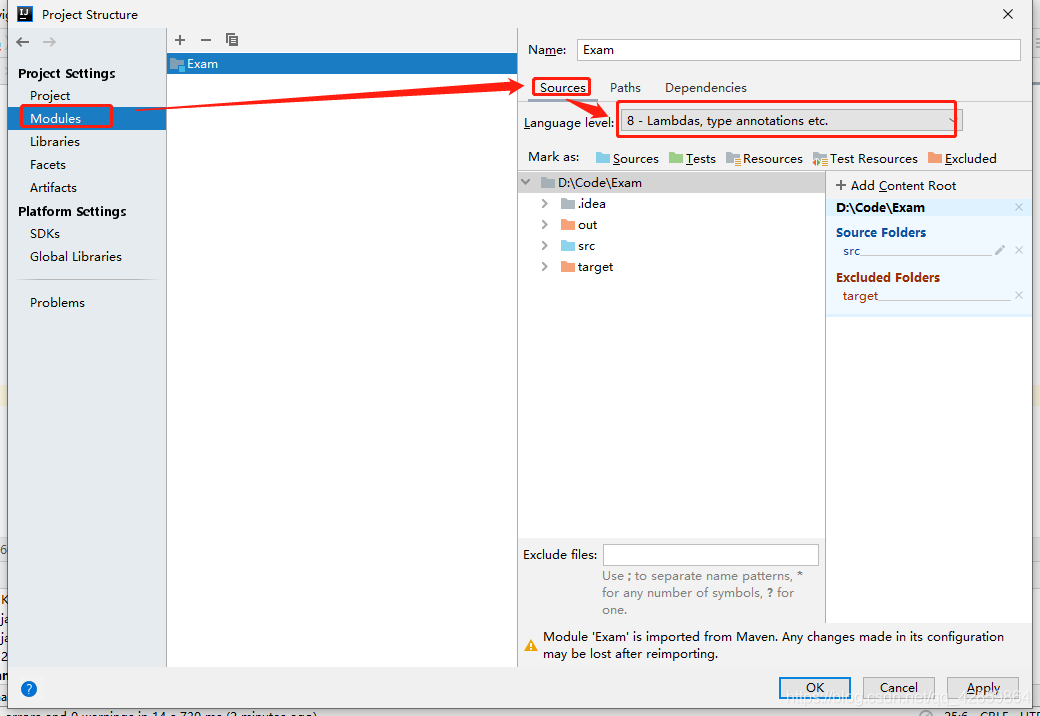
properties文件写法:

jdbc.url=jdbc:mysql:///mybatis\_db?useUnicode=true&characterEncoding=UTF-8

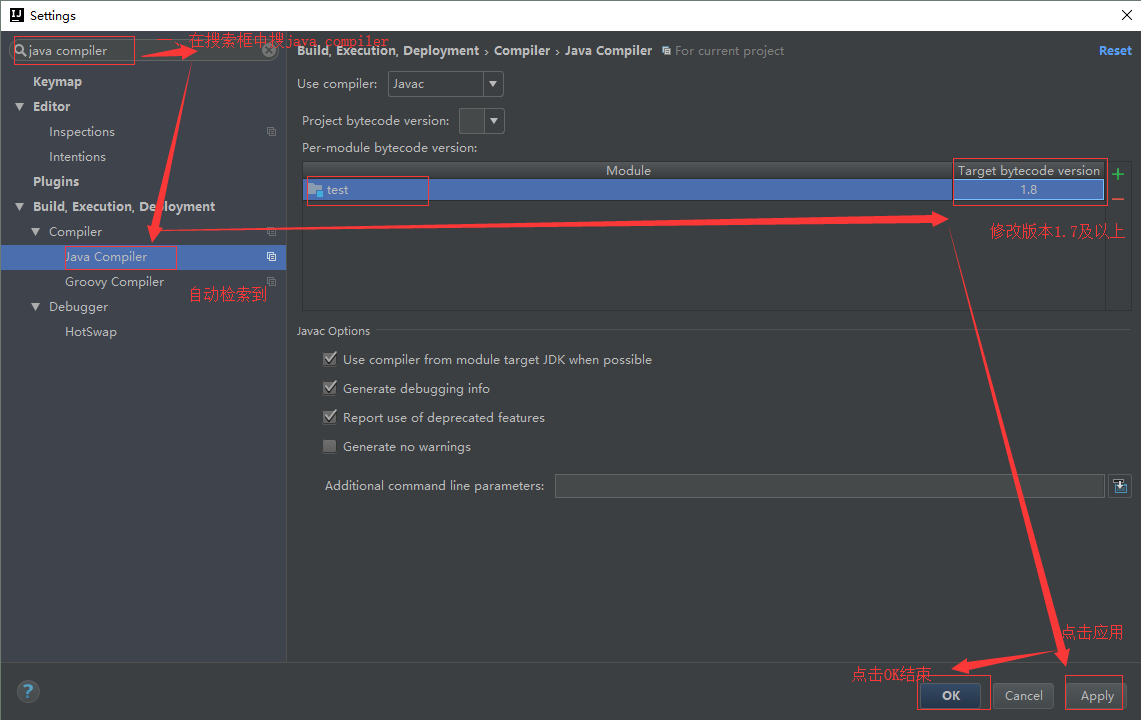
XML文件里的写法:

jdbc.url=jdbc:mysql:///mybatis\_db?useUnicode=true&amp;characterEncoding=UTF-8

<https://blog.csdn.net/weixin_41703383/article/details/79696148?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1.add_param_isCf&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1.add_param_isCf>



在File-Setting-Compiler这里的设置修改



SSM中的这些xml这些配置文件，要怎么设置才能像老师的那样那么方便提示？

Languages & Frameworks/Schemas and DTDs 配置一下。

https://blog.csdn.net/lucky\_ly/article/details/102530214

MyBatis一级缓存

MyBatis二级缓存。因为有跨namespace脏读问题，实际中不用，而是用Redis做第三方缓存

解耦：编译期不依赖，运行期才依赖。

**动态加载**

通过new方法来创建对象都是静态加载，但是通过forName()方法去可以进行动态加载。

**OfficeBetter.java**：这里采用动态加载

class OfficeBetter{

public static void main(String args[]){

try{

//动态加载类在运行时刻。编译的时候并不会报错，他才不会管你是哪个类

Class classWord=Class.forName(args[0]); //利用Java反射机制动态加载

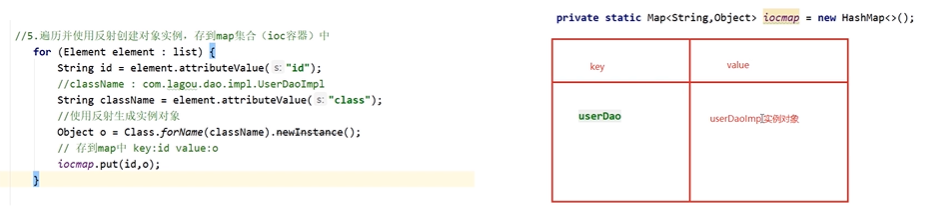
}catch(Exception e){

e.printStackTrace();

}

}

}



# [深入理解SpringIOC容器](https://www.cnblogs.com/zouxiangzhongyan/p/10723545.html)

<https://www.cnblogs.com/zouxiangzhongyan/p/10723545.html>

IDEA Debug时务必足以把断点设在内部执行语句上，否则不会进入Debug模式

src/main/resources/applicationContext.xml Spring核心配置文件

为什么这里IUserDao.class是对象类型? 到底什么东西是对象类型？

根据类型从容器中匹配Bean实例，当容器中相同类型的Bean有多个时，则此方法会报错。

// 2.根据类型在容器中进行查询：有可能报错的情况：根据当前类型匹配到多个实例  
//IUserDao userDao = classPathXmlApplicationContext.getBean("userDao", IUserDao.class);  
IUserDao userDao =

classPathXmlApplicationContext.getBean(IUserDao.class);

DAO持久层

Service业务层

配置文件中后加载的Bean会覆盖前面的同名Bean

不直接new一个DataSource，而是注入一个DataSource，这个小小的改动虽然简单，却带来了一系列好处：

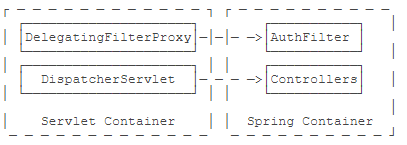
1. BookService不再关心如何创建DataSource，因此，不必编写读取数据库配置之类的代码；
2. DataSource实例被注入到BookService，同样也可以注入到UserService，因此，共享一个组件非常简单；
3. 测试BookService更容易，因为注入的是DataSource，可以使用内存数据库，而不是真实的MySQL配置。

当一个Bean实现了FactoryBean接口后，Spring会先实例化这个工厂，然后调用getObject()创建真正的Bean。getObjectType()可以指定创建的Bean的类型，因为指定类型不一定与实际类型一致，可以是接口或抽象类。

因此，如果定义了一个FactoryBean，要注意Spring创建的Bean实际上是这个FactoryBean的getObject()方法返回的Bean。为了和普通Bean区分，我们通常都以XxxFactoryBean命名。



一个代理模式的引用关系如下：





遇到这个错Spring Boot: Jdbc javax.net.ssl.SSLException: closing inbound before receiving peer's close\_notify。把URL的设置后面加上?useSSL=false

<property name="url" value="${jdbc.url}?useSSL=false"/>

