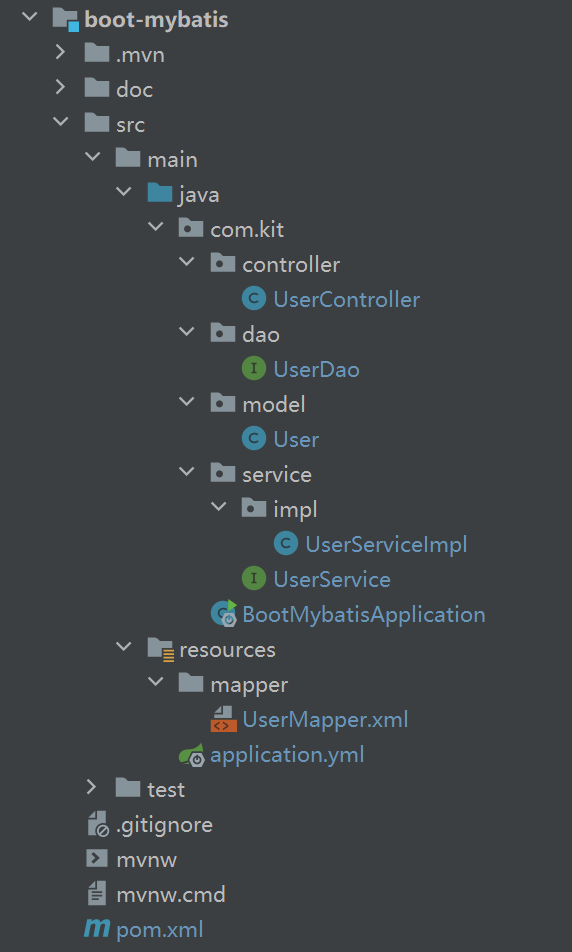
# SpringBoot集成MyBatis-yml方式详解

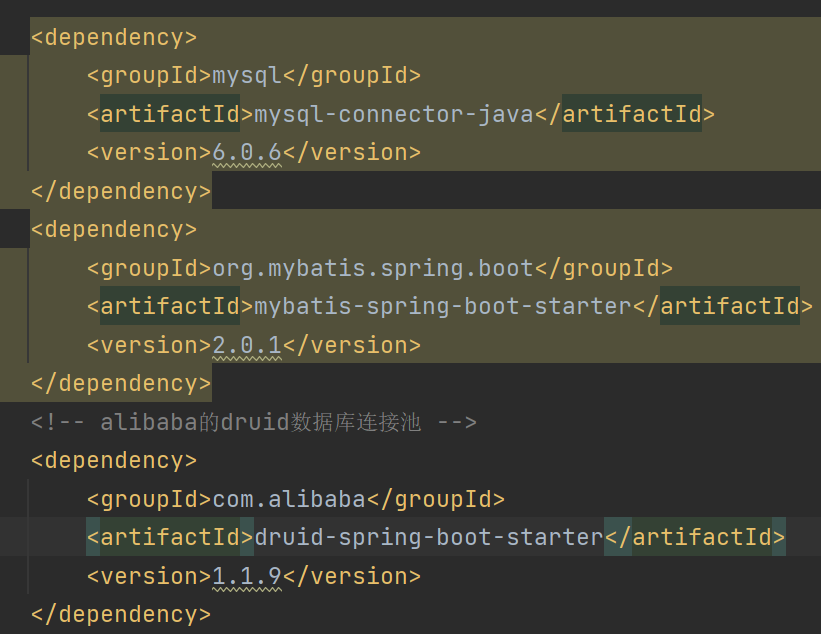
简介：spring boot整合mybatis开发web系统目前来说是市面上主流的框架，每个Java程序和springboot mybatis相处的时间可谓是比和自己女朋友相处的时间都多，但是springboot mybatis并没有得到你的真爱，因为你只是为了养活你女朋友而委曲求全的和spring boot mybatis假意相处。和spring boot mybatis 日夜相处你都不知道 spring boot 怎么和mybatis组合在一起的，yml文件里面的那些配置你都是百度之后复制粘贴进去的，却没有思考过为什么这么配置数据库就生效了？spring就可以加载mybatis了？接下来将会你逐个解释这些问题，缓解一下你和springboot mybatis 假意相处的尴尬。

# springboot整合mybatis

首页搭建了这样一个工程：

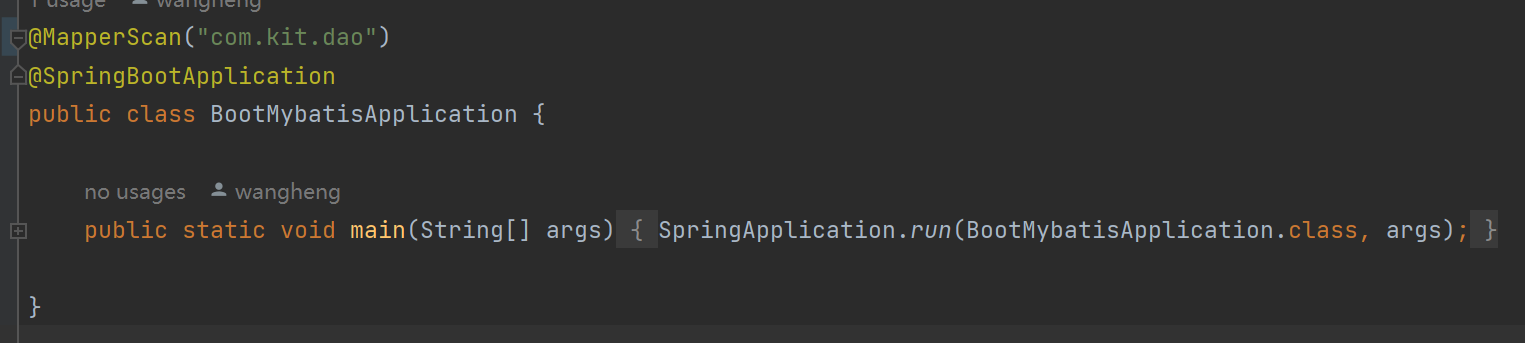


注意pom.xml中加入以下依赖

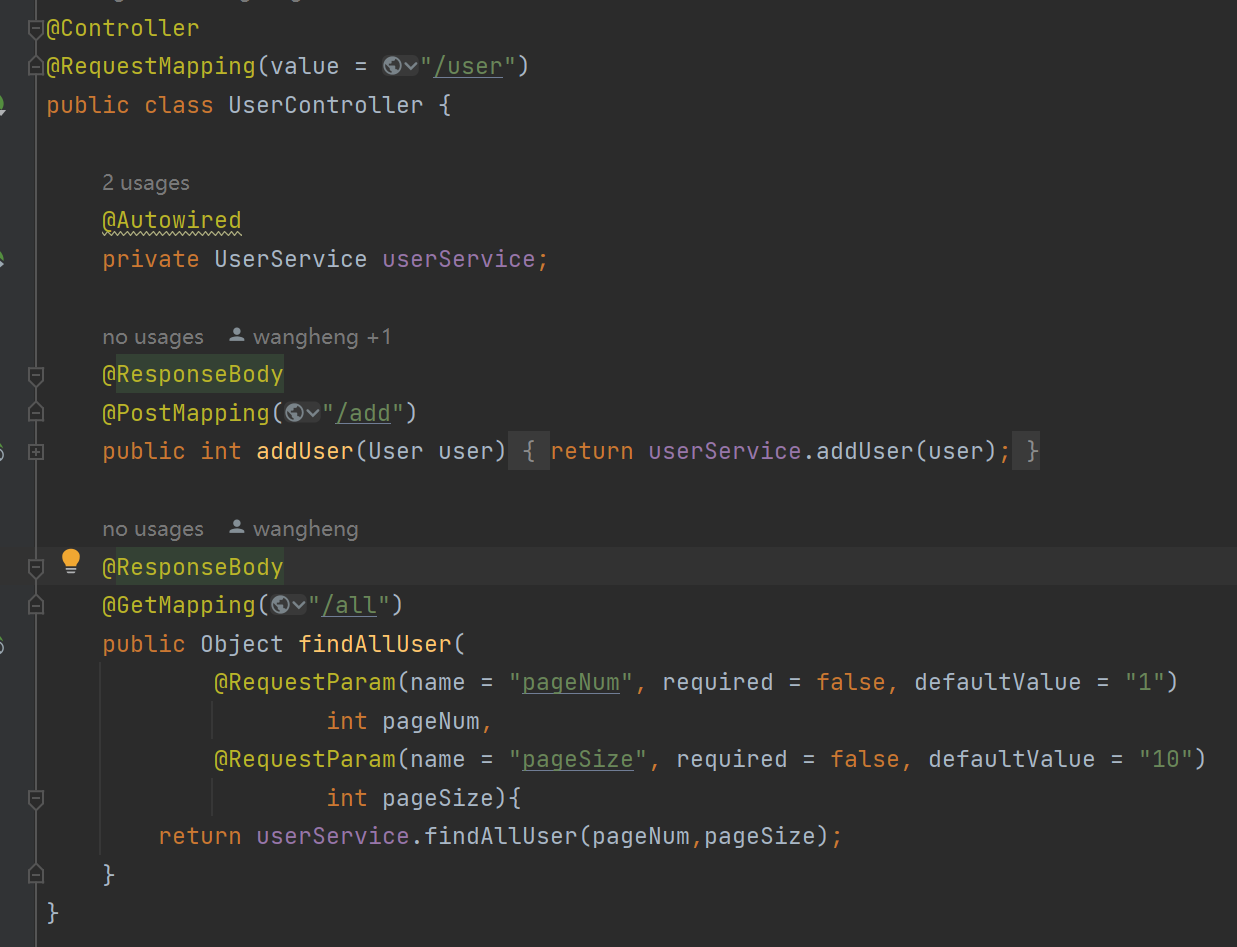


接下来对代码和配置进行部分展示

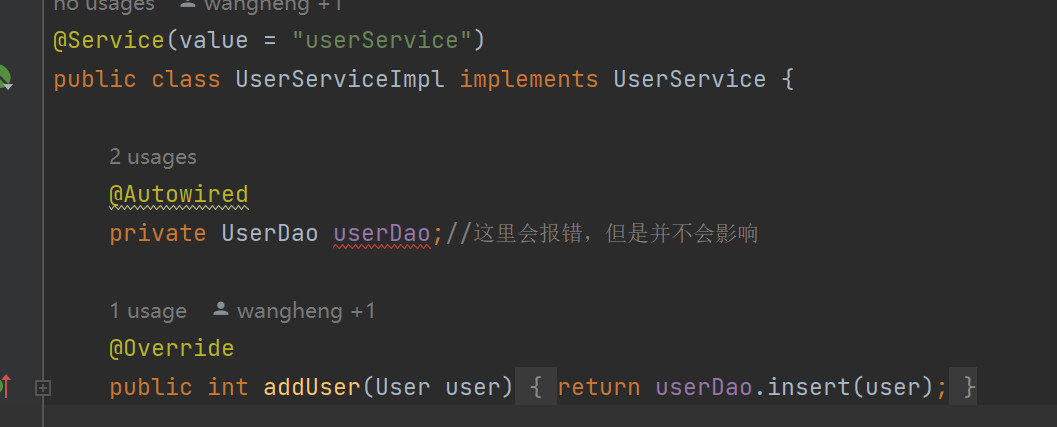
启动类



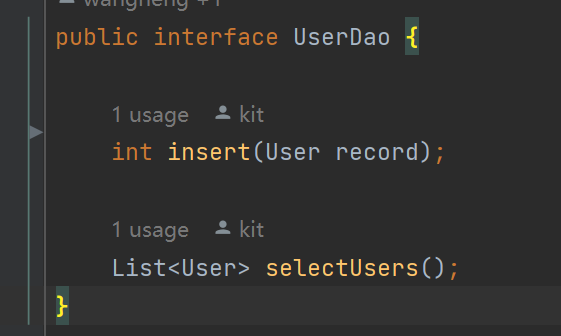
UserController.java



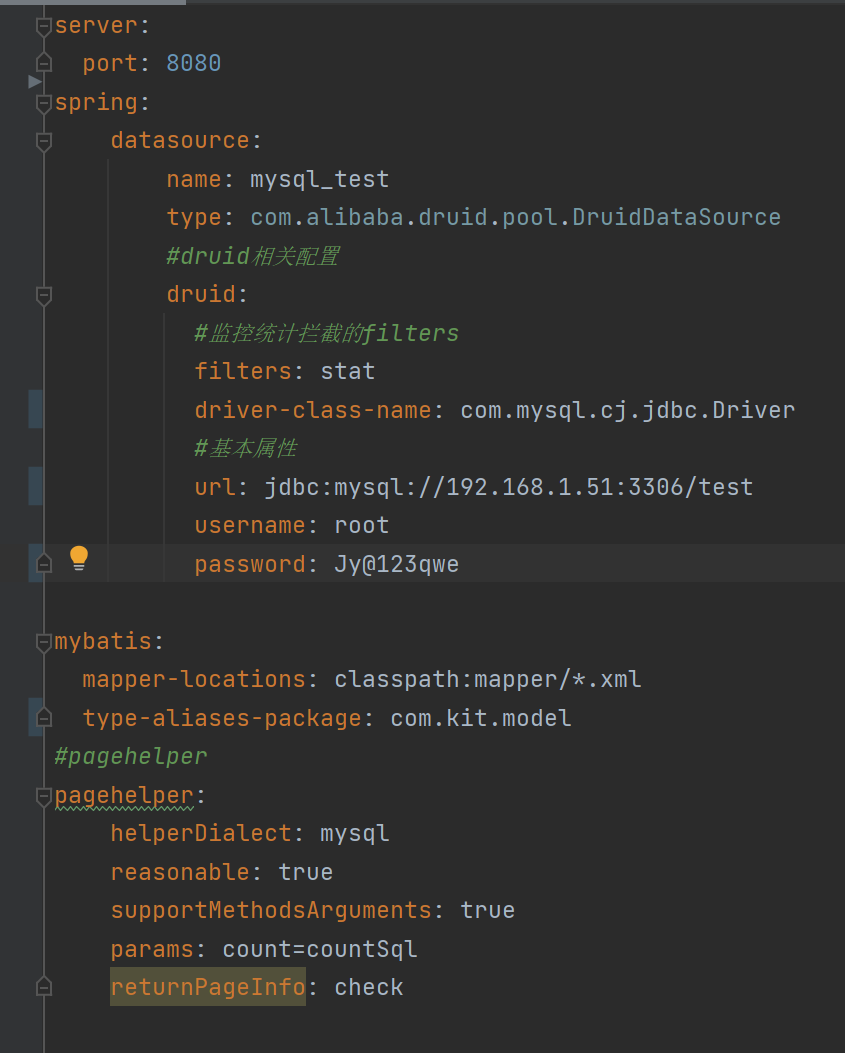
UserServiceImpl.java



UserDao.java



Application.yml配置如下：



另外还需要编写一份UserMapper.xml文件，部分代码细节省略，至此工程搭建完成。

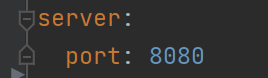
# 二、yml配置详解

项目搭建大家到网上找会有很多案例，简单的案例代码没有什么要讲解的，这里着重讲一下yml配置，很多人使用了很多年都还是不清楚yml里面为什么这么配置，比如spirng.datasources这个哪里来的？配置的参数怎么生效？

1、首先我们需要知道springboot自动装配的原理，解释如下：

springboot定义了一套接口规范，这套规范规定：springboot在启动时会扫描外部引用 jar 包中的META-INF/spring.factories文件，将文件中配置的类型信息加载到。对于外部 jar 来说，只需要按照 springboot定义的标准，就能将自己的功能装置进 springboot。

2、我们看到有个端口配置

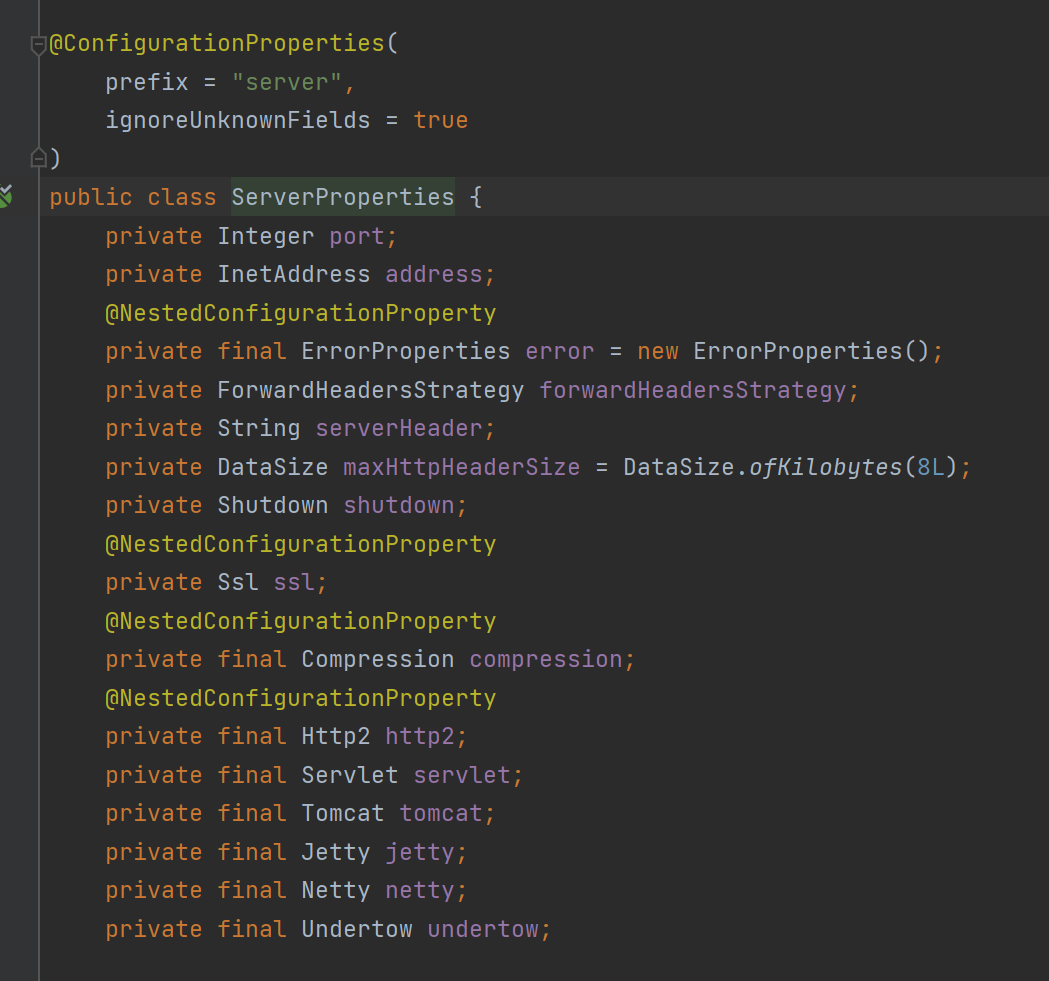


这里我们要看spring-boot-autoconfigure，找到META-INF\spring.factories，

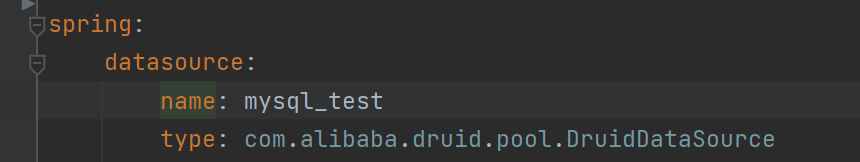
找到自动加载的类中

org.springframework.boot.autoconfigure.web.servlet.ServletWebServerFactoryAutoConfiguration

我们可以看到这里自动加载了ServerProperties.class：



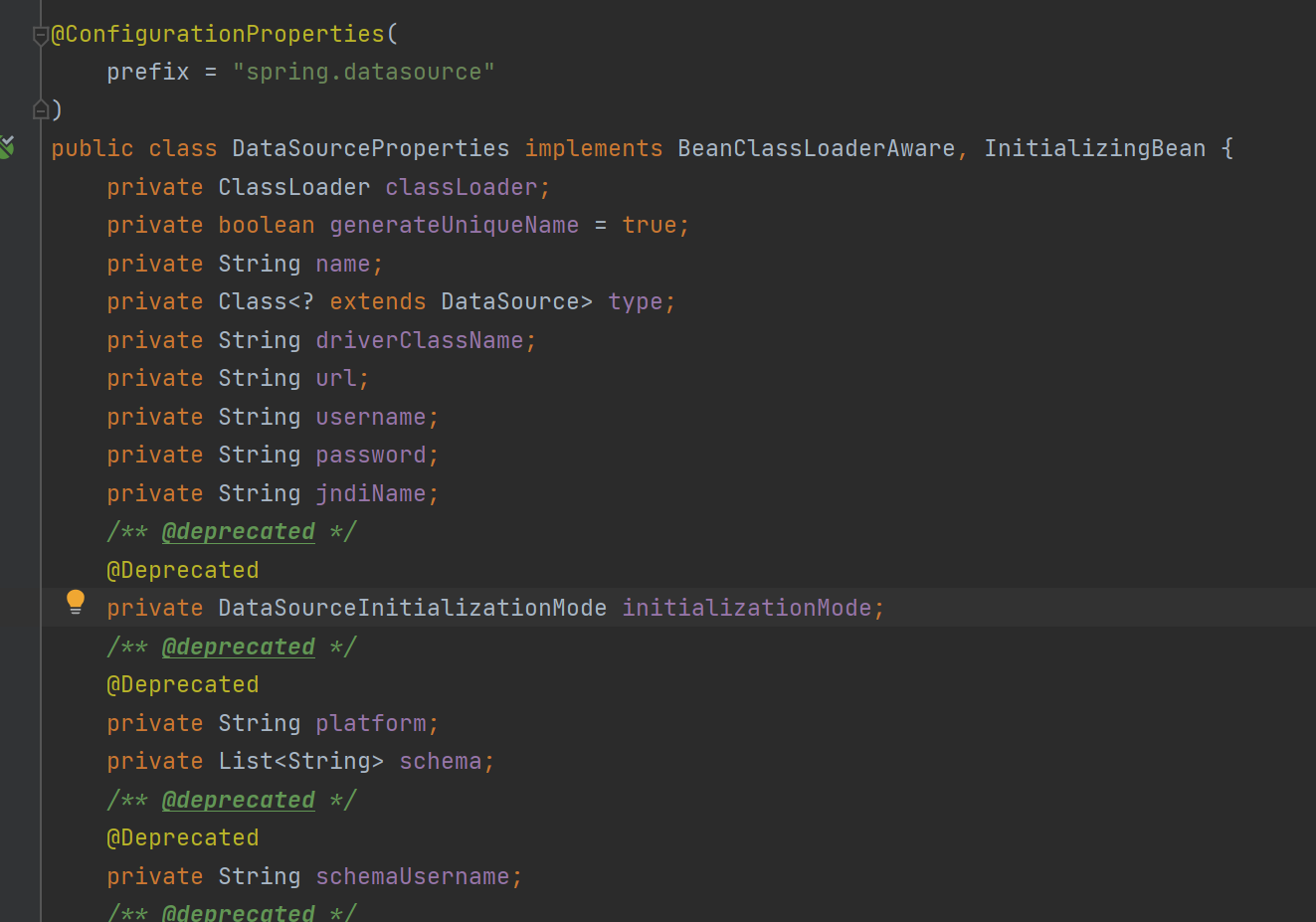
3、接下来我们看到有这一段配置：



这些的话也是在spring-boot-autoconfigure中自动装配的

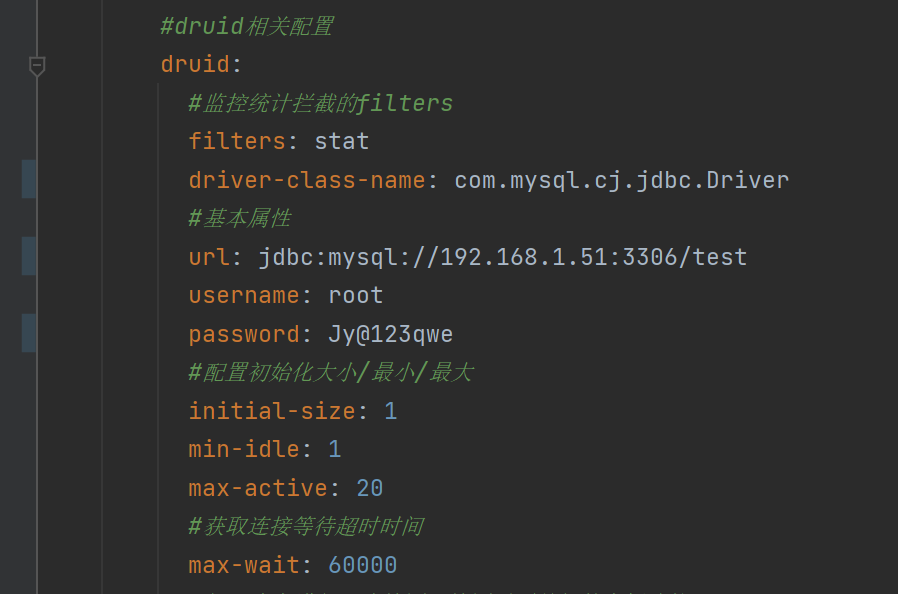
org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration

可以看到自动加载了DataSourceProperties.class：



这些就是我们配置的数据库相关的配置了

1. 接下来看到了数据源的配置



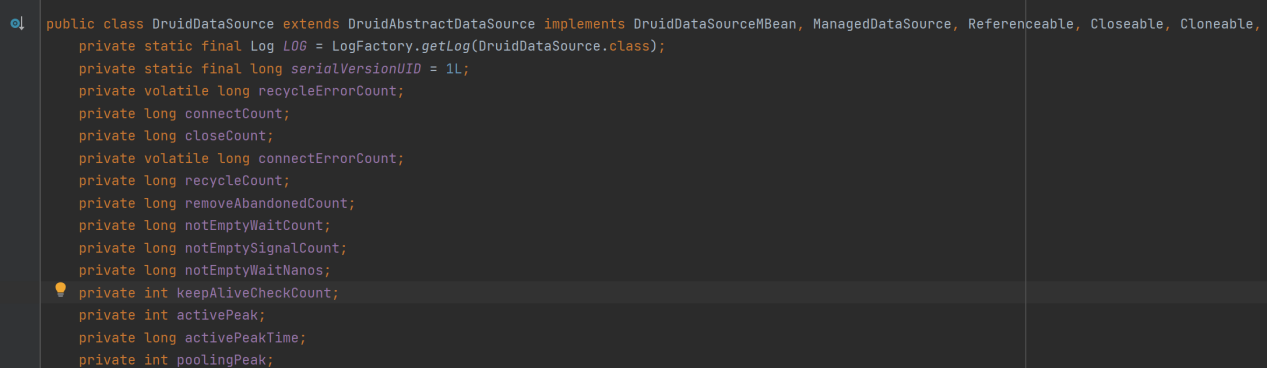
这个是通过第三方的starter加载进来的，我们在程序中引用了一个druid依赖：



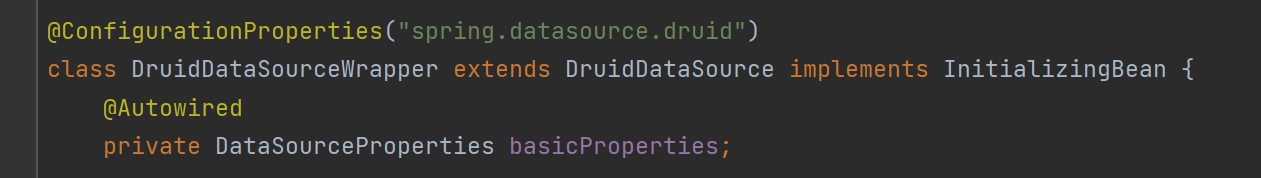
同理我们找到META-INF\spring.factories文件可以发现自动加载了

com.alibaba.druid.spring.boot.autoconfigure.DruidDataSourceAutoConfigure

可以看到这里自动加载了DruidDataSource.class：

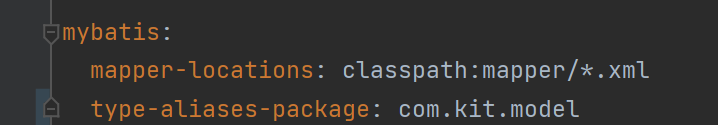


这里DruidDataSource继承了DruidAbstractDataSource类，可以找到子类DruidDataSourceWrapper

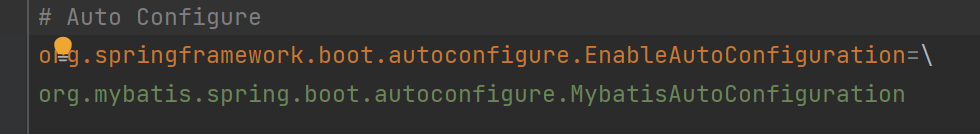


都这里数据库连接池的配置也找到了。

1. 接下来我们看一下mybatis的配置：



这里我们需要找一下mybatis-spring-boot-autoconfigure，我们找到mybatis-spring-boot-starter源码包下面，找到META-INF\spring.factories



我们找到这个类

@org.springframework.context.annotation.Configuration

@ConditionalOnClass({ SqlSessionFactory.class, SqlSessionFactoryBean.class })

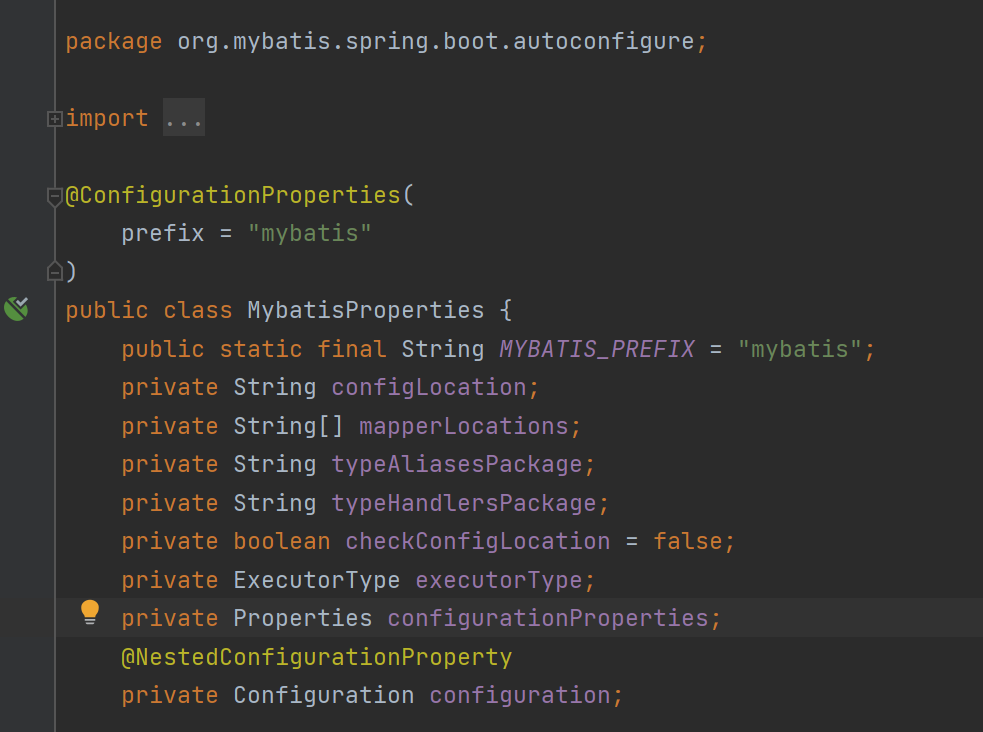
@ConditionalOnBean(DataSource.class)

@EnableConfigurationProperties(MybatisProperties.class)

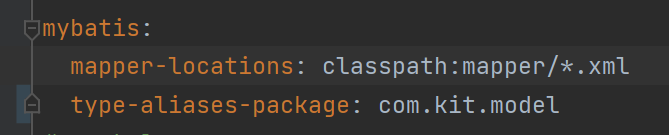
@AutoConfigureAfter(DataSourceAutoConfiguration.class)

public class MybatisAutoConfiguration {}

如下MybatisProperties配置类：



这些对应yml中如下的配置：



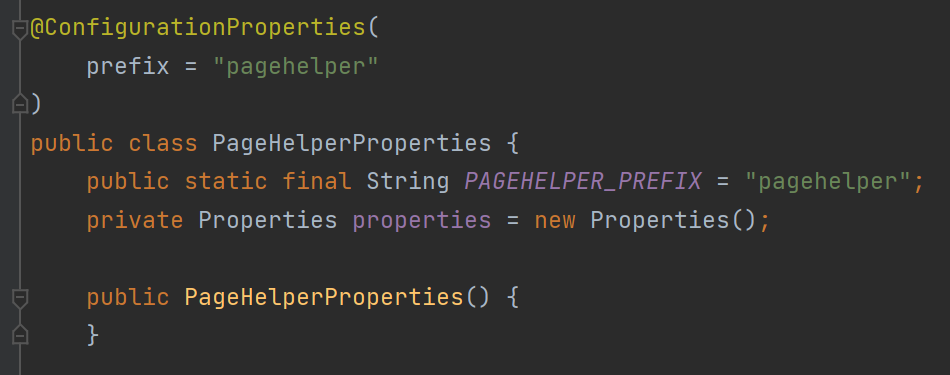
1. 最后还有一个分页插件的配置



这个和druid类似通过第三方的starter加载进来的，我们在程序中引用了一个pagehelper依赖

com.github.pagehelper.autoconfigure.PageHelperAutoConfiguration

这里自动加载了一个PageHelperProperties.class



到此springboot mybatis的配置就讲完了。