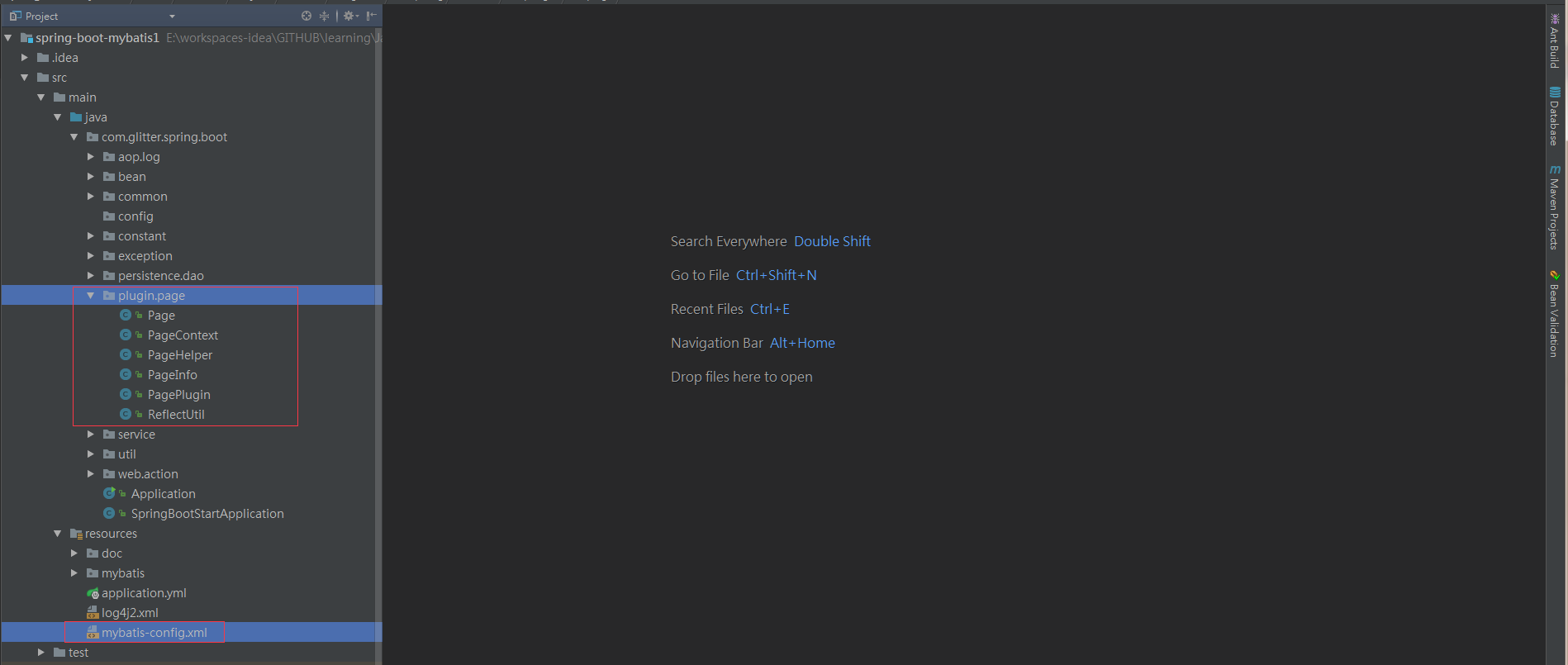
1.导入相关java文件，和mybatis-config.xml文件。



2.增加MybatisConfiguration配置类，使插件受spring管理，那么该bean就会被初始化到项目中，由于插件实现了org.apache.ibatis.plugin.Interceptor接口，所以就可以被mybatis调用到。

package com.glitter.spring.boot.config;  
  
import com.glitter.spring.boot.bean.BaseBean;  
import com.glitter.spring.boot.plugin.page.PagePlugin;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
public class MybatisConfiguration extends BaseBean {  
  
 @Bean  
 public PagePlugin pagePlugin(){  
 return new PagePlugin("mysql");  
 }  
   
}

3.后续的分页效果与之前没有区别。

4.另外，多说一下，原来在mybatis-config.xml中的settings信息，可以在application.yml中进行配置。

**spring:  
 datasource:  
 driver-class-name:** com.mysql.jdbc.Driver  
 **url:** jdbc:mysql://127.0.0.1:3306/demo\_client  
 **username:** root  
 **password:** root  
 **hikari.maxLifetime:** 1765000  
 **hikari.maximumPoolSize:** 15  
  
**mybatis:  
 mapper-locations:** classpath:mybatis/\*.xml  
 **type-aliases-package:** com.glitter.spring.boot.bean  
 **configuration:** *# 这里的配置与mybatis-config.xml的settings属性配置时一一对应的,不同点是这里属性名要换成中划线* **cache-enabled:** false  
 **lazy-loading-enabled:** false  
 **aggressive-lazy-loading:** false  
 **multiple-result-sets-enabled:** true  
 **use-column-label:** true  
 **use-generated-keys:** true  
 **auto-mapping-behavior:** FULL  
 **default-statement-timeout:** 25000  
 **default-fetch-size:** 100  
 **map-underscore-to-camel-case:** true