用idea创建Maven项目

修改setting.xml 里的镜像仓库和本地仓库

用pom文件导入相关jar包，maven会自动导入依赖包，因此spring mvc就只要

spring-web spring-webmvc spring-jdbc三个包就可以开始开发

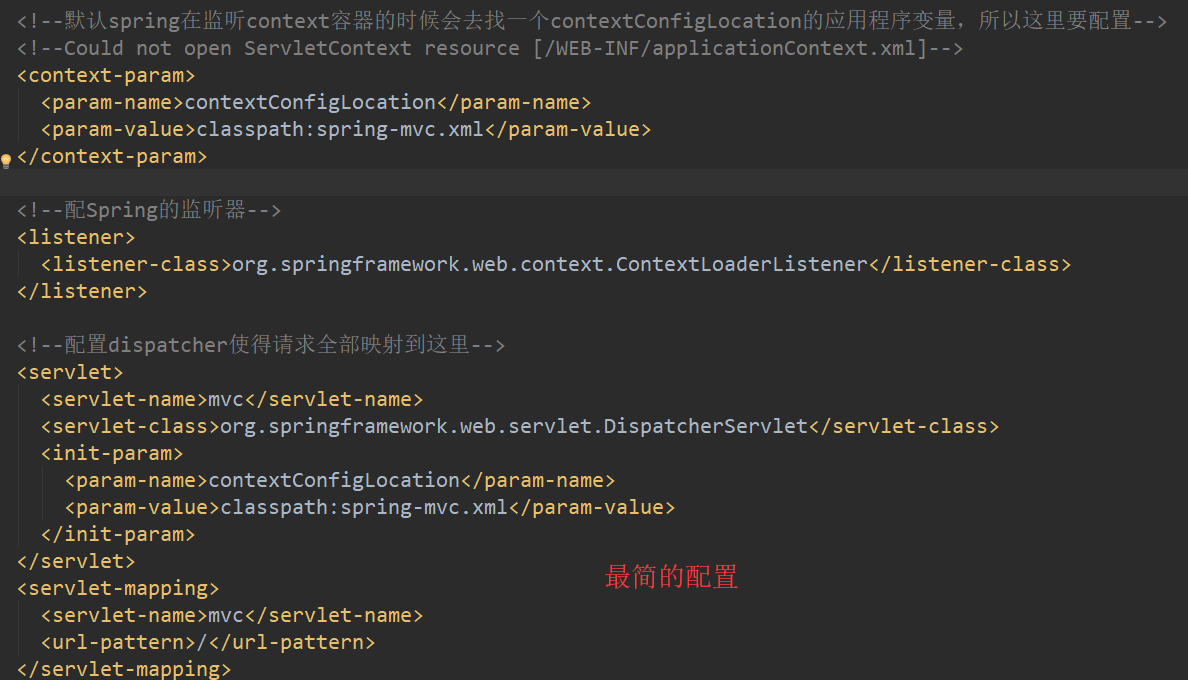
数据包：mysql-connector-java

mybatis包：mybatis

mybatis和spring集成：mybatis-spring

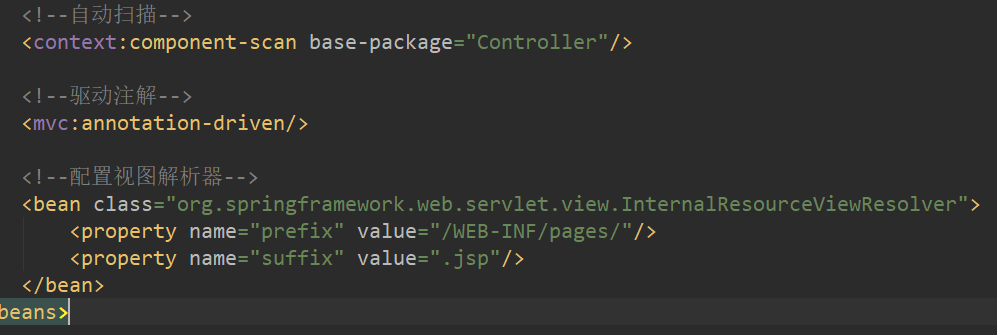
在写java文件的时候 要去tomcat lib目录引入servlet-api.jar包 不然程序会报错

web.xml文件也要配置,



然后编写spring的配置文件：主要就是自动扫描、驱动注解、视图解析器的配置

配置form-data请求要配置一个MultipartResolver





WEB-INF里面的jsp文件在渲染的时候都是从根目录开始渲染的。

使用summernote的时候注意summernote.css文件里的url引用字体问题。

form enctype=form-data的时候 要配置CommonsMultipartResolver并且设置id才可以自动封装,如果属性中有File,一定要是CommonsMultipartFile的类型。否则会注入失败。

这个时候的request其实是一个MultipartHttpServletRequest



上传文件

CommonsMultipartResolver 依赖apache的commons-upload包，并且一定要配置一个id是multipartResolver.同时这个时候的file实例可以用CommonsMultipartFile实例file不可以接收。实例是一个

rejected value [org.springframework.web.multipart.commons.CommonsMultipartFile@6d885b60]

Failed to convert property value of type 'org.springframework.web.multipart.commons.CommonsMultipartFile' to required type 'java.io.File' for property 'file'

将File file改为File[] file

Cannot convert value of type 'org.springframework.web.multipart.commons.CommonsMultipartFile' to required type 'java.io.File' for property 'file[0]

summernote的callback的onImageUpload函数应该是在一添加图片的时候就执行了，默认编辑器会自动把图片转为base64编码，这个时候就需要自己重写这个函数，返回一个新的<image>节点，而不是默认的base64编码字符串。