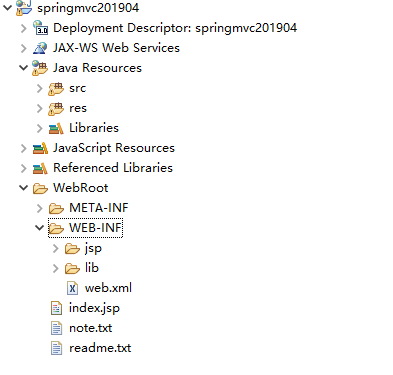
# springMVC搭建过程

## 创建web项目,目录结构如下



## 导入基础jar包和JSON包 放置 WENINO/lib

fastjson-1.2.4.jar

jackson-annotations-2.8.1.jar

jackson-core-2.8.1.jar

jackson-databind-2.8.1.jar

mysql-connector-java-5.0.8-bin.jar

aopalliance-1.0.jar

spring-aop-4.3.18.RELEASE.jar

spring-aspects-4.3.18.RELEASE.jar

spring-beans-4.3.18.RELEASE.jar

spring-context-4.3.18.RELEASE.jar

spring-context-support-4.3.18.RELEASE.jar

spring-core-4.3.18.RELEASE.jar

spring-expression-4.3.18.RELEASE.jar

spring-instrument-4.3.18.RELEASE.jar

spring-instrument-tomcat-4.3.18.RELEASE.jar

spring-jdbc-4.3.18.RELEASE.jar

spring-jms-4.3.18.RELEASE.jar

spring-messaging-4.3.18.RELEASE.jar

spring-orm-4.3.18.RELEASE.jar

spring-oxm-4.3.18.RELEASE.jar

spring-test-4.3.18.RELEASE.jar

spring-tx-4.3.18.RELEASE.jar

spring-web-4.3.18.RELEASE.jar

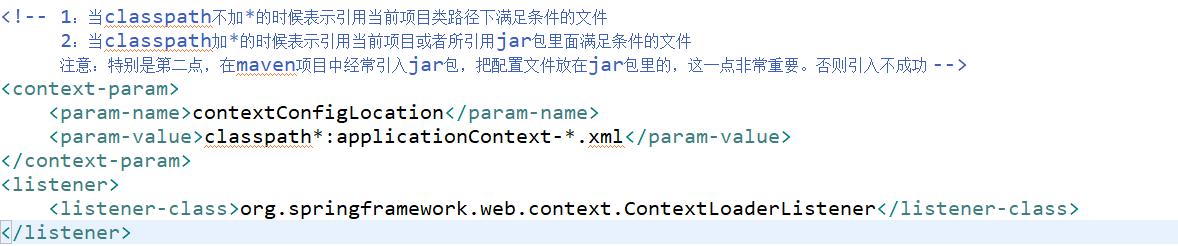
spring-webmvc-4.3.18.RELEASE.jar

spring-webmvc-portlet-4.3.18.RELEASE.jar

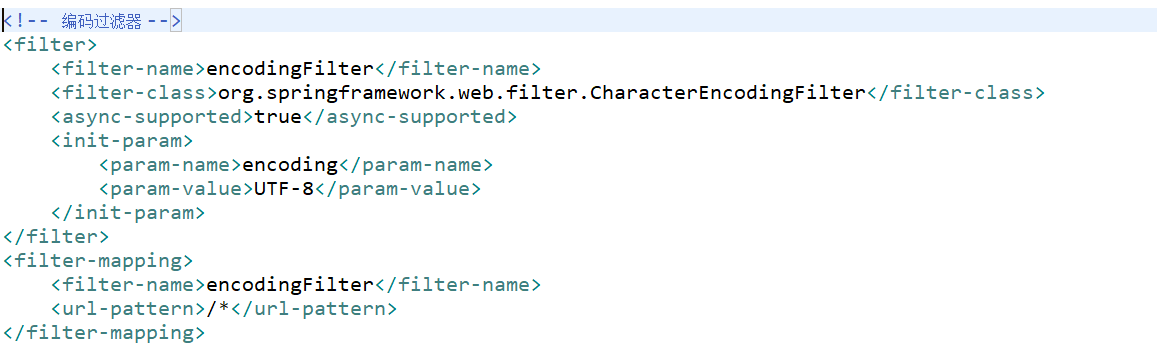
spring-websocket-4.3.18.RELEASE.jar

## 配置 web.xml 文件

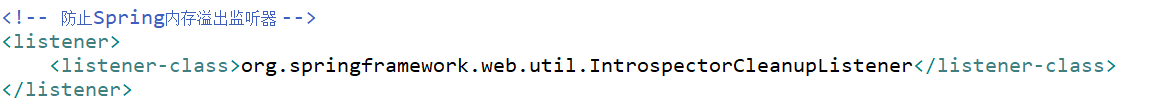
1. 配置springMVC context 上下文监听，用于确保容器加载spring配置



1. 配置全局编码过滤器，确实编码一致



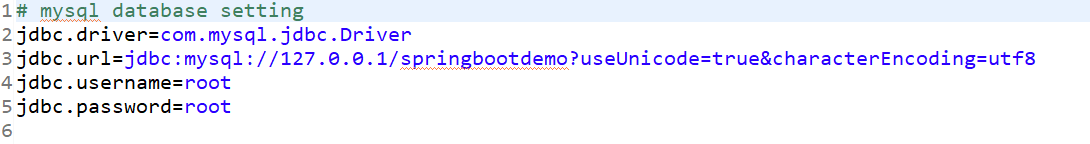
1. 配置spring 内存监听器，防止内存溢出



1. 配置 springMVC dispatcherServlet 拦截处理所有请求



## 配置 config/applicationContext.properties 文件



## 配置 config/applicationContext-spring.xml 文件

（a） 加载porperties 资源文件，启用组件扫描，配置视图解析器



（b） 启用注解驱动，配置消息message转换器



（c） 配置文件上传组件

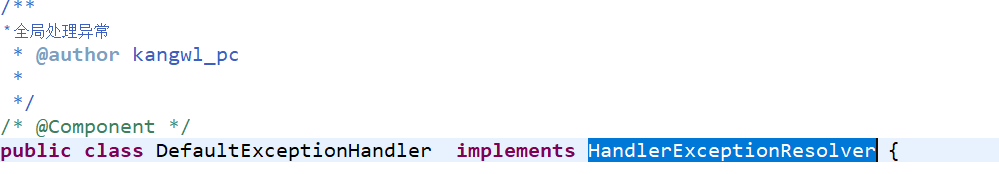


（d） 配置数据源，启用注解事物，启用切面编程

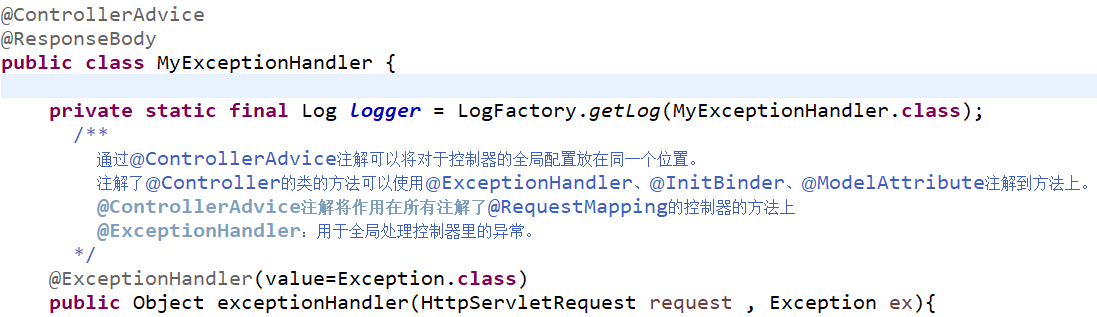


## 全局异常处理

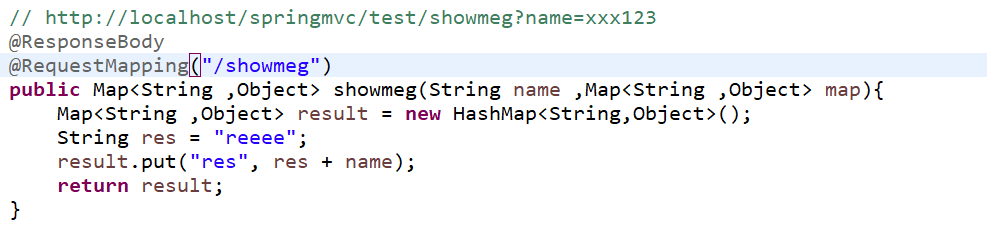
（a）、实现HandlerExceptionResolver接口，返回错误页面视图



（b）、注解方式@ControllerAdvice实现，返回错误信息（JSON）



## 7、创建controller 类，编写测试方法

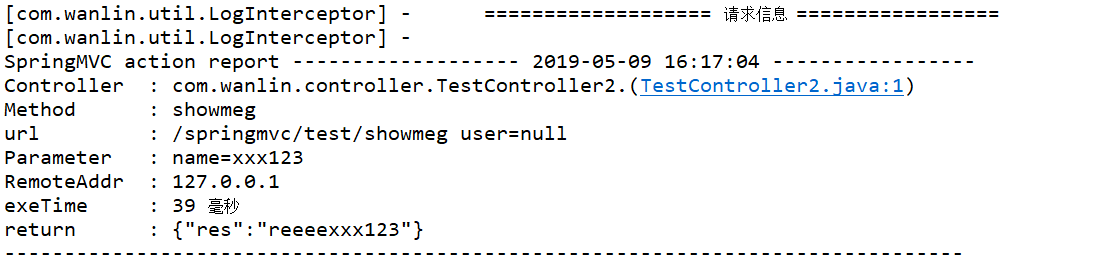


## 8、部署tomcat, 启动服务器，访问url效果如下

（a） 浏览器效果

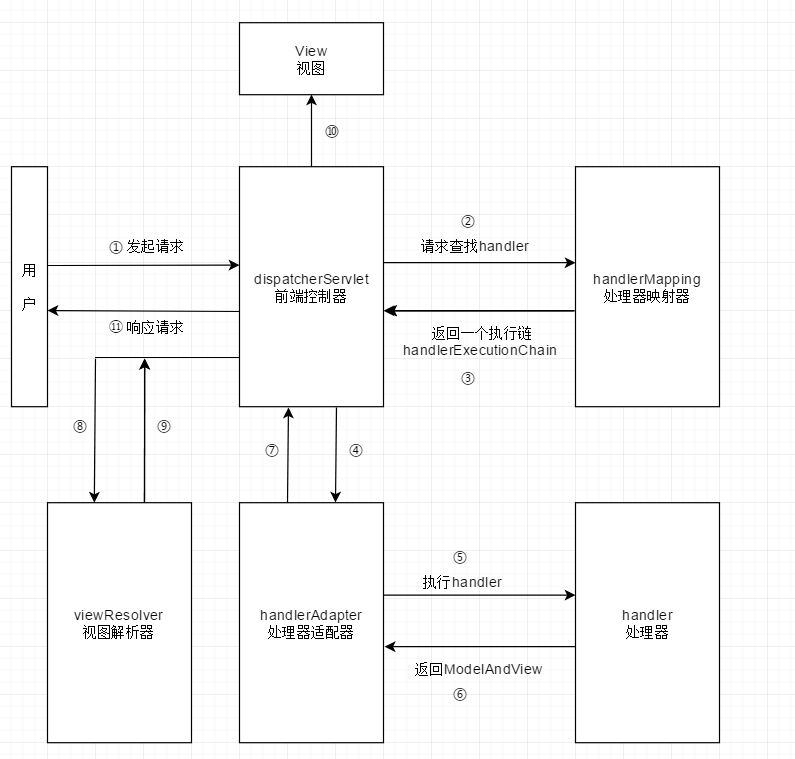


（b） 日志文件效果



# 二、springMVC基本工作流程（原理）

## 基本执行原理图



## 2、 实现过程

① 用户发起请求到前端控制器（dispatcherServlet），该控制器会过滤哪些请求可以访问Servlet,哪些不能访问，并加载springmvc.xml文件

② 前端控制会通过处理器映射器（handlerMapping）获取到对应的handler,获取方式可以是xml配置或者注解

③ 处理器映射器向前端控制器（dispatcherServlet）返回执行链（handlerExcecutionChain）

④ 前端控制器（dispatchServlet）调用处理器适配器（handlerAdapter）执行handler

⑤ 处理适配器执行handler（controller）

⑥ handler执行完毕后，给处理器适配器（handlerAdapter）返回ModelAndView

⑦ 处理器适配器(handlerAdapter)向前端控制器(dispatcherServlet)返回ModelAndView

⑧ 前端控制器请求视图解析器(ViewResolver)去进行视图解析，根据逻辑视图名解析成真正的视图(jsp)，其实就是将ModelAndView对象中存放视图的名称进行查找，找到对应的页面形成视图对象

⑨ 视图解析器（viewResolver）向前端控制器（dispatcherServlet）返回View对象

⑩ 视图渲染，就是将ModelAndView对象中的数据放到request域中，用来让页面加载数据的，即将模型数据填充至视图（View）中

⑪ 响应请求，将请求结果返回给客户端

## **3、springMVC中的几个组件**

前端控制器（DispatcherServlet）：接收请求，分发请求，响应结果

处理器映射器（HandlerMapping）：根据URL去查找处理器

处理器（Handler）：（需要程序员去写代码处理逻辑的）

处理器适配器（HandlerAdapter）：会把处理器包装成适配器，这样就可以支持多种类型的处理器，执行处理器（handler）

视图解析器（ViewResovler）：进行视图解析，多返回的字符串，进行处理，可以解析成视图对象

# 三、springMVC整合mybaties

## 1、导入相应jar包

mybatis-3.4.5.jar

mybatis-spring-1.3.1.jar

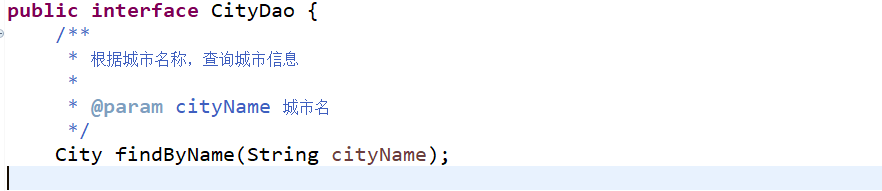
## 2、配置applicationContext-mybaties.xml



## 3、配置 config-mybatis.xml



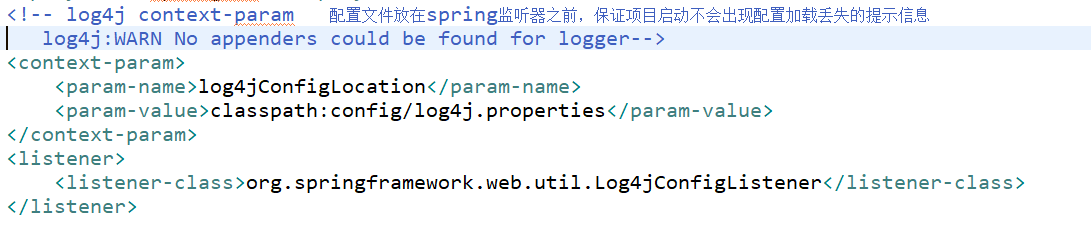
## 4、编写dao接口和mapper.xml文件





# 四、springMVC整合log4j

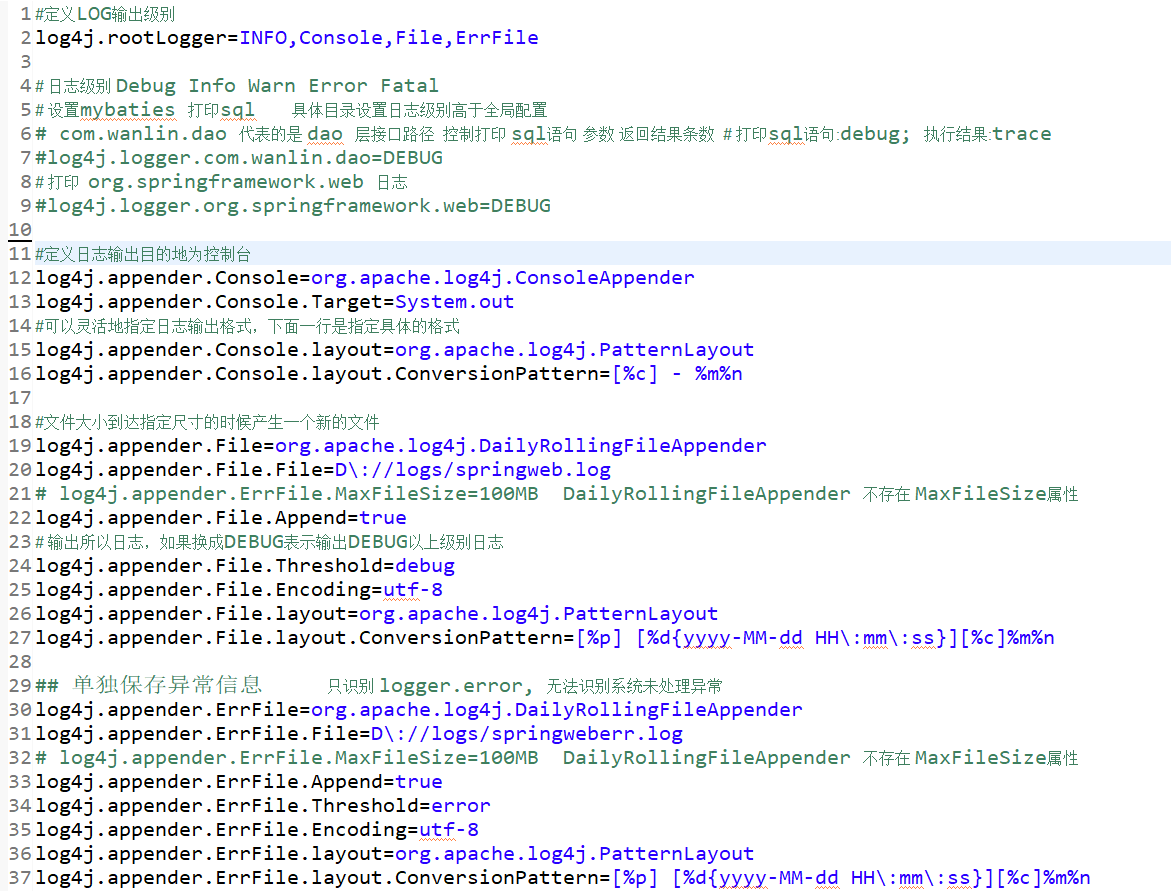
## web.xml 配置日志监听



## 2、导入相应jar包

log4j-1.2.17.jar

## 3、配置log4j.properties文件



# 五、springMVC 整合aspectj

## 1、导入相应jar

aspectjrt-1.9.1.jar

aspectjweaver-1.9.1.jar

## 2、applicationContext-spring.xml 中配置启动支持



## 3、编写实现类 LogInterceptor

