# Springmvc+mybatis框架配置

###### （内含日志+多数据源配置+事务配置+maven配置）

## spring-mvc.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation=" http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.1.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.1.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd ">

<!-- 自动扫描该包，使SpringMVC认为包下用了@controller注解的类是控制器 -->

<context:component-scan base-package="com.andy.manager"/>

<!-- 启动SpringMVC的注解功能，完成请求和注解POJO的映射 -->

<bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter"></bean>

<!-- 定义跳转的文件的前后缀 ，视图模式配置-->

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<!-- 这里的配置我的理解是自动给后面action的方法return的字符串加上前缀和后缀，变成一个 可用的url地址 -->

<property name="prefix" value="/"/>

<property name="suffix" value=".jsp"/>

</bean>

<!-- 配置文件上传，如果没有使用文件上传可以不用配置，当然如果不配，那么配置文件中也不必引入上传组件包 -->

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<!-- 默认编码 -->

<property name="defaultEncoding" value="utf-8"/>

<!-- 文件大小最大值 -->

<property name="maxUploadSize" value="10485760000"/>

<!-- 内存中的最大值 -->

<property name="maxInMemorySize" value="40960"/>

</bean>

<mvc:interceptors>

<!-- 使用bean定义一个Interceptor，直接定义在mvc:interceptors根下面的Interceptor将拦截所有的请求 -->

<bean class="com.andy.manager.common.interceptor.CommonInterceptor"/>

</mvc:interceptors>

</beans>

## spring-mybatis.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.0.xsd

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd">

<context:annotation-config />

<!-- spring mvc 自动扫描注解的时候，不去扫描@Service -->

<context:component-scan base-package="com.andy.manager">

<!--将Controller的注解排除掉 -->

<context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller" />

</context:component-scan>

<util:properties id="app" location="classpath:app-${env}.properties"/>

<!--<bean id="dataSource\_1" class="org.springframework.jndi.JndiObjectFactoryBean">-->

<!--<property name="jndiName" value="#{app.jndiName\_member}"/>-->

<!--</bean>-->

<bean id="dataSource\_1" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<property name="driverClass" value="#{app.member\_jdbcDriver}"/>

<property name="jdbcUrl" value="#{app.member\_jdbcUrl}"/>

<property name="user" value="#{app.member\_jdbcUsername}"/>

<property name="password" value="#{app.member\_jdbcPassword}"/>

<property name="initialPoolSize" value="#{app.member\_jdbcInitialPoolSize}"/>

<property name="minPoolSize" value="#{app.member\_jdbcMinPoolSize}"/>

<property name="maxPoolSize" value="#{app.member\_jdbcMaxPoolSize}"/>

<property name="acquireIncrement" value="#{app.member\_jdbcAcquireIncrement}"/>

<property name="numHelperThreads" value="#{app.member\_jdbcNumHelperThreads}"/>

<property name="maxIdleTime" value="#{app.member\_jdbcMaxIdleTime}"/>

<property name="maxStatements" value="#{app.member\_jdbcMaxStatements}"/>

<property name="maxStatementsPerConnection" value="#{app.member\_jdbcMaxStatementsPerConnection}"/>

<property name="acquireRetryAttempts" value="#{app.member\_jdbcAcquireRetryAttempts}"/>

<property name="idleConnectionTestPeriod" value="#{app.member\_jdbcIdleConnectionTestPeriod}"/>

<property name="testConnectionOnCheckout" value="#{app.member\_jdbcTestConnectionOnCheckout}"/>

</bean>

<!-- spring和MyBatis完美整合，不需要mybatis的配置映射文件 -->

<bean id="sqlSessionFactory\_1" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource\_1"/>

<property name="configLocation" value="classpath:spring/vazh\_mybatis.xml"/>

</bean>

<!-- DAO接口所在包名，Spring会自动查找其下的类 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="com. andy.manager.basinfo.mapper"/>

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory\_1"></property>

</bean>

<!-- DAO接口所在包名，Spring会自动查找其下的类 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="com. andy.manager.basinfo.vip.mapper"/>

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory\_1"></property>

</bean>

<bean name="transactionManager\_1" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

<property name="dataSource" ref="dataSource\_1"></property>

</bean>

<tx:advice id="userTxAdvice\_1" transaction-manager="transactionManager\_1">

<tx:attributes>

<tx:method name="find\*" propagation="SUPPORTS"/>

<tx:method name="get\*" propagation="SUPPORTS"/>

<tx:method name="select\*" propagation="SUPPORTS"/>

<tx:method name="add\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="save\*" propagation="REQUIRED" rollback-for="java.lang.RuntimeException" />

<tx:method name="update\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="modify\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="edit\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="validate\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="del\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="remove\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="clear\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="repair\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="deleteAndRepair\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="change\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="do\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="get\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="find\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="load\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="search\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="datagrid\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="\*" propagation="REQUIRED" read-only="false" />

</tx:attributes>

</tx:advice>

<aop:config>

<!--定义在service包里的任意方法的执行：-->

<aop:pointcut id="pc\_1" expression="execution(public \* com. andy.manager.basinfo.serviceimpl.\*Impl.\*(..))"/>

<aop:advisor pointcut-ref="pc\_1" advice-ref="userTxAdvice\_1"/>

</aop:config>

<!--<bean id="dataSource2" class="org.springframework.jndi.JndiObjectFactoryBean">-->

<!--<property name="jndiName" value="#{app.jndiName\_trans}"/>-->

<!--</bean>-->

<bean id="dataSource2" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<property name="driverClass" value="#{app.acccore\_jdbcDriver}"/>

<property name="jdbcUrl" value="#{app.acccore\_jdbcUrl}"/>

<property name="user" value="#{app.acccore\_jdbcUsername}"/>

<property name="password" value="#{app.acccore\_jdbcPassword}"/>

<property name="initialPoolSize" value="#{app.acccore\_jdbcInitialPoolSize}"/>

<property name="minPoolSize" value="#{app.acccore\_jdbcMinPoolSize}"/>

<property name="maxPoolSize" value="#{app.acccore\_jdbcMaxPoolSize}"/>

<property name="acquireIncrement" value="#{app.acccore\_jdbcAcquireIncrement}"/>

<property name="numHelperThreads" value="#{app.acccore\_jdbcNumHelperThreads}"/>

<property name="maxIdleTime" value="#{app.acccore\_jdbcMaxIdleTime}"/>

<property name="maxStatements" value="#{app.acccore\_jdbcMaxStatements}"/>

<property name="maxStatementsPerConnection" value="#{app.acccore\_jdbcMaxStatementsPerConnection}"/>

<property name="acquireRetryAttempts" value="#{app.acccore\_jdbcAcquireRetryAttempts}"/>

<property name="idleConnectionTestPeriod" value="#{app.acccore\_jdbcIdleConnectionTestPeriod}"/>

<property name="testConnectionOnCheckout" value="#{app.acccore\_jdbcTestConnectionOnCheckout}"/>

</bean>

<!-- spring和MyBatis完美整合，不需要mybatis的配置映射文件 -->

<bean id="sqlSessionFactory2" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource2"/>

<property name="configLocation" value="classpath:spring/vayw\_mybatis.xml"/>

</bean>

<!-- DAO接口所在包名，Spring会自动查找其下的类 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="com. andy.manager.work.\*.mapper"/>

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory2"></property>

</bean>

<bean name="transactionManager\_2" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

<property name="dataSource" ref="dataSource2"></property>

</bean>

<tx:advice id="userTxAdvice\_2" transaction-manager="transactionManager\_2">

<tx:attributes>

<tx:method name="find\*" propagation="SUPPORTS"/>

<tx:method name="get\*" propagation="SUPPORTS"/>

<tx:method name="select\*" propagation="SUPPORTS"/>

<tx:method name="add\*" propagation="REQUIRED" rollback-for="Exception" />

<tx:method name="save\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="update\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="modify\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="edit\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="validate\*" propagation="REQUIRED" rollback-for="Exception" />

<tx:method name="del\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="remove\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="clear\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="repair\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="deleteAndRepair\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="change\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="do\*" propagation="REQUIRED" rollback-for="java.lang.Exception" />

<tx:method name="get\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="find\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="load\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="search\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="datagrid\*" propagation="NOT\_SUPPORTED" read-only="true" />

<tx:method name="\*" propagation="REQUIRED" read-only="false" />

</tx:attributes>

</tx:advice>

<aop:config>

<!--定义在service包里的任意方法的执行：-->

<aop:pointcut id="pc\_2" expression="execution(public \* com. andy.manager.work..serviceimpl..\*Impl.\*(..) )"/>

<aop:advisor pointcut-ref="pc\_2" advice-ref="userTxAdvice\_2"/>

</aop:config>

<tx:annotation-driven transaction-manager="transactionManager\_1"/>

<tx:annotation-driven transaction-manager="transactionManager\_2"/>

</beans>

### logback.xml

<?xml version="1.0" encoding="UTF-8"?>

<configuration>

<appender name="logfile" class="ch.qos.logback.core.rolling.RollingFileAppender">

<File>../logs/mempay\_cms/mempay\_cms.log</File>

<encoder>

<Pattern>[%date] [%-5level] %logger %line --%mdc{client} [%X{TRACE\_LOG\_ID}] %msg%n</Pattern>

<charset>UTF-8</charset>

</encoder>

<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">

<fileNamePattern>../logs/mempay\_cms/mempay\_cms.%d{yyyy-MM-dd}.%i.log</fileNamePattern>

<maxHistory>30</maxHistory>

<TimeBasedFileNamingAndTriggeringPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">

<maxFileSize>512MB</maxFileSize>

</TimeBasedFileNamingAndTriggeringPolicy>

</rollingPolicy>

</appender>

<appender name="console" class="ch.qos.logback.core.ConsoleAppender">

<encoder>

<Pattern>[%date] [%-5level] %logger %line --%mdc{client} [%X{TRACE\_LOG\_ID}] %msg%n</Pattern>

</encoder>

</appender>

<logger name="com.ibatis" level="INFO"/>

<logger name="org.springframework" level="INFO"/>

<logger name="java.sql.PreparedStatement" level="INFO"/>

<logger name="org.springframework.web.servlet.DispatcherServlet" level="INFO"/>

<logger name="com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate" level="INFO"/>

<logger name="java.sql" level="INFO"/>

<logger name="org.apache.commons" level="INFO"/>

<logger name="java.sql.Statement" level="INFO"/>

<logger name="org.springframework.web.context.support.XmlWebApplicationContext" level="INFO"/>

<logger name="com.ibatis.common.jdbc.SimpleDataSource" level="INFO"/>

<logger name="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping" level="INFO"/>

<logger name="java.sql.ResultSet" level="INFO"/>

<logger name="java.sql.Connection" level="INFO"/>

<logger name="com.ibatis.common.jdbc.ScriptRunner" level="INFO"/>

<root level="INFO" >

<appender-ref ref="console"/>

<appender-ref ref="logfile"/>

</root>

</configuration>

### web.xml

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5"

xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">

<!-- Spring和mybatis的配置文件 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring-mybatis.xml</param-value>

</context-param>

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<!-- <async-supported>true</async-supported>-->

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Spring监听器 -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- 防止Spring内存溢出监听器 -->

<listener>

<listener-class>org.springframework.web.util.IntrospectorCleanupListener</listener-class>

</listener>

<servlet>

<servlet-name>SpringMVC</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring-mvc.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

<!-- <async-supported>true</async-supported>-->

</servlet>

<servlet-mapping>

<servlet-name>SpringMVC</servlet-name>

<!-- 此处可以可以配置成\*.do，对应struts的后缀习惯 -->

<url-pattern>\*.do</url-pattern >

</servlet-mapping>

<jsp-config>

<taglib>

<taglib-uri>/diyTaglib</taglib-uri>

<taglib-location>/WEB-INF/tlds/diyTaglib.tld</taglib-location>

</taglib>

</jsp-config>

<welcome-file-list>

<welcome-file>login.jsp</welcome-file>

</welcome-file-list>

</web-app>

### pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com. andy.manager</groupId>

<version>1.0-SNAPSHOT</version>

<artifactId>web</artifactId>

<packaging>war</packaging>

<name>Manager Project Web</name>

<properties>

<spring.version>3.1.1.RELEASE</spring.version>

<mybatis.version>3.2.0</mybatis.version>

<junit.version>4.8.2</junit.version>

<quartz.version>2.2.0</quartz.version>

<lombok.version>0.11.4</lombok.version>

<guava.version>15.0</guava.version>

<netty.version>4.0.10.Final</netty.version>

<ftp.version>1.5.3</ftp.version>

<jedis.version>2.4.1</jedis.version>

<memcach.version>2.6.3</memcach.version>

<spymemcach.version>2.9.1</spymemcach.version>

<axis.version>1.4</axis.version>

<slf4j.version>1.7.5</slf4j.version>

<jxl.version>2.6.10</jxl.version>

<!--平台各模块版本管理 -->

<acc\_utils.version>1.0-SNAPSHOT</acc\_utils.version>

<outservapi\_version>1.0-SNAPSHOT</outservapi\_version>

<outserv\_version>1.0-SNAPSHOT</outserv\_version>

<entrybizfront\_version>1.0-SNAPSHOT</entrybizfront\_version>

<!--项目编码方式 -->

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>${lombok.version}</version>

</dependency>

<dependency>

<groupId>com.google.guava</groupId>

<artifactId>guava</artifactId>

<version>${guava.version}</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>${junit.version}</version>

<scope>test</scope>

</dependency>

<!--log-->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<!-- spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.hibernate.common</groupId>

<artifactId>hibernate-commons-annotations</artifactId>

<version>4.0.4.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.3.5.Final</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>cglib</groupId>

<artifactId>cglib-nodep</artifactId>

<version>2.2</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.6.9</version>

</dependency>

<dependency>

<groupId>com.google.code.findbugs</groupId>

<artifactId>jsr305</artifactId>

<version>2.0.1</version>

</dependency>

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.2.4</version>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-collections4</artifactId>

<version>4.0</version>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.1.39</version>

</dependency>

<!-- logback -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-core</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>log4j-over-slf4j</artifactId>

<version>1.7.5</version>

</dependency>

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.1.3</version>

</dependency>

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc14</artifactId>

<version>10.2.0.4.0</version>

</dependency>

<dependency>

<groupId>c3p0</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.1.2</version>

</dependency>

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.3</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.3</version>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.2</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>asm</groupId>

<artifactId>asm</artifactId>

<version>3.0</version>

</dependency>

<dependency>

<groupId>org.apache.httpcomponents</groupId>

<artifactId>httpclient</artifactId>

<version>4.3</version>

</dependency>

<dependency>

<groupId>net.sf.json-lib</groupId>

<artifactId>json-lib</artifactId>

<version>2.3</version>

<classifier>jdk15</classifier>

</dependency>

<!--<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-asm</artifactId>

<version>3.1.0.M1</version>

</dependency>-->

<!-- mybatis -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>${mybatis.version}</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<version>2.3</version>

</dependency>

</dependencies>

<build>

<finalName>mempay\_cms</finalName>

<resources>

<resource>

<directory>src/main/resources</directory>

<filtering>true</filtering>

</resource>

</resources>

<plugins>

<plugin>

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

<artifactId>maven-deploy-plugin</artifactId>

<version>2.7</version>

<configuration>

<skip>true</skip>

</configuration>

</plugin>

<!--<plugin>-->

<!--<groupId>com.alibaba.citrus.tool</groupId>-->

<!--<artifactId>autoconfig-maven-plugin</artifactId>-->

<!--<version>${autoconfig.plugin.version}</version>-->

<!--<configuration>-->

<!--&lt;!&ndash; 要进行AutoConfig的目标文件，默认为${project.artifact.file}。 <dest>${project.artifact.file}</dest> &ndash;&gt;-->

<!--&lt;!&ndash; 配置后，是否展开目标文件，默认为false，不展开。 <exploding>true</exploding> &ndash;&gt;-->

<!--&lt;!&ndash; 展开到指定目录，默认为${project.build.directory}/${project.build.finalName}。-->

<!--<explodedDirectory> ${project.build.directory}/${project.build.finalName}-->

<!--</explodedDirectory> &ndash;&gt;-->

<!--</configuration>-->

<!--<executions>-->

<!--<execution>-->

<!--<phase>package</phase>-->

<!--<goals>-->

<!--<goal>autoconfig</goal>-->

<!--</goals>-->

<!--</execution>-->

<!--</executions>-->

<!--</plugin>-->

</plugins>

</build>

<profiles>

<profile>

<id>dev</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<properties>

<env>dev</env>

</properties>

</profile>

<profile>

<id>prod</id>

<properties>

<env>prod</env>

</properties>

</profile>

<profile>

<id>test</id>

<properties>

<env>test</env>

</properties>

</profile>

</profiles>

</project>