

7、SSM整合

笔记本： spring

创建时间： 2022/4/4 17:26

作者： 雷丰阳

SSM; Spring+SpringMVC+MyBatis

1、导包

1)、Spring

【aop核心】

com.springsource.net.sf.cglib-2.2.0.jar
com.springsource.org.aopalliance-1.0.0.jar
com.springsource.org.aspectj.weaver-1.6.8.RELEASE.jar
spring-aspects-4.0.0.RELEASE.jar

【ioc核心】

commons-logging-1.1.3.jar
spring-aop-4.0.0.RELEASE.jar
spring-beans-4.0.0.RELEASE.jar
spring-context-4.0.0.RELEASE.jar
spring-core-4.0.0.RELEASE.jar
spring-expression-4.0.0.RELEASE.jar

【jdbc核心】

spring-jdbc-4.0.0.RELEASE.jar
spring-orm-4.0.0.RELEASE.jar
spring-tx-4.0.0.RELEASE.jar

【测试】

spring-test-4.0.0.RELEASE.jar

2)、SpringMVC

【ajax】

jackson-annotations-2.1.5.jar
jackson-core-2.1.5.jar
jackson-databind-2.1.5.jar

【数据校验】

hibernate-validator-5.0.0.CR2.jar
hibernate-validator-annotation-processor-5.0.0.CR2.jar
classmate-0.8.0.jar
jboss-logging-3.1.1.GA.jar
validation-api-1.1.0.CR1.jar

【上传下载】

commons-fileupload-1.2.1.jar
commons-io-2.0.jar

【jstl-jsp标准标签库】

jstl.jar
standard.jar

【验证码】

kaptcha-2.3.2.jar

【springmvc核心】

spring-web-4.0.0.RELEASE.jar
spring-webmvc-4.0.0.RELEASE.jar

3)、MyBatis

【mybatis核心】

mybatis-3.4.1.jar
mybatis-spring-1.3.0.jar 【和Spring整合包】

【ehcache整合】

ehcache-core-2.6.8.jar
mybatis-ehcache-1.0.3.jar
log4j-1.2.17.jar
slf4j-api-1.7.21.jar
slf4j-log4j12-1.7.21.jar

4)、其他的

mysql-connector-java-5.1.37-bin.jar
c3p0-0.9.1.2.jar

2、写配置

0、web.xml配置

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">
    <display-name>7.SSM</display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>

    <!-- 配置Spring容器启动: -->
    <!-- needed for ContextLoaderListener -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <!--指定spring配置文件位置 -->
        <param-value>classpath:spring/applicationContext.xml</param-value>
    </context-param>

    <!-- Bootstraps the root web application context before servlet
initialization -->
    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

    <!-- 配置springmvc的前端控制器 -->
    <!-- The front controller of this Spring Web application, responsible
for handling all application requests -->
    <servlet>
        <servlet-name>springDispatcherServlet</servlet-name>
        <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:spring/applicationContext-mvc.xml</param-
value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <!-- Map all requests to the DispatcherServlet for handling -->
    <servlet-mapping>
        <servlet-name>springDispatcherServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <!-- 两个标准配置-->
    <!-- 字符编码 -->
    <filter>
        <filter-name>CharacterEncodingFilter</filter-name>
        <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
        <init-param>
            <param-name>encoding</param-name>
```

```

        <param-value>utf-8</param-value>
    </init-param>
    <init-param>
        <param-name>forceEncoding</param-name>
        <param-value>true</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>CharacterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<!-- 支持Rest风格的过滤器 -->
<filter>
    <filter-name>HiddenHttpMethodFilter</filter-name>
    <filter-
class>org.springframework.web.filter.HiddenHttpMethodFilter</filter-class>
</filter>
    <filter-mapping>
        <filter-name>HiddenHttpMethodFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

</web-app>

```

1、spring配置

2、springmvc的配置

```

<?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:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
    http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd">

    <!-- SpringMVC只扫描控制器；禁用默认的规则 -->
    <context:component-scan base-package="com.atguigu" use-default-
filters="false">
        <context:include-filter type="annotation"
expression="org.springframework.stereotype.Controller"/>
    </context:component-scan>

    <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/pages/"></property>
        <property name="suffix" value=".jsp"></property>
    </bean>

    <!-- 配置文件上传解析器 -->
    <bean id="multipartResolver"
class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
        <property name="defaultEncoding" value="utf-8"></property>
        <property name="maxUploadSize" value="{1024*1024*20}"></property>
    </bean>

    <!-- 扫静态资源 -->
    <mvc:default-servlet-handler/>
    <!-- 扫动态 -->
    <mvc:annotation-driven></mvc:annotation-driven>

</beans>

```

3、mybatis的配置

整合的关键配置

```

<!--2、配置JdbcTemplate操作数据库。pass -->
<!--3、配置使用mybatis操作数据库 -->
<!-- 可以根据配置文件得到sqlSessionFactory -->
<bean id="sqlSessionFactoryBean"
class="org.mybatis.spring.SqlSessionFactoryBean">
    <!-- 指定配置文件位置 -->
    <property name="configLocation" value="classpath:mybatis/mybatis-
config.xml"></property>
    <property name="dataSource" ref="ds"></property>
    <!-- 指定xml映射文件的位置 -->
    <property name="mapperLocations"
value="classpath:mybatis/mapper/*.xml"></property>
</bean>
<!-- 我们要把每一个dao接口的实现加入到ioc容器; -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <!-- 指定dao接口所在的包 -->
    <property name="basePackage" value="com.atguigu.dao"></property>
</bean>

```

4、其他小伙伴的配置

1)、整合步骤:

- 1)、导入整合包; (能将dao的实现加入到容器中)

3、测试