项目目录

- src
  - -main
    - -java
      - -com.cqw
        - -controller
        - -dao
        - -pojo
        - -service
    - -resources
      - -com.cqw.dao(xml文件的包)
    - -webapp
      - -static
        - -css
        - -img
        - -js
      - -WEB-INF
    - -test
      - -java
  - pom.xml

使用到的jar包

Spring

Spring-webmvc :springmvc

Spring-jdbc :Spring对数据库的管理

Spring-test:Spring对测试的支持(非必选)

aspectjweaver:SpringAOP注解支持

数据库

druid:阿里巴巴的数据库连接池

mysql-connector-java:JDBC驱动

Mybatis

mybatis:mybatis核心

mybatis-spring：mybatis对spring的支持

其他

log4j 1.2.x 日志处理

pagehelper 分页插件

mapper (tk.mybatis) 分页插件

jackson JSON数据的处理

jax.servlet-api 对服务器的支持

jstl JSTL标签

commons-fileupload 文件上传

配置文件

web.xml

```
<!--加载spring的配置文件-->
 <context-param>
 <param-name>contextConfigLocation</param-name>
 <param-value>classpath:springConfig.xml</param-value>
 </context-param>
<!--监听服务器的启动，自动初始化Spring的容器-->
 <listener>
 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
 </listener>
<!--SpringMVC的中央处理器，用于分发请求-->
<servlet>
 <servlet-name>dispatcherServlet</servlet-name>
```

```
<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
<!--配置SpringMVC的配置文件路径-->
<init-param>
<param-name>contextConfigLocation</param-name>
<param-value>classpath:springmvcConfig.xml</param-value>
</init-param>
<!--当服务器启动时，自动加载这个servlet-->
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>dispatcherServlet</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
<!-- 设置post请求的文字编码格式-->
<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>
</filter>
<filter-mapping>
<filter-name>characterEncodingFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
```

Spring的配置文件

```
<!--扫描业务层-->
<context:component-scan base-package="com.cqw.service"/>
<!--加载数据库properties文件-->
<context:property-placeholder location="classpath:jdbc.properties"/>
<!--配置数据库连接池-->
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource">
 <property name="driverClassName" value="${jdbc.driver}"/>
 <property name="url" value="${jdbc.url}"/>
 <property name="username" value="${jdbc.username}"/>
 <property name="password" value="${jdbc.password}"/>
```

```
</bean>
<!--配置SqlsessionFactoryBean，用于创建SqlSessionFactory-->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <!--配置数据源-->
  <property name="dataSource" ref="dataSource"/>
  <!--Mybatis的配置 参照：http://www.mybatis.org/mybatis-3/zh/configuration.html -->
  <property name="configuration">
  <!--开启了通用mapper-->
  <bean class="tk.mybatis.mapper.session.Configuration">
  <!--开启小驼峰(pojo属性)命名和蛇形(数据库)命名的转换 例：a_name转化成aName-->
  <property name="mapUnderscoreToCamelCase" value="true"/>
  <!--开启mybatis的日志,方便查看sql语句-->
  <property name="logImpl" value="org.apache.ibatis.logging.log4j.Log4jImpl"/>
  <!--通用mapper的配置 参照： https://gitee.com/free/Mapper/wikis/3.config?sort_id=208207 -->
  <property name="mapperProperties">
  <value>
  notEmpty=true
  </value>
  </property>
  </bean>
  </property>
  <!--MyBatis的插件配置-->
  <property name="plugins">
  <array>
  <!--多个插件就添加多个Bean-->
  <!--加载pageHelper，分页查询 参照：https://pagehelper.github.io/docs/howtouse/ -->
  <bean class="com.github.pagehelper.PageInterceptor">
  <!--pageHelper的配置-->
  <property name="properties">
  <value>
  helperDialect=mysql
  reasonable=true
  </value>
  </property>
  </bean>
  </array>
```

```
 </property>
</bean>
<!--Mybatis的包扫描-->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
 <property name="basePackage" value="com.cqw.dao"/>
</bean>
<!--开启Spring的事务-->
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
 <property name="dataSource" ref="dataSource"/>
</bean>
<!-- 开启事务注解，使用AOP的技术，对Service层添加了@Transactional的类中的方法进行通知-->
<tx:annotation-driven/>
<!--开启AOP的注解-->
<aop:aspectj-autoproxy/>
```

SpringMVC的配置

```
<!--开启SpringMVC的包扫描-->
<context:component-scan base-package="com.cqw.controller"/>
<!--开启SpringMVC的注解-->
<mvc:annotation-driven/>
<!--配置示图解析器-->
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
 <!--前缀-->
 <property name="prefix" value="/WEB-INF/jsp/"/>
 <!--后缀-->
 <property name="suffix" value=".jsp"/>
</bean>
<mvc:default-servlet-handler/>
<!--配置拦截器-->
<mvc:interceptors>
 <mvc:interceptor>
 <!--拦截器要拦截的请求 /**代表所有请求-->
 <mvc:mapping path="/**"/>
 <!--排除某些接口-->
 <mvc:exclude-mapping path="/login"/>
 <!--拦截的处理-->
 <bean class="com.cqw.interceptor.LoginInterceptor(所代表的类)"/>
 </mvc:interceptor>
```

```
23  </mvc:interceptors>
```

Controller层返回值若想返回指定的请求

```
1  return "forward:请求路径"
```