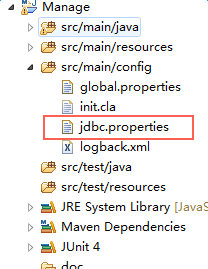
管理后台相关操作

# 1.数据库配置：

## 1.1修改数据库配置：

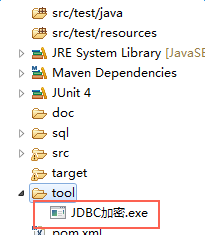
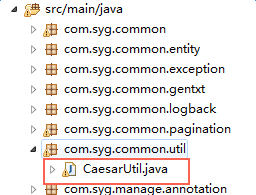
### 1.1.1 配置文件修改



**图1**

在此目录下，将jdbc.properties文件修改为对应的数据库参数后保存。

### 1.1.2 加解密配置文件

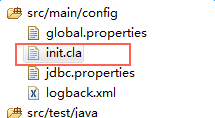


**图2**

针对jdbc.properties文件的加解密提供了两种方式:

方式一:用图2左侧提供的加解密工具类,在工具类的main方法内根绝加密还是解密的需要打开对应的注释。

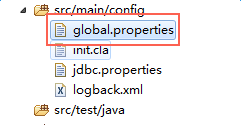
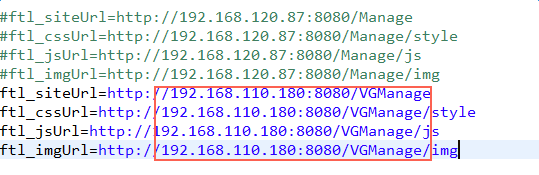
方式二:用图2右侧提供的加解密工具,双击直接运行,将需要加解密的文件拖入后直接操作。



**图3**

**注:使用**方式一进行加解密时需要将控制台的输出内容去覆盖图3中init.cla文件的内容

# 2.更改静态资源访问地址：

**图4**

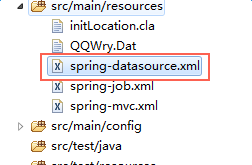
将图4左侧global.properties文件中的静态资源访问地址改成自己主机电脑访问项目的根目录。如图4右图例所示。

# 3.添加新的数据库：

## 3.1修改数据库配置：

首先进行 [数据库的配置](#_1.1修改数据库配置：)，jdbc.properties中添加新的数据库信息再进行加解密。

## 3.2修改配置文件：



**图5**

在如图5所示spring-datasource.xml文件中,对应的添加以下示例代码中的信息,相应需要修改的地方已经标红。

**以下的操作以userDataSource，userSessionFactory，user\_cfg.xml为例**

### 3.1.1 添加数据源:

<!-- 数据源 配置 -->

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

<property name=*"driverClass"* value=*"${user.driverClassName}"* />

<property name=*"jdbcUrl"* value=*"${user.url}"* />

<property name=*"user"* value=*"${user.username}"* />

<property name=*"password"* value=*"${user.password}"* />

<property name=*"initialPoolSize"* value=*"1"* />

<property name=*"minPoolSize"* value=*"1"* />

<property name=*"maxPoolSize"* value=*"3"* />

<property name=*"maxIdleTime"* value=*"600"* />

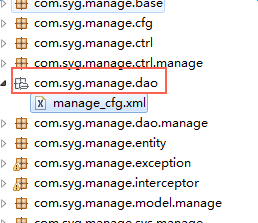
<property name=*"maxStatements"* value=*"100"* />

<property name=*"acquireIncrement"* value=*"1"* />

<property name=*"automaticTestTable"* value=*"t\_c3p0\_Test"*/>

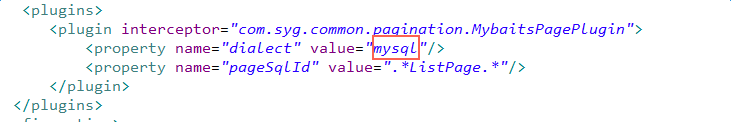
</bean>

### 3.1.2 添加mybatis配置文件:



**图6**

如图6所示，在此目录下添加mybatis配置文件,如新建user\_cfg.xml，文件具体内容与manage\_cfg.xml类似。



**图7**

图7为新建user\_cfg.xml内容，若对应为若数据库使用sqlsever、oracle则需做对应的修改，此处使用mysql。

### 3.1.3 mybatis在spring中的配置:

在[spring-datasource.xm](#_3.2修改配置文件：)l下，添加如下示例代码：

(以下*com.syg.manage.model.user, com.syg.manage.dao.user*为自己所建文件夹)

<!-- SessonFactory 配置 -->

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

<property name=*"dataSource"* ref=*"userDataSource"* />

<property name=*"configLocation"* value=*"classpath:com/syg/manage/dao/user\_cfg.xml"* />

<property name=*"typeAliasesPackage"* value=*"com.syg.manage.model.user"* /> ()

</bean>

<bean class=*"org.mybatis.spring.MyBatisExceptionTranslator"*>

<constructor-arg name=*"dataSource"* ref=*"userDataSource"* />

<constructor-arg name=*"exceptionTranslatorLazyInit"* type=*"boolean"* value=*"false"* />

</bean>

<!-- 对应的自动扫描 配置 -->

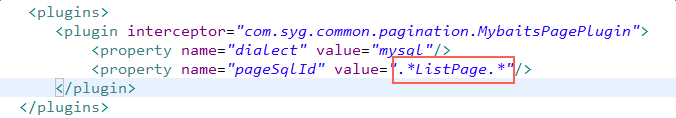
<bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<property name=*"basePackage"* value=*"com.syg.manage.dao.user"* />

<property name=*"sqlSessionFactoryBeanName"* value=*"userSessionFactory"* />

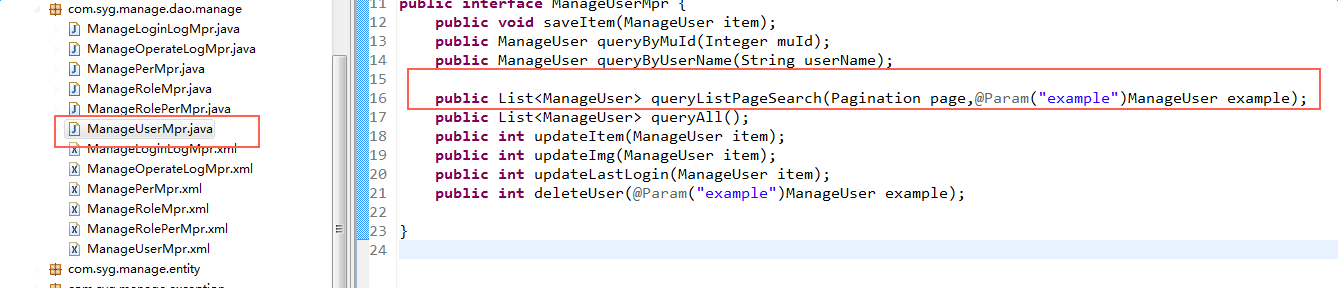
</bean>

# 4.分页操作：



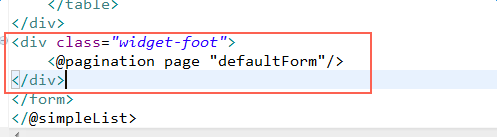
**图8**

如图所示在[user\_cfg.xml](#_3.1.2_添加mybatis配置文件:)中，针对.\*ListPage.\*类似方法名进行了分页拦截(此处也可以自定义)。



**图9**

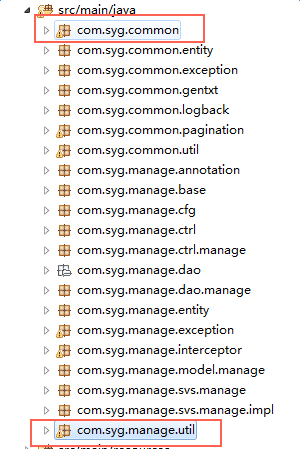
具体使用过程可以参考**src/main/java/com/syg/manage/dao/manage/ManageUserMpr.java**文件中的**queryListPageSearch**方法



**图10**

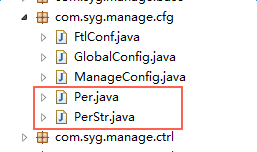
页面中在form表单结束之前添加如图10所示的div层。

# 5.说明：



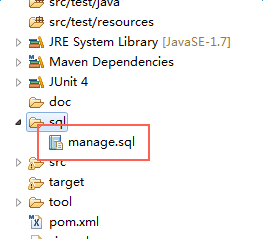
**图11**

图11放置了一些过程中使用到的相关工具类



**图12**

图12与权限相关的枚举类，修改权限参考添加



**图13**

图13后台用户管理相关的基本数据表(Mysql)