# Java JdbcTemplate配置

1. 配置公共数据源， 在WEB-INF文件夹下添加common-beans.xml的spring bean， 添加数据源内容如下

<!-- properties配置连接字符串 -->

<context:property-placeholder location="classpath:application.properties" />

<bean id="dataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="${jdbc.driverClassName}" />

<property name="url" value="${jdbc.url}" />

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

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

</bean>

其中配置信息在properties文件中， 在resources文件夹下添加application.properties文件

1. 在resources文件夹下添加beans文件夹存放bean配置， 添加JTemplate-config.xml的spring bean配置文件，

<import resource="classpath\*:/common-beans.xml" />

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate"

abstract="false" lazy-init="false" autowire="default">

<property name="dataSource">

<ref bean="dataSource" />

</property>

</bean>

1. 添加实体类UserInfo, 与数据库对应
2. 调用代码执行

public String query() {

WebApplicationContext wac = ContextLoader.getCurrentWebApplicationContext();

JdbcTemplate jdbcT = (JdbcTemplate)wac.getBean("jdbcTemplate");

List<UserInfo> books = new ArrayList<UserInfo>();

String sql = "select \* from MyUser";

List list = jdbcT.queryForList(sql);

System.out.println("myjdbctemplate 查询数据记录数:" + list.size());

return "OK";

}

public int delete(int bid) {

WebApplicationContext wac = ContextLoader.getCurrentWebApplicationContext();

JdbcTemplate jdbcT = (JdbcTemplate)wac.getBean("jdbcTemplate");

String sql = "delete from BookInfo where bid =?";

return jdbcT.update(sql, new Object[] { bid });

}

# 其它配置