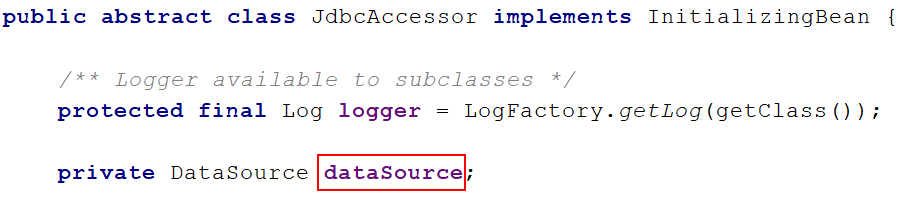
使用JDBC模板，首先需要配置好数据源，数据源直接以Bean的形式配置在Spring配置文件中。

JDBC模板类JdbcTemplate从其父类JdbcAccesor继承了一个属性dataSource，用于接收数据源。





## 1、JdbcTemplate主要提供五类方法：

### execute方法：可以用于执行任何SQL语句，一般用于执行DDL语句；

### update方法及batchUpdate方法：update用于执行新增、修改、删除等语句；batchUpdate用于执行批处理相关语句；

### query方法及queryForXXX方法：用于执行查询相关语句；

int queryForInt(String sql)返回记录数量：

queryForInt(SELECT COUNT(\*) FORM USER);

List<Map<String, Object>> queryForList(String sql)返回一个装有map的list，每

一个map是一条记录，map里的key是字段名。

queryForList(“SELECT A.\* FROM USER A”);

Map<String, Object> queryForMap(String sql)这个查询只能查询一条记录，返回

一个map，key的值是属性的值。

queryForMap(SELECT COUNT(\*) AS keyval FROM USER)；

map.get(“keyval”);

### call方法：用于执行存储过程、函数相关语句。

## 2、JdbcTemplate类支持的回调类：

### 1）预编译语句及存储过程创建回调：用于根据JdbcTemplate提供的连接创建相应的语句

PreparedStatementCreator：通过回调获取JdbcTemplate提供的Connection，由用户使用该Connection创建相关的PreparedStatement；

CallableStatementCreator：通过回调获取JdbcTemplate提供的Connection，由用户使用该Connection创建相关的CallableStatement；

### 2）预编译语句设置回调：用于给预编译语句相应参数设值

PreparedStatementSetter：通过回调获取JdbcTemplate提供的PreparedStatement，由用户来对相应的预编译语句相应的参数值；

BatchPreparedStatementSetter：类似于PreparedStatementSetter，但用于批处理，需要指定批处理大小；

### 3）自定义功能回调：提供给用户一个扩展点，用户可以在指定类型的扩展点执行任何数量需要的操作

CollectionCallback：通过回调获取JdbcTemplate提供的Connection，用户可在该Connecion执行任何数据的操作；

StatementCallback：通过回调获取JdbcTemplate提供的Statement，用户可以在该Statement执行任何数量的操作；

PreparedStatementCallback：通过回调获取JdbcTemplate提供的PreparedStatement，用户可以在该PreparedStatement执行任何数量的操作；

CallableStatementCallback：通过回调获取JdbcTemplate提供的CallableStatement，用户可以在该CallableStatement执行任何数量的操作；

### 4）结果集处理回调：通过回调处理ResultSet或将ResultSet转换成需要的形式

RowMapper：用于将结果集每行数据转换为需要的类型，用户需实现方法mapRow(ResultSetrs,introwNum)来完成将每行数据转换为相应的类型。

RowCallbackHandler：用于处理ResultSet的每一行结果，用户需实现方法processRow(ResultSetrs)来完成处理，在该回调方法中无需执行rs.next()，该操作由JdbcTemplate来执行，用户只需按行获取数据然后处理即可。

ResultSetExtractor：用于结果集数据提取，用户需实现方法extractData(ResultSetrs)来处理结果集，用户必须处理整个结果集；