Spring JDBC

All the classes in Spring JDBC are divided into four separate packages:

core — the core functionality of JDBC. Some of the important classes under this package include JdbcTemplate, SimpleJdbcInsert, SimpleJdbcCall and NamedParameterJdbcTemplate.

datasource — utility classes to access a data source. It also has various data source implementations for testing JDBC code outside the Jakarta EE container.

object — DB access in an object-oriented manner. It allows running queries and returning the results as a business object. It also maps the query results between the columns and properties of business objects.

support — support classes for classes under core and object packages, e.g., provides the SQLException translation functionality

Configuration:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN"

"http://www.springframework.org/dtd/spring-beans-2.0.dtd">

<beans>

<bean id=*"connection"*

class=*"org.springframework.jdbc.datasource.DriverManagerDatasource"*>

<property name=*"driverClassName"*

value=*"com.mysql.jdbc.Driver"* />

<property name=*"url"*

value=*"jdbc:mysql://localhost:3306/student"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"root"* />

</bean>

<bean id=*"jdbcObj"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<constructor-arg>

<ref bean =*"connection"*/>

</constructor-arg>

</bean>

<bean id=*"jobj"* class=*"com.ex.SpringJdbc"*>

<property name=*"jdbcTemplate"*>

<ref bean=*"jdbcObj"*></ref>

</property>

</bean>

</beans>

Basic Queries:

The JDBC template is the main API through which we'll access most of the functionality that we're interested in:

* creation and closing of connections
* running statements and stored procedure calls
* iterating over the ResultSet and returning results
* int result = jdbcTemplate.queryForObject(

"SELECT COUNT(\*) FROM EMPLOYEE", Integer.class);

* public int addEmplyee(int id) {

return jdbcTemplate.update(

"INSERT INTO EMPLOYEE VALUES (?, ?, ?, ?)", id, "Bill", "Gates", "USA");

}

* SqlParameterSource namedParameters = new MapSqlParameterSource().addValue("id", 1);

return namedParameterJdbcTemplate.queryForObject(

"SELECT FIRST\_NAME FROM EMPLOYEE WHERE ID = :id", namedParameters, String.class);