Spring JDBC

Table of Contents

[Agenda 2](#_Toc88751932)

[Spring JDBC 2](#_Toc88751933)

[JDBCTemplate 2](#_Toc88751934)

[DataSource 2](#_Toc88751935)

[JDBCTemplate methods 3](#_Toc88751936)

[Creating a new Spring JDBC project 3](#_Toc88751937)

[Row Mapper Concept | Fetching data 4](#_Toc88751938)

# Agenda

What is Spring JDBC?

Introduction to Spring JDBC?

Problems of JDBC?

How Spring JDBC solves these problems?

# Spring JDBC

Spring JDBC is a powerful mechanism to connect to the database and execute SQL queries

JDBC is an API to connect and transact with database

Problems of JDBC

1. We need to write a lot of code.

- connection open

- statement

- execute

- close

2. Exception handling problem: checked exceptions, multiple try{}catch{} SQLException

3. Repeating of all these codes from class to class, database logic is a time consuming task

Solution of JDBC problems are provided by Spring JDBC.

# JDBCTemplate

Spring JDBC provide a class JDBCTemplate which has all the important methods to perform operations with database

# DataSource

* DataSource Interface
* DriverManagerDataSource is the implementing class
* DataSource d = new DriverManagerDataSource();
* We have to give this object the values such as url, username, password, driver
* We have to supply this dataSource object to JDBC template to perform CRUD

# JDBCTemplate methods

* update() for insert, update, delete queries
* execute() for select queries

# Creating a new Spring JDBC project

1. Create a new Maven projects

2. Insert spring core, spring context, spring jdbc, mysql connector dependencies

**Project: 01SpringJDBC**

**package: spring.jdbc**

CRUD operations:

1. Inserting data to database using Spring JDBC | Best Practice

Project: 01SpringJDBC

package: spring.jdbc, spring.jdbc.entities, spring.jdbc.dao

2. Update Operation using Spring Jdbc

Project: 01SpringJDBC

package: spring.jdbc, spring.jdbc.entities, spring.jdbc.dao

3. Delete Operation using Spring JDBC

Project: 01SpringJDBC

package: spring.jdbc, spring.jdbc.entities, spring.jdbc.dao

4. Select operation

# Row Mapper Concept | Fetching data

RowMapper converts a ResultSet into Object. It maps a row to an object's properties

* Selecting Single Object using Spring Jdbc | Row Mapper
* Selecting Multiple Objects using Spring Jdbc

RowMapper Interface

====================

1. Its is an interface used by JdbcTemplate for mapping rows of a ResultSet on a per-row basis.

2. Implementations of this interface perform the actual work of mapping each row to a result object.

3. It is a functional interface

4. It's abstract method is has T mapRow(ResultSet rs, int rowNum) throws SQLException

----------------------------------------

Spring Jdbc Configuration without XML

Project: 01SpringJDBC

package: spring.jdbc, spring.jdbc.entities, spring.jdbc.dao

----------------------------------

Autowiring with Spring Jdbc

Project: 02SpringJDBCAutoWiring

package: spring.jdbc, spring.jdbc.entities, spring.jdbc.dao