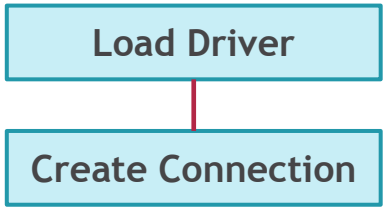
SPRING FRAMEWORK

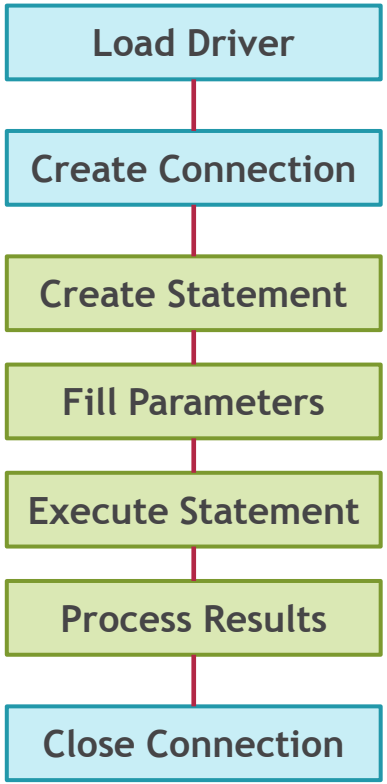
SIMPLE DATABASE ACCESS

APRIL, 2015

SQL QUERY WITH JDBC



SQL QUERY WITH JDBC



SPRING vs YOU

Action	Spring	You
Define connection parameters.		X
Open the connection.	X	
Specify the SQL statement.		X
Declare parameters and provide parameter values		X
Prepare and execute the statement.	X	
Set up the loop to iterate through the results.	X	
Do the work for each iteration.		X
Process any exception.	X	
Handle transactions.	X	
Close the connection, statement and resultset.	X	

DATA ACCESS WITH SPRING

- Template classes
  - `JdbcTemplate`
  - `NamedParameterJdbcTemplate`
  - `SimpleJdbcInsert` and `SimpleJdbcCall`
  - RDBMS Objects
- Consistent exception hierarchy
  - `DataAccessException` as the root exception
  - Runtime exceptions
  - All converted to the same hierarchy

EXAMPLE

DIY

DBLogger

```
- jdbcTemplate: JdbcTemplate;
+ logEvent(...);   // writes to DB
```

EXAMPLE

DIY

DBLogger

```
- jdbcTemplate: JdbcTemplate;
+ logEvent(...);   // writes to DB
```

Derby database and other dependencies:

- ```
  <groupId>org.apache.derby</groupId>
  <artifactId>derby</artifactId>
  <version>10.11.1.1</version>
  ```

- ```
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${spring.ver}</version>
  ```

SPRING XML CONFIGURATION

DIY

```
<bean id="jdbcTemplate"
    class="org.springframework.jdbc.core.JdbcTemplate">
        <constructor-arg ref="dataSource"/>
</bean>
```

SPRING XML CONFIGURATION

DIY

```
<bean id="jdbcTemplate"
    class="org.springframework.jdbc.core.JdbcTemplate">
        <constructor-arg ref="dataSource"/>
</bean>

<bean id="dataSource"
    class="org.springframework.jdbc.datasource.
                                DriverManagerDataSource"
    destroy-method="close">









</bean>
```

SPRING XML CONFIGURATION

DIY

```
<bean id="jdbcTemplate"
    class="org.springframework.jdbc.core.JdbcTemplate">
        <constructor-arg ref="dataSource"/>
</bean>

<bean id="dataSource"
    class="org.springframework.jdbc.datasource.
                                DriverManagerDataSource"
    destroy-method="close">
        <property name="driverClassName"
                        value="${jdbc.driverClassName}"/>
        <property name="url"
                        value="${jdbc.url}"/>
        <property name="username"
                        value="${jdbc.username}"/>
        <property name="password"
                        value="${jdbc.password}"/>
</bean>
```

JDBC.PROPERTIES

DIY

```
jdbc.driverClassName=
        org.apache.derby.jdbc.EmbeddedDriver

jdbc.url=
        jdbc:derby:<PATH_TO_DB>;create=true

jdbc.username=aa

jdbc.password=bb
```

INSERT VALUES

DIY

```
public void logEvent(Event event) {

    jdbcTemplate.update(
  "INSERT INTO t_event (id, msg) VALUES (?,?)",
              event.getId(),
              event.toString());
}
```

QUERY SIMPLE VALUE

```
int count = jdbcTemplate.queryForObject(
      "select count(*) from t_event",
            Integer.class);


// Also possible
int count = jdbcTemplate.queryForInt(sql);
```

QUERY SIMPLE VALUE

```
int count = jdbcTemplate.queryForObject(
      "select count(*) from t_event",
            Integer.class);


// Also possible
int count = jdbcTemplate.queryForInt(sql);


String msg = jdbcTemplate.queryForObject(
      "select msg from t_event where id = ?",
            new Object[]{1}, String.class);
```

QUERY SINGLE OBJECT

```
Event event = jdbcTemplate.queryForObject(
    "select * from t_event where id = ?",
        new Object[]{1},
        new RowMapper<Event>() {
            public Event mapRow(ResultSet rs,
                    int rowNum) throws SQLException {
                Integer id = rs.getDate("id");
                Date date = rs.getDate("date");
                String msg = rs.getString("msg");

                Event event = new Event(id, date);
                event.setMessage(msg);
                return event;
            }
        });
```

QUERY OBJECTS

```
List<Event> events = jdbcTemplate.query (
   "select * from t_event",
      new RowMapper<Event>() {
         public Event mapRow(ResultSet rs,
               int rowNum) throws SQLException {
               //...
               return event;
            }
       });
```

<epam>

THANK YOU

YURIY_TKACH@EPAM.COM

APRIL, 2015