VaanNila

Spring SimpleJdbcTemplate

To use the `SimpleJdbcTemplate` you need to use JDK 1.5 or higher. `SimpleJdbcTemplate` takes advantage of the Java 5 language features like varargs, autoboxing, generics and covariant returns.

```java
package com.vaannila.dao;

import java.sql.ResultSet;
import java.sql.SQLException;

import javax.sql.DataSource;

import org.springframework.jdbc.core.simple.ParameterizedRowMapper;
import org.springframework.jdbc.core.simple.SimpleJdbcTemplate;

import com.vaannila.domain.Forum;

public class ForumDAOImpl implements ForumDAO {

    private SimpleJdbcTemplate simpleJdbcTemplate;

    public void setDataSource(DataSource dataSource) {
        this.simpleJdbcTemplate = new SimpleJdbcTemplate(dataSource);
    }

    @Override
    public void insertForum(Forum forum) {
        String query = "INSERT INTO FORUMS (FORUM_ID, FORUM_NAME,
          FORUM_DESC) VALUES (?,?,?)";
        simpleJdbcTemplate.update(query, forum.getForumId(), forum
                .getForumName(), forum.getForumDesc());
    }

    @Override
    public Forum selectForum(int forumId) {
        String query = "SELECT * FROM FORUMS WHERE FORUM_ID=?";
        return simpleJdbcTemplate.queryForObject(query, new
          ParameterizedRowMapper<Forum>() {
                    @Override
                    public Forum mapRow(ResultSet resultSet, int rowNum)
                       throws SQLException {
                        return new Forum(resultSet.getInt("FORUM_ID"),
                           resultSet.getString("FORUM_NAME"),
                                resultSet.getString("FORUM_DESC"));
                    }
                }, forumId);
    }

}
```

The following snippet shows the `insertForum()` method using the `JDBCTemplate`.

```java
public void insertForum(Forum forum) {
    String query = "INSERT INTO FORUMS (FORUM_ID, FORUM_NAME, FORUM_DESC)
       VALUES (?,?,?)";
    jdbcTemplate.update(query, new Object[] {
       Integer.valueOf(forum.getForumId()),
            forum.getForumName(), forum.getForumDesc() });
}
```

When using `SimpleJDBCTemplate` you can use the variable length arguments instead of an `Object` array. There is no need to explicitly typecast `int` to `Integer`.

```java
@Override
public void insertForum(Forum forum) {
    String query = "INSERT INTO FORUMS (FORUM_ID, FORUM_NAME, FORUM_DESC)
       VALUES (?,?,?)";
    simpleJdbcTemplate.update(query, forum.getForumId(), forum
            .getForumName(), forum.getForumDesc());
}
```

The following snippet shows the `selectForum()` method using the `JDBCTemplate`.

```java
public Forum selectForum(int forumId) {
    String query = "SELECT * FROM FORUMS WHERE FORUM_ID=?";
    return (Forum) jdbcTemplate.queryForObject(query, new Object[] {
       Integer.valueOf(forumId) },
            new RowMapper() {
                public Object mapRow(ResultSet resultSet, int rowNum) throws
                   SQLException {
                    return new Forum(resultSet.getInt("FORUM_ID"),
                       resultSet.getString("FORUM_NAME"),
                            resultSet.getString("FORUM_DESC"));
                }
            });
}
```

Here when using `SimpleJDBCTemplate` you need not explicitly typecast the return object, the `ParameterizedRowMapper` object's type parameter will be taken by default. The return type of the `mapRow()` method is `Forum` instead of `Object` because in Java 5 you can have covariant return types. Since the statement parameter can be of variable length the `forumId` is specified at the end of the list.

```
@Override
public Forum selectForum(int forumId) {
    String query = "SELECT * FROM FORUMS WHERE FORUM_ID=?";
    return simpleJdbcTemplate.queryForObject(query, new
      ParameterizedRowMapper<Forum>() {
                @Override
                public Forum mapRow(ResultSet resultSet, int rowNum) throws
                  SQLException {
                    return new Forum(resultSet.getInt("FORUM_ID"),
                      resultSet.getString("FORUM_NAME"),
                               resultSet.getString("FORUM_DESC"));
                }
            }, forumId);
}
```

You can download and try the example here.

Source :**Download**

Source + Lib :**Download**