So sánh Statement với PreparedStatement, CallableStatement trong JDBC – Java

Posted on *Tháng Một 1, 2017*

So sánh Statement với PreparedStatement, CallableStatement trong JDBC – Java

Statement, PreparedStatement và CallableStatement là 3 loại interace thực thi câu lệnh truy vấn SQL trong JDBC API, trong đó:

- **Statement** – Sử dụng để thực hiện các câu truy vấn SQL tĩnh
- **PreparedStatement** – Sử dụng để thực hiện các câu truy vấn SQL động hoặc có tham số
- **CallableStatement** – Sử dụng để thực thi các stored procedures (Hiểu nôm na là các lệnh định nghĩa sẵn trên database)

Cả 3 interface trên cùng thực hiện công việc khá giống nhau tuy nhiên mỗi loại nên được sử dụng trong từng trường hợp để nâng cao hiệu năng.

Statement

Statement được sử dụng để thực thi các câu lệnh SQL tĩnh, chúng ta không thể truyền tham số vào câu SQL trong thời gian runtime.

Statement có hiệu năng (performance) kém hơn PrepareStatement và CallableStatement .

Statement thường được sử dụngtrong trường hợp câu lệnh SQL chỉ chạy 1 lần, ví dụ sử dụng để thực thi các câu SQL định nghĩa cơ sở dữ liệu – DDL (Data Denifition Language) như CREATE, ALTER, DROP...

Ví dụ:

```
String sql = "CREATE TABLE user_info (" +
    "  id int(11) NOT NULL AUTO_INCREMENT," +
    "  name varchar(45) DEFAULT NULL," +
    "  email varchar(100) DEFAULT NULL," +
    "  address varchar(255) DEFAULT NULL," +
    "  date_of_birth datetime DEFAULT NULL," +
    "  PRIMARY KEY (id)" +
    ")";
Connection con = connectionUtils.getConnection();
Statement stmt = con.createStatement();
stmt.execute(sql);
```

PreparedStatement

PreparedStatement được sử dụng để thực thi các câu truy vấn SQL động hoặc có tham số.

FAQ, NETWORK PROGRAMMING

PreparedStatement thừa kế từ Statement nhưng nó cho phép truyền các tham số vào câu SQL trong thời gian run time.

PreparedStatement được khuyên sử dụng trong trường hợp câu SQL được sử dụng nhiều lần.

Ví dụ câu lệnh INSERT được dùng lại nhiều lần để insert 1 danh sách user vào database.

```
List<User> listUser = new ArrayList<User>();
listUser.add(new User(1, "Rooney", "England"));
listUser.add(new User(2, "Ronaldo", "Brazil"));
listUser.add(new User(3, "Torres", "Spain"));

String sql = "INSERT INTO user_info (id, name, address) VALUES (?, ?, ?);";
PreparedStatement pstmt = con.prepareStatement(sql);

for (User user : listUser) {
  pstmt.setInt(1, user.getId());
  pstmt.setString(2, user.getName());
  pstmt.setString(3, user.getAddress());
  pstmt.execute();
}
```

CallableStatement

CallableStatement được sử dụng để thực thi stored procedures.

CallableStatement được thừa kế từ PreparedStatement và có hiệu năng cao hơn PreparedStatement.

Với CallableStatement, chúng ta có thể truyền 3 loại tham số vào stored procedures là IN (truyền giá trị vào stored procedures), OUT (lấy kết quả trả về từ stored procedures), IN OUT (thực thi cả IN và OUT).

Ví dụ mình tạo một stored procedures thực hiện lấy user theo id:

```
CREATE PROCEDURE getUserById(IN idUser int)
BEGIN
 SELECT * FROM user_info WHERE id = idUser;
END
```

Bây giờ mình sẽ sử dụng CallableStatement để gọi câu stored procedures bên trên:

Truyền id = 1 vào stored procedures getUserById và trả kết quả về ResultSet

```
String sql = "CALL getUserById(?);";
CallableStatement cs = con.prepareCall(sql);
cs.setInt(1, 1);
cs.executeQuery();
ResultSet rs = cs.getResultSet();
while (rs.next()) {
  System.out.println(rs.getInt(1) + "  " + rs.getString(2) + "  " + rs.getString(3));
}
```


- AOP là gì? Aspect Oriented Programming trong Java
- Dependency Injection (DI) là gì? Code ví dụ bằng Java
- Cấu hình giới hạn kích thước file upload cho Tomcat
- Các Scope trong JSP Servlet. Application, Request, Session, Page scope
- Phân biệt save, persist, update, merge, saveOrUpdate trong hibernate
- Sự khác nhau giữa merge với saveOrUpdate trong Hibernate
- Hibernate Batch Processing là gì? Batch Processing trong Hibernate
- Sự khác nhau giữa load() và get() trong Hibernate
- Locking trong Hibernate, so sánh Optimistic lock với Pessimistic lock
- Sự khác nhau giữa String, StringBuffer, StringBuilder?
- JPA là gì? Sự khác nhau giữa JPA với Hibernate
- Một số câu hỏi hay gặp khi phỏng vấn vị trí lập trình Java
- Sự khác nhau giữa openSession() và getCurrentSession() trong Hibernate
- Lập trình hướng đối tượng là gì? Ưu, nhược điểm
- Lập trình cấu trúc là gì? Ưu nhược điểm.
- Lập trình tuyến tính là gì? Ưu

Okay, Done!

Thanks các bạn đã theo dõi bài viết!

References:

http://javaconceptoftheday.com/statement-vs-preparedstatement-vs-callablestatement-in-java/

This entry was posted in *FAQ*, *Network Programming* and tagged *jdbc*. Bookmark the *permalink*.

← JDBC là gì? Kết nối java với mysql bằng JDBC

Hướng dẫn Java JDBC, Ví dụ với Statement trong JDBC →

Trả lời

Email của bạn sẽ không được hiển thị công khai. Các trường bắt buộc được đánh dấu *

Bình luận

Tên *

Email *

Trang web

PHẢN HỒI

nhược điểm.

So sánh lập trình cấu trúc với hướng đối tượng

Servlet Container là gì? Web Server là gì?

Hướng dẫn Java JDBC, Xử lý batch trong JDBC, Code ví dụ Batch Insert

Hướng dẫn Java JDBC: Transaction là gì? Code ví dụ transaction với JDBC

Hướng dẫn Java JDBC, Ví dụ với CallableStatement trong JDBC

Hướng dẫn Java JDBC, Ví dụ với PreparedStatement trong JDBC

Hướng dẫn Java JDBC, Ví dụ với Statement trong JDBC

So sánh Statement với PreparedStatement, CallableStatement trong JDBC – Java

JDBC là gì? Kết nối java với mysql bằng JDBC

strictfp là gì, Từ khóa strictfp trong Java, ví dụ

Connection pool là gì? Khái niệm connection pool trong database

Java 2 là gì? Các version/Phiên bản của Java

Phân biệt giữa Java ME, Java SE và Java EE

So sánh sự khác nhau giữa J2ME, J2SE và J2EE

CHUYÊN MUC

- Algorithm
- Apache
- Apache JMeter
- Apache Kafka
- AWS
- C/C++
- CDI
- Clean Code
- Demo
- Design Pattern
- Docker
- Eclipse
- Elasticsearch
- Excel
- FAQ
- Framework
- Hibernate
- HttpComponents
- Install
- Intellij IDEA
- Java
- Java Basic
- Java Core
- Java8
- JavaScript
- JSF
- JSP-Servlet
- JUnit
- Library

- Linux
- Maven
- MongoDB
- MySQL
- Network Programming
- Node.js
- OOP
- PostgreSQL
- PrimeFaces
- Principle
- Python
- quartz
- Redis
- Security
- Spring
- Spring Boot
- Spring Core
- Spring Data
- Spring Hibernate
- Spring JDBC
- Spring MVC
- Spring Security
- Thymeleaf
- Tomcat
- Uncategorized
- Web Service
- WebSocket
- Wordpress