使用JdbcTemplate的命名参数功能

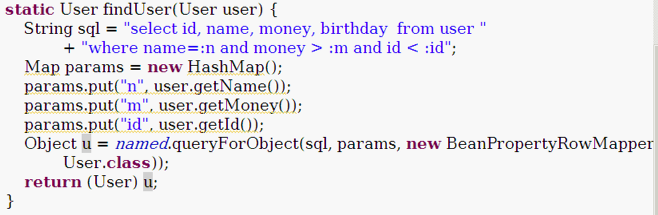
1. 在传入参数的时候，JDBC只能认识用“**？**”作为占位符，但是很容易传入数据时，顺序出现差错。怎么办，可以通过给占位符起个名字，通过名字传入参数，即可解决这个问题。
2. 类NamedParameterJdbcTemplate：就可以处理带有**对占位符起名字**的功能。

**NamedParameterJdbcTemplate *named* = new NamedParameterJdbcTemplate(JdbcUtils.*getDataSource*());**

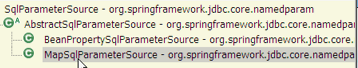
**对占位符起名的方法：**

**利用“：name”名字取代“？”。注意JDBC不能处理该sql语句。**

**通常使用冒号，当然使用&号也是可以的。**



1. 可以利用参数源：两种。
2. **SqlParameterSource**
3. **MapSqlParameterSource**



**SqlParameterSource sps = new BeanPropertySqlParameterSource(user);**

**这句话来代替Map参数。**

**但是sql语句中的占位符的名称必须和User中的属性名相同。**

1. 示例：

**static** NamedParameterJdbcTemplate *named* = **new** NamedParameterJdbcTemplate(

JdbcUtils.*getDataSource*());

**public** **static** **void** main(String[] args) {

User user = **new** User(4, "亚历山大大帝", **new** Date());

User user2 = *findUser*(user);

System.***out***.println(user2.toString());

}

**static** **public** User findUser(User user) {

String sql = "select id,name,birthday from users where id =:id and name = :name";

**SqlParameterSource sps = new BeanPropertySqlParameterSource(user);**

// Map<String, Object> pamramMap = new HashMap<String, Object>();

// pamramMap.put("id", user.getId());

// pamramMap.put("name", user.getName());

Object object = *named*.queryForObject(sql, sps, **new** BeanPropertyRowMapper(User.**class**));

**return** (User)object;

}结果： 

**添加用户**：

**static** **public** **void** addUser(User user) {

String sql = "insert into users(id,name,birthday) values(:id,:name,:birthday)";

SqlParameterSource paramSource = **new** BeanPropertySqlParameterSource(user);

KeyHolder keyHolder = **new** GeneratedKeyHolder();

*named*.update(sql, paramSource, keyHolder);

**int** id = keyHolder.getKey().intValue();

user.setId(id);

Map map = keyHolder.getKeys();

}