



链滴

MyBatis 不定个数参数查询

作者: [ixiaozhi](#)

原文链接: <https://ld246.com/article/1501948718256>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

概述

某些情况下，比如要查询某几个key的value时，需要用 `where id in ()` 进行查询，传入的参数不确定数时，MyBatis 的可以进行特别的配置。

基于 XML 文件配置

利用 XML 的 `foreach` 标签，可以很简易的配置。传入的参数名可以为一个 List，利用 `foreach` 拼接成 SQL。如：

```
<select id="testSearch" resultType="String">
  select value from t_test where id in
  <foreach collection="list" index="index" item="item" open="(" separator="," close=")">
    #{item}
  </foreach>
</select>
```

测试代码：

```
List<String> ids = new ArrayList<String>();
ids.add("test1");
ids.add("test7");
ids.add("test3");
List<String> testBeans = testMapper.testSearch(ids);
```

基于注解 Annotation 配置

使用注解配置的 Mapper 需要使用 `SelectProvider` 来构造 SQL。Mapper 示例如下：

```
import org.apache.ibatis.annotations.SelectProvider;

@SelectProvider(type = TestSelectProviderSQL.class, method = "testSearch")
List<String> testSearch(String... ids);
```

`TestSelectProviderSQL` 类就是用来提示查询所用的 SQL，代码如下：

```
import org.apache.ibatis.jdbc.SQL;
import java.util.Map;
public class TestSelectProviderSQL {
  public String testSearch(Map<String, Object> params) {
    final StringBuffer whereSb = new StringBuffer();
    whereSb.append(" id in (");

    String[] values = ((String[]) params.get("array"));
    for (int i = 0, size = values.length; i < size; i++) {
      if (i == 0) {
        whereSb.append("#{values" + i + "}");
        params.put("values" + i, values[i]);
      } else {
        whereSb.append(",#{values" + i + "}");
        params.put("values" + i, values[i]);
      }
    }
  }
}
```

```

    }
    whereSb.append(")");

    return new SQL() {
        {
            SELECT("value");
            FROM("t_test");
            WHERE(whereSb.toString());
        }
    }.toString();
}
}
}

```

testSearch 参数通过一个 Map 注入，在 Mapper 中传入的参数为一个变长参数，Mapper 对应接的参数在 key=array 中是一个 String 数组。依次取出数组的各个值进行拼接就可以了。调用的测试例如下：

```
List<String> testBeans = testMapper.testSearch("test1","test7","test3");
```

如果 Mapper 中使用普通的参数，如 `List<String> testSearch(String id1, String id2);` 则在 SelectProvider 类中的 Map 数组接收到的值为如下表：

```

<table border="1">
  <tr>
    <td>key</td>
    <td>value</td>
  </tr>
  <tr>
    <td>0</td>
    <td>id1的值</td>
  </tr>
  <tr>
    <td>1</td>
    <td>id2的值</td>
  </tr>
  <tr>
    <td>params1</td>
    <td>id1的值</td>
  </tr>
  <tr>
    <td>params2</td>
    <td>id2的值</td>
  </tr>
</table>

```

使用 `params.get(0)` 或者 `params.get("params1")` 皆可以取出传入的值。需要注意的是，Map 的大是传入参数个数的两倍。

附：排序

我们都知道，这样 `where id in ()` 查出来的顺序和传入的ID顺序是不会一一对应的。对于查询结果的可靠性，我们需要按传入参数 ids 的顺序返回需要的值。对于 SQL 语句可使用如下：

```
select value from t_test where id in ('test1','test7','test3')
order by field(id,'test1','test7','test3')
```

这里取基于Annotation方式的方法对程序进行简单改造，示例如下：

```
import org.apache.ibatis.jdbc.SQL;
import java.util.Map;
public class TestSelectProviderSQL {
    public String testSearch(Map<String, Object> params) {
        final StringBuffer whereSb = new StringBuffer();
        final StringBuffer orderSb = new StringBuffer();
        whereSb.append(" id in (");
        orderSb.append("field(`key`");

        String[] values = ((String[]) params.get("array"));
        for (int i = 0, size = values.length; i < size; i++) {
            if (i == 0) {
                whereSb.append("#{values" + i + "}");
                params.put("values" + i, values[i]);
            } else {
                whereSb.append(",#{values" + i + "}");
                params.put("values" + i, values[i]);
            }
        }
        whereSb.append(")");
        orderSb.append(")");

        return new SQL() {
            {
                SELECT("value");
                FROM("t_test");
                WHERE(whereSb.toString());
                ORDER_BY(orderSb.toString());
            }
        }.toString();
    }
}
```

参考

- MyBatis SelectProvider 使用: <http://www.blogjava.net/dbstar/archive/2011/08/08/35582.html>
- MyBatis Insert List values : <http://stackoverflow.com/questions/17563463/mybatis-insert-list-values>