#{}和${}的区别是什么?

动态 sql 是 mybatis 的主要特性之一，在 mapper 中定义的参数传到 xml 中之后，在查询之前 mybatis 会对其进行动态解析。mybatis 为我们提供了两种支持动态 sql 的语法：#{} 以及 ${}。

面试题:#{}和${}的区别是什么?

1）#{}是预编译处理，${}是字符串替换。

2）mybatis在处理#{}时，会将sql中的#{}替换为?号，调用PreparedStatement的set方法来赋值；mybatis在处理${}时，就是把 ${} 替换成变量的值。

3）使用 #{} 可以有效的防止SQL注入，提高系统安全性。

项目实战中使用,请阅读我博客中Java项目实战分类中的一MySQL中in('5,6,7')只取第一个id为5对应的数据的思考一文，谢谢。

要理解记忆这个题目,我觉得要抓住两点:

（1）$ 符号一般用来当作占位符，常使用Linux脚本的同学应该对此有更深的体会吧。既然是占位符，当然就是被用来替换的。知道了这点就能很容易区分$和#，从而不容易记错了。

（2）预编译的机制。预编译是提前对SQL语句进行预编译，而其后注入的参数将不会再进行SQL编译。我们知道，SQL注入是发生在编译的过程中，因为恶意注入了某些特殊字符，最后被编译成了恶意的执行操作。而预编译机制则可以很好的防止SQL注入。在某些特殊场合下只能用${}，不能用#{}。例如：在使用排序时ORDER BY ${id}，如果使用#{id}，则会被解析成ORDER BY “id”,这显然是一种错误的写法。

真实项目中防止SQL注入:

Mabatis中模糊查询防止sql注入

Mysql:

select * from user where name like concat('%', #{name}, '%')

```
<if test="vo.applyNo != null and vo.applyNo != ''">
    AND ta.apply_no LIKE CONCAT('%',#{vo.applyNo},'%')
</if>
```

Oracle:

select * from user where name like '%' || #{name} || '%'

SQLServer:

select * from user where name like '%' + #{name} + '%'

以下内容是对上述知识的扩展和理解

1.防止恶意SQL语法注入实例

实例一

Stringsql="select*fromtb_namewherename='"+varname+"'andpasswd='"+varpasswd+"'";

如果我们把['or'1'='1]作为varpasswd传入进来.用户名随意,看看会成为什么?

select * fromtb_name = 随意' and passwd = ''or' 1'='1';

因为'1'='1'肯定成立,所以可以任何通过验证

实例二

select * from ${tableName} where name = #{name}

在这个例子中，如果表名为

user; delete user; --

昵称: 佛大Java程序员
园龄: 8个月
粉丝: 2
关注: 2
+加关注

则动态解析之后 sql 如下：

select * from user; delete user; -- where name = ?;

--之后的语句被注释掉，而原本查询用户的语句变成了查询所有用户信息+删除用户表的语句，会对数据库造成重大损伤，极大可能导致服务器宕机。

2.预编译

定义：指的是数据库驱动在发送 sql 语句和参数给 DBMS 之前对 sql 语句进行编译，这样 DBMS 执行 sql 时，就不需要重新编译。

为什么需要预编译?

JDBC 中使用对象 PreparedStatement 来抽象预编译语句，使用预编译

1)预编译阶段可以优化 sql 的执行。

预编译之后的 sql 多数情况下可以直接执行，DBMS 不需要再次编译，越复杂的sql，编译的复杂度将越大，预编译阶段可以合并多次操作为一个操作。

2)预编译语句对象可以重复利用。

把一个 sql 预编译后产生的 PreparedStatement 对象缓存下来，下次对于同一个sql，可以直接使用这个缓存的 PreparedState 对象。

mybatis 默认情况下，将对所有的 sql 进行预编译。

3.mybatis sql 动态解析

mybatis 在调用 connection 进行 sql 预编译之前，会对sql语句进行动态解析，动态解析主要包含如下的功能：

占位符的处理

动态sql的处理

参数类型校验

4.DBMS和DB的关系

DBMS数据库管理系统(databasemanagementsystem)是一种操纵和管理数据库的大型软件，是用于建立、使用和维护数据库（DB）。它对数据库进行统一的管理和控制，以保证数据库的安全性和完整性。用户通过DBMS访问数据库（DB）中的数据。

MySQL是一个关系型数据库管理系统。

数据库是“按照数据结构来组织、存储和管理数据的仓库”。是一个长期存储在计算机内的、有组织的、有共享的、统一管理的数据集合。

参考

https://www.cnblogs.com/ConfidentLiu/p/7142495.html

https://blog.csdn.net/qian_qian_123/article/details/92844194

★本文目录

分类: Java基础

佛大Java程序员
关注 - 2
粉丝 - 2
+加关注

0 0

« 上一篇: MySQL中in('5,6,7')只取第一个id为5对应的数据的思考
» 下一篇: instanceof和isInstance的区别

posted @ 2020-01-20 14:45 佛大Java程序员 阅读(1453) 评论(0) 编辑 收藏

★本文目录