

39_追根究底：思考一下原生 JDBC 操作数据库存在的问题

1、开篇

上节课对 JDBC 做了简单介绍，知道 JDBC 访问数据库的基本流程。然后，通过在原有代码中加入 JDBC 配置和访问类，达到访问数据库的效果。近我们会测试一下上节课编写的代码，同时思考一下原生 JDBC 操作数据库存在的问题。今天的内容：

- JDBC 代码测试
- JDBC 数据库存在的问题

2、JDBC 简介

上节课完成了 JDBC 的代码部分，这节课先测试一下代码是否可以用。依旧需要对代码进行打包、发布、启动等操作，由于 JDBC 的应用是这个项目的过渡技术，也就是说项目最终会使用其他数据库访问技术解决，数据库访问的问题。因此这里不再把打包、发布、启动作为本节课的重点，这相关的工作也请学员自行完成。主要看发布到 ECS 服务器之后，通过 Postman 进行功能的测试。

如图 1 所示，部署并且启动 JDBC 访问数据库的应用以后，通过 Postman 访问对应的 Controller 接口。由于上节课讲到的返回教师信息的 Controller 方法名字叫做 GetAll，其对应的路径是“/demo/jdbc/all”。按照路径访问的到结果，是一个 data 的数组，其内容是教师的详细信息，包括：id、teachername、course、coursename、score 等信息。

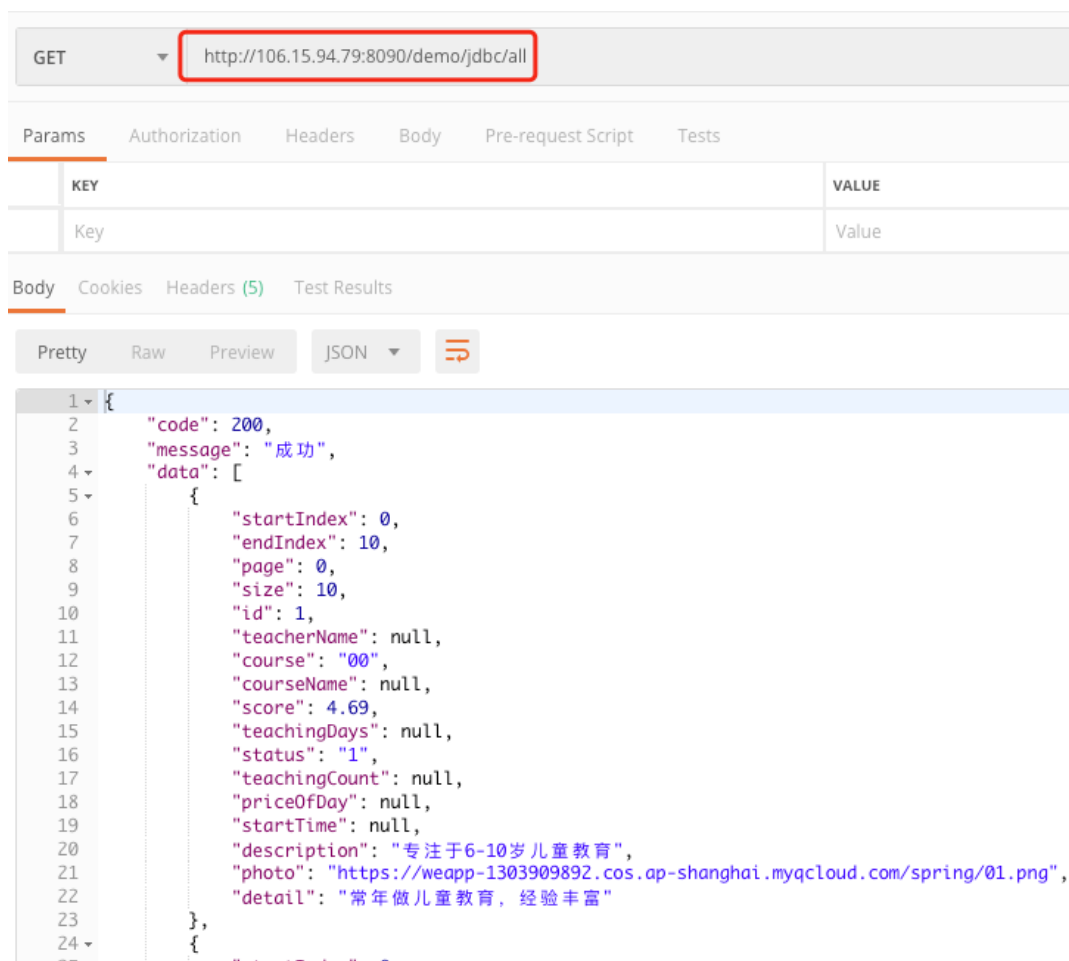


图1 调用 Controller 方法通过 JDBC 访问数据库

3、JDBC 的弊端

数据库连接，使用时就创建，不使用立即释放，对数据库进行频繁连接开启和关闭，造成数据库资源浪费，影响数据库性能。如图 2 所示，回到 `JdbcBaseDao` 类中的 `queryList` 方法，在 `finally` 代码段中须要定义对 `resultSet` 的关闭动作，同时还要对 `prepareStatement` 和 `conn`（数据库连接）进行关闭操作，确实是一系列费事且影响数据库性能的操作。后续我们可以通过数据库连接池功能提升这部分的使用体验。



图2 关闭数据库连接

除了数据库连接的问题，JDBC 中需要将 SQL 语句硬编码到 java 代码中，如果 SQL 语句修改，需要重新编译 java 代码。同时向 PreparedStatement 中设置参数，对占位符号位置和设置参数值，硬编码在 java 代码中，不利于系统维护。如果能够 将 SQL 语句配置在 xml 配置文件中，即使 SQL 变化，不需要对 java 代码进行重新编译。后面的 Mybatis 优化会提到这点。

另外，从 resultSet 中遍历结果集数据时，将获取表的字段进行硬编码，使用如此冗余的操作增加了代码维护的难度。也可以通过 JDBCTemplate 和 Mybatis 的优化解决这个问题。

4、总结

这节课对 JDBC 的代码做了简单的测试，同时反思 JDBC 编写访问数据库代码时会遇到的问题。例如数据库连接的管理和利用问题，SQL 语句硬编码不利于维护的问题，以及返回数据集需要额外处理的问题。这些问题我们会在接下来的课程中逐一给大家介绍，下节课我们会使用 Druid 连接池进行数据操作，提升系统性能。我们下期见，拜拜。