

41_新的思考：基于连接池操作数据库操作存在什么问题

1、开篇

上节课从连接数据库的操作做为切入口，提出了数据库连接池的概念，这个概念的引入帮助提升数据库操作的性能问题。同时在现有的项目中加入了 Druid 连接池配置，并且通过添加 Druid 连接池访问的类，帮助我们使用连接池进行数据库操作。这节课我们会测试本期的代码，然后思考一下在数据库操作中还有哪些可以提升的空间。今天的内容：

- Druid 连接池代码测试
- Druid 连接池还存在哪些问题

2、Druid 连接池代码测试

上节课通过 Druid 数据库连接池在项目中引入了 Druid 连接池，同时在 JdbcDruidBaseDao 中定义了通过 Druid 访问的基本方法，并且在 JdbcDemoController 中调用 JdbcDruidBaseDao 中的访问数据库方法，从而实现通过 Druid 对数据库的访问。

这里就将上次课中编写的代码进行测试。依旧遵循打包、部署、启动的三步原则，然后对这些代码进行测试。通过 Postman 向 controller 发起请求。这里我们选择 jdbcDruidGetAll 方法，其对应的访问路径是 “/jdbcDruid/all”。如图 1 所示，打开 Postman 建议一个 GET 请求，在 URL 中输入

“<http://47.100.160.139:8090/demo/jdbcDruid/all>”，其中 jdbcDruid/all 就是 jdbcDruidGetAll 方法的访问路径。然后，点击 Send 按钮，在下方的输出框就返回了对应的教师列表信息。

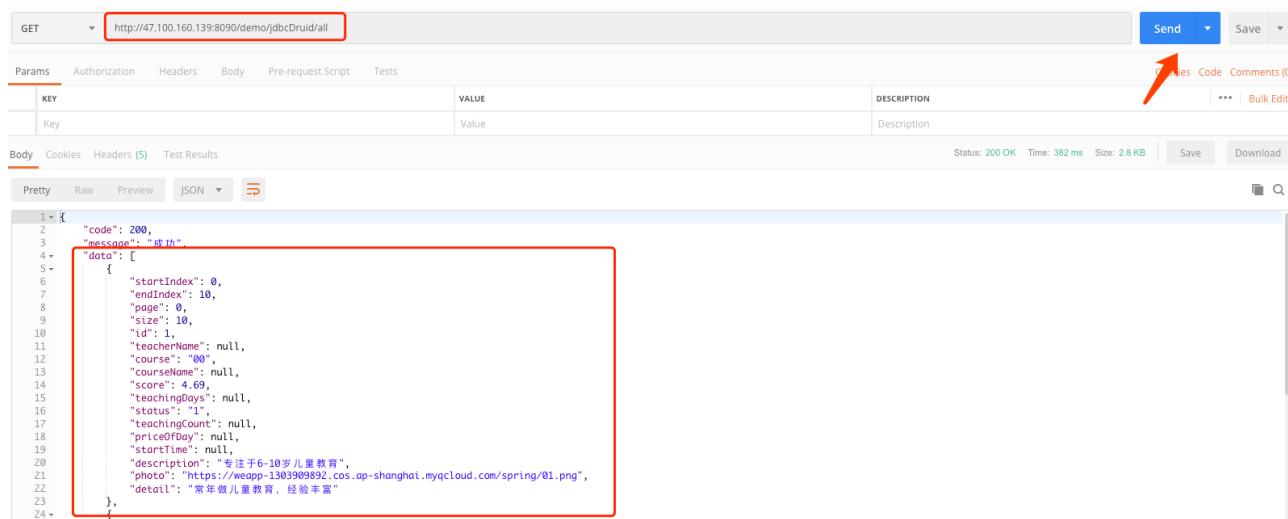


图1 通过 Postman 访问 jdbcDruidGetAll 方法

从返回信息上看，包括开始和结束索引、页数、每页数据量、教师姓名、课程数、课程名、积分、教学天数、教学数量、每天价格、描述和教师图片等信息。

3、Druid 连接池还存在哪些问题

测试完毕以后我回头再看看 Druid 连接池的代码，除了通过数据库连接池的方式解决连接数据库断开数据的资源使用问题。通过数据库连接池的方式，尽量让数据的访问先从连接池中获连接，如果连接池中没有连接的时候，再创建新的连接。但是，针对连接数据库、关闭数据库、使用 PreparedStatement 预编译 SQL 语句，以及 ResultSet 承载需要返回的数据库信息都没有进行优化。换句话说，作为开发者还是需要重复使用这些代码去访问数据库。

下一步，我们需要减少上述这些冗余代码的编写，尽量用更加简化的方式访问数据库，减少数据库查询/编辑缓解的代码引入，以及数据集返回时使用的额外对象。也就是我们下节课要介绍的 JDBCTemplate。

4、总结

本节课将 Druid 连接池实现的数据库访问功能进行了测试，然后思考了 Druid 连接池虽然解决了连接数据库的效率问题，但是依旧存在重复代码应用的问题。为了让

开发人员将更多的经历放到业务逻辑的开发上，在下节课我们会引入 JdbcTemplate。我们下期见，拜拜。