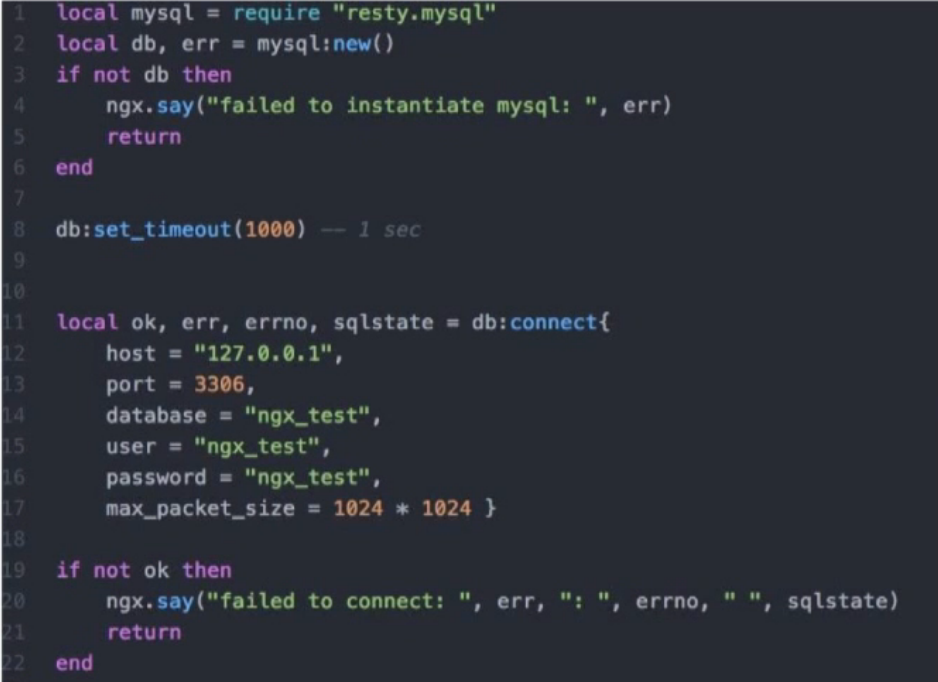
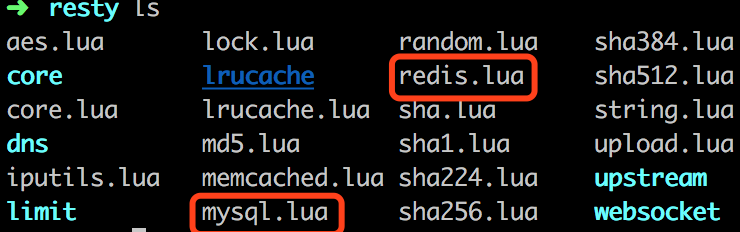
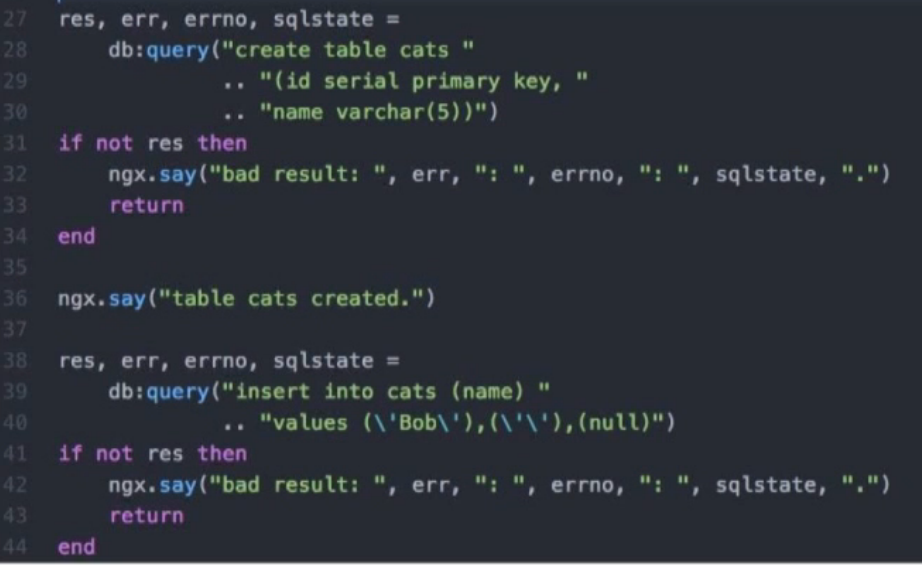
OpenResty学习系列（五）

这一节我将记录一下如何用OpenResty连接MySQL和REDIS这两个常用的数据库。

MYSQL和REDIS的驱动文件可以在nginx官方文档上进行查找，我们先看下代码:首先require了一个mysql文件，我们可以在/usr/local/openresty/lualib/resty/下看到一个mysql.lua的文件，require进来之后，下面就和其他语言的操作类似，我们先new一下，紧接着设置一个timeout，要注意的是openresty里的库大部分是毫秒，但也有少部分是秒，所以我们设置时间的时候一定要看清楚，最好在后面做一个注释说明。后面就进行connect，如果连接失败就记一个错误日志。这些都和其他语言操作一样。



  
下面的代码就有很大的不同，她直接query了一个SQL语句，这点很奇怪，因为我们之前说过，OpenResty是一个异步非阻塞的框架，但我们并没有看到相关操作，比如回调等。他是怎样做的呢，我们用的是cosocket，从而让我们达到写同步逻辑的语句，在内部却执行异步非阻塞的操作的效果。具体原理我们后面再介绍，redis也是类似的操作。



下面跑一次redis连接的例子。我们首先在nginx官方文档上拷贝下redis的test代码到配置文件，这里我们改名为test\_redis。我们reload nginx之后执行，返回正确结果。

我们回顾redis的代码，会发现暴露了很多细节，我们每次连接的时候都要设置一个timeout，每次用完把他放进连接池里等，所以我们在逻辑层操作redis的时候就不够友好，为了解决这个问题，《OpenResty最佳实践》里有一节专门讲了这些，他们把redis进行了二次封装，之后的操作将在二次封装的redis库上进行。

下一节我们将看下OpenResty里面非常重要的一个环节——缓存！

欢迎一起交流学习！

