

Spring多数据源解决方案Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/629/2021_2022_Spring_E5_A4_9A_E6_c104_629761.htm 在很多大型应用中都会对数据进行切分，并且采用多个数据库实例进行管理，这样可以有效提高系统的水平伸缩性。而这样的方案就会不同于常见的单一数据实例的方案，这就要程序在运行时根据当时的请求及系统状态来动态的决定将数据存储在哪一个数据库实例中，以及从哪个数据库提取数据。 Figure 1 数据分割及多数据库架构 通常这种多数据源的逻辑会渗透到业务逻辑中，同时也会给我们使用的数据访问API诸如Hibernate和iBatis等带来不便（需要指定多个SessionFactory或SqlMapClient实例来对应多个DataSource）。 Figure 2 多数据源的选择逻辑渗透至客户端解决方案 Figure 3 采用Proxy模式来封装数据源选择逻辑 通过采用Proxy模式我们在方案中实现一个虚拟的数据源，并且用它来封装数据源选择逻辑，这样就可以有效地将数据源选择逻辑从Client中分离出来。 Client提供选择所需的上下文（因为这是Client所知道的），由虚拟的DataSource根据Client提供的上下文来实现数据源的选择。 Spring2.x的版本中提供了实现这种方式的基本框架，虚拟的DataSource仅需继承AbstractRoutingDataSource实现determineCurrentLookupKey（）在其中封装数据源的选择逻辑。 实例：

```
public class DynamicDataSource extends AbstractRoutingDataSource {
    static Logger log = Logger.getLogger("DynamicDataSource");
    @Override
    protected Object determineCurrentLookupKey() {
        String userId = (String) DbContextHolder.getContext();
        Integer
```

dataSourceId=getDataSourceIdByUserId(userId). return
dataSourceId. } } 实例中通过UserId来决定数据存放在哪个数据
库中。 配置文件示例： gt. gt. gt. gt. gt. gt. gt. gt. gt. gt. gt. gt. gt.
gt. gt. gt. 蔡超 HP 软件架构师 软件架构顾问 SCEA IBM
Certified Solution Designer for OOA&D vUML2
Chaocai2001@yahoo.com.cn 100Test 下载频道开通，各类考试
题目直接下载。 详细请访问 www.100test.com