提升稳定性。Mysql重定向至sqlite的总结

Catalog

[1. 为了提升稳定性以及方便运维测试，将mysql重定向到sqlite 1](#_Toc8947)

[2. 修改 com.mysql.cj.jdbc.NonRegisteringDriver.connect方法 1](#_Toc23701)

[3. 自定义sqlite驱动的链接方法返回conn接口 1](#_Toc26170)

## 为了提升稳定性以及方便运维测试，将mysql重定向到sqlite

主题思路是修改mysql驱动，重定向到sqlite

## 修改 com.mysql.cj.jdbc.NonRegisteringDriver.connect方法

public java.sql.Connection connect(String url, Properties info) throws SQLException {  
  
 if(new File("/cfg/mysql.ini").exists())  
  
 {  
 String dbName=getDbName(url);  
  
 // 修正为SQLite的正确URL格式  
 String newUrl = "jdbc:sqlite:" + dbName + ".db";  
 Connection conn = getConnSqlt(newUrl);  
  
  
 return conn;  
 }

## 自定义sqlite驱动的链接方法返回conn接口

这不能简单使用jdk的conn，因为会循环引用。。多个db驱动有可能混乱引用，所有手动注入conn接口实例

public static CustomSQLiteConnection getConnSqlt(String url) throws SQLException {

return new CustomSQLiteConnection(url);

}

public class CustomSQLiteConnection extends JDBC4Connection {

// 构造函数

// Properties: 创建一个空的 Properties 对象，你可以根据需要向其中添加连接属性。

//timeout etc..

public CustomSQLiteConnection(String url) throws SQLException {

// String dbnm=;

super(url, Objects.requireNonNull(getDbNameSqlt(url)),new Properties()); // 调用父类构造函数

}

// 添加自定义方法

public void customMethod() {

// 自定义逻辑，例如打印连接信息

System.out.println("Custom method executed on SQLiteConnection.");

}

}