Atitit springboot 性能提升总结

Cache context

但是无论context还是sqlfactory 都缓存不了，sqlsession ，conn也不行。。。

换个思路，得到url，重新构建conn得了

SqlSessionFactory sqlSessionFactory = SportApplication.*context*.getBean(SqlSessionFactory.**class**);

Connection connection = sqlSessionFactory.openSession().getConnection();

**package** com. .utils.db;

**import** java.lang.reflect.Field;

**import** java.sql.Connection;

**import** org.apache.ibatis.datasource.pooled.PooledDataSource;

//import org.apache.ibatis.datasource.pooled.PooledConnection;

**import** org.apache.ibatis.session.SqlSession;

**import** com.alibaba.fastjson.JSON;

**import** com.mysql.cj.jdbc.ConnectionImpl;

**import** com.zaxxer.hikari.pool.HikariProxyConnection;

**import** com.zaxxer.hikari.pool.ProxyConnection;

**import** cn.hutool.core.util.ReflectUtil;

**public** **class** DbUtil {

**public** **static** **void** main(String[] args) **throws** Exception {

SqlSession SqlSession1 = MybatisUtil.*getConn*();

System.***out***.println(SqlSession1.getConnection());

System.***out***.println(*getUrl*(SqlSession1.getConnection()));

}

**public** **static** String getUrl(Connection connection) **throws** Exception {

**if** (connection **instanceof** HikariProxyConnection) {

HikariProxyConnection HikariProxyConnection1 = (HikariProxyConnection) connection;

// System.out.println(JSON.toJSON(HikariProxyConnection1)); err to json

//

ProxyConnection ProxyConnection1 = HikariProxyConnection1;

ConnectionImpl ConnectionImpl1 = (ConnectionImpl) com.kok.sport.utils.ReflectUtil

.*getFldVal*(ProxyConnection.**class**, ProxyConnection1, "delegate");

**return** *getUrl*(ConnectionImpl1);

} **else** { //mybatis def

Field f = ReflectUtil.*getField*(connection.getClass(), "h");

f.setAccessible(**true**);

//// import org.apache.ibatis.datasource.pooled.PooledConnection; is invisuvel

Object PooledConnection = f.get(connection);

// UnpooledDataSource UnpooledDataSource1=

Field dataSource = ReflectUtil.*getField*(PooledConnection.getClass(), "dataSource");

dataSource.setAccessible(**true**);

PooledDataSource PooledDataSource1 = (PooledDataSource) dataSource.get(PooledConnection);

// unpooled

// UnpooledDataSource is invisual

// System.out.println(JSON.toJSON(PooledDataSource1));

String url = PooledDataSource1.getUrl();

url=*addUrl*(url,"username" , PooledDataSource1.getUsername());

url=*addUrl*(url,"password" , PooledDataSource1.getPassword());

// System.out.println(url);

**return** url;

}

// return connection.toString();

}

**private** **static** String addUrl(String url, String k, String v) {

**if**(url.contains(k+"="))

**return** url;

**else**

**return** url+"&"+k+"="+v;

}

**private** **static** String getUrl(ConnectionImpl ConnectionImpl1) {

System.***out***.println(ConnectionImpl1.getURL());

System.***out***.println(ConnectionImpl1.getUser());

String pwd = (String) com.kok.sport.utils.ReflectUtil.*getFldVal*(ConnectionImpl1, "password");

System.***out***.println(pwd);

String url = ConnectionImpl1.getURL();

url=*addUrl*(url,"username" , ConnectionImpl1.getUser());

url=*addUrl*(url,"password" , pwd);

// System.out.println(url);

**return** url;

}

}