Atitit mybatis prblm n solu v1 u55

目录

[1.1. 加载任意文职cfg 1](#_Toc16588)

[1.2. 根据dburl获取factory 1](#_Toc32767)

[1.3. Load any mapper file 2](#_Toc29798)

[2. Debug 3](#_Toc31072)

[2.1. Other 4](#_Toc6997)

## 加载任意文职cfg

**public** **static** SqlSessionFactory getSqlSessionFactory() **throws** Exception {

String mybatisCfg\_result = *getCfgTxtV2*();

// System.out.println(mybatisCfg\_result);

InputStream is2 = **new** ByteArrayInputStream(mybatisCfg\_result.getBytes());

// ����sqlSession �Ĺ���

**return** **new** SqlSessionFactoryBuilder().build(is2);

}

## 根据dburl获取factory

**public** **static** SqlSessionFactory getSqlSessionFactoryByDburl(String dburl) {

String mybatisCfg\_result = *getCfgTxtByDburl*(dburl.trim());

// System.out.println(mybatisCfg\_result);

InputStream is2 = **new** ByteArrayInputStream(mybatisCfg\_result.getBytes());

// ����sqlSession �Ĺ���

**return** **new** SqlSessionFactoryBuilder().build(is2);

}

**private** **static** String getCfgTxtByDburl(@NotNull @NonNull String mysqlConnUrl) {

String mybatisCfg\_result = **null**;

**try** {

mybatisCfg\_result = *mybatisCfgTxt*();

} **catch** (IOException e) {

ExUtilV2t33.*throwExV2*(e);

}

// String getCfgFile = SpringUtil.getCfgFile();

String propFilePath = "/cfg";

ConnectionUrlParser connStringParser = ConnectionUrlParser.*parseConnectionString*(mysqlConnUrl);

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

Object url = mysqlConnUrl;

Object usr = UrlUtil.*parse4Q*(connStringParser.getQuery()).get("user");

Object pwd = UrlUtil.*parse4Q*(connStringParser.getQuery()).get("password");

**if** (pwd == **null**)

pwd = "";

url = cn.hutool.core.util.XmlUtil.*escape*(url.toString() + "&allowMultiQueries=true");

mybatisCfg\_result = mybatisCfg\_result.replaceAll("\\$\\{mysql.url}", url.toString());

mybatisCfg\_result = mybatisCfg\_result.replaceAll("\\$\\{mysql.username}", usr.toString());

mybatisCfg\_result = mybatisCfg\_result.replaceAll("\\$\\{mysql.password}", pwd.toString());

**return** mybatisCfg\_result;

// return null;

}

## Load any mapper file

Or change cfg file gvet factory new

// Configuration.addLoadedResource

SqlSessionFactory sqlSessionFactory = MybatisUtil.*getSqlSessionFactory*(li);

// cfg on effect in ini factory

// Configuration c= sqlSessionFactory.getConfiguration();

// for (String mpXmlPathRes : li) {

// c.addLoadedResource(mpXmlPathRes);

// }

**public** **static** SqlSession getConn(String mpXmlPathRes) {

**try** {

// Configuration.addLoadedResource

SqlSessionFactory sqlSessionFactory = MybatisUtil.getSqlSessionFactory();

Configuration c = sqlSessionFactory.getConfiguration();

c.addLoadedResource(mpXmlPathRes);

SqlSession session = sqlSessionFactory.openSession(**true**);

**return** session;

} **catch** (Exception e) {

**throw** **new** RuntimeException(e);

}

}

# Debug

SqlSession conn = MybatisUtil.getConn(li);

Collection<MappedStatement> mappedStatements = conn.getConfiguration().getMappedStatements();

**for** (MappedStatement mappedStatement : mappedStatements) {

System.out.println("--------------------------");

System.out.println("StatementType: " + mappedStatement.getStatementType());

System.out.println("id: " + mappedStatement.getId());

System.out.println("res: " + mappedStatement.getResource());

// System.out.println("getBoundSql:

// "+mappedStatement.getBoundSql(null).getSql());

}

System.out.println(mappedStatements);

## Other

**protected** **static** **void** checkSqlValid(String sql, MappedStatement s) {

**if** (sql == "")

**return**;

**try** {

SQLStatementParser parser = **new** MySqlStatementParser(sql);

// 使用Parser解析生成AST，这里SQLStatement就是AST

SQLStatement statement = parser.parseStatement();

System.***out***.println("---check ok");

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

System.***out***.println(s.getResource());

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

System.***out***.println("---check ok end");

} **catch** (Exception e) {

System.***out***.println("---check err");

System.***out***.println(s.getResource());

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

System.***out***.println("---check err end");

**throw** e;

}

;

}

**private** **static** String getSqlBound(SqlSession conn, String sttID, Map m) {

Configuration Configuration1 = conn.getConfiguration();

MappedStatement MappedStaement1 = Configuration1.getMappedStatement(sttID);

String sqlBound = MappedStaement1.getSqlSource().getBoundSql(m).getSql();

**return** sqlBound;

}