Atitit 获取SqlSessionFactory的三种方式

目录

[第一节 DataSource 方式 1](#_Toc14647)

[第二节 读取sprbt Url方式 ByteArrayInputStream 1](#_Toc22163)

[第三节 Mybatis。Xml方式 3](#_Toc1000)

## DataSource 方式

**public** **static** SqlSessionFactory getSqlSessFac(DataSource dataSourceDb2, String mapPaths) **throws** Exception {

SqlSessionFactoryBean factoryBean = **new** SqlSessionFactoryBean();

factoryBean.setDataSource(dataSourceDb2);

// String mapPath = "classpath:mapper/\*\*/\*.xml";

factoryBean.setMapperLocations(*getRsces*(mapPaths));

**return** factoryBean.getObject();

}

## 读取sprbt Url方式 ByteArrayInputStream

**public** **static** SqlSessionFactory getSqlSessionFactoryBySprbtCfgfile(String mapPaths) {

**try** {

List<String> mpList = Arrays.*asList*(mapPaths.split(","));

String mysqlConnUrl = SpringUtil.*getDbUrlFrmSprbtCfg*();

String mybatisCfg\_result = *mybatisCfgTxtV2*("mybatisCfgTmplt.xml", mysqlConnUrl);

mybatisCfg\_result = mybatisXmlParser.*addmapper2xml*(mpList, mybatisCfg\_result);

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

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

// ����sqlSession �Ĺ���

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

} **catch** (Exception e) {

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

}

}

**public** **static** String mybatisCfgTxtV2(String MybatisCfgPathResDir, String mysqlConnUrl) **throws** Exception {

String mybatisCfg\_result = *mybatisCfgTxt*(MybatisCfgPathResDir);

// String getCfgFile = SpringUtil.getCfgFile();

// String mysqlConnUrl =getMysqlUrlFrmSprbtCfg();

// C:\Users\ATI\.m2\repository\mysql\mysql-connector-java\6.0.6\mysql-connector-java-6.0.6.jar

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;

}

**private** **static** String mybatisCfgTxt(String MybatisCfgPathResDir) **throws** IOException {

// String MybatisCfgPath =getMybatisCfgPath();

// "/" + mybatisNamespace + "mybatis.xml";

// ����mybatis �������ļ�����Ҳ���ع�����ӳ���ļ���

// ClassLoader classLoader = .getClassLoader();

InputStream is = MybatisUtilV55.**class**.getResourceAsStream(MybatisCfgPathResDir);

String mybatisCfg\_result = CharStreams.*toString*(**new** InputStreamReader(is, Charsets.~~UTF\_8~~));

**return** mybatisCfg\_result;

}

## Mybatis。Xml方式

**public** **static** SqlSessionFactory getSqlsessFac(String mybatisCfgfile) **throws** IOException {

InputStream is = Resources.*getResourceAsStream*(mybatisCfgfile);

SqlSessionFactory sqlSessionFactory = **new** SqlSessionFactoryBuilder().build(is);

**return** sqlSessionFactory;

}