**Java执行SQL脚本文件到数据库详解**

本文实例为大家分享了Java执行SQL脚本文件到数据库的具体方式，供大家参考，具体内容如下

**方式一：**直接读取SQL脚本文件的内容，然后传递到SQL中。

代码：RunSqlService：

[?](http://www.jb51.net/article/117176.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | @Autowired  private RunSqlDao runSqlDao;    /\*\*   \* 读取文件内容到SQL中执行   \* @param sqlPath SQL文件的路径：如：D:/TestProject/web/sql/脚本.Sql   \*/  public void runSqlByReadFileContent(String sqlPath) throws Exception {    try {        String sqlStr = readFileByLines(sqlPath);      // System.out.println("获得的文本：" + sqlStr);      if (sqlStr.length() > 0) {        runSqlDao.runSqlBySqlStr(sqlStr);      }    } catch (Exception e) {      e.printStackTrace();      throw e;    }  }    /\*\*   \* 以行为单位读取文件，常用于读面向行的格式化文件   \*/  private String readFileByLines(String filePath) throws Exception {    StringBuffer str = new StringBuffer();    BufferedReader reader = null;    try {      reader = new BufferedReader(new InputStreamReader(          new FileInputStream(filePath), "UTF-8"));      String tempString = null;      int line = 1;      // 一次读入一行，直到读入null为文件结束      while ((tempString = reader.readLine()) != null) {        // 显示行号        // System.out.println("line " + line + ": " + tempString);          str = str.append(" " + tempString);        line++;      }      reader.close();    } catch (IOException e) {      e.printStackTrace();      throw e;    } finally {      if (reader != null) {        try {          reader.close();        } catch (IOException e1) {        }      }    }      return str.toString();  } |

RunSqlDao ：

[?](http://www.jb51.net/article/117176.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | /\*\*   \* @param sqlStr   \*/  public void runSqlBySqlStr(String sqlStr) {    Map<String,Object> map=new HashMap<String,Object>();    map.put("sqlStr", sqlStr);    sqlSessionTemplate.selectList("runSql.runSqlBySqlStr", map);  } |

SQLMap：

[?](http://www.jb51.net/article/117176.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <mapper namespace="runSql">    <select id="runSqlBySqlStr" parameterType="map">  <![CDATA[ ${sqlStr}]]>  </select>    </mapper> |

这种写法：只支持数据的变化（新增、修改、删除）,且SQL文件内容以begin开始，以end结束。无法更新表字段修改等操作。

**方式二；**使用ScriptRunner

代码：RunSqlService：

[?](http://www.jb51.net/article/117176.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | /\*\*   \* 执行sql脚本文件 使用ScriptRunner   \* @param sqlPath SQL文件的路径：如：D:/TestProject/web/sql/脚本.Sql   \*/  public void runSqlByScriptRunner(String sqlPath) throws Exception {    try {      SqlSession sqlSession = sqlSessionFactory.openSession();      Connection conn = sqlSession.getConnection();      ScriptRunner runner = new ScriptRunner(conn);      runner.setEscapeProcessing(false);      runner.setSendFullScript(true);      runner.runScript(new InputStreamReader(new FileInputStream(sqlPath), "UTF-8"));    } catch (Exception e) {      e.printStackTrace();      throw e;    }  } |

这种写法：只能有一行SQL，即一次执行一个SQL语句，否则就会报错。

**方式三：**使用ScriptUtils

代码：RunSqlService：(以下两种方式：脚本.Sql 和RunSqlService 在同一目录下)

方法（1）

[?](http://www.jb51.net/article/117176.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | /\*\*   \* 执行sql脚本文件 使用Spring工具类   \*/  public void runSqlBySpringUtils() throws Exception {    try {      SqlSession sqlSession = sqlSessionFactory.openSession();      Connection conn = sqlSession.getConnection();      ClassPathResource rc = new ClassPathResource("脚本.Sql", RunSqlDao.class);      ScriptUtils.executeSqlScript(conn, rc);    } catch (Exception e) {      e.printStackTrace();      throw e;    }  } |

方法（2）

[?](http://www.jb51.net/article/117176.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | /\*\*   \* 执行sql脚本文件 使用Spring工具类   \*/  public void runSqlBySpringUtils() throws Exception {    try {      SqlSession sqlSession = sqlSessionFactory.openSession();      Connection conn = sqlSession.getConnection();      ClassPathResource rc = new ClassPathResource("脚本.Sql", RunSqlDao.class);      EncodedResource er = new EncodedResource(rc, "utf-8");      ScriptUtils.executeSqlScript(conn, er);    } catch (Exception e) {      e.printStackTrace();      throw e;    }  } |

方法（1），脚本.Sql文件必须是ANSI的，否则执行到数据中汉字是乱码。

方法（2）解决了方法（1）的问题，完美了，喜欢的小伙伴们快拿去享用吧。

以上就是本文的全部内容，希望对大家的学习有所帮助，也希望大家多多支持脚本之家。