**利用Java进行MySql数据库的导入和导出**

转载 2014年07月15日 15:46:51

* 标签：
* [杨凯](http://so.csdn.net/so/search/s.do?q=%E6%9D%A8%E5%87%AF&t=blog) /
* [JAVA](http://so.csdn.net/so/search/s.do?q=JAVA&t=blog) /
* [mysql](http://so.csdn.net/so/search/s.do?q=mysql&t=blog)
* 6433

利用Java来进行Mysql数据库的导入和导出的总体思想是通过Java来调用命令窗口执行相应的命令。

MySql导出数据库的命令如下：

**Sql代码  收藏代码**

1. mysqldump -uusername -ppassword -hhost -Pport exportDatabaseName > exportPath

利用Java调用命令窗口执行命令来进行MySql导入数据库一般分三步走：

第一步：登录Mysql数据库，在登录数据库的时候也可以指定登录到哪个数据库，如果指定了则可以跳过第二步；

第二步：切换数据库到需要导入的目标数据库

第三步：利用命令开始导入

在进行导出的时候，需要注意命令语句的运行环境，如果已经将mysql安装路径下的bin加入到

系统的path变量中，那么在导出的时候可以直接使用命令语句，否则，就需要在执行命令语句的

时候加上命令所在位置的路径，即mysql安装路径想的bin下的mysqldump命令。

基本代码如下：

**Java代码  收藏代码**

1. **import** java.io.IOException;
2. **import** java.io.InputStream;
3. **import** java.io.OutputStream;
4. **import** java.io.OutputStreamWriter;
5. **import** java.util.Properties;
7. /\*\*
8. \* 在进行导出的时候，需要注意命令语句的运行环境，如果已经将mysql安装路径下的bin加入到
9. \* 系统的path变量中，那么在导出的时候可以直接使用命令语句，否则，就需要在执行命令语句的
10. \* 时候加上命令所在位置的路径，即mysql安装路径想的bin下的mysqldump命令
11. \* @author andy
12. \*
13. \*/
14. **public** **class** MySqlImportAndExport {
16. **public** **static** **void** main(String args[]) **throws** IOException {
17. InputStream is = MySqlImportAndExport.**class**.getClassLoader().getResourceAsStream("jdbc.properties");
18. Properties properties = **new** Properties();
19. properties.load(is);
20. //      MySqlImportAndExport.export(properties);//这里简单点异常我就直接往上抛
21. MySqlImportAndExport.importSql(properties);
22. }
24. /\*\*
25. \* 根据属性文件的配置导出指定位置的指定数据库到指定位置
26. \* @param properties
27. \* @throws IOException
28. \*/
29. **public** **static** **void** export(Properties properties) **throws** IOException {
30. Runtime runtime = Runtime.getRuntime();
31. String command = getExportCommand(properties);
32. runtime.exec(command);//这里简单一点异常我就直接往上抛
33. }
35. /\*\*
36. \* 根据属性文件的配置把指定位置的指定文件内容导入到指定的数据库中
37. \* 在命令窗口进行mysql的数据库导入一般分三步走：
38. \* 第一步是登到到mysql； mysql -uusername -ppassword -hhost -Pport -DdatabaseName;如果在登录的时候指定了数据库名则会
39. \* 直接转向该数据库，这样就可以跳过第二步，直接第三步；
40. \* 第二步是切换到导入的目标数据库；use importDatabaseName；
41. \* 第三步是开始从目标文件导入数据到目标数据库；source importPath；
42. \* @param properties
43. \* @throws IOException
44. \*/
45. **public** **static** **void** importSql(Properties properties) **throws** IOException {
46. Runtime runtime = Runtime.getRuntime();
47. //因为在命令窗口进行mysql数据库的导入一般分三步走，所以所执行的命令将以字符串数组的形式出现
48. String cmdarray[] = getImportCommand(properties);//根据属性文件的配置获取数据库导入所需的命令，组成一个数组
49. //runtime.exec(cmdarray);//这里也是简单的直接抛出异常
50. Process process = runtime.exec(cmdarray[0]);
51. //执行了第一条命令以后已经登录到mysql了，所以之后就是利用mysql的命令窗口
52. //进程执行后面的代码
53. OutputStream os = process.getOutputStream();
54. OutputStreamWriter writer = **new** OutputStreamWriter(os);
55. //命令1和命令2要放在一起执行
56. writer.write(cmdarray[1] + "\r\n" + cmdarray[2]);
57. writer.flush();
58. writer.close();
59. os.close();
60. }
62. /\*\*
63. \* 利用属性文件提供的配置来拼装命令语句
64. \* 在拼装命令语句的时候有一点是需要注意的：一般我们在命令窗口直接使用命令来
65. \* 进行导出的时候可以简单使用“>”来表示导出到什么地方，即mysqldump -uusername -ppassword databaseName > exportPath，
66. \* 但在Java中这样写是不行的，它需要你用-r明确的指出导出到什么地方，如：
67. \* mysqldump -uusername -ppassword databaseName -r exportPath。
68. \* @param properties
69. \* @return
70. \*/
71. **private** **static** String getExportCommand(Properties properties) {
72. StringBuffer command = **new** StringBuffer();
73. String username = properties.getProperty("jdbc.username");//用户名
74. String password = properties.getProperty("jdbc.password");//用户密码
75. String exportDatabaseName = properties.getProperty("jdbc.exportDatabaseName");//需要导出的数据库名
76. String host = properties.getProperty("jdbc.host");//从哪个主机导出数据库，如果没有指定这个值，则默认取localhost
77. String port = properties.getProperty("jdbc.port");//使用的端口号
78. String exportPath = properties.getProperty("jdbc.exportPath");//导出路径
80. //注意哪些地方要空格，哪些不要空格
81. command.append("mysqldump -u").append(username).append(" -p").append(password)//密码是用的小p，而端口是用的大P。
82. .append(" -h").append(host).append(" -P").append(port).append(" ").append(exportDatabaseName).append(" -r ").append(exportPath);
83. **return** command.toString();
84. }
86. /\*\*
87. \* 根据属性文件的配置，分三步走获取从目标文件导入数据到目标数据库所需的命令
88. \* 如果在登录的时候指定了数据库名则会
89. \* 直接转向该数据库，这样就可以跳过第二步，直接第三步；
90. \* @param properties
91. \* @return
92. \*/
93. **private** **static** String[] getImportCommand(Properties properties) {
94. String username = properties.getProperty("jdbc.username");//用户名
95. String password = properties.getProperty("jdbc.password");//密码
96. String host = properties.getProperty("jdbc.host");//导入的目标数据库所在的主机
97. String port = properties.getProperty("jdbc.port");//使用的端口号
98. String importDatabaseName = properties.getProperty("jdbc.importDatabaseName");//导入的目标数据库的名称
99. String importPath = properties.getProperty("jdbc.importPath");//导入的目标文件所在的位置
100. //第一步，获取登录命令语句
101. String loginCommand = **new** StringBuffer().append("mysql -u").append(username).append(" -p").append(password).append(" -h").append(host)
102. .append(" -P").append(port).toString();
103. //第二步，获取切换数据库到目标数据库的命令语句
104. String switchCommand = **new** StringBuffer("use ").append(importDatabaseName).toString();
105. //第三步，获取导入的命令语句
106. String importCommand = **new** StringBuffer("source ").append(importPath).toString();
107. //需要返回的命令语句数组
108. String[] commands = **new** String[] {loginCommand, switchCommand, importCommand};
109. **return** commands;
110. }
112. }

上述使用的jdbc.properties文件

**Properties代码  收藏代码**

1. jdbc.username=root
2. jdbc.password=password
3. jdbc.host=localhost
4. jdbc.port=3306
5. jdbc.exportDatabaseName=dbName
6. jdbc.exportPath=d\:\\dbName.sql
7. jdbc.importDatabaseName=test
8. jdbc.importPath=d\:\\dbName.sql