



链滴

SSM 实现网站后台导出 sql 文件备份数据库

作者: [kangaroo1122](#)

原文链接: <https://ld246.com/article/1556340764256>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



网站备份常常是很有必要的，备份的方式有很多。之前不懂怎么备份，网上查了一下，看看怎么通过Java程序执行备份cmd命令，再结合solo的实现，自己的毕设也算是实现了这样的功能。solo用的是Lat e的一个工具类Execs，但是我在用这个工具类的时候，可以ping，但是就是不能做执行备份的cmd命令，可能是自己使用的姿势不对，所以索性自己给改改。

准备

导出是参看了solo的，压缩成zip包再响应前端，压缩用到了一个工具类：jodd-core-5.0.12.jar，下载地址：<https://mvnrepository.com/artifact/org.jodd/jodd-core>

JSP

我这里主要就是一个a标签。

```
<a href="exportSQL">导出数据库</a>
```

Controller

```
//从配置文件获得数据库信息，以备导出数据库
@Value("#{configProperties['jdbc.username']}")
private String dbUser;
@Value("#{configProperties['jdbc.password']}")
private String dbPwd;
@Value("#{configProperties['jdbc.url']}")
private String dbURL;
```

这里连接数据库，是通过jdbc.properties文件得到的，获取值如上。

```
//导出数据库
```

```

@RequestMapping("/exportSQL")
@ResponseBody
public void exportSQL(HttpServletRequest request, HttpServletResponse response) throws FileNotFoundException, IOException {
    int beginIndex = dbURL.lastIndexOf("/") + 1;
    int endIndex = dbURL.indexOf("?");
    //截取，获得数据库名字
    String databaseName = dbURL.substring(beginIndex, endIndex);
    // 设置导出编码为utf8。这里必须是utf8。cmd命令，-u和-p后边没有空格
    String cmd = "mysqldump -u" + dbUser + " -p" + dbPwd + " -localhost --set-charset=utf
--databases " + databaseName;
    try {
        Runtime runtime = Runtime.getRuntime();
        // 调用 MySQL 的 CMD
        Process child = runtime.exec(cmd);
        // 把进程执行中的控制台输出信息写入.sql文件，即生成了备份文件。注：如果不对控制台信息
        行读出，则会导致进程堵塞无法运行
        InputStream in = child.getInputStream();// 控制台的输出信息作为输入流
        InputStreamReader xx = new InputStreamReader(in, "utf8");// 设置输出流编码为utf8。这
        必须是utf8，否则从流中读入的是乱码
        String inStr;
        StringBuffer sb = new StringBuffer("");
        String outStr;
        // 组合控制台输出信息字符串
        BufferedReader br = new BufferedReader(xx);
        while ((inStr = br.readLine()) != null) {
            sb.append(inStr + "\r\n");
        }
        outStr = sb.toString();
        //取得系统缓存目录
        String tmpDir = System.getProperty("java.io.tmpdir");
        //获得当前时间戳
        String date = DateFormatUtil.getStringTime();
        String localFilePath = tmpDir + "db_shms-" + date + ".sql";
        File localFile = new File(localFilePath);
        byte[] data = outStr.getBytes("UTF-8");
        try (final OutputStream output = new FileOutputStream(localFile)) {
            IOUtils.write(data, output);
        }
        File zipFile = ZipUtil.zip(localFile);
        byte[] zipData;
        try (final FileInputStream inputStream = new FileInputStream(zipFile)) {
            zipData = IOUtils.toByteArray(inputStream);
        }
        //设置响应相关的参数
        response.setContentType("application/zip");
        final String fileName = "db_shms-" + date + ".zip";
        response.setHeader("Content-Disposition", "attachment; filename=\"" + fileName + "\"");

        final ServletOutputStream outputStream = response.getOutputStream();
        outputStream.write(zipData);
        outputStream.close();
        in.close();
        xx.close();
        br.close();
    }
}

```

```
        outputStream.flush();
        outputStream.close();
        //删除缓存目录的缓存文件, sql文件和zip文件
        localFile.delete();
        zipFile.delete();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

需要注意的就是cmd命令，在哪些地方需要空格，哪里不需要空格，毕竟是命令，关系到导出是否成功。

关于mysqldump命令的详细参数，可以看这个<https://www.cnblogs.com/liuriqu/p/4207310.html>

XML配置

这里主要就是配置properties文件。

```
xmlns:util="http://www.springframework.org/schema/util"
xsi:schemaLocation="http://www.springframework.org/schema/util
    http://www.springframework.org/schema/util/spring-util.xsd">
```

```
<!-- 加载数据库配置文件 -->
```

```
<util:properties id="configProperties" location="classpath:jdbc.properties" />
```

这样，在Controller就能通过@Value 注解获得配置文件的相应参数值。至于jdbc.properties文件，里就不写了，就是数据库配置的properties文件。