第七节、使用Java操作HDFS

一、准备jar包

> 将$HADOOP_HOME/share/hadoop下的jar添加至开发环境，test包和example以及source包可忽略

1. common

2. hdfs

3. mapreduce

4. yarn

二、解决环境冲突问题

> 解决Could not locate executable null\bin\winutils.exe in the Hadoop binaries

1. 在windows下添加HADOOP_HOME系统环境变量

2. 将winutils.exe拷贝到HADOOP_HOME的bin目录下

3. 如果winutils.exe无法正常运行需要首先修复系统环境

4. 重启Eclipse

三、初始化设置

1. win下添加地址映射

> C:\Windows\System32\drivers\etc\hosts

2. 加载配置

```
1. private static Configuration conf = new Configuration();
```

3. 指定操作地址

```
1. private static final String HADOOP_URL="hdfs://etc01:8020";
```

4. 初始化操作对象

```
1. private static FileSystem fs;
2. 
3. static {
4.     try {
5.         FileSystem.setDefaultUri(conf, HADOOP_URL);
6.         fs = FileSystem.get(conf);
7.     } catch (Exception e) {
8.         e.printStackTrace();
9.     }
10. }
```

5. 查看默认存储块大小

```
1. fs.getDefaultBlockSize();
```

四、核心功能

1. 相对路径

```
1. fs.getHomeDirectory()
```

2. 创建文件夹

> 保证权限校验功能关闭（hdfs-site.xml）

```
<property>
    <name>dfs.permissions</name>
    <value>false</value>
</property>
```

```
fs.mkdirs(new Path("xxx/xxx"));
```

3. 上传文件

```
fs.copyFromLocalFile(src, dst);
```

4. 下载文件

```
fs.copyToLocalFile(src, dst);
```

5. 删除文件

```
if(fs.exists(dstPath)){
    fs.delete(dstPath, true) ;
}
```

6. 使用流向HDFS写入内容

```
try {
    Path path = new Path("xxx/xxx");
    FSDataOutputStream os = fs.create(path, true);
    Writer out = new OutputStreamWriter(os, "utf-8");
    out.write("你好");
    out.close();
    os.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

7. 使用流从HDFS读取内容

```
try {
    Path path = new Path("xxx/xxx");
    FSDataInputStream dataInputStream = fs.open(path);
    InputStreamReader inputStreamReader = new InputStreamReader(dataInputStream, "utf-8");
    BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
    String str = "";
    while ((str = bufferedReader.readLine()) !=null) {
        System.out.println(str);
    }
    bufferedReader.close();
    inputStreamReader.close();
    dataInputStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

8. 使用Process进行命令调用

```
Runtime runtime = Runtime.getRuntime();
try {
    /* 使用Process可接收返回结果 */
    Process process = runtime.exec("hdfs dfs -ls /");
    InputStream is = process.getInputStream();
    BufferedReader reader = new BufferedReader(new InputStreamReader(i
s));
    String line;
    while((line = reader.readLine())!= null){
        System.out.println(line);
    }
    process.waitFor();
    is.close();
    reader.close();
    process.destroy();
} catch (Exception e) {
    e.printStackTrace();
}
```