大数据学习笔记（3）-- hdfs 常用的 java-api

作者：kevinBobo
原文链接：https://ld246.com/article/1546226308573
来源网站：链滴

前面我们已经安装好了hadoop，这下就可以用java来尝试操作它了

引入hadoop依赖

```
<dependencies>
  <dependency>
   <groupId>org.apache.hadoop<groupId>
   <artifactId>hadoop-client<artifactId>
   <version>2.9.2<version>
  <dependency>
  <dependency>
   <groupId>junit<groupId>
   <artifactId>junit<artifactId>
   <version>4.12<version>
  <dependency>
<dependencies>
```

编写测试方法

```
package com.bobo;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.RemoteIterator;
import org.junit.Before;
import org.junit.Test;
```

```
import java.io.IOException;
import java.net.URI;

/**
 * @author bobo
 * @Description:
  * @date 2018-12-29 16:28
 */public class HdfsClientMain {

  private FileSystem fs;

  /**
 * 初始化fs * @throws Exception
  */
  @Before
  public void init() {
  Configuration conf = new Configuration();
  //指定本客户端上传文件到hdfs时需要保存的服本数为2
  conf.set("dfs.replication","3");
  //指定本客户端上传到hdfs时切块的规格大小：64m
  conf.set("dfs.blocksize","128m");
  conf.set("dfs.client.use.datanode.hostname","true");
 try {
 fs = FileSystem.get(new URI("hdfs://namenode:9000"), conf,"root");
  }catch (Exception e){
  e.printStackTrace();
  }
 }
  /**
 * 上传文件到hdfs * @throws IOException
  */
  @Test
  public void uploadFile() throws IOException {
 fs.copyFromLocalFile(new Path("/Users/bobo/Downloads/asdf.txt"),new Path("/"));
  fs.close();
  }

  /**
 * 从hdfs获取文件 * @throws IOException
  */
  @Test
  public void getFile() throws IOException {
 fs.copyToLocalFile(new Path("/hbase-1.2.9-src.tar.gz"),new Path("./"));
  fs.close();
  }

  /**
 * 创建文件夹 * @throws IOException
  */
  @Test
  public void mkdir() throws IOException {
 fs.mkdirs(new Path("/test"));
  fs.close();
  }
```

```
 /**
* 移动文件或者重命名 * @throws IOException
 */
 @Test
 public void mvFile() throws IOException {
fs.rename(new Path("/hbase-1.2.9-src.tar.gz"),new Path("/test/hbase.tar.gz"));
 fs.close();
 }

 /**
* 删除文件 * @throws IOException
 */
 @Test
 public void rmFile() throws IOException {
fs.delete(new Path("/asdf.txt"),true);
 fs.close();
 }

 /**
* 查询目录下的文件 * @throws IOException
 */
 @Test
 public void lsFile() throws IOException {
 RemoteIterator<LocatedFileStatus> files = fs.listFiles(new Path("/"), true);
while (files.hasNext()){
 LocatedFileStatus fileStatus = files.next();
 System.out.println(fileStatus.getPath());
 }
fs.close();
 }

 /**
* 查询目录下的文件和文件夹 * @throws IOException
 */
 @Test
 public void lsFileAndDir() throws IOException {
 FileStatus[] status = fs.listStatus(new Path("/"));
for (FileStatus fileStatus :status) {
 System.out.println(fileStatus.getPath());
 }
fs.close();
 }

 /**
* 读取hdfs中文件的内容
*/
@Test
public void readData() throws IOException {
 FSDataInputStream in = this.fs.open(new Path("/asdf.txt"));
 List<String> strings = IOUtils.readLines(in,"gbk");
 strings.forEach(System.out::println);
 in.close();
 fs.close();
```

```
}

/**
 * 往hdfs中的文件写内容
 */
 @Test
public void createData() throws IOException {
  FSDataOutputStream out = fs.create(new Path("/test.txt"), true);
  out.writeChars("hello hadoop!");
  out.close();
  fs.close();
}

}
```

具体参考代码中的注释