selenium webdriver 读取 excel 进行数据驱动测试

准备：新建一个 excel 文件，文件名为测试类名，sheet 名为测试方法名

excel 第一行为标题，从第二行开始为测试数据

build path：jxl.jar

code：

------begin

```
import java.io.FileInputStream;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

import org.testng.Assert;

import jxl.*;

/**
 * Excel 放在 Data 文件夹下</p>
 * Excel 命名方式：测试类名.xls</p>
 * Excel 的 sheet 命名方式：测试方法名</p>
 * Excel 第一行为 Map 键值</p>
 *  代码参考郑鸿志的 Blog
 * {@link www.zhenghongzhi.cn/post/42.html}
 * @ClassName: ExcelDataProvider
 * @Description: TODO(读取 Excel 数据)
 */
public class ExcelDataProvider implements Iterator<Object[]> {

    private Workbook book          = null;
    private Sheet    sheet         = null;
    private int      rowNum        = 0;
    private int      currentRowNo = 0;
    private int      columnNum     = 0;
    private String[] columnnName;

    public ExcelDataProvider(String classname, String methodname) {
```

```
        try {

            int dotNum = classname.indexOf(".");

            if (dotNum > 0) {
                classname = classname.substring(classname.lastIndexOf(".") + 1,
                        classname.length());
            }
            //从/data 文件夹下读取以类名命名的 excel 文件
            String path = "data/" + classname + ".xls";
            InputStream inputStream = new FileInputStream(path);

            book = Workbook.getWorkbook(inputStream);
            //取 sheet
            sheet = book.getSheet(methodname);
            rowNum = sheet.getRows();
            Cell[] cell = sheet.getRow(0);
            columnNum = cell.length;
            columnnName = new String[cell.length];

            for (int i = 0; i < cell.length; i++) {
                columnnName[i] = cell[i].getContents().toString();
            }
            this.currentRowNo++;

        } catch (Exception e) {
            e.printStackTrace();
            Assert.fail("unable to read Excel data");
        }
    }

    public boolean hasNext() {

        if (this.rowNum == 0 || this.currentRowNo >= this.rowNum) {

            try {
                book.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
            return false;
        } else {
            // sheet 下一行内容为空判定结束
```

```
                if ((sheet.getRow(currentRowNo))[0].getContents().equals(""))
                    return false;
                return true;
            }
        }

        public Object[] next() {

            Cell[] c = sheet.getRow(this.currentRowNo);
            Map<String, String> data = new HashMap<String, String>();
            // List<String> list = new ArrayList<String>();

            for (int i = 0; i < this.columnNum; i++) {

                String temp = "";

                try {
                    temp = c[i].getContents().toString();
                } catch (ArrayIndexOutOfBoundsException ex) {
                    temp = "";
                }

                // if(temp != null&& !temp.equals(""))
                // list.add(temp);
                data.put(this.columnnName[i], temp);
            }
            Object object[] = new Object[1];
            object[0] = data;
            this.currentRowNo++;
            return object;
        }

        public void remove() {
            throw new UnsupportedOperationException("remove unsupported.");
        }
    }

    ------end
```