

## 数据驱动是什么

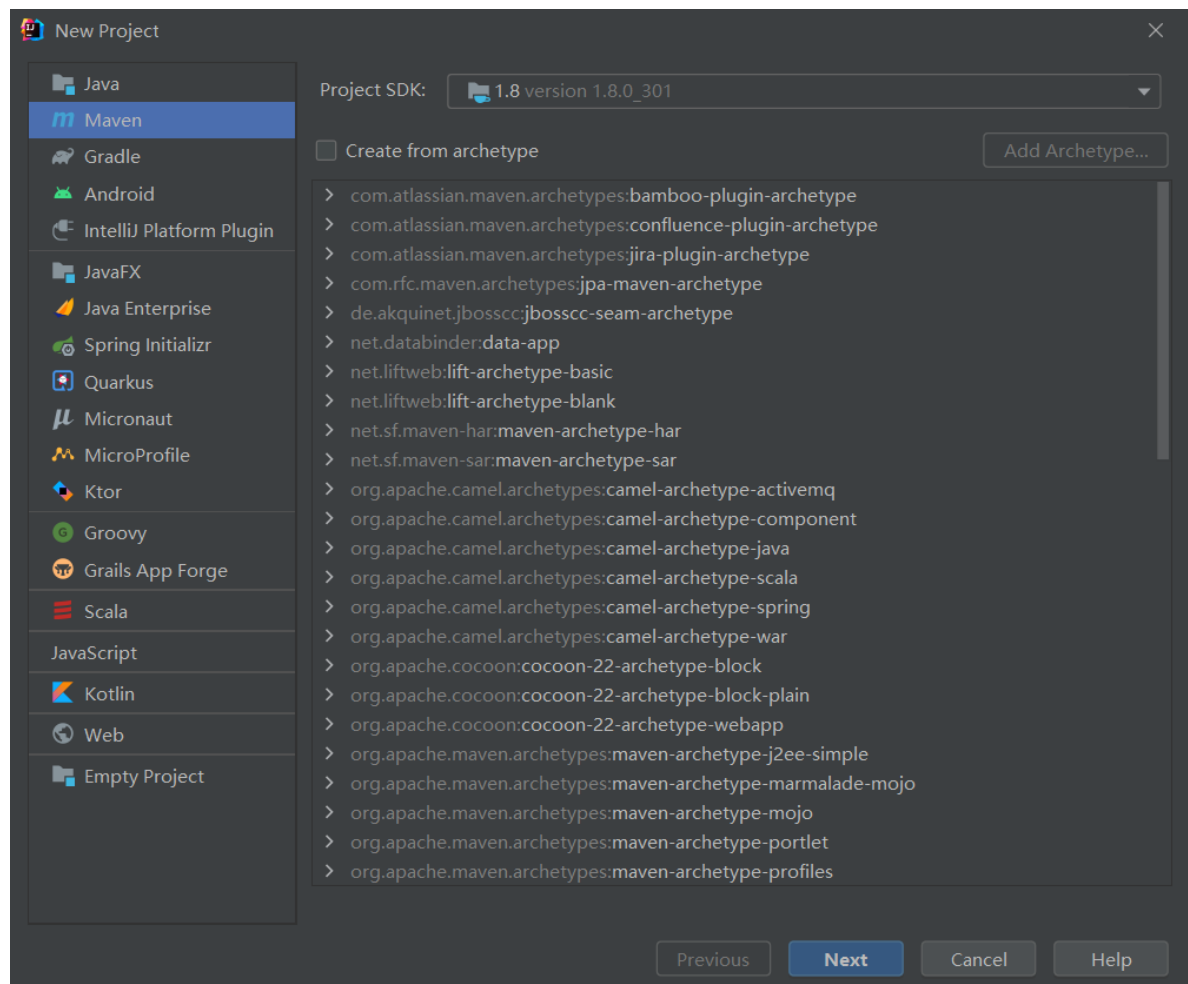
相同的测试脚本使用不同的测试数据，将测试数据与测试行为完全分离，这样的设计模式称为数据驱动，如测试网站的登录使用不同的用户名和密码

### 步骤

- 1、编写测试脚本，包含读取数据功能，数据来源为对象，文件或数据库等
- 2、将测试数据存入外部对象，如对象，文件或数据库等
- 3、运行脚本，读取测试数据
- 4、验证测试结果

## 使用TestNG进行数据驱动

- 1、idea新建maven项目，选择java 1.8



- 2、给项目命名dataProvide\_test，包名后面的example删掉

New Project

Name: dataProvider\_test

Location: E:\project\dataProvider\_test

Artifact Coordinates

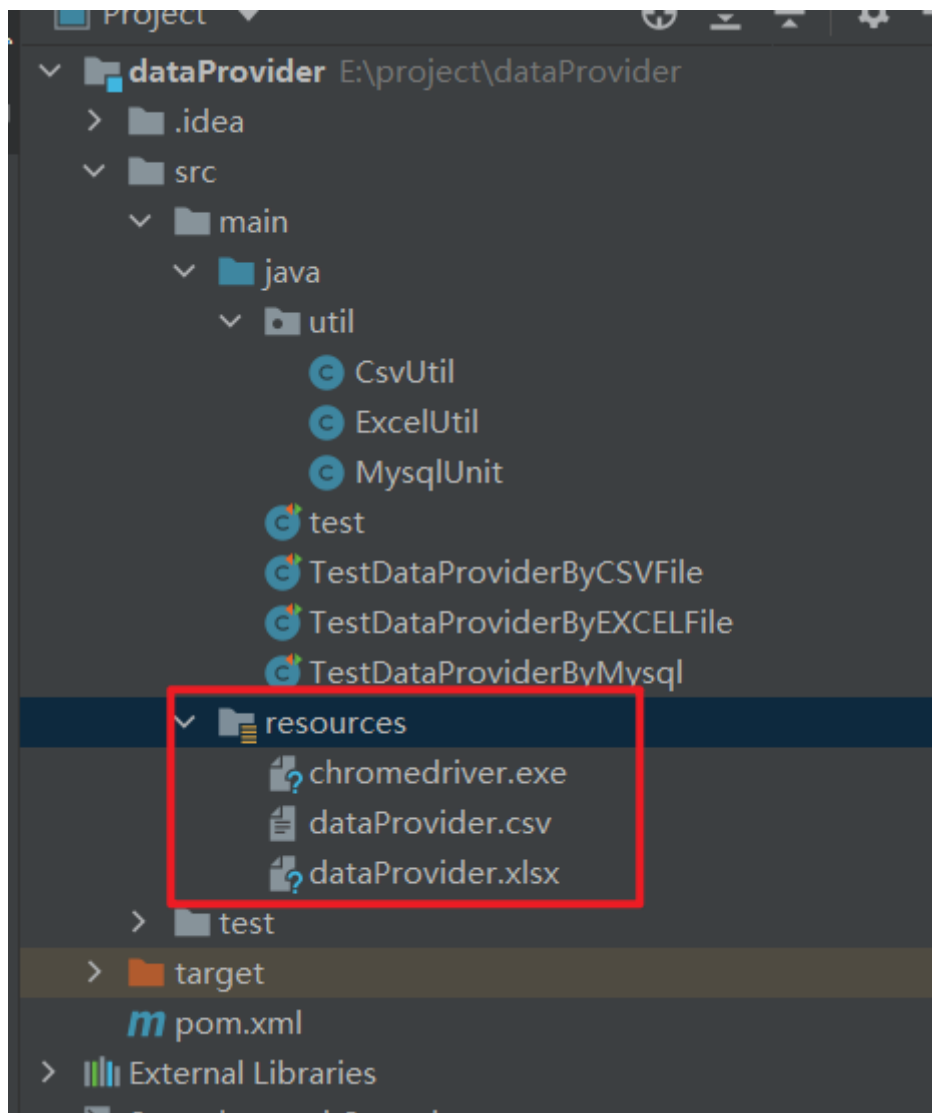
GroupId: org  
The name of the artifact group, usually a company domain

ArtifactId: dataProvider\_test  
The name of the artifact within the group, usually a project name

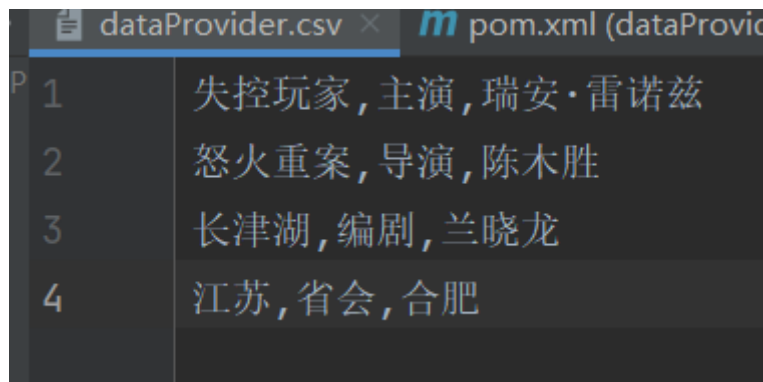
Version: 1.0-SNAPSHOT

Previous Finish Cancel Help

3、把浏览器驱动放入src/main/resources文件夹，驱动从前面的项目中复制过来，csv和xlsx文件自己新建即可



csv内容如下



4、在src/main/java下新建package，命名为util，在包内新建类CsvUtil，作为查询csv文件数据的工具类

```
1 package util;
2
3 import java.io.BufferedReader;
4 import java.io.FileInputStream;
5 import java.io.IOException;
6 import java.io.InputStreamReader;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 public class CsvUtil {
11     public static Object[][] getTestData(String filename)throws IOException{
12         List<Object[]> records = new ArrayList<Object[]>();
```

```

13
14         BufferedReader file = new BufferedReader(new InputStreamReader(
15             new FileInputStream(filename), "UTF-8"));
16         // file.readLine(); //忽略第一行
17         // 遍历表中的每一行
18         String record; //每一行数据
19         while((record = file.readLine()) != null){
20             System.out.println(record);
21             String fields[] = record.split(","); //csv文件的数据均以','分隔，以此为分
            隔，存到数组
22             records.add(fields);
23         }
24         file.close();
25         Object[][] results = new Object[records.size()][];
26         for (int i = 0; i < records.size(); i++) {
27             results[i] = records.get(i);
28         }
29         return results;
30     }
31 }

```

##### 5、在src/main/java下新建类TestDataProviderByCSVFile，作为测试入口

```

1  import org.junit.Assert;
2  import org.openqa.selenium.By;
3  import org.openqa.selenium.WebDriver;
4  import org.openqa.selenium.chrome.ChromeDriver;
5  import org.openqa.selenium.support.ui.ExpectedCondition;
6  import org.openqa.selenium.support.ui.WebDriverWait;
7  import org.testng.annotations.AfterMethod;
8  import org.testng.annotations.BeforeMethod;
9  import org.testng.annotations.DataProvider;
10 import org.testng.annotations.Test;
11 import util.CsvUtil;
12
13 import java.io.IOException;
14
15 public class TestDataProviderByCSVFile {
16     public WebDriver driver;
17     String url = "http://www.baidu.com";
18     public static String filePath = "src/main/resources/dataProvider.csv"; //文件相
        对路径
19
20     @DataProvider(name = "testData")
21     public static Object[][] words() throws IOException{
22         return CsvUtil.getTestData(filePath);
23     }
24     @Test(dataProvider = "testData")
25     public void test(String searchWord1, String searchWord2, String SearchResult)
        {
26         driver.get(url); //打开百度首页
27         driver.findElement(By.id("kw")).sendKeys(searchWord1+" "+searchWord2); //
        在搜索框中输入searchWord1+" "+searchWord2
28         driver.findElement(By.id("su")).click(); //点击"百度一下"按钮
29         //强制等待3S
30         /*try {
31             Thread.sleep(3000);
32         }catch (InterruptedException e){

```

```

33         e.printStackTrace();
34     }*/
35     //显式等待，等待元素加载完毕再进行下一步
36     (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {
37         @Override
38         public Boolean apply(WebDriver d){
39             return d.findElement(By.id("help")).getText().contains("用户反
40 馈");
41         }
42     });
43     Assert.assertTrue(driver.getPageSource().contains(SearchResult)); //断言，
44     //检查页面中是否包含SearchResult元素
45     driver.quit();
46 }
47 @BeforeMethod
48 public void beforeMethod(){
49     driver = new ChromeDriver();
50 }
51 @AfterMethod
52 public void AfterMethod(){
53     driver.quit();
54 }
55 }

```

## 6、在pom.xml文件中加入依赖包

```

1  <dependency>
2      <groupId>org.seleniumhq.selenium</groupId>
3      <artifactId>selenium-java</artifactId>
4      <version>3.141.59</version>
5  </dependency>
6  <dependency>
7      <groupId>org.testng</groupId>
8      <artifactId>testng</artifactId>
9      <version>RELEASE</version>
10     <scope>test</scope>
11 </dependency>
12 <dependency>
13     <groupId>junit</groupId>
14     <artifactId>junit</artifactId>
15     <version>4.13.1</version>
16     <scope>compile</scope>
17 </dependency>
18 <dependency>
19     <groupId>org.testng</groupId>
20     <artifactId>testng</artifactId>
21     <version>7.4.0</version>
22     <scope>compile</scope>
23 </dependency>

```

## 7、在TestDataProviderByCSVFile文件中

@DataProvider(name = "testData") 和 @Test(dataProvider = "testData")的名字必须一致，用以数据绑定，否则运行将报没有名为testData的数据

CsvUtil.getTestData(filePath) 为从CsvUtil类中调用getTestData函数，获取csv文件中的数据

```

@DataProvider(name = "testData")
public static Object[][] words() throws IOException{
    return CsvUtil.getTestData(filePath);
}

@Test(dataProvider = "testData")
public void test(String searchWord1, String searchWord2, String SearchResult){
    driver.get(url); //打开百度首页
    driver.findElement(By.id("kw")).sendKeys(...charSequences: searchWord1+" "+searchWord2);
    driver.findElement(By.id("su")).click(); //点击"百度一下"按钮
}

```

8、打开百度，找到输入框并把前两个词输入，进行等待，等待有两种方式

(1) 强制等待

(2) 显式等待，即等待某一元素加载完毕再进行下一步

```

@Test(dataProvider = "testData")
public void test(String searchWord1, String searchWord2, String SearchResult){
    driver.get(url); //打开百度首页
    driver.findElement(By.id("kw")).sendKeys(...charSequences: searchWord1+" "+searchWord2); //在搜索框中输入"JUnit"
    driver.findElement(By.id("su")).click(); //点击"百度一下"按钮
    //强制等待3S
    try {
        Thread.sleep(millis: 3000);
    } catch (InterruptedException e){
        e.printStackTrace();
    }
    //显式等待，等待元素加载完毕再进行下一步
    (new WebDriverWait(driver, timeOutInSeconds: 10)).until(new ExpectedCondition<Boolean>() {
        @Override
        public Boolean apply(WebDriver d) { return d.findElement(By.id("help")).getText().contains("用户反馈"); }
    });
    Assert.assertTrue(driver.getPageSource().contains(SearchResult));
}

```

最后进行断言，检测是否含有SearchResult元素

9、运行TestDataProviderByCSVFile测试

Assert.assertTrue(driver.getPageSource().contains(searchResult));

可以看到最后一个是明显错误的，因为该断言语句检测的是页面的全部元素，手动输入“江苏 省会”，搜索“合肥”，会发现“合肥”元素是真的存在的

江苏省财政厅政府信息公开指南》,为公民、法人和其他组织获取省财政厅的政府信息提...  
czt.jiangsu.gov.cn/col/col5121... 百度快照

3 江苏省的区划调整,30年时间,省会为何搬迁了4个城市?  
4 2019年12月10日 南直隶地区则改称为江南省,省会在江宁府。江南省经济发达、人口众多,仅仅是苏州府和松江府,就承担了全国10%左右的赋税。清朝对江南省并不放心,在康熙4年,清朝把江南省拆成了两个部...  
5 转瞬间落日残烟 百度快照

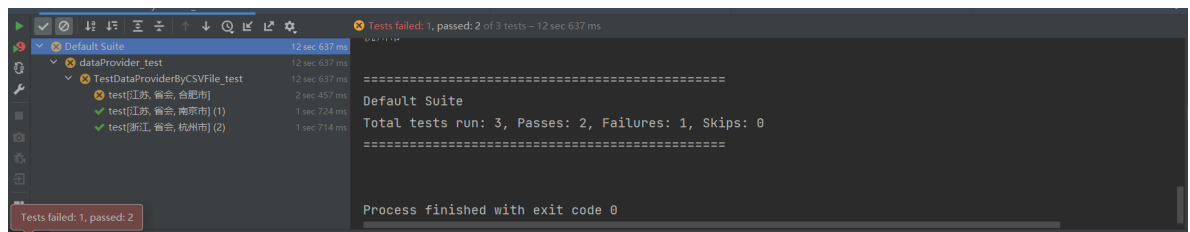
6 江苏省会的最新相关信息  
7 南京是省外安徽马鞍山特别提醒,“徽京”又出圈了 网易  
8 马鞍山是安徽省属的地级市,南京则是隔壁省江苏省的省会。但马鞍山距离安徽省会合肥超过100公里,但到江苏省会南京只有几十公里,坐高铁不到20分钟,每天往返安徽马鞍山...  
9 苏州:难圆的省会之梦 网易  
10 为什么南京会被调侃是安徽的省会?有考虑过合肥... 腾讯  
中国最短命的省会,仅坚持1年半就被摘除,却是中... 网易  
合肥,看了济南南京才知道,做强省会,山东江苏比... 书成有话说

其他人还在搜

为了缩小检查范围，可以用之前的XPath进行定位

```
1 String temp = driver.findElement(By.xpath("//*  
  [@id=\"1\"]/div/div/div/div[2]/div[2]/a")).getText();  
2 System.out.println(temp);  
3 Assert.assertTrue(temp.equals(searchResult));
```

## 10、修改csv文件，再次运行



可以看到目前是我们想要的结果了。

## 使用Excel文件进行数据驱动测试

1、在src/main/resources文件夹下新建一个dataProvider.xlsx存储测试数据，文件内容如下

A	B	C	D
失控玩家	主演	瑞安·雷诺兹	
怒火重案	导演	陈木胜	
长津湖	编剧	兰晓龙	

2、在src/main/java/util下新建类ExcelUtil，作为查询Excel文件数据的工具类

```
1 package util;  
2  
3 import org.apache.poi.hssf.usermodel.HSSFDDateUtil;  
4 import org.apache.poi.hssf.usermodel.HSSFWorkbook;
```

```

5  import org.apache.poi.ss.usermodel.*;
6  import org.apache.poi.xssf.usermodel.XSSFCell;
7  import org.apache.poi.xssf.usermodel.XSSFWorkbook;
8
9  import java.io.*;
10 import java.text.SimpleDateFormat;
11 import java.util.ArrayList;
12 import java.util.List;
13
14 public class ExcelUtil {
15     public static Object[][] getTestData(String filename, String sheetName) throws
IOException {
16         File file = new File(filename);
17         FileInputStream inputStream = new FileInputStream(file);
18         Workbook workbook = null;
19         // 获取文件拓展名
20         String fileExtensionName = filename.substring(filename.indexOf("."));
21         if(fileExtensionName.equals(".xlsx")){
22             workbook = new XSSFWorkbook(inputStream);
23         }
24         else if(fileExtensionName.equals(".xls")){
25             workbook = new HSSFWorkbook(inputStream);
26         }
27
28         Sheet sheet = workbook.getSheet(sheetName);
29         int rowCount = sheet.getLastRowNum()-sheet.getFirstRowNum(); //获取总行数
30         List<Object[]> records = new ArrayList<Object[]>();
31         SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");
32         Cell cell;
33         for (int i = 0; i <= rowCount; i++) {
34             Row row = sheet.getRow(i);
35             String fields[] = new String[row.getLastCellNum()];
36             for (int j = 0; j < row.getLastCellNum(); j++) {
37                 cell = row.getCell(j);
38                 if(cell != null){
39                     //如果是数字类型
40                     if(cell.getCellType()==CellType.NUMERIC){
41                         //判断是不是日期格式
42                         if(DateUtil.isCellDateFormatted(cell)){
43                             fields[j] =
simpleDateFormat.format(cell.getDateCellValue());
44                         }
45                     }
46                     //如果是字符串类型
47                     if(cell.getCellType()==CellType.STRING){
48                         row.getCell(j).setCellType(CellType.STRING);
49                         fields[j] = row.getCell(j).getStringCellValue();
50                     }
51                 }
52             }
53             records.add(fields);
54         }
55     }
56
57     Object[][] results = new Object[records.size()][];
58     for (int i = 0; i < records.size(); i++) {
59         results[i] = records.get(i);
60     }

```



```

61         return results;
62     }
63 }

```

### 3、在pom.xml中导入依赖包

```

1  <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
2      <dependency>
3          <groupId>org.apache.poi</groupId>
4          <artifactId>poi</artifactId>
5          <version>5.1.0</version>
6      </dependency>
7  <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
8      <dependency>
9          <groupId>org.apache.poi</groupId>
10         <artifactId>poi-ooxml</artifactId>
11         <version>5.1.0</version>
12     </dependency>

```

### 4、在src/main/java下新建类TestDataProviderByCSVFile，作为测试入口

```

1  import org.junit.Assert;
2  import org.openqa.selenium.By;
3  import org.openqa.selenium.WebDriver;
4  import org.openqa.selenium.chrome.ChromeDriver;
5  import org.openqa.selenium.support.ui.ExpectedCondition;
6  import org.openqa.selenium.support.ui.WebDriverWait;
7  import org.testng.annotations.AfterMethod;
8  import org.testng.annotations.BeforeMethod;
9  import org.testng.annotations.DataProvider;
10 import org.testng.annotations.Test;
11 import util.CsvUtil;
12 import util.ExcelUtil;
13
14 import java.io.IOException;
15
16 public class TestDataProviderByCSVFile {
17     public WebDriver driver;
18     String url = "http://www.baidu.com";
19     public static String filePath = "src/main/resources/dataProvider.xlsx";
20     public static String sheetName = "Sheet1";
21
22     @DataProvider(name = "testData")
23     public static Object[][] words() throws IOException{
24         return ExcelUtil.getTestData(filePath, sheetName);
25     }
26     @Test(dataProvider = "testData")
27     public void test(String searchWord1, String searchWord2, String SearchResult)
28     {
29         driver.get(url); //打开百度首页
30         driver.findElement(By.id("kw")).sendKeys(searchWord1+" "+searchWord2); //
31         在搜索框中输入"JUnit"
32         driver.findElement(By.id("su")).click(); //点击"百度一下"按钮
33         //强制等待3S
34         /*try {
35             Thread.sleep(3000);
36         } catch (InterruptedException e) {
37             e.printStackTrace();
38         }
39         Assert.assertEquals(SearchResult, "JUnit");
40     }
41 }

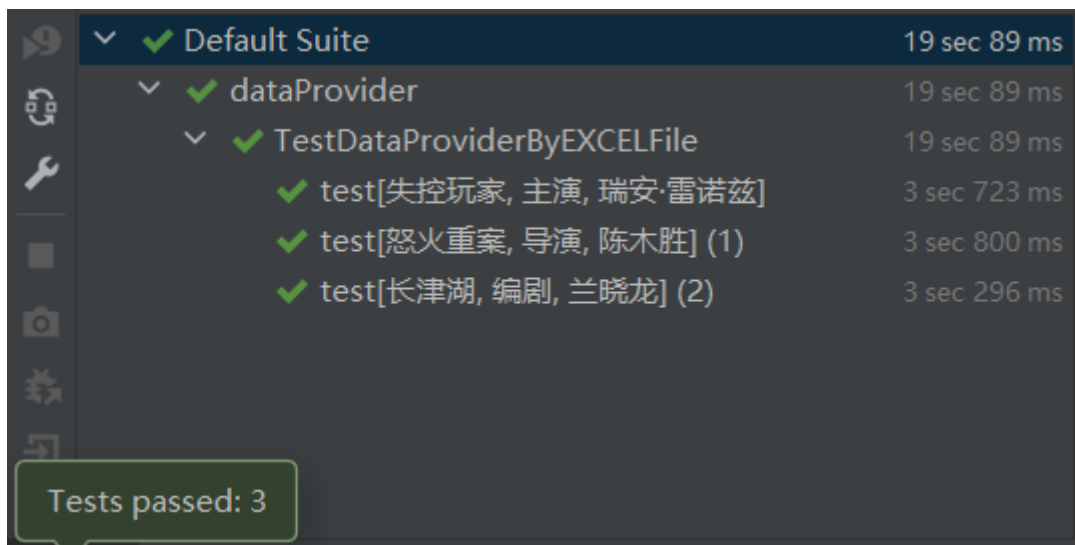
```

```

34         }catch (InterruptedException e){
35             e.printStackTrace();
36         }*/
37         //显示等待，等待元素加载完毕再进行下一步
38         (new WebDriverWait(driver,10)).until(new ExpectedCondition<Boolean>() {
39             @Override
40             public Boolean apply(WebDriver d){
41                 return d.findElement(By.id("help")).getText().contains("用户反
42                 馈");
43             }
44         });
45         Assert.assertTrue(driver.getPageSource().contains(SearchResult));
46         // driver.quit();
47     }
48     @BeforeMethod
49     public void beforeMethod(){
50         System.setProperty("webdriver.chrome.driver",
51             "src/main/resources/chromedriver");// chromedriver服务地址
52         driver = new ChromeDriver();
53     }
54     @AfterMethod
55     public void AfterMethod(){
56         driver.quit();
57     }
58 }

```

5、运行测试，最后得到我们想要的结果



## 使用MySQL数据库实现数据驱动测试

### 测试逻辑

- 1、打开百度首页，从数据库测试数据表中读取每行数据的前两列数据作为搜索词汇。
- 2、在搜索框中输入两个搜索关键词，中间有一个空格。
- 3、单击“搜索”按钮。
- 4、断言搜索结果页面是否包含数据库测试数据表中每行数据的第三列数据内容，包含则认为测试执行成功，否则认为测试执行失败。

## 测试环境准备

### MySQL安装

Windows10下Mysql安装

安装链接([https://blog.csdn.net/weixin\\_42683679/article/details/81428591](https://blog.csdn.net/weixin_42683679/article/details/81428591))

驱动部分不用看

驱动部分直接在pom.xml中添加

```
1 <dependency>
2     <groupId>mysql</groupId>
3     <artifactId>mysql-connector-java</artifactId>
4     <version>6.0.6</version><!-- 选择合适自己的版本 -->
5 </dependency>
```

### 数据库数据表建立

- 1、使用cmd命令，执行mysql -uroot -p可以直接回车也可以直接在后面加上你的数据库的密码登录MySQL数据库。
- 2、执行 `create database dataProvider;`，创建一个数据库，用于保存测试数据。
- 3、执行命令 `use dataProvider;`，切换到dataProvider数据库进行后续操作。
- 4、执行如下SQL语句创建测试数据表。

```
CREATE TABLE testdata (Movie_Name char(30),Movie_Property char(30),Expect_result char(30));
```

- 5、执行如下SQL语句插入4条数据，用于后续测试。

```
INSERT INTO testdata VALUES("千与千寻","上映日期","2001");
```

```
INSERT INTO testdata VALUES("功夫","主演","周星驰");
```

```
INSERT INTO testdata VALUES("泰坦尼克号","主演","莱昂纳多");
```

每条SQL语句执行后显示"Query OK,1 row affected"，表示执行成功。

输入 `select * from testdata;` 查看是否创建成功。

```
[mysql> select * from testdata;
+-----+-----+-----+
| Movie_Name | Movie_Property | Expect_result |
+-----+-----+-----+
| 千与千寻   | 上映日期       | 2001          |
| 功夫       | 主演           | 周星驰        |
| 泰坦尼克号 | 主演           | 莱昂纳多      |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

### 代码测试

- 1、在src/main/java/util下新建类MysqlUnit，来读取数据库中数据

```
1 package util;
```

```

2
3 import java.io.BufferedReader;
4 import java.io.FileInputStream;
5 import java.io.IOException;
6 import java.io.InputStreamReader;
7 import java.sql.*;
8 import java.util.ArrayList;
9 import java.util.List;
10
11
12 public class MysqlUnit {
13     public static Object[][] getTestData(String tableName) throws IOException {
14         String driver = "com.mysql.cj.jdbc.Driver";
15         String url = "jdbc:mysql://127.0.0.1:3306/dataProvider?
serverTimezone=GMT&&useSSL=false";
16         String user = "root";
17         String password = "root";
18         List<Object[]> records = new ArrayList<Object[]>();
19         try {
20             Class.forName(driver);
21             Connection conn = DriverManager.getConnection(url, user, password);
22             if(!conn.isClosed())
23                 System.out.println("数据库连接成功");
24             Statement statement = conn.createStatement();
25             String sql = "select * from " + tableName;
26             ResultSet rs = statement.executeQuery(sql);
27             ResultSetMetaData resultSetMetaData = rs.getMetaData();
28             int cols = resultSetMetaData.getColumnCount();
29             while(rs.next()){
30                 String fields[] = new String[cols];
31                 int col = 0;
32                 for (int colIdx = 0; colIdx < cols; colIdx++) {
33                     fields[col] = rs.getString(colIdx+1);
34                     col++;
35                 }
36                 records.add(fields);
37                 System.out.println(rs.getString(1)+" "+rs.getString(2)+"
"+rs.getString(3));
38             }
39             rs.close();
40             conn.close();
41         } catch (ClassNotFoundException e){
42             System.out.println("未能找到驱动类");
43             e.printStackTrace();
44         } catch (SQLException e){
45             e.printStackTrace();
46         } catch (Exception e){
47             e.printStackTrace();
48         }
49
50         Object[][] results = new Object[records.size()][cols];
51         for (int i = 0; i < records.size(); i++) {
52             results[i] = records.get(i);
53         }
54         return results;
55     }
56 }

```

## 2、在src/main/java下新建类TestDataProviderByMysql，作为测试入口

```
1  import org.junit.Assert;
2  import org.openqa.selenium.By;
3  import org.openqa.selenium.WebDriver;
4  import org.openqa.selenium.chrome.ChromeDriver;
5  import org.openqa.selenium.support.ui.ExpectedCondition;
6  import org.openqa.selenium.support.ui.WebDriverWait;
7  import org.testng.annotations.AfterMethod;
8  import org.testng.annotations.BeforeMethod;
9  import org.testng.annotations.DataProvider;
10 import org.testng.annotations.Test;
11 import util.ExcelUtil;
12 import util.MysqlUnit;
13
14 import java.io.IOException;
15
16
17 public class TestDataProviderByMysql {
18     public WebDriver driver;
19     String url = "http://www.baidu.com";
20     public static String tableName = "testdata";
21
22     @DataProvider(name = "testData")
23     public static Object[][] words() throws IOException{
24         return MysqlUnit.getTestData(tableName);
25     }
26     @Test(dataProvider = "testData")
27     public void test(String searchWord1, String searchWord2, String SearchResult)
28     {
29         driver.get(url); //打开百度首页
30         driver.findElement(By.id("kw")).sendKeys(searchWord1+" "+searchWord2); //
31         在搜索框中输入"JUnit"
32         driver.findElement(By.id("su")).click(); //点击"百度一下"按钮
33         //显示等待，等待元素加载完毕再进行下一步
34         (new WebDriverWait(driver,10)).until(new ExpectedCondition<Boolean>() {
35             @Override
36             public Boolean apply(WebDriver d){
37                 return d.findElement(By.id("help")).getText().contains("用户反
38                 馈");
39             }
40         });
41         Assert.assertTrue(driver.getPageSource().contains(SearchResult));
42         // driver.quit();
43     }
44     @BeforeMethod
45     public void beforeMethod(){
46         System.setProperty("webdriver.chrome.driver",
47         "src/main/resources/chromedriver");// chromedriver服务地址
48         driver = new ChromeDriver();
49     }
50     @AfterMethod
51     public void AfterMethod(){
52         driver.quit();
53     }
54 }
```

如果出现错误 [Authentication plugin 'caching\_sha2\_password' cannot be loaded]

可以使用下列代码修改数据库密码加密方式 `ALTER USER 'yourusername'@'localhost' IDENTIFIED WITH mysql_native_password BY 'yourpassword';`

3、执行代码得到我们想要的结果

▼	✓	Default Suite	30 sec 470 ms
▼	✓	softTest	30 sec 470 ms
▼	✓	TestDataDrivenByMysqlDatabase	30 sec 470 ms
	✓	test[千与千寻, 上映日期, 2001]	7 sec 124 ms
	✓	test[功夫, 主演, 周星驰] (1)	5 sec 8 ms
	✓	test[泰坦尼克号, 主演, 莱昂纳多] (1)	8 sec 675 ms