Atitit 动态js内容爬虫的抓取总结

WebDriver+ff+chrome 无头模式 推荐

1. Htmlunit （经测试不行 1

1.1. phantomjs-2.1.1 不推荐，已经落后

经测试抓到部分，但无法抓取数据表格部分 2

package com.kok.sport.utils.mockdata;

import java.io.File;

import java.io.IOException;

import java.util.Date;

import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;

import org.apache.logging.log4j.LogManager;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.firefox.FirefoxOptions;

import org.openqa.selenium.htmlunit.HtmlUnitDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

//import org.openqa.selenium.remote.DesiredCapabilities;

public class WebdriverFirefox {

static org.apache.logging.log4j.Logger logger = LogManager.getLogger(WebdriverFirefox.class);

public static void main(String[] args) throws Exception {

// declaration and instantiation of objects/variables

String gkdv = "D:\\prj\\spdjs\\node\_modules\\geckodriver\\geckodriver.exe";

if(new File(gkdv).exists())

System.setProperty("webdriver.gecko.driver", gkdv);

else

System.setProperty("webdriver.gecko.driver", "/geckodriver");

FirefoxOptions chromeOptions=new FirefoxOptions();

//设置 chrome 的无头模式

chromeOptions.setHeadless(Boolean.TRUE);

//启动一个 chrome 实例

// webDriver = new ChromeDriver(chromeOptions);

WebDriver driver = new FirefoxDriver(chromeOptions);

// org.openqa.selenium.remote.s

// DesiredCapabilities IEcaps = DesiredCapabilities.htmlUnit();

//INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS

// IEcaps .setCapability(HtmlUnitDriver.INVALIDSELECTIONERROR,true);

// WebDriver driver = new HtmlUnitDriver(BrowserVersion.FIREFOX);

// Set implicit wait

driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);

// comment the above 2 lines and uncomment below 2 lines to use Chrome

System.setProperty("webdriver.chrome.driver",

"D:\\prj\\spdjs\\node\_modules\\chromedriver\\lib\\chromedriver\\chromedriver.exe");

// WebDriver driver =new ChromeDriver();

// InternetExplorerDriver lang.UnsatisfiedLinkError:

// C:\Users\Administrator\AppData\Local\Temp\webdriver3771754343790130835.tmp:

// Can't load IA 32-bit .dll on a AMD 64-bit platform

// org.openqa.selenium.ie.InternetExplorerDriver.class

// new ChromeDriver();

String baseUrl = "https://live.leisu.com/";

String expectedTitle = "Welcome: Mercury Tours";

String actualTitle = "";

// launch Fire fox and direct it to the Base URL

driver.get(baseUrl);

// get the actual value of the title

actualTitle = driver.getTitle();

/\*

\* compare the actual title of the page with the expected one and print the

\* result as "Passed" or "Failed"

\*/

if (actualTitle.contentEquals(expectedTitle)) {

System.out.println("Test Passed!");

} else {

System.out.println("Test Failed");

}

Thread.sleep(12000);

while (true) {

String pageSource = driver.getPageSource();

Leisu.processTime(pageSource);

FileUtils.write(new File("d:\\cache\\leisu.htm"), pageSource,false);

try {

FileUtils.write(new File("/leisu.htm"), pageSource,false);

} catch (Exception e) {

logger.error(e);

}

Thread.sleep(7000);

logger.info(new Date());

System.out.println(new Date());

}

// close Fire fox

// driver.close();

// System.out.println("f");

}

}

# Ref

Atitit.数据检索与网络爬虫与数据采集的原理概论

1. 信息检索 1

1.1. 《信息检索导论》(（美）曼宁...)【简介\_书评\_在线阅读】 - dangdang.html 1

1.2. 《现代信息检索(原书第2版)（由信息检索领域的代表人物撰写，及时掌握现代信息检索关键主题的详细知识）》(（智）贝泽耶茨...) 2

2. 网络爬虫 2

2.1. 第8章 web爬取199 3

2.2. 《用Python写网络爬虫》([澳]理查德...) 4

3. 数据采集 4

3.1. 《Python网络数据采集》(...)【简介\_书评\_在线阅读】 - dangdang.html 4

4. 爬虫框架与工具 5

5. 参考资料 5

Atitit 雷速比赛数据抓取 爬虫抓取总结

经过测试headless webview不能抓取到数据，只有使用带头模式webview firefox webdriver可以获取到雷速数据

目录

1. Htmlunit （经测试不行 1

1.1. phantomjs-2.1.1 经测试抓到部分，但无法抓取数据表格部分 2

1.2. Webdriver+ff （ok 3

Atitit 网络爬虫与数据采集器的原理与实践attilax著 v2

1. 数据采集 1

1.1. http lib 1

1.2. HTML Parsers， 1

1.3. 怎么制作爬虫 2

1.4. 网页爬虫组成： 3

1.4.2. 爬虫进阶 爬虫的进阶就是需要与数据来源方斗智斗勇 3

2. 实现类库框架 4

3. 问题与难点（html转txt) 4

4. 参考资料 4

爬虫的一些知识点

目录

1. 网络爬虫 1

2. 产生背景 垂直领域搜索引擎 2

3. 1 聚焦爬虫工作原理以及关键技术概述 3

4. 涉及技术 3

4.1. 下载网页 一般是通过net api 3

4.2. 分析网页（html分析，接口可能有json 3

5. 分类 3

5.1. 通用网络爬虫（General Purpose Web Crawler）、聚焦网络爬虫（Focused Web Crawler）、增量式网络爬虫（Incremental Web Crawler）、深层网络爬虫（Deep Web Crawler）。 4

5.2. 聚焦网络爬虫 4

5.3. Deep Web 爬虫 4

6. 网页分析算法 5

6.1. 网页分析算法可以归纳为基于网络拓扑、基于网页内容和基于用户访问行为三种类型。 5

7. 言归正传，java实现网络爬虫一般有五种方法 7

7.1. 2.基于HttpURLConnection类编写爬虫：java se的net包的核心类，主要用于http的相关操作。简单 7

7.2. 3.基于apache的HttpClient包编写爬虫：由net包拓展而来，专为java网络通信编程而服务。常用 7

7.3. 5.基于Selenium或者是WebDriver之类的有头（有界面）浏览器。。适合于复杂界面 8

8. 核心代码范例 8

8.1. 下载网页 8