

HttpUnit 本质上相当于一个后台的透明的浏览器引擎，使用java中的HttpUnit可以实现模拟点击按钮，抓取网页元素，实现动态爬虫，之前一直使用jsoup进行爬虫，不过需要动态处理，发现了httpUnit这个好的框架。

1.几行代码实现获取网页源代码。

```
final WebClient webClient=new WebClient();
final HtmlPage page=webClient.getPage("http://www.baidu.com");
System.out.println(page.asText()); //asText()是以文本格式显示
System.out.println(page.asXml()); //asXml()是以xml格式显示
webClient.closeAllWindows();
```

2.模拟文本框与模拟按钮提交，下面我们模拟实现模拟百度搜索关键字。

// 得到浏览器对象，直接New一个就能得到，现在就好比说你得到了一个浏览器了

```
WebClient webclient = new WebClient();
```

// 这里是配置一下不加载css和javaScript，因为httpunit对javascript兼容性不太好

```
webclient.getOptions().setCssEnabled(false);
webclient.getOptions().setJavaScriptEnabled(false);
```

// 做的第一件事，去拿到这个网页，只需要调用getPage这个方法即可

```
HtmlPage htmlpage = webclient.getPage("http://baidu.com");
```

// 根据名字得到一个表单，查看上面这个网页的源代码可以发现表单的名字叫“f”

```
final HtmlForm form = htmlpage.getFormByName("f");
```

```
// 同样道理，获取”百度一下“这个按钮
final HtmlSubmitInput button = form.getInputByValue("百度一下");
// 得到搜索框
final HtmlTextInput textField = form.getInputByName("q1");
//搜索我的id
textField.setValueAttribute("th是个小屁孩");
// 输入好了，我们点一下这个按钮
final HtmlPage nextPage = button.click();
// 我把结果转成String
String result = nextPage.asXml();

System.out.println(result); //得到的是点击后的网页
```