**基于DevTools协议+Chromium headless的客户端爬虫框架**

参考：<http://blog.csdn.net/cteng/article/details/51346094>

之前的做法是使用PhantomJS以及一个html+嵌套iframe包含目标网站URL+跨域dom操作的简单性能优化。

PhantomJS实现下面的核心需求：  
（1）无头模式，然而PhantomJS内核基于老版本的QtWebKit，与最新版本的Chromium代码相比，版本太老了，很多特性用不了（虽然目前国内网站应该也还没开始用上这些？比如ServiceWorker、CSS Custom Properties、Web Components等等）  
（2）可以等待PhantomJS将页面load完成之后，注入JS执行，这之间需要支持下列核心特性：  
    2.1 可以通过注入JS代码的方法导入外部JS资源，一般就是动态插入一个<script>元素，并设置src属性；  
    2.2 可以用注入的JS代码来自动化地控制页面的scroll，或者根据特定网站的代码直接调用相关的JS Event Handler函数  
    2.3 当然，PhantomJS应该允许从外部直接触发用户输入事件，比如根据某个CSS Selector Path定位到某个Element，然后触发其click事件，浏览器本身通常不支持这么做，但是浏览器的扩展，比如WebDriver这样的原本设计用于自动化Web测试的工具支持这么做

现在的问题是，DevTools支持不支持这些核心需求？

DevTools是一种基于WebSocket通信的协议，请求响应数据都是JSON格式，可以用一个桌面版本的Chrome PC版浏览器连接到手机上的Chrome for Android浏览器进行远程调试，但假如我并不需要这种UI界面呢？这里的客户端爬虫框架需要的是自动化处理，而不是UI+用户交互，DevTools理论上应该是可以支持的，但是可能存在某些限制或使用上的不便，需要进一步进行封装（用NodeJS）？

Check List：逐一检查DevTools协议支持不支持上面的3个核心特性需求？？？

Chromium内核中的headless顶级模块好像是从M49版本开始加入的，它提供了多进程的、无头模式的、支持DevTools的自动化支持。

headless模块的编译设置里去除了对于gpu的依赖。（但是不清楚它实现了多大程度上的无头模式？？？如果真的完全支持的话，照理说应该可以成功编译出一个Windows命令行可执行文件，并且只依赖于网络和文件IO、多线程/IPC这些不需要UI界面参与的系统API）

问题（TODO）：  
（1）需要Git获取最新版本的Chromium内核代码，并使用新的GN，而不是gyp进行编译

PS：不过我最初设想的客户端爬虫框架是给页面DOM树提供一套新的脚本binding（比如使用Lua），或者也可以实现成一种DSL专用爬虫任务描述语言。这有好处：

（1）不需要用JS写繁琐的代码  
（2）页面原生的JS代码完全感觉不到有爬虫在抓取数据的存在，但是外部注入JS的方法理论上避免不了

但是缺点是：  
（1）实现难度大，可能需要在内核代码级别做修改，增加一套新的bindings API，但我好像从来没听说过如何bind一个JavaScript以外的脚本语言？  
（2）这个新的脚本语言仍然是受限制的：比如说，它不应该修改dom，但是即使如此，考虑到页面原生JS可能正在并发的修改dom，这个地方不能有读写冲突；同时，它应该提供根据某种语法定位元素的API（可以重用CSS Selector的概念）；它应该支持dom range对象的定制化的序列化数据导出；同时，它需要允许以编程的方式（脚本自动化）注入用户输入事件，让页面原生JS代码“以为”发生了scroll、gesture、click等UI交互操作。

客户端爬虫也需要能够加载“广告过滤拦截”功能，这个倒是可以在headless代码的基础上修改实现的，但是现有方法html+iframe处理36kr网站的时候就出问题了：它在设置iframe的srcdoc属性为html字符串的时候，过滤掉了所有的<script>元素，这导致页面内容不能正常渲染为dom，原因是36kr现在使用了react.js+json数据来做render。

其实如果直接基于那个json数据来做爬虫数据导出也不是不可以。。。