


蒋国纲的技术博客

技术为解决问题而生

公告



蒋国纲(Guogang Jiang)
热能与动力工程专业毕业，曾从事过金融、游戏、信息安全、工业控制、仪器仪表、移动应用及车联网等领域的软件开发工作，曾在Windows, Windows Mobile, Linux等平台从事过大量C++编程工作，如今在.net, java, objective c, javascript等开发技术里均有涉猎。目前主要从事企业管理类软件开发。工作之余喜欢写写总结，写写博客。
旧博客地址：
2006-2009 CSDN.com
2009-2011 CppBlog.com
昵称: guogangj
园龄: 5年
粉丝: 271
关注: 2
+加关注

搜索

找找看

谷歌搜索

最新随笔

1. ASP.NET MVC防范CSRF最佳实践

2. 让ASP.NET接受有“潜在危险”的提交

3. 读完成稿

4. 再见Windows C++

5. 用Maven部署war包到远程Tomcat服务器

6. 用IntelliJ IDEA创建Gradle项目简单入门

7. Java的异步HttpClient

8. 高性能的关键: Spring MVC的异步模式

9. .net的简易多线程处理

10. 快速了解必要的网络知识

我的标签

dotNet(15)

IT技巧(13)

开发的故事(11)

JavaWeb(11)

ASP.net/MVC(8)

Web API(8)

职业感悟(8)

WPF(7)

其它(7)

Unix/Linux(6)

更多

随笔档案

2017年1月 (1)

2016年12月 (2)

2016年8月 (1)

2016年5月 (5)

2016年4月 (1)

2016年3月 (5)

2016年2月 (3)

2016年1月 (2)

2015年11月 (1)

2015年7月 (1)

2015年1月 (2)

Chrome插件（Extensions）开发攻略

随笔-112 文章-0 评论-432

本文将从个人经验出发，讲述为什么需要Chrome插件，如何开发，如何调试，到哪里找资料，会遇到怎样的问题以及如何解决等，同时给出一个个人认为的比较典型的例子——获取网页内容，和服务器交互，再把信息反馈给用户。OK，准备开始吧，我尽量把文章写得好看点，以免读者打瞌睡。

目录

1. 为什么需要

2. 为什么是Chrome

3. 需要准备什么

4. 如何开始

5. Page Action

6. Chrome插件结构

7. 学习资料

8. 我的例子

9. 调试
 - 调试Content Scripts
 - 调试Background
 - 调试Popup
 - 一些问题

10. 总结

为什么需要

简单地说，浏览器插件，可以大大的扩展你的浏览器的功能。包括但不仅限于这些功能：捕捉特定网页的内容，捕捉HTTP报文，捕捉用户浏览动作，改变浏览器地址栏/起始页/书签/Tab等界面元素的行为，与别的站点通信，修改网页内容.....给你增加许多想象空间，试想想看，你可以用它来识别一些网站上的广告代码，并直接把这些代码删掉，这样你就不会受到广告的困扰了，没错，如你所愿，这样的插件别人已经开发好了，你可以直接用。不过，也要说浏览器插件的弊端，那就是：会带来一些安全隐患，也可能让你的浏览器变得缓慢甚至不稳定。

为什么是Chrome

因为Chrome的插件开发起来最简单，总体上看没什么新的技术，开发语言就是javascript，web前端工程师能很快上手；而Firefox的插件开发则复杂许多，涉及到环境的搭建和一些WEB以外的技术；IE的插件开发就更复杂了，需要熟悉C++和COM技术，当然还要装微软的Visual Studio。

这里有篇老外写的文章，对比Chrome、Opera和Firefox的插件开发的：<http://blog.nparashuram.com/2011/10/writing-browser-extensions-comparing.html>。

应该说Chrome和Opera的插件的开发都不难，但Firefox的则比较棘手，也许你要问，那为什么Firefox的插件是最丰富的？我想这有些历史原因，Chrome出来毕竟比较晚，另外几种浏览器提供的插件的功能也是不尽相同的，OK，我们还是言归正传吧。

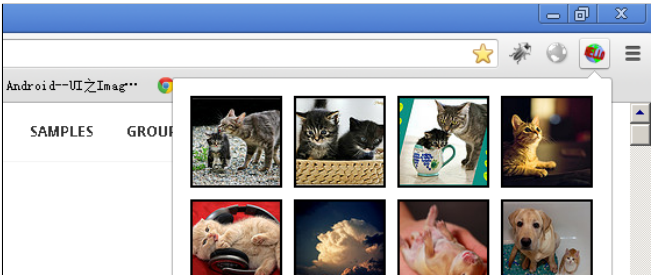
需要准备什么

几乎是零需求。Chrome浏览器和一个文本编辑器即可，文本编辑器最好是带语法高亮的那种。谷歌对我们做技术的人来说真是太度了。

如何开始

强烈建议看看官方的说明：<https://developer.chrome.com/extensions/getstarted.html>。

文章不长，照着文章去做，完成后，你就成功开发了第一个Chrome插件，这个插件会弹出一个小窗口，上面显示些阿猫阿狗的小图片。如图：



2014年11月 (2)

2014年10月 (5)

2014年9月 (1)

2014年8月 (4)

2014年5月 (1)

2014年4月 (1)

2014年3月 (6)

2014年2月 (2)

2014年1月 (2)

2013年12月 (1)

2013年10月 (6)

2013年8月 (1)

2013年7月 (2)

2013年6月 (1)

2013年5月 (4)

2013年4月 (4)

2013年3月 (8)

2013年2月 (3)

2013年1月 (6)

2012年12月 (1)

2012年11月 (1)

2012年10月 (4)

2012年9月 (2)

2012年8月 (2)

2012年7月 (5)

2012年6月 (1)

2012年4月 (2)

2012年3月 (8)

2012年2月 (2)

相关博客

最新评论

1. Re:Java的异步HttpClient

不是纯异步 execute会阻塞。。。 --molixiaoge

2. Re:给ASP.net Web API的

Controllor分类

```
config.Routes.MapHttp(
    name: "DefaultArea",
    routeTemplate: "{area}/{controller}/{id}",
    defaults: new {
        area = "DefaultArea"
    });

config.Routes.MapHttp(
    name: "about/{area}/{controller}/{id}",
    routeTemplate: "about/{area}/{controller}/{id}",
    defaults: new {
        area = "DefaultArea"
    });
```

我配置了两个路由，默认加了一个Controller

Controllers

Home

Products

DefaultAreaControlCont



这个插件一共有4个文件：

- manifest.json - 所有插件都要有这个文件，这是插件的配置文件，可看作插件的“入口”。
- icon.png - 小图标，推荐使用19*19的半透明png图片，更好的做法是同时提供一张38*38的半透明的png图片作为大图标，在我后面提供的例子中，我就是那么干的。
- popup.html - 就是你所看到的那个阿猫阿狗的弹出页面。
- popup.js - 阿猫阿狗页面所引用的javascript文件。

这里千万千万注意了，我当初没仔细看popup.html里有一小段注释，这一小段注释说：**出于安全考虑，javascript必须与html分开存放**。而我想嘛，一个小测试程序，没必要分开吧，直接写一起不就行了吗？结果javascript死活执行不了，我翻来覆去找不到原因，还以为弹出的小窗口不支持javascript，在网上搜索了半天又没有结果，最后才发现是这个原因，浪费了许多时间，这个事情也一定程度上说明了：细节决定成败。

manifest.json中的内容也非常显而易见，我选择其中几个属性讲一下：

```
{
  "manifest_version": 2,

  "name": "One-click Kittens",
  "description": "This extension demonstrates a browser action with kittens.",
  "version": "1.0",

  "permissions": [
    "https://secure.flickr.com/"
  ],
  "browser_action": {
    "default_icon": "icon.png",
    "default_popup": "popup.html"
  }
}
```

"manifest_version": 现在应该总是2。

"permissions": 很重要的东西，即允许插件做哪些事情，访问哪些站点，假如一个插件的"permissions"里写有"http://*.hacker.com/"，那么这个插件就被允许访问hacker.com上的所有内容，包括可能会把你的一些个人信息提交给hacker.com，危险性不言而喻，查看一个插件能访问那些站点的方法是：在chrome的地址栏里输入"chrome://extensions/"（注意：这个页面我们之后要频繁用到，请收藏一下），然后点对应插件的旁边的那个“权限”，如：

Firebug Lite for Google Chrome™ 1.4.0.11967

Firebug Lite for Google Chrome, supported by the Firebug Working Group [权限 访问网站](#)

ID : bmagokdooijbeehmknfglimnifench

检查视图 : [background.html](#)

☐ 在隐身模式下启用

Proxy SwitchySharp 1.10.2

轻松快捷地管理和切换多个代理设置。基于 "Proxy 访问网站"

ID : dpplabbmogkhghncfbfdeeoekofdjegm

目前权限

Firebug Lite for Google Chrome™

该程序可以：

- 访问您在所有网站上的数据
- 访问您的标签页和浏览活动

关闭

"default_popup": 用来指定点击小图标后弹出的小窗口中默认显示的是哪个html，这个弹出的小窗口就叫做“popup”。

"browser_action": 这是一个浏览级别的动作，也就是说，不管你现在在访问哪个页面，那个小按钮总是显示出来，而我们的插件如果仅仅是针对某些页面的话，就不适合用这个"browser_action"了。下面我们来弄一个只有访问博客园（www.cnblogs.com）才会出现的小按钮。

Page Action

http://www.cnblogs.com/guogangj/p/3235703.html

2/8

chrome-plugin-page-action-demo./z

这个插件只有4个文件，其中两个还是图标，那就只剩下一个必须的manifest.json和一个background.js了。

manifest.json:

```
{
  "manifest_version": 2,
  "name": "cnblogs.com viewer",
  "version": "0.0.1",
  "background": { "scripts": ["background.js"] },
  "permissions": ["tabs"],
  "page_action": {
    "default_icon": {
      "19": "cnblogs_19.png",
      "38": "cnblogs_38.png"
    },
    "default_title": "cnblogs.com article information"
  }
}
```

注意：这里是“page_action”而不是“browser_action”属性了。

“permissions”属性里的“tabs”是必须的，否则下面的js不能获取到tab里的url，而这个url是我们判断是否要把小图标show出来的依据。background是什么概念？这是一个很重要的东西，可以把它认为是chrome插件的主程序，理解这个很关键，一旦插件被启用（有些插件对所有页面都启用，有些则只对某些页面启用），chrome就给插件开辟了一个独立的javascript运行环境（又称作运行上下文），用来跑你指定的background script，在这个例子中，也就是background.js。

background.js

```
function getDomainFromUrl(url){
    var host = "null";
    if(typeof url == "undefined" || null == url)
        url = window.location.href;
    var regex = /.*/:/([^\/*]*).*/;
    var match = url.match(regex);
    if(typeof match != "undefined" && null != match)
        host = match[1];
    return host;
}

function checkForValidUrl(tabId, changeInfo, tab) {
    if(getDomainFromUrl(tab.url).toLowerCase() == "www.cnblogs.com"){
        chrome.pageAction.show(tabId);
    }
};

chrome.tabs.onUpdated.addListener(checkForValidUrl);
```

代码中，我们使用了一个正则表达式去匹配url，获取出其中的domain部分，如果domain部分是“www.cnblogs.com”的话，就把小图标show出来，效果如下：



当然了，你现在点那个小图标的话，是没有任何反应的，我没有像官方提供的那个例子那样提供了popup。OK，现在是时候描述下chrome插件的结构了。

Chrome插件结构

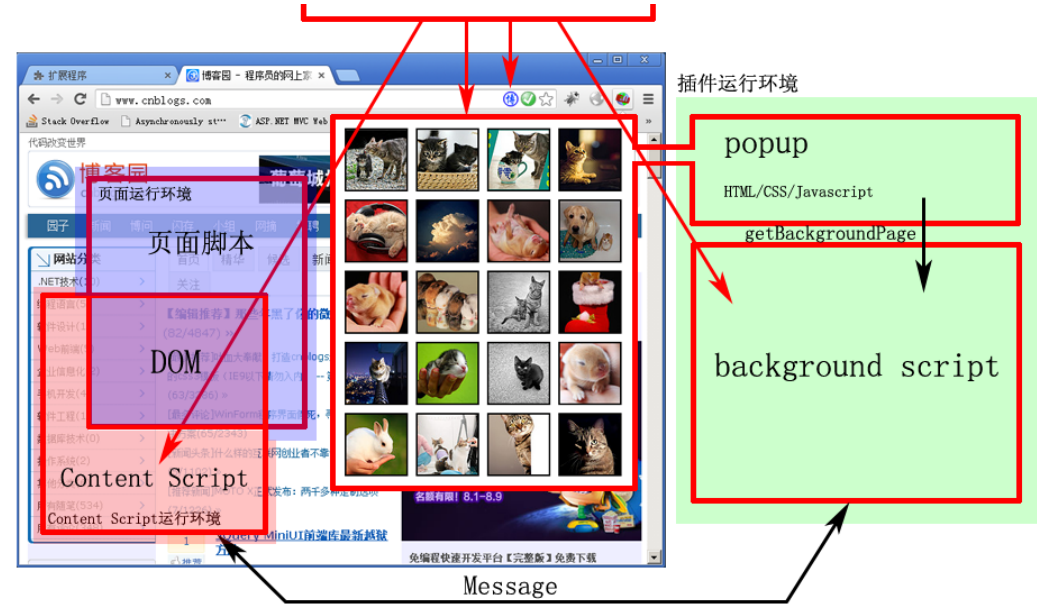
需要声明的是，这个结构图是我自己画的，代表我对Chrome插件的理解，可能并不全面，甚至还不是很准确，但找来找去找不到现成的，只好自己动手，如有谬误，请不吝指出。

配置/入口

manifest.json

阅读排行榜

- 1. Chrome插件（Extensions）开发攻略(64700)
- 2. 搞个这样的APP要多久？(26833)
- 3. Mac OS X上IntelliJ IDEA 13与Tomcat 8的Java Web开发环境搭建(15819)
- 4. 用IntelliJ IDEA创建Gradle项目简单入门(15688)
- 5. 如何实现RESTful Web API的身份验证(13279)



如图，manifest.json作为插件的配置文件，同时可以看作程序的“入口”，因为它指定了显示什么图标，background script有哪些文件，content script又有哪些文件，pop up的页面是什么，等等。

什么是popup，什么是background script，相信大家都清楚了，那什么是content script呢？content script就是我们注入到页面中的脚本，插件允许我们往网页中注入脚本，这是一个多么让人有想象力的功能，其功能之强大无需多解释，总的来说，就是让我们全面干预页面的内容！也许你马上会想到，这可能带来很大的安全隐患，没错，有些恶意插件会窃取你的页面信息，而有些有漏洞的插件则可能让你遭受跨站脚本注入（XSS）的攻击；另一个可能你会想到的问题是：往页面中注入自己的脚本，难道不会跟页面原本的脚本发生冲突吗？能想到这点说明你真的很厉害，如果我们的注入脚本和页面原本的脚本处于同一个运行环境中，确实会发生冲突，所以，Chrome是另外开辟了一个独立的运行空间，供我们的Content Script使用的，Content Script能访问DOM的内容，但却不能访问页面原本的脚本（我是说直接访问不行），反之，页面原本的脚本也不能直接访问Content Script。在图中，浅红色的背景块代表Content Script的运行环境，而浅蓝色的背景块代表页面运行

环境，另外插件的运行环境我用浅绿色表示，注意，这是三个不同的运行环境，调试的时候你会充分体会到它们的不同。

那么，Content Script会在什么时候运行呢？默认情况下，是在网页加载完了和页面脚本执行完了，页面转入空闲的情况下（Document Idle），但这个是可以改变的，详情可参考https://developer.chrome.com/extensions/content_scripts.html，查看其中的“run_at”。

由于处于不同的运行环境中，Content Script和Background Script不能直接互相访问，那它们之间如何通信？通过Message！这个之后的代码中会有。

学习资料

理解了Chrome插件结构之后，我相信你完全有能力开发一款自己的插件了，当然了，你得自己去google一些资料，这里我就分享下我的方法。

首先，官方的资料一定得看看，<https://developer.chrome.com/extensions/index.html>，这个上面的资料得大致浏览一下（不需要全部仔细看），这样你能够明白一些术语，知道如何去寻找你的解决方案。

再则，官方提供的例子，可以看看，<https://developer.chrome.com/extensions/samples.html>，我发现上面的例子有些已经不能用于新版的Chrome了，但没关系，你只要找你想要的就行了，也不用一个个尝试，就根据你的需要，挑选几个你感兴趣的看看即可。

遇到问题，怎么办？当然是用google去查找问题，但这里我最最强烈推荐stackoverflow.com，这简直是解决问题的神器！不多解释了，用过便知。

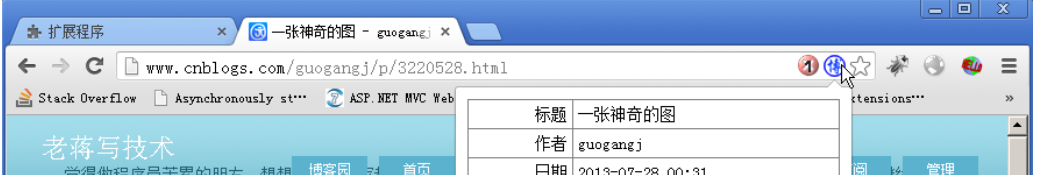
学习过程基本上就是：看个大概，写点代码，调试调试。就可以了。哦，大前提当然是你得有javascript的基础。（你：呵呵，你在逗我吧！）

我的例子

[chrome-plugin-cnblogs-article-information.zip](#)

[chrome-plugin-cnblogs-article-info-server.zip](#)（服务器端，PHP代码）

好，轮到我的例子登场了。它的功能是这样的：当你浏览博客园的时候，它会启动并尝试获取你浏览的文章的信息（标题、作者和日期），再通过往另一个服务器发送请求的方式，记录和获取你第一次访问这篇文章的时间，把这个时间连带文章的信息，显示在popup上。听起来挺无聊的功能，但关键是为了演示嘛，如图：



见符赋性静处白奈的朋友，愿您

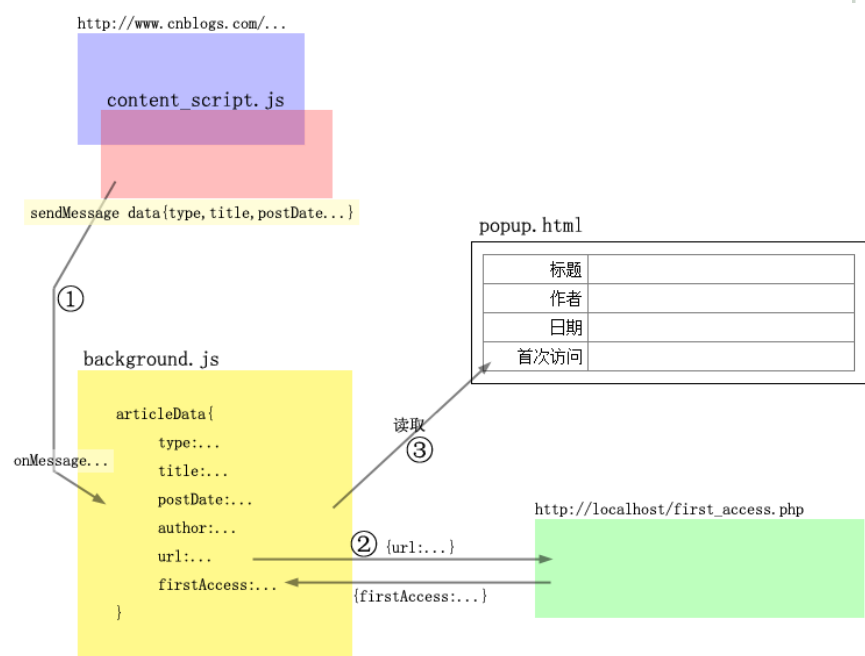
首次访问2013-08-01 16:33

公告

随笔-56 文章-0 评论-77

这个插件一共有9个文件，新出现的文件有两个（其它相信大家很熟悉了），一个是“content_script.js”，这就是前面提到的Content Script，获取和修改页面的内容就靠它了；另一个是“jquery-2.0.0.min.js”，大名鼎鼎的jQuery，我很喜欢用的js库，其理念是“write less, do more”，能帮我减少很多代码，这是目前最新的2.0.0版，这个版本跟以前的1.x.x的最大差别就是不再支持IE6、7和8，我个人是十分赞同这种做法的，微软的旧版浏览器都成了Web技术发展的绊脚石了，而且这次我们用的是Chrome浏览器，果断选择最新版了。

另外还有一个服务器端，为了让问题简化，这次我用了php代码，一个php文件就是整个处理了，没有太多繁杂的配置，简洁，这是php最大的优势。系统结构如图：



抓取网页的内容得依靠content_script.js，然后通过sendMessage/onMessage和background.js交换数据，background.js将url信息通过ajax（XMLHttpRequest）发送给localhost，获取此页面的第一次访问的时间，最后，用户点小图标，popup.html出现，popup.html会读取（代码在popup.js中）background.js中的articleData的数据，把它显示出来。这就是整个过程。

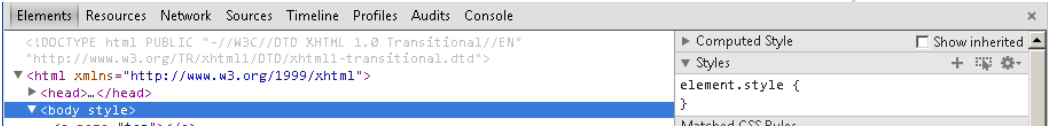
我抓取网页数据的方式并不能确保所有的博客园的文章都能被正常获取，这跟用户使用的博客模板有关系，但我尝试下来大多数文章还是可以抓取的，我不去适应所有的模板了，毕竟这只是个演示的demo。

另外还需说明的一点是我使用了jQuery做XMLHttpRequest，post的内容不是传统的html表单形式，而是json数据，所以在服务器端这边，就不能直接用\$_REQUEST获取，而是通过读取“php://input”的内容获取。顺便谈谈个人对web api的一个看法：“统一”大于“灵活”，这是我的观点，我确定我的接口的格式是json，使用utf-8编码，于是就一直用下去，调用者不用考虑用XML还是html表单还是别的，开发者也不必多考虑，让这成为一种统一的约定，在团队协助和以后的开发中会很省事。

调试

程序开发，必定要涉及到调试，记得我刚开始做WEB开发前，问一些做了好久WEB开发的朋友，你们是怎样做javascript调试的，我发觉大多数人竟然回答：用alert一点点试吧——不是不行，是太原始，太低效了，对吧？其实Chrome直接支持javascript的调试，拥有了Chrome，就相当于拥有了一个强大的javascript调试器了。

Chrome打开开发者工具的方法是<Ctrl>+<Shift>+<I>（Windows版），大致如下：

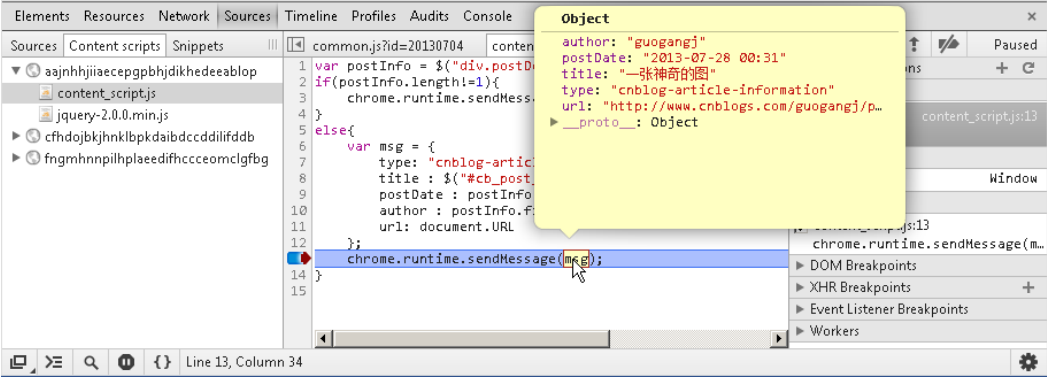




我们这次需要关心的有“Elements”、“Sources”和“Console”这三个标签。Elements是用来做DOM分析的，功能有点类似Firebug，帮助我们分析页面的内容；而Sources，是我们用来调试javascript的；Console则是我们的Log的输出窗口，也是一个调试利器。

调试Content Script

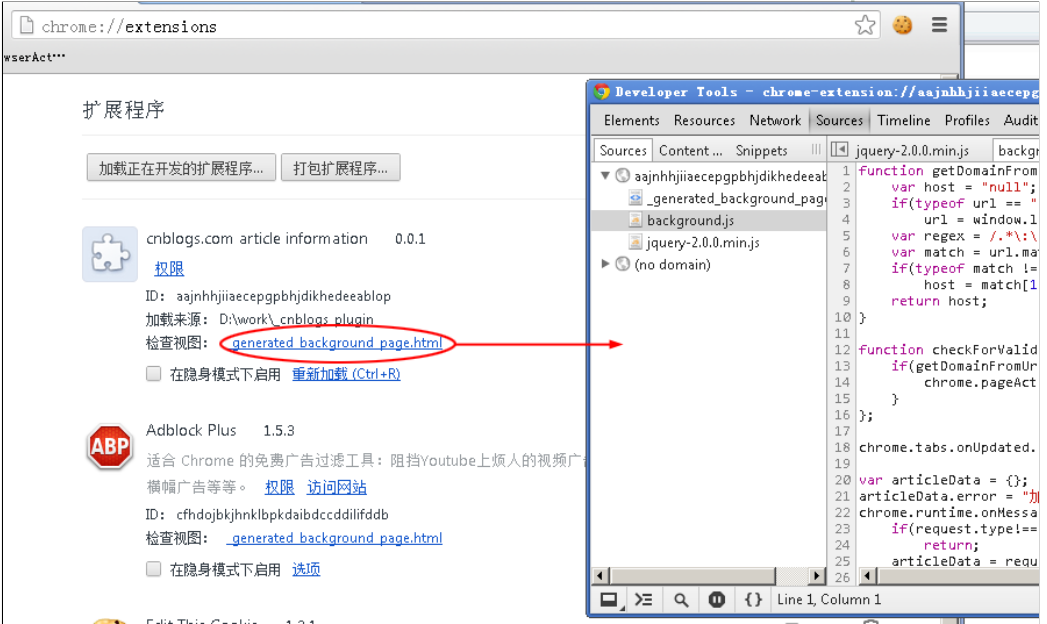
如我提供的这个例子，可在Sources的“Content Scripts”下看到“content_script.js”然后设断点，执行到断点处时，Chrome会挺住，你可以观察到上面的值，如图：



太cool了，请问你还要一点alert吗？

调试Background

由于background和content script并不在同一个运行环境中，因此上面的方法是看不到Background的javascript的。要调试Background，还需要打开插件页，也就是“chrome://extensions”。点对应的插件的“generated background page.html”，就出现了调试窗口，接下来的操作就跟前面的类似了。如图：



至于你看到ID，“aajnhhjia.....”这一长串东西，这是chrome自动安排的一个ID。

调试Popup

虽然Popup和Background是处于同一运行环境中，但在刚才的Background的调试窗口中是看不到Popup的代码的。调试Popup还需要这样：



然后.....就跟前面差不多了。

一些问题

也许有时候你会发觉调试器不是很灵，至少我用下来感觉如此，比如你可能发现断点设不了，或者断点不起作用，或者看不到你自己的javascript文件。我的方法是在插件页中，把对应的插件的“已启用”这个复选框去掉，

再重新勾上，然后再点一下“重新加载(Ctrl+R)”，通常能解决问题。当然了，还有些很古怪的问题，还不好重现，总体的解决思路就是重新载入一下，实在不行的话重启浏览器，或者清除浏览器缓存什么的，再试试看。

在做插件调试的时候我还遇到一个十分郁闷的问题，那就是我的Chrome使用了“Go Agent”，关于Go Agent是用来干嘛的，这个嘛，可以去google一下，我相信绝大多数程序员都会喜欢上它.....可由于使用了这个东西，很可能会导致插件的XMLHttpRequest工作不正常，而且可能你会思索半天也找不到原因，好吧，暂时把Go Agent停用掉，甚至可能你需要重启下Chrome——我的经验。

总结

我还是想说，我觉得Google对我们程序员来说是个很大度的公司，在Chrome这个产品上面就可见一斑。利用Chrome插件技术，我们可以做许多有用的东西，通过本文，相信你已经知道如何去开发一款Chrome插件了，当然了，Chrome插件的功能是很强大的，我用到的仅是冰山一小角。要深入，当然还需要更加充分地利用google和stackoverflow.com了。

标签: HTML/Javascript

好文要顶

关注我

收藏该文

guogangj
关注 - 2
粉丝 - 271
[+加关注](#)

690

« 上一篇: 一张神奇的图
» 下一篇: [防止开发人员获取到敏感数据（SQL Server的数据加密简介）](#)

posted @ 2013-08-04 00:35 guogangj 阅读(64701) 评论(58) 编辑 收藏

[< Prev](#) [1](#) [2](#)

评论列表

- #51楼 2014-11-19 13:18 光之使者

不错，学习了！

支持(0) 反对(0)
- #52楼 2014-11-30 21:54 zhanjindong

赞！正准备写一个简单的插件推送网页到kindle上去。

支持(0) 反对(0)
- #53楼 2015-02-11 07:31 suifengtec

赞

支持(0) 反对(0)
- #54楼 2015-03-13 11:07 qdujunjie

你好，你的文章很好，至少我之前不知道怎么调试popup.js现在知道了。

但是我还是有个问题，就是popup.html和popup.js怎么能取到当前页面的tab的info呢？有办法吗？可以回复我邮件吗：qdujunjie@126.com

支持(0) 反对(0)
- #55楼 2015-03-13 11:07 qdujunjie

另外content_script.js如何操纵当前页面？比如设置当前页面的背景色？

支持(0) 反对(0)
- #56楼 2015-12-13 11:08 乔布儿

赞！

支持(0) 反对(0)
- #57楼 2016-07-30 10:41 java_small_ant

chrome插件是否可以获取开发者工具下面Frames栏目下的所有资源？

支持(0) 反对(0)
- #58楼 2016-12-13 15:29 混世魔王

有个问题，页面一进入或者加载的时候获取到网页内容没问题，但是如果在页面内点击其他博客的链接，content_script不会去抓取新的页面内容。内容就不更新了，请问有没有解决办法？

支持(0) 反对(0)

[< Prev](#) [1](#) [2](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】Google+GitHub联手打造前端工程师课程
- 【推荐】票选最美云上大数据暨大数据技术峰会

**最新IT新闻：**

- 以深度长文出名的Medium，也“改变初心”做快餐式内容了
 - 支付宝医保移动支付再下一城，深圳之后落地镇江
 - 铁汉柔情钢铁侠：为小迷妹改变特斯拉营销策略
 - 京东说今年要开10000家专卖店，苏宁说这个只能当笑话听听
 - 创业公司，请不要害怕员工跳槽！
- » 更多新闻...

**最新知识库文章：**

- 垃圾回收原来是这么回事
 - 「代码家」的学习过程和学习经验分享
 - 写给未来的程序媛
 - 高质量的工程代码为什么难写
 - 循序渐进地代码重构
- » 更多知识库文章...