

my_business的专栏

平凡的IT人，平淡的IT生活，希望这里的点滴，能成为其中不多的亮点

个人资料



????&????

访问: 368910次

积分: 4642

等级: BLOG > 5

排名: 第5128名

原创: 83篇

转载: 3篇

译文: 0篇

评论: 206条

博客专栏



VxWorks深入浅出

文章: 8篇

阅读: 28922



挨踢人生

文章: 11篇

阅读: 58187



设计模式攻略

文章: 27篇

阅读: 44387



Windows8开发专栏

文章: 24篇

阅读: 128596

文章分类

设计模式系列 (27)

挨踢生涯 (12)

技术杂谈 (9)

window8 App开发 (25)

技术人的充电时刻，24个精彩研发案例在等你

程序员2月书讯

社区有奖问答--一起舞动酷炫的iOS动画

Chrome插件开发进阶

标签: chrome插件 action browser chrome function extension

2012-07-03 12:29 25654人阅读 评论(4) 收藏 举报

分类: 技术杂谈 (8)

版权声明：本文为博主原创文章，未经博主允许不得转载。

好奇心使然，体验了下chrome的插件开发（按照英文的意思chrome extension应该称为chrome扩展），发现还真是个不错的东东，在浏览google开发文档的同时也做了一些尝试，脑子不好使，不记录下来没准过一周就淡忘了，索性把自己的一些尝试记录下来跟大家分享一下吧。

本文会围绕chrome插件以下的5个概念来进行说明：

- background page（后台页面）
- browser action（浏览器功能扩展）
- content script（内容脚本）
- page action（页面功能扩展）
- NPAPI plugin（NPAPI插件）



另外，本文并不是chrome插件的入门文档，或许需要你对chrome extension有一定了解才容易理解这些内容，所以如果你完全没接触过chrome extension，建议你先看以下文章：

<http://code.google.com/chrome/extensions/getstarted.html>

1. background page

在manifest.json文件中可以如此定义background：

```
[javascript]
01. {
02.   "background": {
03.     "scripts": ["background.js"]
```

http://blog.csdn.net/my_business/article/details/7711525

1/6

[又爱又恨的C++ \(6\)](#)[VxWorks系列 \(8\)](#)

文章搜索

文章存档

[2014年10月 \(1\)](#)[2014年02月 \(4\)](#)[2013年10月 \(4\)](#)[2013年09月 \(1\)](#)[2013年08月 \(1\)](#)[展开](#)

阅读排行

[Chrome插件开发进阶](#)

(25651)

[老生常谈，正确使用men](#)

(20666)

[【windows8开发】深入i](#)

(15855)

[一个苏州IT人的5年挨踢经](#)

(11430)

[一个苏州IT人的5年挨踢经](#)

(11194)

[使用GDB调试Android N](#)

(10614)

[【VxWorks系列】基于VI](#)

(9161)

[【Windows8开发】Metri](#)

(8915)

[【windows8开发】异步](#)

(8152)

[【windows8开发】开发](#)

(8047)

评论排行

[【windows8开发】深入i](#)

(20)

[一个苏州IT人的5年挨踢经](#)

(17)

[一个苏州IT人的5年挨踢经](#)

(16)

[【Windows8开发】关于'](#)

(11)

[工作7年，从《一个苏州人](#)

(9)

[【windows8开发】C++习](#)

(8)

[一个苏州IT人的5年挨踢经](#)

(7)

[【C++模版之旅】项目中](#)

(7)

[【Windows8开发】异步](#)

(7)

[老生常谈，正确使用men](#)

(6)

推荐文章

[* CSDN日报20170225——《六年程序生涯》](#)[* Android"挂逼"修炼之行---解析公众号文章消息和链接文章消息如何自动打开原理](#)[* 技术晋升的评定与博弈](#)[* 安居客Android项目架构演进](#)[* Apk脱壳圣战之---如何脱掉"梆梆加固"的保护壳](#)

最新评论

[【windows8开发】C++开发Met](#)
[遮住你的眼睛: 请问必须是Windows 8 系统和vs2011吗](#)[老生常谈，正确使用memset](#)
[kellen_f: 不懂为什么第二个里面初始化的值为1就不对了呢???](#)[【C++模版之旅】静态多态\(模版](#)
[bianoa: 如果基类是作为一个函数的形参，那这种方式就会显得](#)

04. }

05. }

当然也可以通过属性"page"替换"scripts"来引入.html文件，不过scripts和page只能两选其一
"background": { "page": "background.html" }

顾名思义，可以理解为背景页面脚本，或者直接解释为后台脚本。background用来处理插件本身的一些逻辑，比如插件加载时需要执行的处理，运行中需要统一维护的数据等等，background只会在插件加载的时候运行一次，你可以在这个过程中让它绑定一些运行中的事件。比如background中可以直接访问chrome.browserAction对象来设置和定义browserAction，如：

[javascript]

01. chrome.browserAction.setIcon({path:"icon.png"});

也可以绑定browserAction的点击事件定义事件响应处理：

[javascript]

01. chrome.browserAction.onClicked.addListener(function(){
02.
03. });

还可以在background中调用chrome.tabs.create()来创建新的tab。

除了跟browserAction进行交互以外，与content script也可以进行协作，这点让我们在content script的部分中再来说明。

2. browser action

browser action可以理解为对浏览器现有菜单功能的扩展。可以在manifest.json文件中对b

[javascript]

01. {
02. "browser_action": {
03. "default_icon": "icon19.png",
04. "default_title": "Google Mail",
05. "default_popup": "popup.html"
06. }
07. }

如此设置下，在插件加载时浏览器菜单栏就会有如上图所示的桔色按钮

(icon19.png)，单击此按钮会弹出popup.html描述的窗口。前面我们已经说明过在background中可以访问browser action，那反过来呢？这个更直接，在browser action中通过background对象可以直接调用background中定义的方法或对象，如下所示，假设在background.js中定义了testBG函数，那么在popup.html中可以这样访问：

[javascript]

01. var bg = chrome.extension.getBackgroundPage();
02. bg.testBG();

在popup中还可以注入代码到web page中，但只限于对dom的访问和修改。

[javascript]

01. chrome.tabs.executeScript(null, {code:"document.body.style.backgroundColor=blue"});

3. content script

很不灵活了

老生常谈，正确使用memset
xiaohun52: good，第一个总结
就是我想要的答案，还赚了下面
这么总结

【C++模版之旅】神奇的Traits
kelvin_yin: // 当T为float时，
Compute方法的参数为float，返
回类型为inttemplate <...

一个苏州IT人的5年挨踢经历-----
Chaos-js: 11年之前这个薪资水
平，同样的技术水平，现在没有
20k，也没啥意思了

【C++模版之旅】神奇的Traits
hh_xj: 赞

【C++模版之旅】项目中一次活
华秋实: AssignData里最好判断
一下data和_data是否相等

一个苏州IT人的5年挨踢经历-----
gaofeisz: 想了解下美名软件在嵌
入式软件这一块怎么样？

一个苏州IT人的5年挨踢经历-----
flyer_2001: 博主牛人，过来膜拜
下。我在博主提到过的2家呆
过。

你肯定不会仅满足于对浏览器菜单功能进行扩展，很多时候也需要跟页面本身进行交互，这是就需要用到内容脚本content script，content script跟页面page共用同一份页面的dom，也就是说content script可以直接去访问或修改当前页面的dom，但是注意了，它们只是共享了dom的访问，js处理本身却是在两个不同的沙盒中运行的，所以并不能互调各自的js代码。

如下处理是在当前页面中插入一个div节点：

```
[javascript]
01. var element = document.body.firstChild;
02. var div = document.createElement("div");
03. document.body.insertBefore(div, element);
```

那么在content script中能跟background交互吗？当然。首先在content script中可以通过content script监听消息：

```
[javascript]
01. chrome.extension.onRequest.addListener(function(request, sender, sendResponse) {});
```

background发送消息：

```
[javascript]
01. chrome.tabs.sendMessage(tab.id, data, function(data) {});
```

反之，
background监听消息：

```
[javascript]
01. chrome.extension.onRequest.addListener(function(request, sender, sendResponse) {});
```

content script侧发送消息：

```
[javascript]
01. chrome.extension.sendMessage(data, function(data) {});
```

content script除了跟background可以交互，跟web page本身也可以有信息交互。一方面content script可以直接访问page的dom，同时还可以通过dom的Event来跟页面进行交互。

content script中绑定事件：

```
[javascript]
01. document.addEventListener("EventName",function() {});
```

web page中如何发送事件给content script呢？如下：

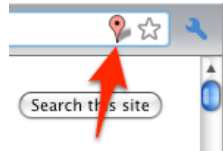
```
[javascript]
01. var ev = document.createEvent('HTMLEvents');
02. ev.initEvent('EventName', false, false);
03. document.dispatchEvent(ev);
```

反之，web page中绑定事件，content script中触发事件的话同上面处理一样，也成立。

4. page action

page action跟browser action有所区别，它会加载和显示在地址栏的里面，所以一般会跟当前访问的URL地址进行交互，比如你可以让浏览器在访问所有带“mail”字符串的URL

时显示page action。具体的显示位置如下图：



如果想使用page action功能，需要在manifest.json中定义如下属性：

```
[javascript]
01. "page_action" : {
02.     "default_icon" : "icon.png",
03.     "default_title" : "Page Action"
04. }
```

我们可以把对page action的设置和处理放在background page中，从而直接在background中通过chrome.pageAction来设置page action，比如如下代码实现了当所访问URL中有mail字符串时就显示page action的icon这样的功能：

```
[javascript]
01. chrome.tabs.onUpdated.addListener(function (tabId, changeInfo, tab) {
02.     if (tab.url.indexOf("mail") > -1) {
03.         chrome.pageAction.show(tabId);
04.     }
05. });
```

5. NPAPI Plugin

JS虽然越来越牛叉了，但毕竟存在局限性，有些处理目前用JS来做毕竟还不现实，比如视频插件，大数据计算等，这类处理大部分还是会通过C++等语言封装的组件来实现的，那么在JS的插件里怎么调用这些C++的组件？因为chrome支持了NPAPI，那这点就不成问题了。（关于什么是NPAPI，一来不是本文介绍的重点，二来我也没详细研究过，有兴趣的自己google吧）

在这里就介绍下在chrome插件中怎么去结合NPAPI插件。首先需要在manifest.json中定义如下关于插件的属性：

```
[javascript]
01. "plugins": [
02.     { "path": "plugin1.dll", "public": true },
03.     { "path": "plugin2.dll" }
04. ]
```

当chrome插件加载后，就可以在html页面中使用所定义的NPAPI插件了，假设插件的mime-type是x-plugin1时我们可以这样去调用NPAPI插件：

```
[html]
01. <embed type="application/x-plugin1" id="pluginId">
02. <script>
03.     var plugin = document.getElementById("pluginId");
04.     var result = plugin.PluginMethod(); //调用plugin中定义的方法
05. </script>
```

好了，就写这么多，谨以此文整理了下自己的思路，也希望能给大家带来一些收获。

上一篇

【设计模式攻略】行为型模式之Visitor模式

下一篇

【设计模式攻略】行为型模式之Mediator模式

我的同类文章

技术杂谈（8）					
• 使用GDB调试Android NDK...	2013-03-29	阅读	10610	• Performance verifying for S...	2013-03-04 阅读 1309
• A simple case of unit testing	2013-02-25	阅读	943	• 构建工具scons让一切变得...	2012-08-01 阅读 5434
• apache代理解决Javascript...	2012-05-11	阅读	5180	• Sed命令详解	2012-04-17 阅读 522
• javascript引擎PK:V8 vs Spi...	2012-03-23	阅读	5210	• 服务器模型分析与验证	2012-03-06 阅读 2270

参考知识库



Hadoop知识库
6168 关注 | 554 收录



Apache Spark知识库
6406 关注 | 401 收录

猜你在找

- ArcGIS for JavaScript
- ComponentOne Studio for WinForms基础课程
- 移动端游戏架构设计
- Learning AngularJS
- 韦东山嵌入式Linux第一期视频
- Chrome插件开发中的消息传递
- Chrome插件开发之二 添砖加瓦
- Chrome插件开发教程
- chrome 插件开发之 --manifest.json
- 谷歌chrome插件开发好文章

查看评论

4楼 qdujunjie 2015-03-16 13:34发表



这样调用
chrome.browserAction.onClicked.addListener(function(tab) {});

会有问题报错：
Uncaught TypeError: Cannot read property 'onClicked' of undefinedbackground.js:200 (anonymous function)

这是怎么回事？

3楼 TouTouch 2014-01-02 14:00发表



非常感谢，解决了我的一些疑惑

2楼 ProvenceXX 2013-07-10 21:11发表



想问一下关于chorme插件的问题，麻烦加一下好友呗？
Q: 252121382

1楼 miameng 2013-04-01 15:14发表



请问怎么能控制content script在页面异步数据都加载完毕再运行呢？？为什么我写的extension都是在页面动态数据（如图片，评论）加载之前运行呢？？真的困扰我很久 希望您看到了指点我一二。。万分感谢

发表评论

用户名: martinkro

评论内容:

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 