**开发自己的Chrome扩展插件**



随着Google Chrome浏览器近几年蓬勃发展，极速的浏览体验，快速的版本更新迭代，稳定的性能，简介的界面，早已取代了微软IE浏览器的霸主地位。成大多数用户的默认浏览器。

对于早期的前端开发者们来讲，经历IE时代的折磨，各种兼容与调试困难，经历过FireFox浏览器崛起Firebug前端开发者的福音。当Chrome来临的时候，从起初的新奇到如今的依赖与陪伴，开发调试工具强大、扩展应用程序类型与数量丰富众多，成为前端开发者的必备利器。

Apps文档全面细致，人人都能根据自己需要开发与扩展Chrome的功能。今天就来简单介绍一下Chrome插件的开发步骤与方法，只需要我们掌握简单的前端开发知识HTML、CSS、JavaScript，就能完成整个插件开发。

**基础知识：**

Chrome扩展包后缀为 .crx，解压缩以后为HTML文件、CSS样式文件、JavaScript脚本文件、图片等静态文件及manifest.json组成。Chrome扩展插件的入口可以在浏览器的工具栏或地址栏中，当用户点击后激活进行下一步的操作或相关信息的展示，也可以干脆没有图标，在后台静默运行。

**开始第一个插件：**

下面根据具体的一个实例来看一下，我们开发一个用来生成当前浏览网址二维码的插件，从而可以用手机扫描访问PC端的网页，免去了用户在手机端输入网址的步骤。

**生成二维码的方法：**

这里使用一个google的提供一个接口，只需要传入对应的网址就能生成该网址的二维码。

接口如下：

http://chart.apis.google.com/chart?cht=qr&chs=200x200&chl=http://www.qq.com

cht qr这里二维码，还有其他类型如p3为三维饼图

chs 生成二维码的尺寸大小为 200x200 px

chl 参数传入要生成二维码的网址

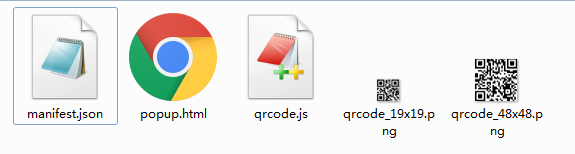
**插件编写：**

需要一下几个文件，及二维码图标图片。

manifest.json 插件配置文件，

popup.html 二维码输出文件，

qrcode.js 生成二维码的方法文件，



**manifest.json**

{

"manifest\_version": 2,

"name": "二维码生成器",

"version": "1.1",

"description": "一个用来把当前浏览页面地址生成二维码的小插件",

"icons": {

"48": "qrcode\_48x48.png"

},

"browser\_action": {

"default\_icon": "qrcode\_19x19.png",

"default\_popup": "popup.html"

},

"permissions":[

"tabs"

]

}

配置文件name定义了扩展文件名称，version定义了版本等。

name 插件名称

version 插件的版本

description 插件描述

icons 插件的在管理界面的图标

default\_icon 插件的图标

default\_popup 点击以后的弹层，用来展示生成二维码的容器

permissions 声明tabs权限

**popup.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<style>

html,body,div{margin:0;padding:0;}

#qrcode{min-width: 100px;min-height: 100px;}

</style>

</head>

<body>

<!-- 用来存放二维码图片的dom -->

<div id="qrcode"></div>

<!-- /用来存放二维码图片的dom -->

<!-- 生成二维码的方法 -->

<script src="qrcode.js"></script>

<!-- /生成二维码的方法 -->

</body>

</html>

html页面的代码省略了很多内容，比如title标签，对chrome扩展插件来说标题在这里是使用不到的，当然你写上也不会报任何的错误。

**qrcode.js**

chrome.tabs.getSelected(null, function (tab) {

if(tab && tab.url){

var html = "<img src='http://chart.apis.google.com/chart?cht=qr&chs=200x200&chl=" + tab.url + "&chld=H|1'>";

document.getElementById("qrcode").innerHTML = html;

}else{

document.getElementById("qrcode").innerHTML = "获取URL出错~";

}

});

这里通过Chrome.tabs获取到当前页面的url，tab对象还包含其他信息如果页面标题等。再通过之前的生成二维码的接口<http://chart.apis.google.com/chart>，来生成二维码的图片地址，回填到popup页面dom里面。

**添加至Chrome**

至此插件功能已编写完成，现在看如果看如何载入到Chrome的插件列表中。

菜单 → 更多工具 → 扩展程序 （或者直接数据地址：chrome://extensions）

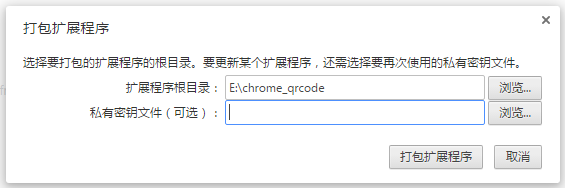
 

选中开发者模式，点击加载扩展程序按钮，选择以上开发文件存放的目录，就可以看到我们载入插件了，如果有错误会在载入文件时有对应的错误提示，如有本地的修改和更新点击重新加载即可。

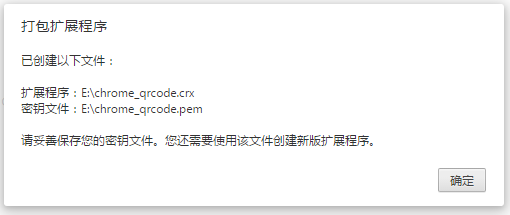


使用效果如上，赶快拿出手机扫描一下吧。

**打包生成**



点击打包扩展程序，选择程序目录。第一次打包程序没有密钥文件，会生成密钥文件 \*.pem。



Chrome\_qrcode.crx文件为打包完成的文件，把文件发送给小伙伴就可以使用你开发的插件了。

**发布到Chrome网上商店：**

Google的网上商店地址，<https://chrome.google.com/webstore/category/apps>，要上传到网上商店需要付费的开发者用户。申请地址：<https://chrome.google.com/webstore/developer/dashboard/>

**经验说明：**

manifest.json文件名是固定，名称不能修改。

popup.html里面的JavaScript文件不能写道页面内，需要使用外链的方式写到单独文件里面qrcodes.js文件。

附录：

Chrome应用商店的地址：<https://chrome.google.com/webstore/category/apps>

开发者文档的地址：<https://developer.chrome.com/extensions>