

简单了解HTML5中的Web Notification桌面通知

这篇文章发布于 2016年07月6日, 星期三, 00:43, 归类于 [JS相关](#)。阅读 96925 次, 今日 31 次 [83 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=5474>
本文全文转载需购买版权(750¥), 摘要引流则免费, 具体参见[这里](#)

一、传统的通知实现

通知可以说是web中比较常见且重要的功能, 私信、在线提问、或者一些在线即时通讯工具我们总是希望第一时间知道对方有了新的反馈, 这个时候, 就需要页面给予即使的通知。

在以前, 我们的通知实现主要是通过闪烁页面的标题内容来实现, 实现原理其实很简单, 就是定时器不断修改 `document.title` 的值。

例如:

```
setInterval(function() {
    var title = document.title;
    if (isShine == true) {
        if (/新/.test(title) == false) {
            document.title = '【你有新消息】';
        } else {
            document.title = '【          】';
        }
    } else {
        document.title = titleInit;
    }
}, 500);
```

结果标题就会这样:



眼见为实, 您可以狠狠地点击这里: [标题栏闪动提示JS演示demo](#)

窗体失焦的时候, 标题就会闪。

这里有一个小的知识点, 就是浏览器窗体获得焦点和失去焦点, Chrome和FireFox浏览器是 `window` 的 `onfocus`, `onblur` 方法; 而IE浏览器则是 `document` 的 `onfocusin`, `onfocusout` 方法, 因此有:

```
window.onfocus = function() { };
window.onblur = function() { };

// for IE
document.onfocusin = function() { };
document.onfocusout = function() { };
```

如果复杂场景, 请使用 `addEventListener` 和 `attchEvent` 进行事件绑定。

然而, 这种提示有个致命的缺陷, 就是用户的浏览器要一直是张开的。比方说用户浏览器最小化, 标题

就看不见，自然就无法及时get到有新消息的信息。

好了，新技术的出现不会是无缘无故的，总是为解决某一类问题或需求出现的。Web Notification就可以很好地解决上面的痛点。

二、HTML5 Web Notification桌面通知特点

HTML5 Web Notification通知是属于桌面性质的通知，有点类似于显示器右下角蹦出的QQ弹框，杀毒提示之类的，跟浏览器是脱离的，消息是置顶的。

比方说，你私信女神说“同事放我鸽子，独立日的票多了一张，浪费了可惜，要不送你？”，然后浏览器爽快地关闭……的话，你是收不到消息的（zxx: 此处有更新），只能最小化，或非当前窗口，然后你就可以安安心心地看《余罪》，没错，就安安心心地看。跟你讲，只要女神一回复，立马，桌面右下角就会出现：“女神说：好啊！”的通知提示，你就可以秒回！女神看到这秒回的速度，心里想的一定是：“这傻小子，一定是一直盯着屏幕等我消息……”剧情就完全不一样了！



更新于2016-07-12

浏览器关闭是否可以接收到Web Notification通知，如果说“收不到”，肯定是不准确的，因为我自己时不时可以收到facebook的通知。但是，恕我愚钝，一直没想明白原理。难道Facebook和Chrome酱有合作，如果是这样那就是特例；如果不是，那通知是哪里发出的呢？

求达人解惑！

自己更新于2018-08-04

Facebook的通知是应该是使用的FCM，也就是Firebase云信息传递，是一种跨平台消息传递解决方案。

使用Service Worker注册对象，调这个对象的pushManager.subscribe的方法让浏览器弹一个框，询问用户是否允许接受消息通知。

如果点击允许的话，浏览器就会向FCM请求生成一个subscription（订阅）的标志信息，然后把这个subscription发给服务端存起来，用来发Push给当前用户。服务端使用这个subscription的信息调web push提供的API向FCM发送消息，FCM再下发给对应的浏览器。然后浏览器会触发Service Worker的push事件，让Service Worker调showNotification显示这个push的内容。操作系统就会显示这个Push。

然而，Firebase国内被墙，默认情况下，无法推送消息，大大影响了此功能的收益。要想有用，需要一个境外服务器中转，或用户自己梯子已经架起来了。

兼容性

IE14以及其他桌面浏览器都支持Web Notification，目前移动端的支持情况并不好，Android部分支持，iOS Safari浏览器全军覆没(至9.3版本)：

Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		29						
		45					4.3	
		49					4.4	
	46	50			9.2		4.4.4	
13	47	51	9.1	38	9.3	8	50	50
14	48	52	10	39				
	49	53	TP	40				
	50	54						

移动端要是支持的话，我猜，可能会跟原生app的提示效果类似，直接出现在屏幕上，感觉到时候应该蛮吊！

三、HTML5 Web Notification语法

`window.Notification`

如果浏览器支持Web Notification，不考虑私有前缀，则window.Notification就会是一个有很多静态属性和实例方法的函数。基本上，Web Notification所有的语法都是围绕Notification这个函数来进行的。

显然，通知这种事情是有可能扰民的，因此，必须经过用户同意才行。因此：

1. Notification.requestPermission()

这是一个静态方法，作用就是让浏览器出现是否允许通知的提示，window系统Chrome浏览器目前的UI效果是这样的：

语法目前有新旧两种，下面这个是最近规范上更新的基于promise的语法：

```
Notification.requestPermission().then(function(permission) { ... });
```

下面这个是基于简单的回调：

```
Notification.requestPermission(callback);
```

其中 `callback` 是可选参数，根据MDN的说法，Gecko 46开始舍弃了这种语法，但是，我自己使用Firefox 47测试，跑得很正常（有part4的截图为证）。难道Firefox的版本号不等同于Gecko的版本号？

无论是then中的还是直接callback函数的参数都是一样的，表示当前是否允许。只会是 `granted`，`denied`，或 `default`。

其中 `granted` 表示用户允许通知，`denied` 表示用户嫌弃你，`default` 表示用户目前还没有管你。

```
Notification.requestPermission().then(function(result) {
    // result可能是是granted, denied, 或default.
});
```

2. Notification.permission^[只读]

这是一个静态属性。表示是否允许通知，值就是上面的 `granted`，`denied`，或 `default`。

默认情况下，`Notification.permission` 的值是 `'default'`：

因此，`Notification.requestPermission()` 的回调方法中，可以不使用 `result` 参数，直接使用 `Notification.permission` 获取当前的通知状态。

现在，用户允许出通知了，我们也知道可以出通知了，下面剩下的就是显示通知了。

3. new Notification(title, options)

通过new构造，显示通知。其中 `title` 是必须参数，表示通知小框框的标题内容，`options` 是可选参数，对象，支持的参数以及释义见下表：

属性名	
	释义
dir	默认值是 <code>auto</code> , 可以是 <code>ltr</code> 或 <code>rtl</code> , 有点类似 <code>direction</code> 属性。表示提示主体内容的水平书写顺序。
lang	提示的语言。没看出来有什么用。大家可以忽略之~
body	提示主体内容。字符串。会在标题的下面显示。比方说上面的“好啊！(害羞.gif)”。
tag	字符串。标记当前通知的标签。
icon	字符串。通知面板左侧那个图标地址。
data	任意类型和通知相关联的数据。
vibrate	通知显示时候，设备震动硬件需要的振动模式。所谓振动模式，指的是一个描述交替时间的数组，分别表示振动和不振动的毫秒数，一直交替下去。例如[200, 100, 200]表示设备振动200毫秒，然后停止100毫秒，再振动200毫秒。
renotify	布尔值。新通知出现的时候是否替换之前的。如果设为 <code>true</code> , 则表示替换，表示当前标记的通知只会出现一个。注意都在这里“当前标记”没？没错， <code>true</code> 参数要想其作用，必须 <code>tag</code> 需要设置属性值。然后，通知就会像这样覆盖： <div><div></div></div> 而不会是默认的叠高楼： <div><div></div></div>
silent	布尔值。通知出现的时候，是否有声音。默认 <code>false</code> , 表示无声。
sound	字符串。音频地址。表示通知出现要播放的声音资源。
noscreen	布尔值。是否不再屏幕上显示通知信息。默认 <code>false</code> , 表示要在屏幕上显示通知内容。 <div><div></div></div>
sticky	布尔值。是否通知具有粘性，这样用户不太容易清除通知。默认 <code>false</code> , 表示没有粘性。根据我自己的猜测，应该和 <code>position</code> 的 <code>sticky</code> 属性值类似。

4. Notification.close()

通知显示了，如何关闭呢？可以通过调用 `Notification.close()` 实例方法，实际上，通知如果你放着不管，一段时间后就会自动隐藏，具体多久不详，我估摸着5秒有的。

5. 事件句柄

Notification.onclick

点击通知，然后.....

Notification.onerror

通知显示异常，然后。例如，明明 `Notification.permission` 是 `default`，你还让我显示。

下面这些呢有必要独立出来，虽然现在是支持挺好的，但是，由于目前规范并没有把它们列入其中，所以，未来有可能浏览器就不支持了。

Notification.onclose

通知关闭了，然后..... 无论是用户手动关闭，还是直接 `Notification.close()` 关闭都会触发该事件。

Notification.onshow

通知显示的时候，该干嘛干嘛~~

6. 其他属性值

除了 `Notification.permission` 外，`Notification` 还有很多其他只读属性值，但是，基本上和上面的 `options` 参数一致，返回的值也是 `options` 和默认值的合并值（如果浏览器支持的话）。

Notification.title^[只读]

Notification.dir^[只读]

Notification.lang^[只读]

Notification.body^[只读]

Notification.tag^[只读]

Notification.icon^[只读]

Notification.data^[只读]

Notification.silent^[只读]

Notification.title^[只读]

Notification.timestamp^[只读]

通知创建或者可以使用的时间。

Notification.noscreen^[只读]

Notification.renotify^[只读]

Notification.sound^[只读]

Notification.sticky^[只读]

Notification.vibrate^[只读]

四、HTML5 Web Notification实例演示

知道了作用和API，下面就可以来个实例演示下，看看究竟真面目如何。

您可以狠狠地点击这里：[浏览器通知Web Notifications实例demo](#)

点击demo页面按钮：

如果你是第一次，会有如下提示：

通过后，就会出现妹子相关的通知信息了，例如，FireFox浏览器下（使用的是callback回调实现）：

此时，你浏览器最小化，或者页面刷新，该通知都是纹风不动的。点击通知，可以和页面进行交互，例如，本demo显示了一段文字：

核心代码如下（显示和点击）：

```
if (Notification.permission == "granted") {
    var notification = new Notification("Hi, 帅哥: ", {
        body: '可以加你为好友吗? ',
        icon: 'mm1.jpg'
    });

    notification.onclick = function() {
        text.innerHTML = '张小姐已于' + new Date().toString().split(' ')[0] + '加你为好友! ';
        notification.close();
    };
}
```

完整测试代码参见demo。

至此，一个可以应用于实际开发的Web Notification demo就跃然纸上了。好像facebook已经在使用Web Notification了。

更新于**2017-07-14**

最新的Chrome的Notification要想有效果需要 `https` 协议才行，因此，上面的demo可能就没有效果。如果这样，可以[点击这里](#)感受效果。

点击按钮后的提示效果类似：

五、结束语

细心的小伙伴可能发现文章的转载协议变了，以前都是转载请注明出处，但是，现在，不允许随便转载了。因为有些网站的做法我实在忍无可忍，如果转载一些经典的文章，我还可以理解，但是，几乎把整站都搬过去，那就不是知识传播了，而是劳动果实的窃取，为己谋利的商业做法了。

比方说本文，设计demo，验证API细节，截图，制作Gif以及文章书写，那可是连续3天1点多睡觉熬出来的，累积花费十几个小时。你说，被某些人大手一挥，分分钟弄过去，不放原文地址，或者地址做得像彩蛋一样，难以轻易发现。那对原作者是没有一点点的尊重啊！

所以，从本篇文章开始，也就是2016年7月开始，使用新的转载协议，主要是用来约束那些随便盗取他人

劳动知识成功的商业站点。对于，一心向善，致力于推动行业发展的小伙伴们，可以联系我进行当面授权。

尊重知识产权，共创和谐社会！



《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。



« [PNG格式小图标的CSS任意颜色赋色技术](#)

[CSS3 appearance大全鉴赏以及是否影响box-sizing](#) »

猜你喜欢

- [ES6 JavaScript Promise的感性认知](#)
- [了解JS中的全局对象window.self和全局作用域self](#)
- [借助Service Worker和localStorage缓存及离线开发](#)
- [借助HTML5 details,summary无JS实现各种交互效果](#)
- [页面可用性之outline轮廓外框的一些研究](#)
- [js下拉菜单实现与可访问性问题的一些思考](#)
- [jQuery powerFloat万能浮动层下拉层插件](#)
- [span与a元素的键盘聚焦性以及键盘点击性研究](#)
- [小卖弄：纯CSS实现的outline切换transition动画效果](#)
- [CSS :focus伪类JS focus事件提高网站键盘可访问性](#)
- [div模拟textarea文本域轻松实现高度自适应](#)

分享到: 1

标签: [focus](#), [HTML5](#), [Notification](#), [Promise](#)

发表评论 (目前83条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

« 先前评论

1. 95e2说道:

2019年01月6日 04:40

请问通知图标是否可以使用svg? 如果可以, 是否可以写成data:image/svg+xml.....的格式?

[回复](#)



张 鑫旭说道:

2019年01月6日 18:20

你可以试试。

[回复](#)



2. mfk说道:

2019年01月4日 11:34

<https://zhangxinxu.github.io/https-demo/notification/basic.html>

我记得以前可以, 今天又试了下, 怎么不行了, 点了没任何反应。我浏览器Chrome 版本 70.0.3538.77

[回复](#)



3. icefour说道:

2018年11月2日 10:37

mac版chromium内核61的notification 弹出提示, 会在左边带上应用图标, 有没有办法去掉这个图标呢? (设置了icon之后就会同时显示2个图标)

[回复](#)



4. charles说道:

2018年10月19日 18:09

很棒!

[回复](#)



5. zmqf说道:

2018年07月10日 17:34

有没有设置唤醒浏览器的属性?

[回复](#)



墨明棋妙说道:

2018年09月18日 16:12

google、火狐、360、qq浏览器等浏览器都在设置里找通知, 然后设置允许所有网站显示桌面通知
设置链接:

360

se://settings/content

qq浏览器

qqbrowser://settings/search#%E9%80%9A%E7%9F%A5

火狐

about:preferences#searchResults

谷歌

chrome://settings/content

[回复](#)



6. 随风说道:
2018年07月6日 11:25

您好, 请问下 我的网站协议是http的, 在Chrome上面不兼容, 但是在Firefox上面是可以的, 请问下有什么解决的方法么, 更改https可能公司层面不会允许, 求指教

回复

7. 啊手动阀说道:
2018年06月6日 17:47

您好, 怎样让通知框不自动关闭?

回复

veveue说道:
2018年07月2日 11:33

requireInteraction: 表示通知应保持有效, 直到用户点击或关闭它, 而不是自动关闭。默认值为false。
<https://developer.mozilla.org/zh-CN/docs/Web/API/notification/Notification>

回复

8. York说道:
2018年05月15日 13:28

Notification API 必须是https才可以?

回复

9. dandelion说道:
2018年01月18日 18:18

非https的站点, chrome 的 Notification.permission 为 denied, 且无法更改, 只能从http升级为https协议吗, 不知道是否有什么办法解决呢

回复

Daisy说道:
2018年02月24日 11:39

同问, 不知道这个问题您有没有解决

回复

Steven说道:
2018年02月26日 17:37

同问啊, 有人解决吗?

回复

yvanzhu说道:
2018年04月23日 15:49

申请个https不麻烦的

helo说道:
2019年01月11日 19:56

如果只是内网使用的话, 其实可用localhost 或者是127.0.0.1访问, 是可行的

回复

York说道:
2018年05月15日 13:29

同问, 不知道这个问题您有没有解决

[回复](#)

Fw说道:

2019年01月7日 11:07

我本地起nginx测试http的也正常能用啊

[回复](#)



10. 移动端notification测试说道:

2017年12月25日 15:45

在移动端上面测试，接口函数和属性都是存在的，alert（Notification.permission）出来是denied，但是调用Notification.requestPermission()却没有弹出授权提示，两种写法都试了，安卓UC V11.5和QQ V 8.0都是没反应，也给了浏览器通知栏权限还是不行，本地localhost和https环境都不行，张哥，何解？是兼容问题吗？

[回复](#)



李小龙说道:

2018年02月12日 09:37

让用户使用移动版firefox

[回复](#)



PeopleSea 说道:

2018年02月25日 08:15

火狐也不行

[回复](#)



« 先前评论

最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的？](#)
- » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) ⁽¹⁹⁰⁾
- » [未来必热：SVG Sprite技术介绍](#) ⁽¹¹⁹⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹⁵⁾
- » [HTML5终极备忘大全（图片版+文字版）](#) ⁽⁹³⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸⁶⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸²⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁹⁾
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) ⁽⁷⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁷⁶⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷⁶⁾

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) ⁽⁷⁶⁾
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) ⁽⁶⁴⁾
- » [看，for..in和for..of在那里吵架！](#) ⁽⁶⁰⁾
- » [是时候好好安利下LuLu UI框架了！](#) ⁽⁴⁷⁾
- » [原来浏览器原生支持JS Base64编码解码](#) ⁽³⁵⁾
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) ⁽³³⁾
- » [炫酷H5中序列图片视频化播放的高性能实现](#) ⁽³¹⁾
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) ⁽³⁰⁾
- » [windows系统下批量删除OS X系统.DS_Store文件](#) ⁽²⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽²⁶⁾

猜你喜欢

- [ES6 JavaScript Promise的感性认知](#)
- [了解JS中的全局对象window.self和全局作用域self](#)
- [借助Service Worker和cacheStorage缓存及离线开发](#)
- [借助HTML5 details,summary无JS实现各种交互效果](#)
- [页面可用性之outline轮廓外框的一些研究](#)
- [js下拉菜单实现与可访问性问题的一些思考](#)
- [jQuery powerFloat万能浮动层下拉层插件](#)
- [span与a元素的键盘聚焦性以及键盘点击性研究](#)
- [小卖弄：纯CSS实现的outline切换transition动画效果](#)
- [CSS :focus伪类JS focus事件提高网站键盘可访问性](#)
- [div模拟textarea文本域轻松实现高度自适应](#)