

HTML5 postMessage iframe跨域web通信简介

这篇文章发布于 2012年02月29日, 星期三, 12:41, 归类于 [Web综合](#)。阅读 62886 次, 今日 22 次 [10 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=2229>

一、简单概要

web通信（洋名：web messaging）是一种文档中独立的浏览上下文间的DOM不会被恶意的跨域脚本暴露数据分享方式。

得得得，术语啊什么的，比看到凤姐还头疼。有必要把上面一句话拆开讲：

- web通信是一种数据分享方式（有屁话之嫌）；
- 通信的主体是“浏览上下文”（这是纳尼？）；
- 哦，“浏览上下文”呢是“一个将 [Document](#) 对象呈现给用户的环境”，你可以近似理解为平常我们看到的某个页面所处的环境；
- web通信不会有DOM被恶意暴露的危险；
- 目前应用比较多的就是iframe之间的通信（这是我自个儿额外加的😄）。

当我们谈论web通信的时候，实际上谈论的是两个略有不同的系统：跨文档通信(**cross-document messaging**)和通道通信(**channel messaging**)。跨文档通信就是我们国内更为熟知的HTML5 `window.postMessage()` 应用的那种通信；通道通信也被称为“`MessageChannel`”。伴随着server-sent事件以及[web sockets](#), 跨文档通信和通道通信成为HTML5 通信接口“套件”中有一部分。

浏览器支持(至2012-02-27)

web通信已经被Opera, Chrome, Safari支持，尽管Safari≤5.1.2的版本有bug. IE8 部分支持跨文档通信：只能和iframe通信，不支持新窗口通信。IE10 将支持通道通信。Firefox目前支持跨文档信息，但是并不支持通道通信。

浏览器支持(至2013-04-24)

Firefox 20目前仍然不支持通道通信，IE10确认支持通道通信。

二、通信事件

这里要先介绍下 `message` 事件对象。为何？


因为，无论是跨文档通信(**cross-document messaging**)、通道通信(**channel messaging**)、服务器发送事件(**server-sent events**)或是网络套接字(**web sockets**)都要执行 `message` 事件。因此理解之时很有帮助滴。就像是你无论娶要林志姐还是“风又变人”姐，都要过你妈这关，因此，了解你妈的偏好就很有帮助啦！

`Message`事件的定义可参见[这里](#)（此链接建议不用点，英文+术语→大头儿子），该事件包含5个只读属性：

data
包含任意字符串数据，由原始脚本发送
origin
一个字符串，包含原始文档的方案、域名以及端口(如： <code>http://domain.example:80</code>)

lastEventId
一个字符串，包含了当前的消息事件的唯一标识符。
source
原始文件的窗口的引用。更确切地说，它是一个WindowProxy对象。
ports
一个数组，包含任何MessagePort对象发送消息。

在跨文档通信和通道通信中，`lastEventId` 的值一般是个空字符串；`lastEventId` 应用在服务器端发送事件上。发送信息中如果没有ports, 则 `ports` 属性值就是个长度为0的数组。

`MessageEvent` 继承DOM事件接口，且属性共享。然而，通信事件并没有冒泡，不能取消，也没有默认行为。

三、跨文档通信

跨文档通信的使用跟我们平时实际生活中的邮件接收等类似。发送→接收。

文字化的描述不利于理解。所以，先从一个例子开始吧。

例子很简单，页面上两个iframe框架，左侧的可以输入信息，点击确认按钮后，输入的信息可以在右侧的iframe中显示。

您可以狠狠地点击这里：[两个iframe之间的跨文档通信demo](#)

例如，我们在左侧输入“白静被害”，点击按钮后，右侧就有对应显示，参见下面截图：



上面已经提过，跨文档通信被IE8+浏览器支持，因此，本demo在IE8浏览器下也是有效果滴：



OK，上面例子简单很，发送以及接收！

发送核心JS代码如下：

```
window.parent.frames[1].postMessage(message, '*');
```

说明：

1. `window.parent.frames[1]` 指的就是demo正页中的第二个iframe. 然后使用 `postMessage` 方法发送数据。
2. `postMessage` 方法支持两个参数，具体参考下表：

message
发送的数据
targetOrigin
发送数据的来源。

其实还有个可选的第三参数 `transfer`，不过用在通道通信中，这个放在后面讲。

`postMessage` 方法中的 `message` 参数不仅仅可以是字符串，结构对象、数据对象（如：`File` 和 `ArrayBuffer`）或是数组都是可以的。很厉害吧，不过遗憾的是，IE8/IE9/FireFox3.6及其以下版本只支持字符串数据。

`targetOrigin` 参数指接收文档的来源。除非接收信息浏览上下文来源于提供的 `targetOrigin` 中的一个匹配，否则浏览器是不会发送消息的。当然，您可以像上面的demo一样，绕开这类限制，直接使用 `"*"` 通配符，但显然，这是把主公暴露在反贼的刀口之下啊（不安全的信息泄露）。本demo为了便于理解，去除不必要的干扰，所以才使用了 `"*"` 通配符，您在实际使用的时候务必指定目标来源。

3. 您还可以通过使用 `"/"` 来限制信息只能同源发送。不过，在文字落成的这个时候，只有Opera浏览器支持之。
4. 还有一个需要注意的就是偶们指定来源的时候，后面不要带上斜杠。也就是要使用：

```
window.postMessage('发送信息.','http://example.zhangxinxu.com');
```

而不是：

```
window.postMessage('发送信息.','http://example.zhangxinxu.com/');
```

跨文档的浏览器窗体通信

上面的demo实在一个文档页面内完成的。实际上跨文档通信也可以在不同窗体之间完成。

您可以狠狠地点击这里：[不同窗体间的跨文档通信demo](#)

在现代浏览器下，点击男生或女生按钮（例如点击男生按钮），打开的新页面子啊2秒后显示了对应的通信信息。如下截图展示（截自FireFox 10）：



这个例子不仅仅展示了窗口见的通信，还有一点很重要的，就是展示了特定时间控制窗体信息的发送。

demo主页面有个名为 `message` 的全局变量，当点击男生按钮的时候，这个变量值变成“我是男生，帅气的男生！”；点击女生按钮则是“我是女生，漂亮的女生！”。消息的发送来自收到打开页面的 `'ready'` 通信信息。

代码稍稍长了点，这里就不展示了。主demo页面JS代码都有，且高亮显示了，有中文注释，相信不难理解的。

本来还想举个实际应用的例子，例如iframe高度自适应问题。不过，一是懒；二是担心躲得深不易发现。这里赞先免了。有精力的时候可以专门说说。

本文一开始就提过，IE8不支持窗体通信，但是，细心的你可能发现IE9浏览器下也没有效果。到不是别的，而是貌似IE9还没有提供 `e.currentTarget.openner` 接口，使得demo瞌睡去了。

四、通道通信

消息通道提供了一个直接，双向浏览上下文之间的通信手段。跟跨文档通信一样，DOM不直接暴露。取而代之，管道每端为端口，数据从一个端口发送，另一个变成输入（反之亦然）。

消息通道是有用的，特别是跨多个起源的沟通。请考虑以下情形：人人网上(<http://renren.com>)嵌入了一个第三方的游戏页面（通过iframe的形式，如“人人餐厅”），同时，这个第三方的游戏页面(<http://game.com>)又需要从另外一个通讯录网站(<http://address.com>)获取用户的通讯信息。咋办？

也就是说通讯录站点要发送信息给游戏站点，根据跨文档通信，我们让父页面作为代理（也就是这里的人人网页面）（类似第一个demo）。然而，这种做法意味着通讯录站点需要有和人人网页面一样的信任级别。人人网这个社交站点需要信任每一个请求，或者为我们过滤（应该指：一个一个指定）。

但是，使用渠道通信，通讯录站点(<http://address.com>)和游戏站点(<http://game.com>)可以直接沟通。

MessageChannel和MessagePort对象

当我们创建了一个 `MessageChannel` 对象，我们实际上创造了两个相互关联的端口。一个端口保持开放，为发送端。另外一个被转发到其他浏览上下文。

每一个端口就是一个 `MessagePort` 对象，包含3个可用方法：

<code>postMessage()</code>
通过通道发送消息
<code>start()</code>
开始在端口上分派接受的信息
<code>close()</code>
关闭端口

`MessagePort` 对象还有 `onmessage` 事件属性，可被用来定义事件句柄而不是事件监听。

实例

上面过于术语的东西我自己都看不明白，还是实例好说话。

您可用狠狠地点击这里：[通道通信应用demo](#)

文章以开始的兼容性部分已经提过，FireFox浏览器目前还不支持通信通道，因此，上demo需要在Opera或是Chrome浏览器下打开。



demo页面操作与第一个demo类似，左侧输入信息，点击按钮提交。

例如在Opera浏览器下：



当然，并不是每一次都是OK的，有时候会弹出“端口不可用的提示”，原因不详，估计可能与端口被占用有关，遇到这种情况，稍后尝试即可。

简析

上面的demo动用了三个页面：主页面和两个iframe页面。下面说说每个页面都做了些什么：

首先是第一个iframe页面（demo左侧有表单提交的那个）。其任务有两个：一是告诉主页面，我加载好了；二是扩大并确定端口，表单提交时用做发送用。

```
// 告诉主页面，我加载好了
window.parent.postMessage('发送页加载完毕', 'http://www.zhangxinxu.com');
```

```
// 扩大并确定端口
port = evt.ports[0];

// 端口发送数据
port.postMessage(message);
```

然后是第二个iframe页面（demo右侧那个，用来显示信息）。其任务有三个，一是创建 `MessageChannel` 通道对象；二是告诉主页面，我加载好了，并把端口传过去；三是显示发送信息。

```
// 创建一个新的 MessageChannel 对象
var mc = new MessageChannel();

// 给父级发送一个端口
window.parent.postMessage('显示页加载完毕', 'http://www.zhangxinxu.com', [mc.port1]);

// 显示发送的信息
mc.port2.addEventListener('message', messageHandle, false);
mc.port2.start();
```

最后是主页面。其任务很简单就一个：告诉第一个iframe页面，端口已经打开了（第一个iframe就可以确

定跟第二个iframe通信的端口了）。

```
// 将端口告诉其他文档
window.frames[0].postMessage('端口打开','http://www.zhangxinxu.com', evt.ports);
```

于是，三者关联配合，就搞定啦！

五、其他资源

slideshare上有个web通信的文档，这里展示下，有兴趣的童鞋可以快速浏览下。

HTML5 Web Messaging

更多来自Mike Taylor的演示文档

六、参考文章

- An Introduction to HTML5 web messaging
- HTML5/browsers（中文）
- 利用HTML5的window.postMessage实现跨域通信

《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](#)。



« 翻译：即将到来的CSS私有前缀灾难

小tip: 使用CSS(Unicode字符)让inline水平元素换行 »

猜你喜欢

- 小tip: iframe高度动态自适应
- iframe和HTML5 blob实现JS,CSS,HTML直接当前页预览

- reflection.js-实现图片投影倒影效果js插件
- HTML CSS列表元素ul,ol,dl的研究与应用
- 让所有浏览器支持HTML5 video视频标签
- CSS3&HTML5各浏览器支持情况一览表
- 翻译 - 逐渐消失的Flash网站
- HTML5 css reset
- 翻译-你必须知道的28个HTML5特征、窍门和技术
- 时鲜技术：图像的像素化处理
- 揭秘视频网站video视频倍速播放的实现

分享到： 1

标签： HTML5, iframe, MessageChannel, MessagePort, postMessage, web通信, 浏览上下文, 跨文档通信, 通道通信

发表评论（目前10条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<input type="text"/>	
<input type="button" value="提交评论"/>	

1. vino24说道：

2016年07月11日 09:34

你好，我想请教下webview之间可以用postMessage传递消息，最近在做这样一个需求，在浏览器中可以成功的进行通信，但是放到APP里面的时候发送方消息发出去了，但是接收页面怎么也收不到消息。

具体的代码见：<http://stackoverflow.com/questions/38258648/cant-receive-message-send-by-postmessage-method-on-webview>

代码的调了很多次还是没成功，有可能是容器的问题。如果是容器问题有什么解决办法吗？

[回复](#)



2. zhoucunt说道：

2016年05月16日 15:16

已阅

[回复](#)



3. otarim说道：

2013年10月30日 16:38

信道看起来相当不错~

[回复](#)



4.

xyanyue说道:

2013年03月29日 17:16

很屌

回复


5.

yuhan说道:

2013年03月26日 17:36

按说的写的，接收页面e.currentTarget.opener这个成null了，这是什么问题

回复


6.

airoschou说道:

2013年03月20日 09:24

我还是不懂这个能做什么?

回复


7.

stefan说道:

2012年04月8日 13:15

最后一个通道通信太受用了，的确是一个解决多方信任关系的好办法！

回复


8.

短发哥说道:

2012年03月12日 16:02

这里面讲的方法是js的吗，postMessage搜不到啊? 是怎么与html5联系起来的

回复


9.

lafa说道:

2012年03月6日 19:59

冒泡。。。配图太生动了。。。

回复


10.

猫猫儿说道:

2012年02月29日 14:39

看了一遍，有点晕。。前景很好。但是支持性现在很纠结

回复



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](#) ⁽¹⁹³⁾
- » [未来必热：SVG Sprite技术介绍](#) ⁽¹²⁰⁾
- » [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](#) ⁽⁸⁰⁾
- » [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 scrollToView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

- [小tip：iframe高度动态自适应](#)
- [iframe和HTML5 blob实现JS,CSS,HTML当前页预览](#)
- [reflection.js-实现图片投影倒影效果js插件](#)
- [HTML CSS列表元素ul,ol,dl的研究与应用](#)
- [让所有浏览器支持HTML5 video视频标签](#)
- [CSS3&HTML5各浏览器支持情况一览表](#)
- [翻译 - 逐渐消失的Flash网站](#)
- [HTML5 css reset](#)
- [翻译-你必须知道的28个HTML5特征、窍门和技术](#)
- [时鲜技术：图像的像素化处理](#)
- [揭秘视频网站video视频倍速播放的实现](#)