

# HTML5: Communication API

1. [浏览器支持](#)
2. [API 参考](#)
3. [DEMO](#)
4. [注意事项](#)
5. [兼容解决方案](#)
6. [参考文献](#)

## 一、浏览器支持

Platform	MAC						
Browser	<a href="#">OPERA</a>	<a href="#">FIREFOX</a>	<a href="#">SAFARI</a>	<a href="#">CHROME</a>	<a href="#">OPERA</a>	<a href="#">FIREFOX</a>	
Version	10.63	3.6	5	7	10.63	3.6	4.0
Post Message	√	√	√	√	√	√	√
HTML5 XMLHttpRequest Level 2	-	√	√	√	-	√	√

## 二、API 参考

### 1. API SPEC

Post Message: <http://www.whatwg.org/specs/web-apps/current-work/multipage/comms.html>

XMLHttpRequest Level 2: <http://www.w3.org/TR/2010/WD-XMLHttpRequest2-20100907/>

### 2. SCRIPT

#### (1) 浏览器检测:

##### <1> Post Message

```
var hasXHR_2 = typeof window.postMessage !== undefined;
```

##### <2> XMLHttpRequest

```
var hasXHR_2 = typeof (new  
XMLHttpRequest()).withCredentials !== undefined;
```

#### (2) Scripting API:

<1> Post Message:

- 发送方:

```
var win = document.getElementById('iframe').contentWindow;
win.postMessage('hi, sofish', 'http://sofish.de');
```

- 接收方:

```
window.addEventListener('message',function(evt){
    if(evt.orgin === 'http://source/domain/') {
        element.textContent = evt.data;
    }
}, true)
```

postMessage(message, targetOrigin, [, ports]);	描述
message	要发送的数据
targetOrigin	{String}, 目标域
ports	端口数组, 可选

message 事件	描述	只读
evt.data	返回接收到的数据	√
evt.origin	返回 <a href="#">server-sent event</a> 和 cross-document messageing 中发送数据的域	√
evt.source	数据源的 window object	√
evt.ports	返回数据源的端口数组对象	√

<2> XMLHttpRequest

New XMLHttpRequest Level 2 <a href="#">progress event</a> names	描述
loadstart	请求开始
progress	正在下载或发送
abort	请求停止。使用 abort() 停止请求

error	请求失败 注：我们可以用它来替换 onreadystatechange 来检测错误状态
load	请求成功 注：我们可以用它来替换 onreadystatechange 来检测成功状态
timeout	超时
loadend	请求已经完结，无论状态是成功还是失败

### 三、DEMO

#### (1) Post Message:

注：新建一个 HTML 文件，本地运行吧

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Sample</title>
</head>
<body>

  <iframe id="iframe" src="http://sofish.de/file/html5/
communication/postmessage.html" height="200" width="500"></iframe>

  <form id="form" action="">
    <input id="message" type="text" />
    <input type="submit" value="send it" />
  </form>

  <script type="text/javascript">

    var win = document.getElementById('iframe').contentWindow;
    document.getElementById('form').onsubmit = function(e){

win.postMessage(document.getElementById('message').value,
'http://sofish.de');
    e.preventDefault();
    }

  </script>

</body>
</html>
```

## (2) XMLHttpRequest Level 2 : [DEMO](#)

### 四、注意事项

- 即使 `PostMessage` 的数据是来源于可信站点，也要隐患，因为这些数据可能是这篡改的，一些最佳实践：
  - 不要轻易执行第三数提供的字符串。在自己的程序中，应避免使用 `eval` 来处理某些字符串，相反，可以使用 `JSON`，`JSON parser` 会把它变得安全；
  - 不要直接把接收的数据作为页面结构直接使用 `element.innerHTML = e.data`，我们可以先让接收的数据变成文本 `element.textContent = e.data`。
- 关于 `XMLHttpRequest` 的内容很多，更多的可以看看以下扩展阅读：
  - [MDC: XMLHttpRequest](#)
  - [cross-site xmlhttprequest with CORS](#)
  - [Http Access Control](#)

### 五、兼容解决方案

- [Alice Solution: 跨域](#)
- [IE8 XDomainRequest object](#)

### 六、参考文献

- BOOK: 《Pro HTML5 Programming》
- [HTML5: Communication](#)
- [Cross-Window Messaging](#)
- [MDC: window.postMessage](#)
- [XMLHttpRequest Level 2](#)
- [Cross-domain Ajax with Cross-Origin Resource Sharing](#)
- [MDC: XMLHttpRequest](#)
- [HTML5 & CSS3 Support, Web Design Tools & Support](#)