

...cont'd

3

Now, add a function to send a message

```
function sendMsg( ) {  
    const cage =  
        document.getElementById( 'cage' ).contentWindow  
    cage.postMessage( 'Message Received from: ' +  
        document.domain , 'http://example.com' )  
}
```

4

Create another HTML document that incorporates a heading, a paragraph to display the document domain, and an empty paragraph – in which to receive a message

```
<h1>Receiver</h1>  
<p id="host" >Iframe Page Domain: </p>  
<p id="msg"></p>
```



receiver.html

5

Begin a script to display the domain that is hosting this second HTML document, and add a message event listener

```
document.getElementById( 'host' ).innerText +=  
document.domain  
window.addEventListener( 'message', readMsg )
```

6

Now, add a function to write a received message

```
function readMsg( event ) {  
    if ( event.origin === 'http://localhost' )  
        document.getElementById( 'msg' ).innerText = event.data  
}
```

7

Save the documents on different domains, then click the button to magically send a cross-document message

A screenshot of two browser windows. The left window, titled 'Sender', has the URL 'localhost/sender.html'. It contains a red header 'Receiver' and the text 'Main Page Domain:localhost'. Below it is an empty text area. A blue button labeled 'Send Message' is at the bottom. The right window, also titled 'Sender' and with the same URL, has a red header 'Receiver' and the text 'Main Page Domain:localhost'. Below it is a text area containing 'Message Received from:localhost'. A blue button labeled 'Send Message' is at the bottom. The cursor is hovering over the 'Send Message' button in the left window.



Discover more about the Messaging API online at html.spec.whatwg.org/multipage/web-messaging.html