網頁外鏈用了target="_blank"，結果悲劇了

前端開發 6天前

以下文章來源於1024譯站，作者大道至簡

> **來自公眾號：1024譯站**

今天給大家分享一個Web 知識點。如果你有過一段時間的Web 開發經驗，可能已經知道了。不過對於剛接觸的新手來說，還是有必要了解一下的。

我們知道，網頁裡的 `a` 標籤默認在當前窗口跳轉鏈接地址，如果需要在新窗口打開，需要給 `a` 標籤添加一個 `target="_blank"` 屬性。

```
<a href="http://kaysonli.com/" target="_blank">1024译站</a>
```

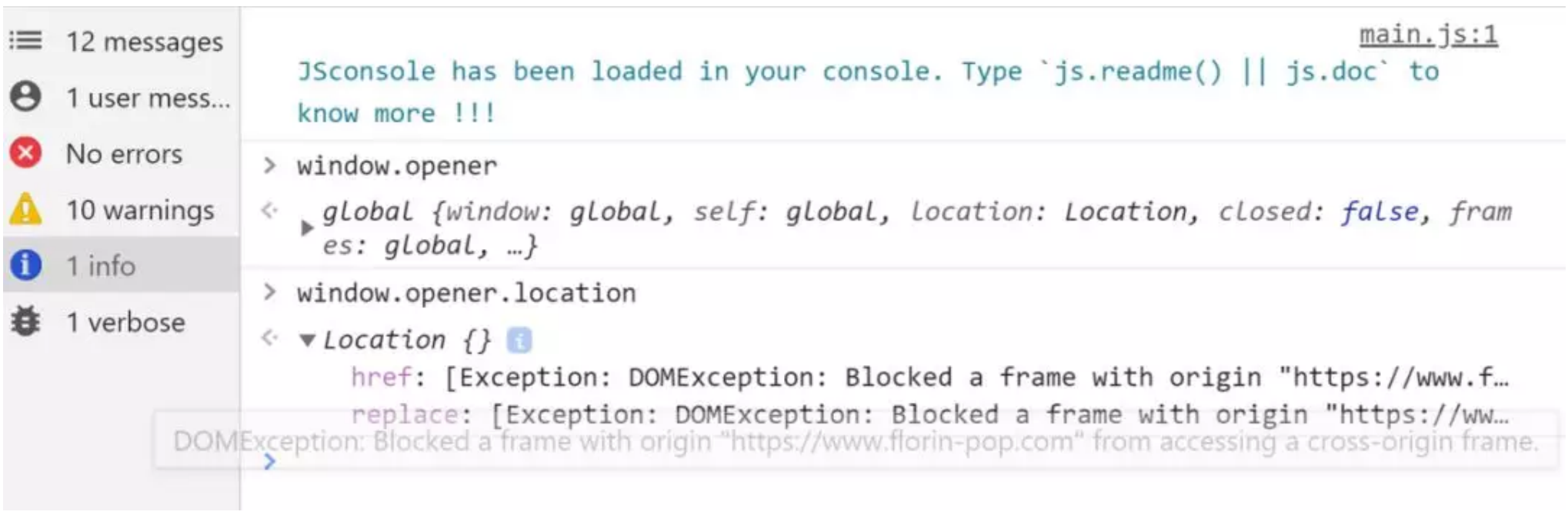
順便提下一個有意思的現象，很早之前我就發現，國外網站傾向於在當前頁跳轉，而國內網站喜歡打開新窗口。不信你們可以去驗證下。我不知道這是交互設計上的文化差異，還是技術上的開發習慣。

當然，這兩種方式各有優缺點。當前頁跳轉顯得操作比較有連貫性，不會貿然打斷用戶的注意力，也會減少瀏覽器的窗口（tab 頁）數量。但是對於需要反復回到初始頁面的場景來說，就很麻煩了。比如搜索結果頁面，通常需要查看對比幾個目標地址，保留在多個窗口還是比較方便。

今天要說的不只是用戶體驗上的差別，而是涉及**安全**和**性能**。

安全隱患

如果只是加上 `target="_blank"` ，打開新窗口後，新頁面能通過 `window.opener` 獲取到來源頁面的 `window` 對象，即使跨域也一樣。雖然跨域的頁面對於這個對象的屬性訪問有所限制，但還是有漏網之魚。



這是某網頁打開新窗口的頁面控制台輸出結果。可以看到 `window.opener` 的一些屬性，某些屬性的訪問被攔截，是因為跨域安全策略的限制。

即便如此，還是給一些操作留下可乘之機。比如修改 `window.opener.location` 的值，指向另外一個地址。你想想看，剛剛還是在某個網站瀏覽，隨後打開了新窗口，結果這個新窗口神不知鬼不覺地把原來的網頁地址改了。這個可以用來做什麼？釣魚啊！等你回到那個釣魚頁面，已經偽裝成登錄頁，你可能就稀里糊塗把賬號密碼輸進去了。

還有一種玩法，如果你處於登錄狀態，有些操作可能只是發送一個 `GET` 請求就完事了。通過修改地址，就執行了非你本意的操作，其實就是CSRF攻擊。

性能問題

除了安全隱患外，還有可能造成性能問題。通過 `target="_blank"` 打開的新窗口，跟原來的頁面窗口共用一個進程。如果這個新頁面執行了一大堆性能不好的JavaScript代碼，佔用了大量系統資源，那你原來的頁面也會受到池魚之殃。

解決方案

盡量不使用 `target="_blank"` ，如果一定要用，需要加上 `rel="noopener"` 或者 `rel="noreferrer"` 。這樣新窗口的 `window.openner` 就是 `null` 了，而且會讓新窗口運行在獨立的進程裡，不會拖累原來頁面的進程。不過，有些瀏覽器對性能做了優化，即使不加這個屬性，新窗口也會在獨立進程打開。不過為了安全考慮，還是加上吧。

我特意用自己的博客網站http://www.kaysonli.com/試了一下，點擊裡面的外鏈打開新頁面， `window.openner都` 是 `null` 。查看頁面元素發現， `a` 標籤都加上了 `rel="noreferrer"`。博客是用Hexo生成的，看來這種設置已經成了基本常識了。

另外，對於通過 `window.open` 的方式打開的新頁面，可以這樣做：

```
var yourWindow = window.open();
yourWindow.opener = null;
yourWindow.location = "http://someurl.here";
yourWindow.target = "_blank";
```

希望這個小技巧對你有用。

- 編號1199，輸入編號直達本文
- ●輸入m獲取文章目錄

推薦↓↓↓

Web開發