移动端 click 事件 300ms 延迟的前世今生

白蜀黍 (/u/0a00ea99f32a) ＋关注
2017.03.28 12:36* 字数 2596 阅读 276 评论 1 喜欢 8
(/u/0a00ea99f32a)

> 原文首发于 baishusama.github.io (https://link.jianshu.com?t=http://baishusama.github.io/2017/03/27/%E7%A7%BB%E5%8A%A8%E7%AB%AF-click-%E4%BA%8B%E4%BB%B6%E7%9A%84-300ms-%E5%BB%B6%E8%BF%9F/)，欢迎围观～

存疑

最开始，我遇到的其实是“移动端遮罩层滑动穿透”的问题。

在查找“滑动穿透”问题相关资料的时候，我搜到了很多 click 300ms 延迟的问题。我那个时候有些不知所云，因为我自己并没有真实遇到过 300ms 延迟现象，也就没怎么在意。

时至今日想动笔总结遇到了若干次的“滑动穿透”问题的时候，搜集资料的偶然间得以解惑 300ms 的前世今生。

移动端 `click` 的 300ms 延迟

那么，这 300ms 延迟到底是从哪里来的呢？ (https://link.jianshu.com?t=http://www.telerik.com/blogs/what-exactly-is.....-the-300ms-click-delay)

时间要追溯到 2007 年初代 iPhone 发布前夕，苹果为了解决“如何用手机这种小尺寸屏幕来显示 PC 端网页”这个问题，提出了很多聪明的约定（convention）。而后因为 iPhone 的大获成功，这些约定被各大手机浏览器争相效仿。

这些约定之中，**双击缩放（Double Tap to Zoom）** 就是 300ms 的“元凶”——当用户在页面上 click 的时候，浏览器为了判断这个用户操作是单击还是双击，会等待 300-350ms 。如果 300ms 内，发生了第二次 click 事件，那么视为双击；否则为单击，等 300ms 时间过去之后，才触发 click 事件。

在那个还不存在响应式设计和双指缩放（Pinch to Zoom）的时代，这个延迟是一个合理的预防措施。但不幸的是，**这 300ms 的延迟已经成为用户觉得 web 应用比 native 应用更慢、性能不及后者的主要原因之一。**诸如，链接、按钮、多选框等基于 click 交互的元素，以及 JS 对 click 事件的监听，都因此受到影响。

幸运的是，浏览器开发商（vendor）和开发者都注意到了这个问题，提出了一些解决方案。

解决方案

方案一、👎 禁用缩放

- 代码：

```
<meta name="viewport" content="user-scalable=no">
<!-- 或者 -->
<meta name="viewport" content="initial-scale=1, minimum-scale=1, maximum-scale=1">
```

- 原理：双击是为了缩放，如果禁用缩放，那么就没双击什么事儿了，也不需要额外等待 300ms 了。
- 支持情况：在 Android 平台上，由 Chrome 最先提出，FireFox、Opera 等浏览器也相继支持；IOS 9.3 开始一度支持，IOS10 开始不再支持。
- 缺点：Safari 不支持。而且，禁用缩放会**损害移动端网页的可用性和可访问性**。例如，可能无法放大网页中的一张图片或一段字体较小的文字。

> 这里要注意区分：“双击缩放”（Double Tap to Zoom）和“双指缩放”（Pinch to Zoom）。为了兼顾消除 300ms 延迟和不损害可用性和可访问性，**我们应该抛弃双击缩放、拥抱双指缩放。**

方案二、👍 视窗宽度设置为设备宽度

- 代码：

```
<meta name="viewport" content="width=device-width">
```

- 原由：正如 Chromium Code Reviews (https://link.jianshu.com?t=https://codereview.chromium.org/18850005/) 上说的，`viewport` 的 `width` 设置得小于等于 `device-width` 的页面，是针对移动端优化过的或者是响应式的站点，其内容足够清晰，双击缩放失去了意义。因此，为包含上面这行代码的页面**禁用双击缩放**。同时，双指缩放得以保留，从而也就**没有可用性和可访问性问题**了。
- 支持情况：自 Chrome 32 开始，FF、IE/Edge 也随后支持了；2016 年 3 月，IOS 9.3 开始支持。
- 推荐使用！

> 该解决方案的“禁止双击缩放”是遵守如下规则 (https://link.jianshu.com?t=https://trac.webkit.org/changeset/191644/webkit)的：
>
> - 当页面设置了视窗宽度为设备宽度且是初始尺寸（页面尚未缩放），此时，双击缩放才是被禁止的。
> - 如果视窗尺寸不是初始尺寸（页面已经缩放），双击缩放是被允许的。
> - 为了在用户结束缩放后仍能 fast-click ，缩小时，只能缩小到初始尺寸，而不是最小尺寸。

方案三、👍 指针事件（Pointer Events）

- 代码：

```
a, button, .myelements {
    -ms-touch-action: manipulation; /* IE10  */
    touch-action: manipulation;     /* IE11+ */
}
```

- 根据规范 (https://link.jianshu.com?t=https://w3c.github.io/pointerevents/#the-touch-action-css-property)：CSS 属性 `touch-action` 决定了触摸输入（touch input）能否触发 UA （User Agent）支持的默认行为。这包括但不限于诸如平移或缩放等行为。
- 根据 MDN (https://link.jianshu.com?t=https://developer.mozilla.org/en-US/docs/Web/CSS/touch-action)： `touch-action` 的 `manipulation` 值激活了平移和双指缩放手势，而禁用了双击缩放等非标准的手势。
- 支持情况：在 Can I Use (https://link.jianshu.com?t=http://caniuse.com/#search=touch-action) 上可以看出，除了 Opera Mini 不支持、FF 需要手动启用和 Android 4.x 的自带浏览器有些迷之外，其他浏览器支持良好。
- 推荐使用！

在只有 IE 支持指针事件的初期，诞生了不少指针事件的 polyfill 解决方案 (https://link.jianshu.com?t=https://thx.github.io/mobile/300ms-click-delay#%E6%8C%87%E9%92%88%E4%BA%8B%E4%BB%B6%E7%9A%84-polyfill)。在仍不支持指针事件的浏览器上，这是一种变通的方式。

`shim` VS `polyfill`

1. 一个 shim 是一个库，它将一个新的 API 引入到一个旧的环境中，而且仅靠旧环境中已有的手段实现。
2. **polyfill 就是浏览器 API 的 shim 。** (https://link.jianshu.com?t=https://www.wikiwand.com/en/Polyfill) 它用于实现浏览器并不支持的原生 API 的代码，是**抹平**新旧浏览器对原生 API 支持**差异**的封装。通常，polyfill 会先检查当前浏览器是否支持某个 API，如果不支持的话就加载它自己的实现，然后新旧浏览器就都可以使用这个 API 了。相当于“打补丁”，“刮腻子”。

方案四、👍 轻量级库 FastClick (https://link.jianshu.com?t=https://github.com/ftlabs/fastclick)

- 代码：

```
window.addEventListener( "load", function() {
    FastClick.attach( document.body ); // 直接绑定到 <body> 上可以确保整个应用都能受益
}, false );
```

- 原理：FastClick 在检测到 `touchend` 事件的时候，会通过 DOM 自定义事件 (https://link.jianshu.com?t=https://developer.mozilla.org/en-US/docs/Web/API/CustomEvent)立即触发一个模拟的 `click` 事件，并把浏览器 300ms 之后真正触发的 click 事件阻止掉。
- 无冲突：当 FastClick 检测到当前页面使用了基于 `<meta>` 标签或者 `touch-action` 属性的解决方案时，会静静地看别的解决方案装逼 (https://link.jianshu.com?t=https://github.com/ftlabs/fastclick#when-it-isnt-needed)。
- 唯一的缺点：文件大小占 10 KB……
- 推荐使用！

关于“始作俑者” Safari

起承转折

(2013) 300 毫秒点击延迟的来龙去脉 (https://link.jianshu.com?t=https://thx.github.io/mobile/300ms-click-delay)一文中提到的 IOS 特有的**双击滚动（Double Tap to Scroll）：仍存在**、并没有像原文猜测的那样消失。（亲测 IOS 10.2.1 Safari 已设置 `<meta name="viewport" content="width=device-width">` 的页面在屏幕上或下 1/4 处双击仍能滚动。）

起初看到「2016 年 3 月发布的 IOS 9.3 移除了 300ms 延迟、从而实现了“fast-tap” (https://link.jianshu.com?t=https://developer.apple.com/library/content/releasenotes/General/WhatsNewInSafari/Articles/Safari_9_1.html#//apple_ref/doc/uid/TP40014305-CH10-SW8)」时，我还欣慰地想道：最先提出“双击缩放”约定的苹果，在最后也顺应了历史潮流嘛。但是接着看到「IOS10 无视了禁用缩放（user-scalable） (https://link.jianshu.com?

t=https://developer.apple.com/library/content/releasenotes/General/WhatsNewInSafari/Articles/Safari_10_0.html#//apple_ref/doc/uid/TP40014305-CH11-SW1)」我的内心瞬间黑人问号：“？？？”。

后来，静静地看了两篇文章（Safari zoom gesture's comeback in iOS 10 (https://link.jianshu.com?t=http://cloudless.studio/articles/39-safari-zoom-gesture-s-comeback-in-ios-10) 和 How to disable viewport scaling in iOS 10? You don't. (https://link.jianshu.com?t=https://wouterdeschuyter.be/blog/how-to-disable-viewport-scaling-in-ios-10-you-dont-941140811)），做了点 `<meta>` 标签的测试。

测试结果

测试环境：IOS 10.2.1

1. 只设置 `<meta name="viewport" content="user-scalable=no">` 和不设置没有任何区别——`user-scalable=no` 被完全无视。
2. 只设置 `<meta name="viewport" content="width=device-width">` ，和方案二里的描述一致，仍可以在初始尺寸下禁用双击缩放。
3. 只设置 `<meta name="viewport" content="initial-scale=1.0">` ，初始状态和“测试2”很像，但是仍存在双击缩放，即仍有 300ms 延迟。
4. 设置 `<meta name="viewport" content="width=device-width, initial-scale=1.0">` 或者 `<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">` ，和只设置 `width=device-width` 并无显著差异。
5. 为某个 `<a>` 链接设置 `touch-action: manipulation;` ，可以禁用该元素上的双击缩放。

要点如下：

1. `user-scalable=no` 完全起不到禁止缩放的作用， `width=device-width` 仍能且仅能禁止双击缩放。
2. 只设置 `meta` 无法完全禁用缩放，双指缩放总是可行的。

暮然回首

冷静下来后，重新审视上述变故，发现其实是两回事。前面提到过“我们应该抛弃双击缩放、拥抱双指缩放”，苹果没有打破这个原则。只是，**苹果出于可访问性考虑，直接任性地完全无视了 `user-scalable=no` 。**

> **Accessibility**
> Pinch-to-zoom is always enabled for all users. The viewport setting for user-scalable is ignored. (https://link.jianshu.com?t=https://developer.apple.com/library/content/releasenotes/General/WhatsNewInSafari/Articles/Safari_10_0.html#//apple_ref/doc/uid/TP40014305-CH11-SW1)

当然，这导致了觉得应该一切尽在掌控、想要完全禁用缩放以避免破坏布局的开发者的怨言。如果，你**还是想完全禁用缩放，可以参考** SO 上的这个回答 (https://link.jianshu.com?t=http://stackoverflow.com/questions/37808180/disable-viewport-zooming-ios-10-safari/38573198#38573198)。

解惑

最开始提到过，我至今没有遇到过这个问题。对这个现象我推理如下：

我的肾机在开发移动端的半年间只在近期做过一次系统升级（目前已升到 10.2.1）。之前使用的具体的版本号已经无从得知了（P.S. 如果有谁知道怎么查看肾机本机上的版本更新历史，请务必告诉我233），但是更新到 IOS10 之前，我一直有使用 9.3+ 才支持的 Night Shift 功能，也就是说升级之前的系统版本号肯定在 9.3 或者以上。

而我写移动端页面的时候，惯例会 `meta:vp` 然后 `Tab` 生成 `<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">` 。在 IOS10 之前，这行代码还是能够禁用页面的缩放的，也就不存在 300ms 的延迟问题了。

这就是为什么之前我本机测试的时候一直没有遇到传说中的 300ms 延迟现象的原因了。

参考

1. (2013) 300 毫秒点击延迟的来龙去脉 (https://link.jianshu.com?t=https://thx.github.io/mobile/300ms-click-delay)
2. (2014) 5-ways-prevent-300ms-click-delay-mobile-devices (https://link.jianshu.com?t=https://www.sitepoint.com/5-ways-prevent-300ms-click-delay-mobile-devices/)
3. (2015) Implement viewport-width-based fast-click heuristic (https://link.jianshu.com?t=https://trac.webkit.org/changeset/191644/webkit)
4. (2016) 无线端浏览器 click 事件 300ms 延迟 (https://link.jianshu.com?t=http://qiudeqing.com/mobile_web/2016/05/21/mobile-browser-click-300ms-delay.html)
5. (2016) 300ms-tap-delay-gone-away (https://link.jianshu.com?t=https://developers.google.com/web/updates/2013/12/300ms-tap-delay-gone-away)