

移动页面加速google的AMP和百度的MIP简介

这篇文章发布于 2017年09月10日, 星期日, 22:47, 归类于 [Mobile相关](#)。阅读 16168 次, 今日 6 次 [16 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=6404>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

一、关于移动页面加速

google先出了个AMP – Accelerated Mobile Pages (加速的移动页面), 后来百度仿照出了个MIP– Mobile Instant Pages – 移动网页加速器。

大致意思是说, 你做网页呀, 如果遵循我制定的一些规则, 你的页面加载速度会更快, 据吹会有15%~85%的性能提升。

听上去好像很厉害, 但是, 这里的“遵循我制定的一些规则”可不是类似YSlow之类的性能优化建议, 那可是真真实实伤筋动骨的修改, 从成本权衡来讲, 修改老页面是不靠谱的, 需要重新整套适配这些搜索引擎的页面。

我去, 就为了提高一点虚幻的加载速度, 就花这么多成本去重新搞一套页面, 无论是对开发者还是企业而言, 实在是下不了手啊, 所以如果说单纯就是为了加速, 无论是AMP还是MIP的推广和普及必将遭受重大的阻碍, 因为无需伤筋动骨就可以进行网页优化的手段很多, 虽然优化效果没有使用AMP或MIP那么极致, 但觉得性价比高很多。

然而, 无论是AMP还是MIP, 还有一个潜在的, 并没有大肆宣扬的好处, 那就是可以影响搜索引擎的排名 (也就是SEO)。至于有多影响, 并没有明说。但是以我本人不太专业的看法, 原来你是在搜索引擎第二页的, 使用了AMP或MIP后跑到第一页, 绝对是没问题的。中国人最讲究面子, 你如此看得起我, 用我推出的技术, 我不给你点好处, 我自己良心也过不去呀, 对不对? ——加个闪电标志, 有身份; 排名提前, 有好处。

嘿嘿, 这下问题就搞大了, 这搜索引擎的排名啊, 那可是直接跟钱挂钩的呀, 注意这里的“直接”; 网站加载速度虽然也与钱挂钩, 但这个它不直接, 太隐蔽, 所以上司啊老板啊都更看重网站的搜索引擎排名, 而不是加载速度。

二、google的AMP和百度的MIP快速上手

google的AMP文档地址是: <https://www.ampproject.org/docs/>

百度的MIP文档地址是: <https://www.mipengine.org/doc/00-mip-101.html>

由于AMP和MIP干的是同样的事情, 并且借鉴痕迹明显, 因此两者实现的套路基本是一样的, 例如拿快速创建一个AMP或MIP页面举例:

1. 首先创建一个标准的HTML文件:

- 在 `<html>` 标签中增加 `mip` 标识
- 编码为 `utf-8`
- 添加 `meta-viewport`, 用于移动端展现

```
<!DOCTYPE html>
```

```

<html mip>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
    <title>Hello world</title>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>

```

2. 添加运行环境:

在AMP中使用AMP需要的JS（外链），以及CSS（内联），在MIP中使用MIP需要的CSS和JS（都是外链）。

然后文件放置的位置也有所区别如下，AMP的JS放置在头部，而MIP和传统规则一致，CSS放在头部，JS放在底部，HTML结构示意图分别如下：

```

<!DOCTYPE html>
<html mip>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
    <link rel="stylesheet" type="text/css" href="https://mipcache.bdstatic.com/static/v1/mip.css">
    <title>Hello world</title>
  </head>
  <body>
    <h1>Hello world!</h1>
    <script src="https://mipcache.bdstatic.com/static/v1/mip.js"></script>
  </body>
</html>

```

```

<!doctype html>
<html amp lang="en">
  <head>
    <meta charset="utf-8">
    <script async src="https://cdn.ampproject.org/v0.js"></script>
    <title>Hello, AMPs</title>
    <link rel="canonical" href="http://example.ampproject.org/article-metadata.html">
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
    <script type="application/ld+json">
      {
        "@context": "http://schema.org",
        "@type": "NewsArticle",
        "headline": "Open-source framework for publishing content",
        "datePublished": "2015-10-07T12:02:41Z",
        "image": [
          "logo.jpg"
        ]
      }
    </script>
    <style amp-boilerplate>body{-webkit-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-moz-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-ms-animation:-amp-start 8s steps(1,end) 0s 1 normal both;animation:-amp-start 8s steps(1,end) 0s 1 normal both;}</style>

```

```

1 both}@-webkit-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-
moz-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-ms-keyframes
-amp-start{from{visibility:hidden}to{visibility:visible}}@-o-keyframes -amp-start{from
{visibility:hidden}to{visibility:visible}}@keyframes -amp-start{from{visibility:hidden
}to{visibility:visible}}</style><noscript><style amp-boilerplate>body{-webkit-animati
on:none;-moz-animation:none;-ms-animation:none;animation:none}</style></noscript>
</head>
<body>
  <h1>welcome to the mobile web</h1>
</body>
</html>

```

3. 添加关联标签:

首先一样的是都需要在头部 `<head>` 引入关联标签:

```
<link rel="canonical" href="$SOME_URL">
```

这里的 `$SOME_URL` 指向的就是当前页面对应的普通页面地址, 此时浏览器会继承原页面(移动端)的点击权重, 同时加速页将作为搜索引擎的首选导流页面。

百度的MIP还提供了这个:

```
<link rel="miphtml" href="$MIP_URL">
```

添加在普通页面的 `<head>` 标签中, 用来告诉搜索引擎加速页面地址是哪一个。

谷歌的则是:

```
<link rel="amphtml" href="$AMP_URL">
```

如果我们的加速页面没有原页面, 则 `href` 需要指向当前页面地址。

4. 添加样式:

在移动加速网页中, 所有的自定义CSS全部建议内联, 在MIP中使用 `mip-custom` 属性标识, 在AMP中使用 `amp-custom` 标识, 例如:

```
<style mip-custom>
  h1 { color: red;}
</style>
```

```
<style amp-custom>
  /* any custom style goes here */
  body {
    background-color: white;
  }
  amp-img {
    background-color: gray;
    border: 1px solid black;
  }
</style>
```

5. 替换禁用的HTML标签:

在使用移动页面加速技术的时候, 很多标签是不能使用的。例如 ``、`<video>`、`<iframe>` 等都是不能使用的, 需要使用其他标签替换或者直接禁止使用 (如 `<iframe>`), 例如在MIP中

，`` 标签需要替换成 `<mip-img>`，在AMP中 `` 标签需要替换成 `<amp-img>`。

```
<mip-img layout="responsive" width="350" height="263" src="https://www.mipengine.org/static/img/mip_logo_3b722d7.png"></mip-img>
```

6. 使用内置的组件：

出于对代码质量和性能的考虑，移动加速页面是不允许自定义JavaScript代码的。如果有交互效果，必须使用JavaScript，则需要使用MIP或者AMP内置的组件。

无论是谷歌的AMP还是百度的MIP，其组件都分为内置组件和扩展组件（个性化组件），所谓内置组件指的是不需要再引入额外JS的组件，而扩展组件需要再引入JS文件。

MIP的组件文档[参见这里](#)，AMP的组件文档[参见这里](#)。

两者内置的组件都非常的多，具体到某个具体组件还是有一些差异的，比方说多图片轮播组件carousel在MIP中属于内置组件，但是在AMP中则是扩展组件，需要引入额外的JS：

```
<script async custom-element="amp-carousel" src="https://cdn.ampproject.org/v0/amp-carousel-0.1.js"></script>
```

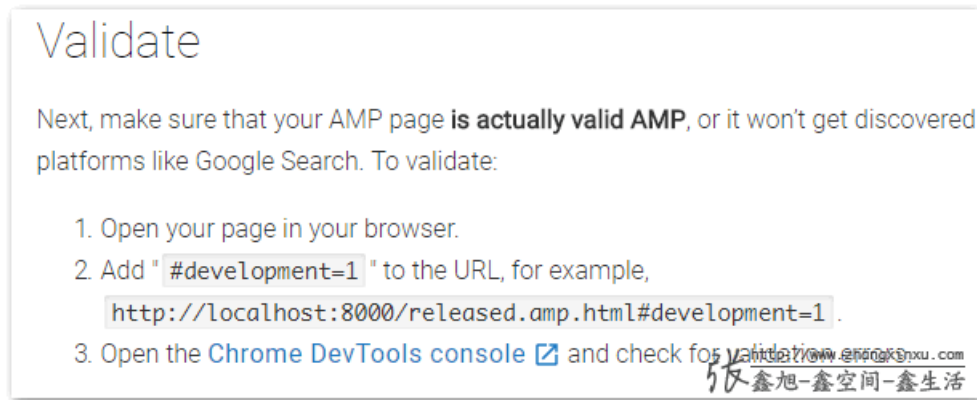
7. 预览和验证：

无论是谷歌的AMP还是百度的MIP都提供了预览和验证功能，百度MIP是在线工具：

校验代码，使用 [MIP 校验工具](#)。

预览线上 URL 异步打开效果，使用 [MIP 预览工具](#)。

谷歌AMP则是借助google chrome开发者工具：



以上几步流程就是演示的如果构建一个简单的移动加速页面。

更复杂以及更详尽的一些使用，可以自行参考官方文档，或者等一段时间，看看我能不能有一些实践经验分享。

三、建议优先支持百度的MIP

虽然谷歌的AMP和百度的MIP原理和思路类似，但毕竟代码上还存在很多差异，想要全部都支持，一种是借助工具，另外一种就是两套都开发。但是很显然如果两套都支持，那成本真的是相当高了。

所以对国内的用户和产品而言，我的建议是优先支持百度的MIP，因为毕竟在移动端使用谷歌进行搜索的人并不多，占比非常有限。在面对商业利益面前，所谓的爱恨情仇我们其实应该放在一边，更关注的是现实利益，所以如果想要进行移动网页加速支持，直接走MIP即可！

举个例子，拿分享组件而言，AMP分享的就是twitter，Facebook，组件也是YouTube之类的，对国内用户

而言，价值真的是非常有限。而百度的MIP分享则本地化要更加友好。

四、网页移动加速原理

很多小伙伴想了解移动加速的原理。

加速主要在两方面：

1. 任何页面，你CSS内联，交互简单化，不使用渲染速度不快的HTML标签，都会变快的，与AMP或者MIP并没有直接关系；
2. 搜索引擎厂商的高性能CDN。我们审视下按照移动加速规则处理后的网页，基本上就是一个与开发人员和开发企业没有任何关系的页面，所有CSS内联，所有JS都使用AMP或者MIP提供的JS，也就是说，无论是网页HTML还是JS全部都可以缓存在搜索引擎厂商的CDN上，相当于免费使用大厂的CDN。要知道使用CDN都是要收费的，使用了使用AMP或者MIP后，这部分钱可以省了。

以上就是加速原理。其实，和我们平常移动端网页优化规则是一致的，只是AMP和MIP进行的规则强约定。

五、结束语

无论是谷歌的AMP还是百度的MIP对我们网页的交互限制非常大，对于一些非常复杂的一面，想要原封不动的转换为移动加速页面，难度是非常高的，成本是非常大的。考虑到这种现实问题，我们实际的做法通常有两个方向：一是实现移动加速页面的时候调整原页面的交互设计为简单模式，例如普通的小说阅读页面支持上下左右滑屏加载阅读，如果变成AMP或者MIP页面，就只能变成点击下一章链接去加载，UI布局也需要简化，例如去除设定阅读纹理，白天黑夜模式之类的功能；另外一个是对原本就比较简单的页面进行移动加速处理，例如博客文章，新闻内容，广告页面等。

对于并未广泛应用的新技术而言，我个人建议是先从一些简单的页面入手，通过数据验证，MIP加速确实能够明显增加公司的收益，再大规模应用在其他页面上，有助于规避成本风险。

好，以上就是本文的全部内容，感谢阅读，欢迎纠错！



《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](#)。



« 使用canvas实现和HTML5 video交互的弹幕效果

请使用千位分隔符(逗号)表示web网页中的大数字 »

猜你喜欢

- 找到适合自己的前端发展方向
- 回流与重绘：CSS性能让JavaScript变慢？
- 翻译 - CSS Sprites:实用技术还是生厌之物？
- 炫酷H5中序列图片视频化播放的高性能实现
- HTML5扩展之微数据与丰富网页摘要

- 翻译 - 高性能网站需避免的7个错误
- 你到了第几层? 图片式标题、按钮与隐藏文本
- PhoneGap/Cordova控制iOS7状态栏的显隐/颜色
- mobilebone.js-mobile移动web APP单页切换骨架
- 小tip: 使用meta实现页面的定时刷新或跳转
- 开源移动端元素拖拽惯性弹动以及下拉加载两个JS

分享到: 1

标签: AMP, MIP, SEO, 优化, 性能, 移动开发

发表评论 (目前16条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. 大表哥说道:

2018年12月15日 09:36

mip卒, 2018年12月14日

[回复](#)



2. 码客说道:

2018年06月22日 16:39

还要考虑两套标准, 真是服了这两家~

[回复](#)



3. 香保说道:

2018年05月25日 22:33

学习一下。

[回复](#)



4. leslieSie说道:

2018年04月30日 14:59

据我所知目前框架属于侵入式的框架, 不建议使用, 而且没有生态, 受限大

[回复](#)



5.

曾小乱说道：

2018年01月12日 16:22

可以和 vue 或者 react 框架结合使用吗？目前我了解到并不行。

回复

abc说道：

2018年02月1日 18:31

我也想在react中加入mip。

回复


6.

房屋贷款利率说道：

2017年09月27日 15:57

非常有用，值得学习，谢谢

回复


7.

vfv001说道：

2017年09月12日 16:27

项目用过mip,限制还是比较多的

回复


8.

mantou说道：

2017年09月12日 12:19

我觉得最大的加速原因没有说：就是多页面共享资源。
访问过其他页面，缓存了自己页面的资源。

回复


9.

yugasun说道：

2017年09月12日 10:13

非常赞同博主说法，标准虽然能提供规范，但是如果不是自由组织而是互联网巨头指定的标准，那就可能是一种垄断了。

回复


10.

shinhwa说道：

2017年09月12日 09:07

问个问题，如果所有js都必须使用内置的，那页面自带的业务逻辑怎么办，也必须使用内联的js写？

回复

zphhhh说道：

2018年08月3日 16:53

内联js也不行，利用插件写。

回复


11.

Jackson说道：

2017年09月11日 15:52

厉害了！

回复



12. 前端小武说道：
2017年09月11日 15:50
前排留名...
[回复](#)
13. nenzo说道：
2017年09月11日 09:41
只能用在移动端吗
[回复](#)
- Inori说道：
2017年09月11日 13:49
移动端的效果格外显著的意思
[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](#) (190)
- » [未来必热: SVG Sprite技术介绍](#) (119)
- » [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](#) (79)
- » [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

- [找到适合自己的前端发展方向](#)
- [回流与重绘：CSS性能让JavaScript变慢？](#)
- [翻译 - CSS Sprites:实用技术还是生厌之物？](#)
- [炫酷H5中序列图片视频化播放的高性能实现](#)
- [HTML5扩展之微数据与丰富网页摘要](#)
- [翻译 - 高性能网站需避免的7个错误](#)
- [你到了第几层？图片式标题、按钮与隐藏文本](#)
- [PhoneGap/Cordova控制iOS7状态栏的显隐/颜色](#)
- [mobilebone.js-mobile移动web APP单页切换骨架](#)
- [小tip: 使用meta实现页面的定时刷新或跳转](#)
- [开源移动端元素拖拽惯性弹动以及下拉加载两个JS](#)