前端图片优化方案

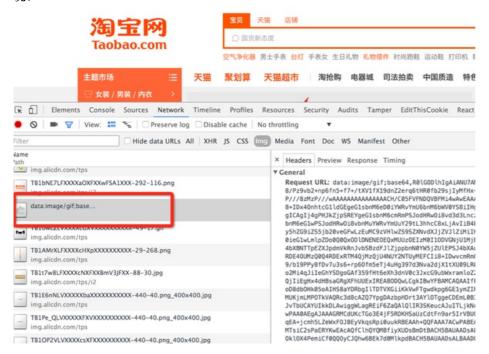
前端图片优化的方案:

- 1. 使用base64编码代替图片
- 2. 合并图片sprite
- 3. 图片延迟加载
- 4. 使用css、svg、canvas或iconfont代替图片
- 5. 加载对应尺寸图片
- 6. 预加载
- 7. 更好的图片格式

我们拿淘宝网来做案例分析:

1. 使用base64编码代替图片

场景:将图片转换为base64编码字符串inline到页面或css中,适用于图片大小小于2KB,页面上引用图片总数不多的情况。

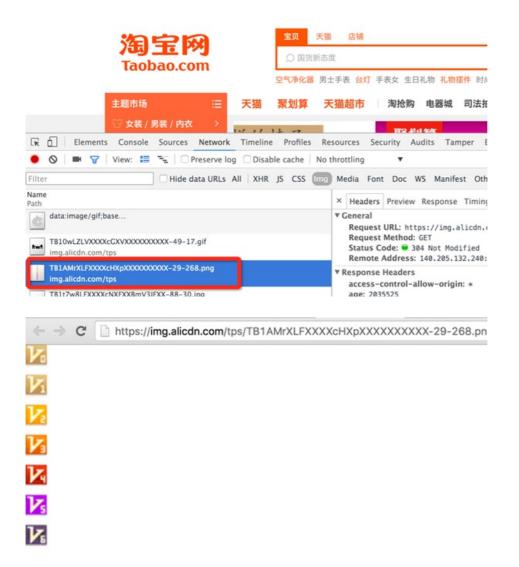


2. 合并图片sprite

场景:将多个页面上用到的背景图片合并成一个大的图片在页面中引用 ,可以有效的较少请求个数。但是合并的大图也不要太大。

比如淘宝首页的这个sprite图片

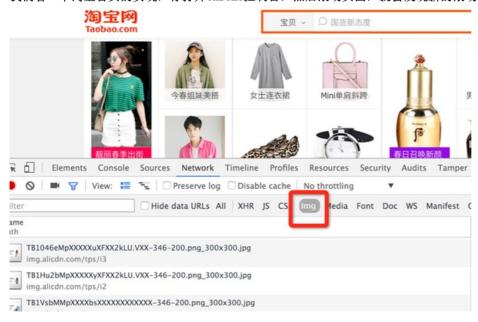
我们打开这个图片看一下



3. 图片延迟加载

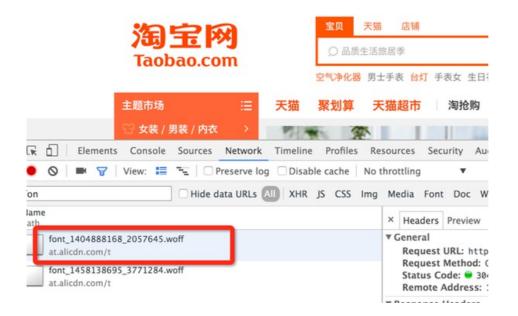
场景:对应图片比较多的页面,可以考虑通过js来实现图片的延迟加载,先让一部分图片优先请求下来,当用户滚动页面的时候进一步加载图片。

我们看一下淘宝首页的实现,你打开chrome控制台,然后滚动页面,就会发现新的滚动就会导致大量图片的新请求



4. 使用css、svg、canvas或iconfont代替图片

场景: css方式可以用来绘制相对简单的团来代替图片,一般使用before或者after伪元素来丰富图案的复杂度。使用css, svg, iconfont减少图片尺寸,请求数据少。



5. 加载对应尺寸图片

场景:不同终端对同一个图片需求不一样,可以根据终端加载不同的图片来节省没必要的流量。 查看淘宝首页图片的网络加载,会通过cdn技术,加上对应要加载的图片参数,比如60x60,加载需要的尺寸。



6. 预加载

场景:默认情况下浏览器会对页面中和当前域名(正在浏览网页的域名)不在同一个域的域名进行预获取,并且缓存结果,这就是隐式的DNS Prefetch。如果想对页面中没有出现的域进行预获取,那么就要使用显示的DNS Prefetch了,也就是使用link标签。

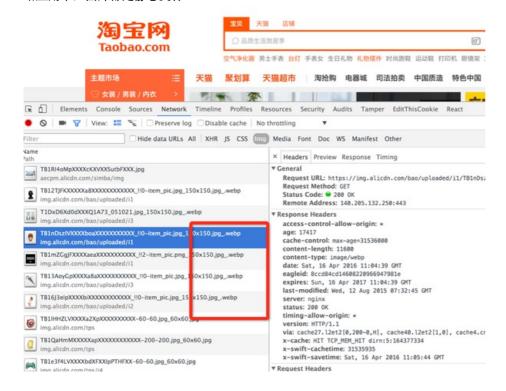
我们在查看淘宝网的前端代码可以看到前端加上了dns-prefetch,因为前端静态资源的域名和主域名http://taobao.com不一致



7. 更好的图片格式

场景:使用webp、bpg、sharpP等新图片格式来替代原始格式,这些具有更好的压缩比。

我们看淘宝这边,部分图片使用了webp格式,WEBP与JPG相比较,编码速度慢10倍,解码速度慢1.5倍,而绝大部分的网络应用中,图片都是静态文件。



大淘宝基本把我所列举的优化方式都用上。

具体项目中根据自己的使用场景选择你需要的优化图片优化方案即可!

文/野次(简书作者)

原文链接: http://www.jianshu.com/p/f7725a55df20

著作权归作者所有,转载请联系作者获得授权,并标注"简书作者"。