**IE 团队的前端性能优化建议**

2009-3-18 ~ 2009-3-20，微软面向 Web 开发和 Web 设计人员的 MIX 09 年度大会在拉斯维加斯举行，本届大会的主题是”The Next Web Now”，关于开发技术和设计怎么更加有效结合起来，服务未来互联网的发展趋势。120多场精彩的课程，3000多来自各国的专家参会。

视频及PPT[在此](http://videos.visitmix.com/MIX09/All)下载，虽然其中多数是微软的产品宣讲，不过也有一些很有价值的分享，比如：

* C23F[Windows Internet Explorer 8 in the Real World: How Is Internet Explorer 8 Used](http://videos.visitmix.com/MIX09/C23F)
* C24F[Measuring Social Media Marketing](http://videos.visitmix.com/MIX09/C24F)
* C26F[Designing the Windows 7 Desktop Experience](http://videos.visitmix.com/MIX09/C26F)
* T25F[Web Development Using Microsoft Visual Studio: Now and in the Future](http://videos.visitmix.com/MIX09/T25F)

今天主要介绍的是 IE 团队给出的前端性能优化建议，下面是摘要翻译，有兴趣的话可以看下[原PPT](http://mschannel9.vo.msecnd.net/o9/mix/09/pptx/t53f.pptx)

* 出处：[Building High Performance Web Applications and Sites](http://videos.visitmix.com/MIX09/T53F)
* 演讲人：John Hrvatin，Internet Explorer 项目经理
* 翻译整理：[ytzong](http://www.99css.com)

**中心思想：**

* 适用于所有浏览器
* 没有魔术般的解决方案
* 考虑可维护性

**Top 100 sites IE8 CPU 使用情况：**

* 84% – 布局，渲染，格式化…
* 16% – JScript & DOM

**Top AJAX sites IE8 CPU 使用情况：**

* 67% – 布局，渲染，格式化…
* 33% – JScript & DOM

从以下4个方面来进行优化：

1. **CSS 性能：**
   * 尽量减少样式
     + 未用到的样式增加了下载量
     + 浏览器必须解析并匹配所有选择器
       - 失败代价很高！
   * 简化选择器
     + 复合的元素选择器会慢
     + 尽可能
       - 使用 class 或 ID 选择器
       - 元素选择器尽可能简单
       - 使用子(child)选择器代替后代(descendent)选择器
       - 不要混用 RTL 和 LTR
     + 减少样式使这一切变得简单

DEMO：

差：

table tr td ul li {color: green;}

好：

li#pass {color: green;}

差：

ul li {color: purple;}

好：

ul > li {color: purple;}

* + 不要使用 expressions
    - 慢 – 通常被这样评价
    - IE8 标准模式不支持
  + 尽量减少页面重新布局
    - 移动内容体验不好
    - 浏览器执行不必要的任务
  + 简化!

1. **优化符号解析(Symbol Resolution)**(译注：原文更详细)
   * Watch for expensive name lookups
   * 缓存多次查找的为本地变量
   * **当需要时才优化**
   * **考虑可维护性**
2. **JavaScript 代码低效**(译注：原文更详细)
   * Use the native JSON object
   * Turn large switch statements into lookups
   * Avoid property access methods
   * Minimize DOM interaction
   * Use querySelectorAll for groups
   * **当需要时才优化**
   * **考虑可维护性**
3. **HTTP 性能：**典型的访问
   * 从服务端/缓存请求
     + JS
     + CSS
     + 图片
     + HTML
   * 浏览器
     + 布局
     + 执行脚本
     + 更多下载

针对此的优化：

* + 不要拉伸图片
  + 分别合并JS/CSS
  + 合并图片(sprites)

重复访问

* + 有条件的 HTTP 请求
    - 简单的 HTTP 请求

Pragma: no-cache

* + - 时间条件

If-modified-since: date,time

* + 请求
  + GET /images/banner.jpg HTTP/1.1
  + Host: www.microsoft.com
  + If-Modified-Since:
  + Wed, 22 Feb 2006 04:15:54 GMT
  + 响应
  + HTTP/1.1 304 Not Modified
  + Content-Type: image/jpeg
  + Last-Modified:
  + Wed, 22 Feb 2006 04:15:54 GMT
  + 提供缓存内容
    - 时间条件

Expires: date,time

Max-age: #seconds

* + 请求
  + GET /images/banner.jpg HTTP/1.1
  + Host: www.microsoft.com
  + 响应
  + HTTP/1.1 200 OK
  + Content-Type: image/jpeg
  + Expires: Fri, 12 Jun 2009 02:50:50 GMT
  + 请求
  + GET /images/banner.jpg HTTP/1.1
  + 响应
  + 无响应：从缓存请求服务

要点

* + 减少请求数
    - 外联JS/CSS/图片
    - 使用缓存
  + 减小请求量
    - 使用 HTTP 压缩
    - 使用条件请求
    - 避免阻塞因素
      * 将 JS 放在 HTML 底部

Tags: [性能](http://www.99css.com/?tag=%e6%80%a7%e8%83%bd), [翻译](http://www.99css.com/?tag=%e7%bf%bb%e8%af%91)