

用事实说话！AJAX应用程序开发七宗罪 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/275/2021_2022__E7_94_A8_E4_BA_8B_E5_AE_9E_E8_c67_275165.htm

AJAX bandwagon是个好去处。它带给你更快、更高效、更强动态的应用。但它也有自身的缺陷。初一看，具备一些常识似乎就能避免这些缺陷，在一定程度上，的确如此。但从DHTML起源来看

，AJAX应用程序充满了结构性差异。不论你在应用程序开发工作中掌握了多少常识，从别人犯的错误中吸取教训也是有好处的。我们称这些错误为“七宗死罪”，但它们不能代表全部的错误。事实上，在你犯这些致命过失之前，你可能首先犯了一些较轻的错误。因此我们从这里着手。这是每个人都可能犯的错误。这些错误是多么普遍，通过Google搜索一下，你就可以发现大部分的错误。七宗轻罪 1.滥用Back按钮这是每个人都会犯的错误。Back按钮在很多网页程序中已经成为用户的期望。很多AJAX研发新手在开发AJAX应用程序时，出于多种原因都敏捷地使用着Back按钮。首先

，JavaScript对于它来说不是最友好的语言。其次，AJAX设计中需要一种全新的思考方式。对于AJAX研发新手“后退”显然不是最好的选择。“后退”是一种你更新页面，或更常见的，你需要在特定情况下进行“撤销”时才用到的功能。在进行编码前应认识到这些，或者你可能重复做功。2.忘记告诉用户当前发生的状况AJAX工作原理中的一部分是其不使用常规的网页用户界面加载程序。因此，你需要明确设计一些可视的提示，使用户了解正在发生的状况。3.忽视链接这也是AJAX的标准失误：漏做了程序外部用户可以剪贴的URL链

接。我们都曾经多少次复制过URL链接然后将其发送给别人? 当你在使用AJAX时, 提供给你的用户有用URL链接的唯一方法是, 手动提供给他们。为什么? 因为在AJAX程序中, 服务器不提供JavaScript动态生成的这个页面! 不要忽视你的用户可能感兴趣的这个网络应用中最普通的功能。既然服务器不支持, 那你花点功夫为用户提供URLs。

4. 用内容控制替代控制

如果你正在寻找动态的内容控制, 那么对传统的客户服务器交互作用的突破对你来说可是件好事。但这也是一项罪过: 在重写一个页面的某精确位置上的内容以调整用户的交互式体验时确实确实能够很好的控制, 但这也将使你的页面不完整。通常, 我们专注于处理页面的某一部分, 而忘记服务器不刷新页面。这会导致页面凌乱, 用户体验降低, 当他们察看页面时可能看到过时的页面! 把你的注意力放在整个页面。确保出现动态页面的内容都得到更新。

5. 累死蜘蛛

AJAX的优势在于无需重装就可以提供给页面的大量的文本。AJAX的缺陷在于无需重装就可以提供给页面的大量的文本。如果应用被设置为对搜索引擎友好, 那么, 你能够想象会出现什么情况。无论页面中出现了什么, 请务必在最上面植入足够稳定的文本, 为蜘蛛们去玩耍吧。产生乱码文本AJAX不支持很多字符集。这不是涉及生死的局限性, 但忘记它能够产生真的问题。最基本的字符集是UTF-8。不论JavaScript发送什么, 别忘记正确地编码, 并且根据内容设置服务器端的字符集。没有为使用不支持JavaScript的浏览器的用户提供提示--有些浏览器不支持JavaScript, 用户一时不能明白出了什么状况。请给他们提供提示。实话实说, 其中大部分是常识性问题。真正的问题都很容易让人忽视的。造成内存泄漏任何长期从事开发

工作的人都知道循环引用，并了解其给内存管理带来的危害。AJAX所使用的JavaScript是内存管理的语言。这意味着JavaScript具有内置的信息包收集功能，因此能够抽取不再有引用路径使用的变量并重新分配这些变量所使用的内存。作为基本工作原理这很好，但是在模型对象和察看元素之间互相引用时，由于这些循环引用，你就不能依靠这个功能来实现你的内存使用最优化。从原则上讲，对象为零，则元素为零，但是如果这时从元素到对象的向后引用，那么信息包搜集器不会动这些对象。现在，问题出来了：在文件对象模型中，任何文件树中的DOM节点都可能被树中的其它元素引用，不论其是否被其他对象所引用！因此任何在信息包收集器中经过标注的被DOM节点向后引用的对象，在这一方向必须为空，否则其内存就会一直处于已分配状态。不了解“异步”的含义--异步很容易让不熟悉它的用户感到紧张。但是如果您为这些用户所设计的网页应用程序属于桌面应用程序，那么他们肯定不会感到不安。这是一个至关重要的设计点。大部分网页应用功能与桌面副本非常类似。但是在网页应用中，用户期望这种虚幻的特征导致他们截然不同。用户在与网页浏览器打交道时会带有非常不同的偏见和期望，而对于桌面应用中他们并没有这样的行为。因此，尽管页面与服务器之间频繁的响应会非常好，非常高效，页面能够同时对自身进行修订，但是这将会使用户头晕眼花。因此，您需要遵守两条守则，要考虑到进入用户视觉范围内的每个变化：如果对于用户来说不是很迫切的更新，那么要使升级更为温和，不会转移。如果更新对于用户与应用的交互非常重要，那么更新要清楚而明显。使服务器在黑暗中用户端与服务器的交

流减少是一大问题，而以前并非如此。在以前，服务器端的应用了解全部情况并且能够看到全部状况：每个例外，每次重新加载，每个事件多能被看到并记录下来，当然服务器也知道客户端是什么状况，因为服务器会记录下屏幕上显示的一切。在AJAX应用程序中，问题不是这样。当有事件发生时，这些事件是与服务器相互独立的，也就是说，当客户端出现问题时，服务器端并不会马上知道。在某个位置发现和记录客户端发生的事件以及例外，使服务器能够尽快追踪需要干涉的问题。用GET偷懒GET用于重新找回数据.POST用于对GET设置。不要在不适当的时候使用GET，即使你认为这样做没有危害。GET操作改变状态，改变状态的链接会令用户感到困惑.大部分都认为链接的作用是导航，而不是功能。不兼容数据类型--JavaScript不是.net Framework框架中的一部分。尽管这的确令人伤心，但这呈现给我们一个我们可能会碰到的问题：确定JavaScript能够理解其运行平台上的数据类型，反之，对于.NET或其他都是如此。可能会有多种转换器，你需要把它们找出来。例如，Ajax.NET Pro资料库，提供能够转换.NET和JavaScript对象符号的转换器。一些应用程序不知道何时关闭--无需刷新页面的内容动态生成如果没有关闭时间将会非常糟糕。你见过多少比美国国会议事录还长的网页?如果网页无限延长无疑会是用户的噩梦，只要想想用户会怎么看待永不停止的应用程序就知道了。让您的网络应用具有动态效果，但是一定要在可行的限度之内。保持你的JavaScript远离你的DOM请记住AJAX建立在模型-视觉-控制(Model-View-Controller)结构之上。请认真地对待这点。JavaScript属于模型层面，DOM属于视觉层面，而控制器是

他们的婚姻顾问。保证让你的网络文件独立于JavaScript之外(这样有利于不支持JavaScript的用户)--除了当内容自身只在用户使用JavaScript时才有意义及可操作性。在这种情况下，用JavaScript创建内容。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com