|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Kupu：开源的Web富文本编辑器** | | | | [chenkw](mailto:chenkaiwang@yahoo.com.cn) 翻译   更新：2005-05-05 22:32:12  版本: 1.0 |  |  | |
| |  | | --- | | 原文：[Rich Web Text Editing with Kupu](http://www.onlamp.com/lpt/a/5792) 作者：Robert Jones 原文发表时间：04/28/2005 译者：chenkw  　　我编写web应用程序也有些年头了，但一直苦于找不到一个称心的方法让用户输入并提交任意文本块。HTML表单的TEXTAREA标记可以处理普通文本，但我的应用程序大多涉及格式更丰富的内容。一个获得富文本（rich text）的办法是，让用户在字处理器中编辑内容然后将文档提交到服务器，但文档提交后便无法随意处理。原来我最需要的编辑器，就是一个可以嵌入到任何网页中的WYSIWG（What You See Is What You Get，所见即所得）编辑器。天哪，我终于找到了我的理想，它就是----[Kupu](http://kupu.oscom.org/)。 　　Kupu是一个开源的JavaScript应用程序，它实现了一个灵活的、全功能（full-featured）的HTML编辑器，不需要任何插件就可以在网页中运行。它的主要应用是作为内容管理系统（如Zope、Plone）的嵌入式编辑器，提供创建网页的功能。它的设计非常灵活，你几乎可以毫不费力地将其嵌入到任何web应用程序中。 　　当然，你首先需要明白了它的工作原理。和其他一些优秀的软件一样，Kupu只提供了有限的文档，而且源代码中的注释也少得可怜。我可以理解为什么会这样。开发人员总是将希望有限的时间用于编写新代码，而非撰写文档。然而，这大大降低了其工作对社区的影响。开源软件除了具有可获得性（availability），还应该具有易用性（accessibility）。 　　我撰写本文，希望提高Kupu的易用性，并演示将它嵌入你的应用程序是一个多么简单的过程。它是一个非常优秀的程序，应该被更多的人接受。  **JavaScript**  　　Kupu由Paul Everitt、Guido Wesdorp、Philipp von Weitershausen等人编写，它是JavaScript程序设计的一个精彩示例。用这种语言编一个最简单的应用程序对我来说都是一种考验，Kupu让我惊叹不已。 　　由于人们意识到XMLHttpRequest的价值，JavaScript目前似乎正处于一个复兴阶段。Google Suggest以及最近给人深刻印象的Google Maps都采用这种技术，Kupu也不例外。XMLHttpRequest是浏览器和服务器之间的一种后台通信方式，这样就不需要在每次数据交换时刷新整个页面。更多内容请参考Drew McLellan的文章[Very Dynamic Web Interfaces](http://www.xml.com/pub/a/2005/02/09/xml-http-request.html)。此外Joel Webber在其blog里撰文剖析了[Google Maps的实现技术](http://jgwebber.blogspot.com/2005/02/mapping-google.html)。 　　不幸的是，使用JavaScript，特别是和XMLHttpRequest一起，会让你卷入浏览器兼容性的漩涡。Kupu目前仅在Mozilla 1.3.1、Internet Explorer 5.5和Netscape Navigator 7.1或更高版本上测试通过。它尚不能在Konqueror、Operah和Safari上正确运行，但新版本有望支持这些浏览器。  **安装Kupu**  　　基本安装是非常简单的。你只需要到 [http://kupu.oscom.org/download/](http://kupu.oscom.org/download/ ) 下载压缩包(450K)，然后解压到网站的一个可访问的目录下就行了。它将创建kupu/目录，其下包含一些文件和子目录。你不需要进行make或者其他类似的操作。但我建议你先不要看doc/子目录下的内容。 　　在浏览器中输入<your path>/kupu/common/kupu.html（<your path>指kupu所在的路径，你需要补充完整这个URL）就可以看到编辑器界面了。你也可以访问我的[Kupu编辑器测试页面](http://www.craic.com/oreilly/kupu/kupu/common/kupu.html)。结果如图1所示。  http://www.javaresearch.org/members/chenkw/kupu_1.jpg  　　图1 Kupu示例页面  　　该页面由三部分组成：工具栏和左右面板。顶部的工具栏包含了编辑器标准功能按钮。右侧面板由一些块（block）组成，每个块为相应功能提供额外信息。左侧面板就是你正在编辑的文档。在本示例中，左侧面板预装入了简要介绍Kupu设计理念的文档。 　　在左侧面板内某处单击然后就可以开始输入。如果你使用Firefox浏览器，而上述操作没有反应，那么请按F7键。按F7可以启动“插入浏览”（caret browsing），修从而复这个问题。这个界面有些怪异，但还是挺直观的。选择一些文本然后改变其颜色和格式。创建一些链接。插入源为某URL的图片并改变其大小。最好再插入一个表格并增删其行列。这个演示程序并不支持保存操作，它将触发一个错误。我将在下文介绍如何修复这个错误。 　　Kupu目前的版本也许不能满足你对它作为WYSIWYG编辑器的全部期望。例如，你无法改变字体，只能插入Web上的图片。还有一些功能是不正常的，至少在我的机器上如此。但其兼容性对多数应用程序来说已经绰绰有余了。若你不想涉及插件或者Java applet，Kupu绝对是一个让人振奋的成就。它仅仅是一个包含了JavaScript的网页。 　　Kupu的实际价值在于它可以嵌入到其他应用程序中。嵌入方式有两种。最简单的用法是使用POST方法将数据作为一个CGI脚本参数值发送到服务器。另一个用法稍有点复杂，但它是Kupu小组的选择，那就是使用PUT方法和一个简单的CGI脚本将数据发送到web站点的一个文件中。  **使用Kupu**  　　嵌入Kupu利用其功能是以在页面中包含许多JavaScript文件为代价的。即使像common/kupu.html这样的基本页面中也包含了让人望而生畏的代码。更要命的是，包含文件的链接都是相对的，这会让你很头疼，若你想在Kupu源程序结构上花些时间的话。你其实不想碰这些代码，但是你需要建立自定义内容和Kupu之间的关联并让一切正常运行。幸运的是，你可以使用<base>标记，这是一个简单办法。 　　将Kupu发布版放到你的站点树（web tree）的某子目录下。然后在站点树的其他地方为你的网页建立一个目录。将模板页（kupu/common/kupu.html or kupu/common/kupuform.html）复制过去，并在页面的<head>段中添加<base>标记，指向发布版的common目录。例如：  <head> <base href="http://www.craic.com/oreilly/kupu/kupu/common/"> </head>  　　浏览器将在所有相对链接前附加该URL。注意，它必须是一个包括主机名的完整URL路径；这个规则适用于该页面中的所有相对URL，而不仅是Kupu用到的。在插入图片、样式表、或链向本站其他页面时，务必注意<base>标记的影响。  **以PUT方法使用Kupu**  　　整合Kupu的第一种方法是使用PUT方法上传文件到web服务器。PUT虽然也是http协议的一种方法，但在知名度上无法与GET和POST相提并论。PUT允许上传完整的文件，然后将其置入站点树中。当然，你也可以使用HTML表单和CGI脚本来完成同样任务，但PUT的目的就是为了简化这种过程。[《Publishing Pages with PUT》](http://www.apacheweek.com/features/put)一文提供了一些背景知识，该文发表于1997年。 　　也许大家很少使用PUT方法的主要原因就是其固有的不安全性了。PUT方法有允许任何人提交任何文件到服务器的安全隐患。你需要仔细配置服务器，避开任何潜在的安全漏洞。任何一个小小的失误都可能给攻击者创造莫大机会。相反，你可以考虑使用POST方法打造一个封闭、安全的系统，通过CGI脚本与之交互。这将要求更多编程工作，但其风险更易于管理。 　　一个PUT请求将发送两行头信息，然后才是请求内容。CONTENT\_LENGTH告诉服务器发送的数据长度。PATH\_TRANSLATED指定数据写入哪个文件。你可以将其视为浏览器从服务器获取一个页面的逆过程。Apache httpd配置默认接受PUT请求，但你还需要配置一个CGI脚本来处理接收的文件。下面是CGI脚本配置示例，你只需要根据你的情况替换掉脚本名就可以了：  Script PUT /cgi-bin/kupu/handle\_put.cgi  　　以下是一个简单的CGI脚本实现。注意，该脚本中没有考虑安全检查。任何人都可以向运行了该脚本的站点上传文件。在公网站点上安装该脚本时务必添加安全限制。切记！（本示例并非我的站点上运行的应用程序）  #!/usr/bin/perl -w # handle\_put.cgi # Basic PUT Handling routine with NO SECURITY!  if($ENV{'REQUEST\_METHOD'} ne 'PUT') {   errorMsg("Request method is not PUT"); }  my $filename = $ENV{'PATH\_TRANSLATED'};  if(not $filename) {   errorMsg("PATH\_TRANSLATED was empty"); }  my $length = $ENV{'CONTENT\_LENGTH'};   if(not $length) {   errorMsg("CONTENT\_LENGTH was empty"); }  # Add Security Checks Here! Limit access to certain # directories and limit size and/or type of file.  E.g.  if($length > 100000) {   errorMsg("CONTENT\_LENGTH is too large"); }  # Read in the uploaded data my $content = '';  my $nread = read(STDIN, $content, $length); # Make the output more readable by adding newlines $content =~ s/\>\</\>\n\</g> $filename" || errorMsg("Unable to open $filename"); print OUT $content; close OUT;  # The 204 code signals the transfer was OK but does not  # update the current page - so you stay in the editor print qq[Status: 204\n]; print qq[Content-type: text/html\n\n]; print $content; exit;  sub errorMsg {   my $msg = shift;   print qq[Content-type: text/html\n\n];   print qq[<html><head><title>Error</title></head>\n];   print qq[<body>\nError: $msg<br>\n</body></html>\n];   exit; }  　　OK，服务器已经就绪。现在可以设置Kupu使用PUT了。你不需要在网页中指定CGI脚本，因为Apache配置文件中已经预先做好了配置。你所需要做的只是在页面中指定服务器上用于写入数据的文件。将kupu/common/kupu.html复制到你的目录，按照上文的说明添加<base>标记，这样你就可以从发布版目录中装入JavaScript库了。 　　看一眼这个页面的源代码。300多行看了就晕，要命的是其中不乏非HTML内容。幸运的是，现在只需要其中两行，一行在在头部，另一行在尾部。现在并不需要触动其他内容。在kupu.html的源代码尾部指定页面装入编辑器。这里，我使用了一个空白页面（kupublank.html）。  <iframe id="kupu-editor" frameborder="0" src="kupublank.html" scrolling="auto">  　　接着指定服务器上写入数据的文件。查找<kupuconfig>，从上往下数大约30行左右就是：  <kupuconfig> <dst>http://www.craic.com/oreilly/kupu/no\_such\_file.html</dst>  　　将<dst>标记中的内容替换为目标文件的URL。web服务器必须拥有该目录的写权限。注意，你指定的是单个文件。在这些示例中，你无法在运行时定义文件名。 　　浏览器装入刚配置的编辑器后，输入一些文本，编辑其格式。若想保存文件，只需要单击Kupu工具栏上的磁盘图标。结果如图2所示，好像没什么变化：  http://www.javaresearch.org/members/chenkw/kupu_2.jpg  　　图2 编辑文本  　　现在转到刚才被定义为目标的URL。你将看到和刚才的编辑时完全一样的结果。当然，目标页面是没有工具栏的，如图3所示：  http://www.javaresearch.org/members/chenkw/kupu_3.jpg  　　图3 新创建的页面  　　这是一个简单示例，你马上就可以发现一些缺点。例如，你每次保存都会覆盖目标文件。虽然Kupu开发人员偏爱PUT方法，但若将编辑器嵌入到自己的应用程序，特别是CMS（内容管理系统）中，你可能发现在表单中使用编辑器将提供更多的控制。  **在HTML表单中使用Kupu**  　　第二种方法是使用表单包装Kupu代码。这需要更多的努力，因为你需要一个CGI脚本来处理数据。不过，相对PUT技术来说，它提供更多的数据控制。PUT只是简单地对编辑器中的东西作了一下镜像。 　　下面介绍一个HTML表单示例（kupu/common/kupuform.html）。该页在浏览器中的显示结果和上一个示例（kupu.html）的一样。页面源代码非常相似，但关键之处略有不同。从上到下数，在第20行左右，你可以找到一个<form>标记：  <form action="http://debris.demon.nl/printpost" method="POST">  　　这个URL定义了处理上传文本的脚本。将它替换为你的脚本。从下往上数大约8行，你可以找到一个<iframe>标记：  <iframe id="kupu-editor" frameborder="0"        src="fulldoc.html" scrolling="auto">  　　src属性指定了启动编辑器时装载的文档。本例的装载文档是fulldoc.html。 　　若想使用这个表单，只需要改变这些行。首先，将kupuform.html复制到你的目录，并像上例一样添加一个<base>标记来访问JavaScript函数库。在<iframe>标记中指定一个空白文件kupublank.html，这样编辑器启动时就显示空白文档。 　　你需要在服务器上运行一个CGI脚本来处理提交的数据。下面是一个简单的脚本（kupu\_echo.cgi），它读入表单传来的数据（该数据随kupu参数传递），然后直接将该数据返回给用户。Kupu生成的HTML是没有换行的，这给查看源代码带来不便，所以示例脚本中在每个标记后面插入一个换行符。  #!/usr/bin/perl -w # kupu\_echo.cgi  use CGI; my $cgi = CGI->new();  my $text = $cgi->param('kupu'); $text =~ s/\>/\>\n/g;  print $cgi->header(); print $text;  　　将该脚本放到站点的cgi-bin/目录中，将赋予可执行权限：chmod a+x kupu\_echo.cgi，然后将其URL替换到<form>标记。这两个标记改变后应该和下面的类似：  <form action="http://www.craic.com/cgi-bin/kupu/kupu\_echo.cgi" method="POST">  ...  <iframe id="kupu-editor" frameborder="0"        src="kupublank.html" scrolling="auto">  　　在浏览器里打开kupuform.html，你会看到编辑器的左面板为空白。随便敲入一些文本，修改其格式。单击工具栏中的磁盘图标将编辑结果上传到CGI脚本。如果你想即刻获得满足，那么请访问我的站点上的演示页面。http://www.craic.com/oreilly/kupu/kupuform.html 　　这个过程并不是很麻烦。你只需要改变模板页面的两行代码，然后编写一个简单的CGI脚本。最重要的是，你连一行JavaScript都不需要触动。当然，这只是一个简单示例，但它帮助你度过了学习曲线的第一个阶段。  **一个规模较大的示例**  　　我编写了一个基本的网志程序（blogging application），在表单中嵌入Kupu来创建新纪录，这是一个稍微有点复杂的例子。 　　它由一个包含了所有Kupu代码的HTML模板文件，以及一个处理记录提交并显示所有记录链接的CGI脚本。这是一个很简单的网志程序，但它演示了如何将编辑器整合到一个包含其他控件的表单。它还介绍了如何修改Kupu模板。 　　图4是该程序的截图，图上显示已经加入了几条记录。  http://www.javaresearch.org/members/chenkw/kupu_4.jpg  　　图4 Kupu作为weblog编辑器  　　除了编辑器之外，示例中还包含一个普通的表单控件<input>，用于输入该纪录的标题。使用表单来整合Kupu，你在充分利用编辑器功能的同时，还可以加入自己的表单控件。我对模板页作了一些修改，对Kupu的工具栏作了一下调整。你可以到Kupu的主页上去看看不同的CMS系统的截图，它们都针对自己的应用作了一些修改。如果你真想那么做，那么你需要理解这个JavaScript函数库，而且明白它是如何使用XML进行配置的。不过，这已经大大超出本文的范围了。 　　本程序的源代码太长，就不包含在这里了，你可以到[我的Kupu站点](http://www.craic.com/oreilly/kupu/)下载。 　　查看[我的Kupu模板页时](http://www.craic.com/oreilly/kupu/blog_template.html)，你会发现一些形如<!-- ### -->的注释，这表明在该处我修改了原始的kupuform.html模板页。CGI脚本中替换了模板中的Perl风格的变量，如$srcFile。（的确，用PHP将更简单！）我也包含了[Perl CGI脚本的源代码](http://www.craic.com/oreilly/kupu/kupu_blog_cgi.html)。 　　你也可以在我的站点上体验一下[基于Kupu的网志系统](http://www.craic.com/cgi-bin/kupu/kupu_blog.cgi)。（我已经删掉了所有记录。）  **总结**  　　本文的示例仅仅涉及Kupu的皮毛，但它为你自行体验Kupu或理解Kupu中的XML和JavaScript铺设了道路。这些程序使用了XMLHttpRequest。XMLHttpRequest逐渐在web开发中受到重视，它是一项值得关注的技术。   　　Robet Jones经营着Craic Computing，这是一家坐落于西雅图的小型生物仪器公司，为生物技术行业提供高级软件和数据分析服务。多年的程序设计经验使他成了名副其实的分子生物学家。 | |