

Edición de páginas Web con FCKeditor

FÁCIL ARREGLO

Las páginas Web modestas suelen tener información desactualizada debido a la falta de tiempo del webmaster para mantener todas las páginas al día. Incluso algo tan sencillo como una corrección ortográfica puede llevar días. FCKeditor libera al webmaster dejando que el usuario se encargue. **POR JAMES MOHR**

Además de mantener páginas Web personales, también soy webmaster voluntario de un club deportivo local. Una queja común en el club es que nadie se toma su tiempo en proporcionar información actualizada. Algunas de las secciones de la página no proporcionan ninguna información en absoluto, simplemente porque nadie ha tenido tiempo de mantener las páginas actualizadas. Aún teniendo el conocimiento técnico para mantener las páginas, simplemente no disponía de tiempo material para mantener cada una de ellas. Necesitaba una manera de que a los miembros

del club les resultase sencillo poner la información online ellos mismos.

La manera más sencilla para que un novato mantenga una página Web es escribir el texto con algún editor de textos y entonces subir los archivos con un cliente de FTP. Desafortunadamente, la mayor parte de la gente de nuestro club tenía problemas incluso con esto. Por tanto, para conseguir que el mayor número de personas proporcionase información actualizada, se requería un mecanismo lo más sencillo posible para actualizarla. Se creó una base de datos para noticias y eventos, permitiendo a los usuarios añadir información en formularios basados en Web. Sin embargo, no encontraba una solución fácil para permitir a los usuarios modificar el contenido de las páginas. Hasta que encontré FCKeditor.

FCKeditor es un editor de texto que reside en el servidor y es accesible a través de un navegador Web convencional. Al integrar FCKeditor en la página Web, se crean los medios adecuados para que los usuarios editen sus propias páginas Web.

Incluso los usuarios más inexpertos se sintieron cómodos al trabajar con el editor de texto, por lo que FCKeditor proporcionó la solución que era incluso más sencilla que un entorno de edición basado en Zope o en Wiki. El reto, por supuesto, era entender cómo integrar FCKeditor en una página Web. En este artículo, des-

cribimos cómo editar online con FCKeditor.

Editor Completo

FCKeditor (véase Figura 1) tiene todas las funcionalidades que la mayoría de la gente necesita en un procesador de textos, incluyendo una completa barra de herramientas. Bajo esta barra encontramos el área de texto WYSIWYG. FCKeditor funciona con casi todos los navegadores en cualquier entorno.

En el servidor, FCKeditor proporciona soporte para los lenguajes de desarrollo Web más comunes, como PHP, Perl, ColdFusion, Java e incluso ASP.Net. Esto permite una integración perfecta con nuestra página ya existente.

FCKeditor está disponible en una docena de idiomas. Por defecto, el idioma se detecta a través de la configuración de nuestro navegador, aunque es fácil configurar un idioma específico en el archivo de configuración.

La mayor desventaja de FCKeditor es la falta de documentación. Lo poco que existe hace repetidas referencias a los ejemplos proporcionados (véase el cuadro "Archivos de Ejemplo"). Aunque los ejemplos nos ayudan a ver lo que hace el programa, este método de documentar requerirá que buceemos a través de demasiado código para descubrir cómo personalizar el programa.

Instalación

Podemos encontrar la última versión de FCKeditor en su página Web [1]. En el momento de escribir estas líneas, la última versión disponible era la Version 2 Release Candidate 2. Deberíamos conseguir sin duda la RC2 si estamos ejecutando un servidor Apache bajo Linux, pues existían algunos problemas con la RC1.

Una vez hayamos obtenido el paquete de instalación, creamos en primer lugar un directorio *FCKeditor* bajo la raíz de documentos de nuestro servidor Web y



descomprimos el paquete dentro de este directorio. Es posible cambiar la ubicación, pero si lo hacemos, tendremos que realizar demasiados cambios a los archivos de configuración (véase el cuadro titulado “Fijando la Ubicación”).

Cada uno de los cuatro ejemplos de PHP proporcionados en el paquete (véase el cuadro titulado “Archivos de Ejemplo”) muestran los diferentes aspectos del editor. Por ejemplo, uno de estos ejemplos nos permite elegir la barra de herramientas, otro el idioma, etc. Cada uno de los componentes puede añadirse a cualquier página, de manera que podemos construir un editor personalizado que nos permita seleccionar cualquiera de las opciones.

Hemos elegido uno de los archivos de muestra y lo hemos utilizado como punto de arranque para nuestra configuración. El ejemplo que hemos elegido parece ser el de más sencilla adaptación de la configuración. En nuestro caso, no vamos a querer que los usuarios puedan toquetear en la configuración del editor ya que tratamos de mantener el sistema lo más sencillo posible. Por tanto, hemos eliminado todo el código extra relacionado con la configuración del editor. También hemos deshabilitado la autodetección del idioma. Si la página está en español, fijaremos el lenguaje por defecto a español:

```
$oFCKEditor->Config 2
['AutoDetectLanguage'] =>
false ;
$oFCKEditor->Config 2
['DefaultLanguage'] = 'es' ;
```

Si usted está configurando FCKEditor para su propio entorno, elija un archivo de ejemplo que se adapte a su situación y adapte la configuración a sus necesidades.

Carga de Textos en el Editor

El primer paso para integrar FCKEditor en una página Web existente es definir un medio para cargar el texto en el editor. El atributo *Value* contiene el texto que aparezca en la ventana del editor en ese momento. Podemos definir el texto a aparecer en la ventana del editor simplemente con configurar el atributo *Value*:

```
$oFCKEditor->Value = 2
'Tu texto' ;
```

Podemos fijar este valor como cadena vacía para empezar con el navegador vacío, pero una de las mejores características de FCKEditor es la capacidad de editar páginas ya existentes. Por lo tanto, necesitamos averiguar cómo conseguir introducir el contenido de un archivo existente en el editor. La manera más sencilla es usar la función de PHP *function file_get_contents()*:

```
$oFCKEditor->Value = 2
file_get_contents2
("ruta_a_fichero") ;
```

Advertencia: *file_get_contents()* puede cargar una URL o fichero directamente. La URL puede contener includes por el lado del servidor, marcos o cualquier otro tipo de cosas. Por lo tanto, el archivo que termine en nuestro navegador puede no ser el que realmente queremos editar. El método más seguro, en nuestra opinión, es evitar URLs. Antes de pasar el nombre de archivo a *file_get_contents()*, verificamos que estamos cargando un archivo local.

Incluso usando este método, los contenidos son bastante estáticos. Necesitamos una manera de pasar dinámicamente el nombre de archivo para poder cargarlo. Podemos hacer esto de manera sencilla pasando el nombre de archivo como parte de una consulta:

```
http://linux.local/FCKEditor2
/jimmoedit.php?filename=2
index.html
```

Ahora, dentro del script del editor (*jimmoedit.php* en este ejemplo), leemos la variable de nombre de archivo:

```
$filename = 2
$_GET['filename'];
```

Debemos tener cuidado aquí, ya que es posible crear una consulta que haga

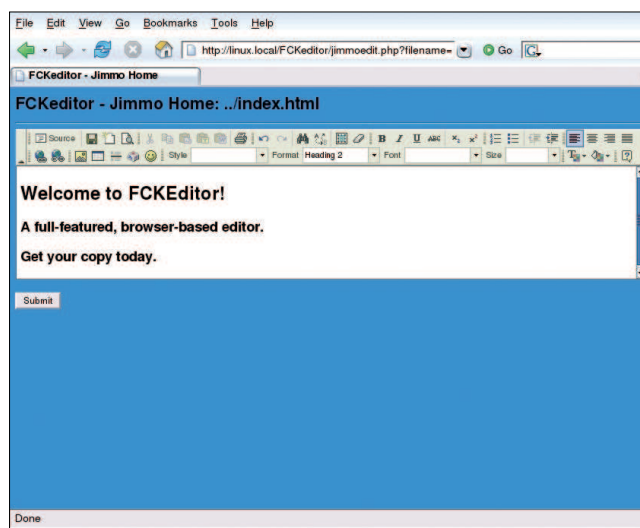


Figura 1: Ventana principal de FCKEditor.

mucho más que simplemente cargar un archivo en nuestra página web. Por ejemplo, un usuario malicioso podría especificar el archivo *.htaccess* o cualquier otro archivo sensible de nuestro sistema. Por lo tanto, deberíamos verificar siempre que el valor escrito en *\$filename* es válido y lógico antes de cargar el archivo.

Acceso al Editor

A menudo, tener usuarios tecleando una cadena de consulta a mano es demasiado trabajoso, por lo que necesitamos una manera más fácil de pasar el archivo. De cómo lo hagamos depende cómo se generará la página, quién está autorizado a editarla, etc. En la página que yo administro, las páginas se generan usando PHP,

Archivos de Ejemplo

Encontraremos unos cuantos ejemplos para cada lenguaje de desarrollo en el directorio *_samples*. Nosotros hemos usado PHP, por lo que hemos utilizado los archivos del directorio *php* y hemos copiado uno de los archivos de ejemplo desde *_samples/php* hasta el directorio raíz de *FCKEditor*. Luego hemos copiado los archivos *sampleposteddata.php* y *samples/sample.css*. En nuestro caso, hemos usado *sample2.php* y le hemos llamado *jimmoedit.php* (por motivos de simplicidad). Este es el archivo que necesitamos cargar, y donde tenemos que empezar a hacer los primeros cambios. También hemos renombrado el archivo *css* como *jimmo.css* y hemos cambiado el archivo *jimmoedit.php* consecuentemente.

por lo que puedo verificar si el usuario está conectado o no. Si lo está, el enlace "Edite esta página" contendrá la URL completa del editor, incluyendo la ruta relativa hasta el archivo adecuado.

En otra página en la que no se usa PHP ni otros lenguajes de script, las cosas se tornan un poco más complejas, ya que no hay manera directa de ver si el usuario está conectado. La solución es poner el editor en un directorio protegido con una contraseña en el servidor usando la autenticación básica estándar de Apache. Podemos determinar el nombre del usuario conectado así:

```
$current_user = ?
$_SERVER["PHP_AUTH_USER"];
```

Tras verificar que el usuario está conectado, podemos determinar el grupo al que pertenece el usuario. Esto es necesario debido a que el sistema determina los archivos que puede editar el usuario basándose en las filiaciones a grupos del usuario. Por ejemplo, una persona en el grupo voleibol podrá editar únicamente páginas dentro del directorio voleibol.

Como los nombres de directorio son los mismos que los nombres de grupo, es sencillo averiguar el directorio correcto. Podemos entonces usar varias funciones de PHP para buscar el directorio y crear una lista HTML de archivos disponibles, generando los enlaces correspondientes que pasan los nombres de los archivos a editar. Una ventaja clave de este método es que podemos filtrar cualquier archivo al que los usuarios no deban tener acceso o que no deban editar.

Uso del Editor

El editor en sí se comporta de la misma manera que cualquier otro editor. Por ejemplo, seleccionando texto y pulsando el botón adecuado, podemos cambiar el texto a negrita o cursiva, cambiar el tamaño de fuente y demás. Esto no sólo aplica las etiquetas estándar. FCKEditor inserta distintos estilos en la etiqueta. El editor incluso proporciona un modo de código fuente, que nos permite editar el código HTML directamente.

Una característica que realmente impresiona es el editor de tablas. Cuando pulsamos sobre el botón, aparece la ventana emergente mostrada en la Figura 2. Como en otros editores de HTML, podemos definir las dimensiones de la tabla

(tanto número de celdas como píxeles), bordes, relleno de las celdas, etc.

Una vez que se inserta la tabla, podemos cambiar un buen número de propiedades. Si seleccionamos la tabla pulsando con el botón izquierdo, podemos arrastrar los lados o las esquinas para cambiar el tamaño. Pulsando con el botón derecho del ratón, podremos insertar o eliminar filas, columnas e incluso celdas. Podemos también dividir filas o columnas, e incluso cambiar las propiedades de la tabla entera. Otras funcionalidades nos permiten crear listas, añadir enlaces e imágenes o cambiar los colores de fondo.

Navegación por el Servidor

FCKEditor proporciona un mecanismo de navegación por los archivos de servidor. Esta exploración se complementa con el uso de "conectores" para varios lenguajes. Por ejemplo, el conector y el código para el editor de PHP se sitúan en *editor/filemanager/browser/default/connectors/php*.

Definimos qué conector debe usar el navegador en el archivo *fckconfig.js*. Existen dos variables en el archivo *fckconfig.js* que tendremos que fijar: *FCKConfig.LinkBrowserURL* y *FCKConfig.ImageBrowserURL*. Por defecto, estas variables apuntan al conector ASP, por lo que también tendremos que

quitar la marca de comentario al conector PHP.

Por defecto FCKEditor busca los archivos en diferentes directorios en función de su tipo. Por ejemplo, podríamos querer un directorio especial para archivos y otro para imágenes. Para que la ruta no dependa del tipo de recurso, tendremos que cambiar el archivo *connectors/php/io.php*. Primero, en la función *GetUrlFromPath()* cambiamos la línea:

```
return $GLOBALSU
["UserFilesPath"] ?
. $resourceType . $folderPath ;
```

por lo siguiente

```
return $GLOBALSU
["UserFilesPath"] ?
. $folderPath ;
```

Luego, en la función *ServerMapFolder()* cambiamos la línea

```
$sResourceTypePath = ?
$GLOBALS["UserFilesDirectory"] ?
. $resourceType . '/' ;
```

por lo siguiente

```
$sResourceTypePath = ?
$GLOBALSU
```

Fijando la Ubicación

Aunque las instrucciones señalan que descomprimamos el paquete en el directorio *FCKEditor* bajo la raíz de archivos de nuestro servidor Web, en principio podemos situar el paquete en cualquier lugar. Para cambiar la ubicación, tendremos que configurar la propiedad *BasePath* del objeto editor. Por ejemplo, en el archivo *jimmoedit.php* encontraremos una entrada como la siguiente:

```
$oFCKeditor->BasePath = ?
"/FCKeditor/";
```

Nótese que antes de esta entrada, ya hemos creado una nueva instancia del editor. Por tanto aquí estamos especificando el atributo *BasePath* para esa instancia (desde el momento en que *\$oFCKeditor* es la instancia actual del objeto editor).

Si echamos un vistazo a los comentarios del ejemplo, veremos que *BasePath* se determina con base en el directorio *_samples*. Tanto si fijamos la propiedad aquí o dentro del archivo, esto crea la

clase editor. Para PHP, esto sería el archivo *fckeditor.php*. (Además de crear el objeto *FCKEditor*, el archivo *fckeditor.php* es también responsable de crear el HTML que define el marco que es el editor en sí mismo, además del código necesario para crear la barra de herramientas, y todo lo demás). Dentro de la función constructora para la clase editor, fijaremos el atributo *BasePath* (así como otros atributos útiles). En este ejemplo, al estar la ruta incrustada en el constructor, deberíamos comentar o eliminar el código extra dentro del archivo *jimmoedit.php*.

A continuación, tenemos que cambiar la ruta hasta el editor real. Esta ubicación se incluye al principio del archivo *jimmoedit.php* usando la ruta relativa:

```
include("../..../fckeditor.php") ;
```

Al estar el archivo editor ahora en el directorio raíz de FCKEditor (el mismo que el archivo *fckeditor.php*), tendremos que eliminar la ruta relativa.

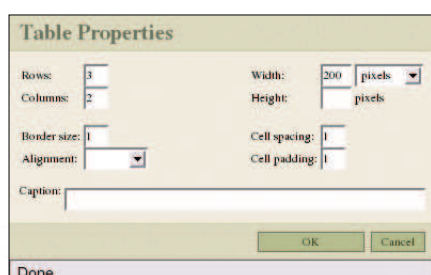


Figura 2: Configuración de tablas HTML.

```
["UserFilesDirectory"] . '/' ;
```

A continuación tendremos que cambiar la ruta de los archivos de usuario. Esto se hace en el archivo *connector.php*. Cerca del comienzo del archivo, encontraremos una referencia que tiene una pinta tal que:

```
$GLOBALS["UserFilesPath"] =  
'/UserFiles/' ;
```

Cambiamos la ubicación a un directorio de nuestra elección pariente a la raíz de documentos de nuestro servidor.

Guardado del Archivo Editado

Una vez hayamos terminado de editar el archivo, pulsamos en el botón *Submit*, con el que enviamos el texto a la acción definida (igual que cualquier formulario Web). Por defecto es *sampleposteddata.php*, que nosotros hemos renombrado como *posteddata.php*.

Si observamos el archivo de ejemplo de envío, todo lo que está haciendo es tomar el contenido del área de entrada del formulario y mostrarlo. Sin embargo, no hay referencia al archivo original, simplemente porque los ejemplos no se escribieron para leer el contenido de los archivos. Para permitir al script que guarde el archivo, debemos pasarle el nombre del archivo. Podemos hacer esto fijando una variable oculta de la siguiente forma:

```
<input type="hidden" name="  
"filename" value="  
<?=$filename?>">
```

Ahora sabemos en la página de envío que archivo ha sido editado.

Una cosa a reseñar es que todo lo que hace el script de ejemplo de envío es proporcionar un marco de trabajo para que nosotros creemos nuestro propio meca-

nismo para guardar el archivo. En el script de ejemplo, todas las barras invertidas añadidas en el formulario se eliminan y el código HTML se convierte en entidades HTML con la siguiente línea:

```
$postedValue =  
htmlspecialchars(  
stripslashes( $value ) ) ;
```

Esto significa que todas nuestras etiquetas pierden su significado especial. Por ejemplo, el signo menor que (<) se convierte a *<*. Esto se hace de manera que podemos ver el código HTML generado. De otra manera, el código se interpreta por nuestro navegador.

Aunque esto es bueno como demostración y como prueba, no encontramos especial valor en mostrar el código HTML, particularmente si nuestros usuarios no van a ser entendidos en la materia. En su lugar, podemos proporcionar el código del editor como vista previa. Podemos crear un formulario bajo la vista previa con un simple botón que guarde el archivo.

Téngase en cuenta que una vez hayamos guardado el archivo, la versión antigua ya no estará disponible. Para evitar daños irreparables al original, hemos creado un subdirectorio *archive* en cada directorio. Antes de guardar el archivo, copiamos el archivo al directorio *archive*, añadiendo la información de fecha y hora al nombre del archivo. De esta manera, si alguien destruye el archivo, el original se puede recuperar con facilidad.

Del mismo modo, debemos tener en cuenta el entorno en el cual se va a mostrar el archivo. Si el archivo se carga directamente, alguien podría añadir un *include* de la parte servidor o algún código PHP para obtener permisos indebidos. Por tanto deberíamos revisar el código antes de guardarlo.

Guardar el archivo llegados a este punto es muy sencillo. Tenemos el nombre del archivo en la variable *\$filename* y el contenido en *\$FCKeditor1*. Usamos la función apropiada de PHP para guardar el archivo.

Configurar el Editor

Podemos configurar la apariencia y funcionalidad del editor para adaptarlo a nuestra página Web específica. Por ejemplo, podemos usar nuestra propia hoja de estilo para cambiar la apariencia. Por

defecto, el editor usa la hoja de estilo *_samples/sample.css* a la que se accede en los archivos de ejemplo a través de la ruta relativa *./sample.css*.

En nuestro caso, concluimos que la barra de herramientas por defecto con todas sus maravillosas funcionalidades era demasiado para nuestros usuarios. El objetivo era proporcionar una manera sencilla de editar los archivos online, por lo que necesitábamos una manera de limitar lo que estaba disponible. Por defecto, FCKEditor proporciona dos conjuntos diferentes de herramientas: *Default* y *Basic*. El conjunto *Basic* sólo proporciona media docena de opciones, lo que era realmente demasiado poco. Necesitábamos algo intermedio.

Afortunadamente, FCKEditor proporciona una manera muy fácil de cambiar las herramientas que están disponibles. En el archivo *fckconfig.js*, encontraremos que las barras se definen del siguiente modo:

```
FCKConfig.ToolbarSets  
["Basic"] = [ ... ];
```

La manera más fácil de crear nuestra propia barra de herramientas es copiar una barra existente, renombrarla adecuadamente y añadir o eliminar herramientas según nuestro parecer. Dentro de los corchetes, observaremos las funcionalidades que se proporcionan con esta barra de herramientas específica. Si incluimos un nuevo conjunto de corchetes, insertamos un separador entre los conjuntos de herramientas (no debemos olvidar la coma entre los conjuntos de herramientas). Al incluir '-' dentro de un conjunto de herramientas, se inserta un separador que, por algún motivo, es más pequeño.

Conclusión

Aunque FCKEditor proporciona muchas funcionalidades, este artículo debería proporcionar suficientes conocimientos básicos para empezar. Investigando a través de los diferentes archivos de configuración, encontraremos muchas opciones más para cambiar FCKEditor para que se adapte a nuestras necesidades. ■

RECURSOS

[1] Página de FCKEditor: www.fckeditor.net