

Edycja stron WWW za pomocą programu FCKeditor

RAZ, DWA I GOTOWE

Małe serwisy WWW często zawierają przestarzałe informacje, ich twórcy nie mają bowiem czasu na ciągle uaktualnianie stron. Nawet coś tak prostego, jak poprawa błędów w pisowni, może zająć całe dni. Program FCKeditor zdejmuję obowiązek aktualizacji z barków administratora serwisów, pozwalając przejąć inicjatywę użytkownikom. **JAMES MOHR**

Oprócz zarządzania prywatnymi stronami WWW, prowadzę też – jako ochotnik – serwis lokalnego klubu sportowego. Członkowie klubu narzekają często, że nikomu nie chce się uzupełniać bieżących informacji. Niektóre części strony nie zawierają w ogóle żadnych danych – tylko dlatego, że nikt nie ma czasu na ich uaktualnianie. Mam wiedzę techniczną wystarczającą do zarządzania serwisem, ale nie mogę poświęcić czasu na opiekę nad każdą stroną. Potrzebowałem prostego rozwiązania, które umożliwiłoby członkom klubu samodzielne wprowadzanie informacji online.

Dla początkującego użytkownika najprostszym sposobem aktualizowania strony WWW jest pisanie kodu w dowolnym edytorze i wysyłanie plików na serwer za pomocą klienta FTP. Niestety, większość ludzi z naszego klubu nawet z tym ma kłopoty. Tak więc, aby umożliwić jak największej liczbie osób umieszczanie informacji na stronach, musiałem zrobić coś, aby aktualizowanie ich było jak najprostsze. Utworzyłem bazę danych wiadomości i nadchodzących wydarzeń, pozwalając użytkownikom dodawanie informacji do formularzy internetowych. Nie znalazłem jednak prostego rozwiązania do aktualizowania innych stron; przynajmniej dopóki nie trafiłem na program FCKeditor.

FCKeditor to edytor tekstowy rezydujący na serwerze i dostępny przez zwykłe okno przeglądarki. Integrując go z serwisem WWW stworzy-

łem wygodne rozwiązanie, umożliwiające użytkownikom edycję ich własnych stron. Nawet najbardziej niedoświadczeni użytkownicy bez problemu radzili sobie z edytorem tekstowym, FCKeditor okazał się więc narzędziem jeszcze prostszym w obsłudze, niż oferujące podobne funkcje Zope czy środowiska typu Wiki. Wyzwanie polegało, oczywiście, na wymyśleniu, jak zintegrować go z serwisem. Sposób realizacji tego zadania opisuję w niniejszym artykule.

Edytor o pełni możliwości

FCKeditor (Rysunek 1) ma wszystkie cechy, których większość ludzi wymaga od narzędzia do przetwarzania tekstu, w tym także udostępniający wszystkie funkcje pasek narzędzi. Pod paskiem znajduje się pole tekstowe typu WYSIWYG. FCKeditor działa z niemal każdą przeglądarką i w każdym środowisku.

Instalowany na serwerze, FCKeditor obsługuje najpopularniejsze języki służące do tworzenia stron internetowych: PHP, Perla,

ColdFusion, Javę, a nawet ASP i ASP.Net. Umożliwia to bezproblemową integrację z istniejącym serwisem WWW.

FCKeditor dostępny jest w dwudziestu kilku wersjach językowych. Domyślnie język wykrywany jest na podstawie ustawień przeglądarki, ale zdefiniowanie określonego języka w pliku konfiguracyjnym jest dość proste.

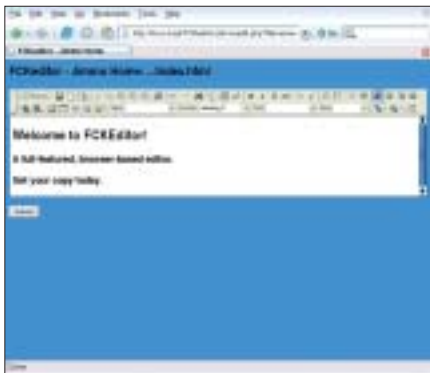
Największą wadą programu jest brak dokumentacji. Zawiera ona tylko powtarzalne odnośniki do dołączonych do programu przykładów (patrz pasek boczny „Sample Files”). Przykłady te pomagają wprawdzie zorientować się w możliwościach programu, takie podejście do dokumentacji sprawia jednak, że aby odkryć, jak dostosować program do swoich potrzeb, trzeba przebrnąć przez sporą ilość kodu.

Instalacja

Najnowszą wersję FCKeditora pobrać można ze strony WWW programu [1]. W czasie pisania tego tekstu jest to wersja 2 Release

Przykładowe pliki

W katalogu `_samples` znaleźć można kilka przykładów wykorzystania każdego języka. Ja używam PHP, wykorzystałem więc pliki znajdujące się w podkatalogu `php` - skopiowałem jeden z nich do katalogu głównego FCKeditor. Następnie skopiowałem pliki `sampleposteddata.php` i `_samples/sample.css`. Wykorzystałem plik `sample2.php` i nazwałem go (dla uproszczenia) `jimmoedit.php`. Ten plik będzie trzeba wczytać i w nim właśnie należy zacząć wprowadzać jakiegokolwiek zmiany. Zmieniłem także nazwę arkusza stylów na `jimmo.css` i odpowiednio zmodyfikowałem plik `jimmoedit.php`.



Rysunek 1: FCKeditor – główne okno programu.

Candidate 2. Jeżeli korzystasz z serwera Apache działającego pod Linuxem, koniecznie wybierz wersję RC2, ponieważ poprzednia powodowała pewne problemy.

Po ściągnięciu pakietu instalacyjnego utwórz katalog *FCKeditor* w katalogu głównym dokumentów (DocumentRoot) na swoim serwerze WWW i rozpakuj pakiet do tego katalogu. Możesz zmienić położenie programu, ale będzie to wymagało wprowadzenia kilku zmian do plików konfiguracyjnych (patrz pasek „Setting the Location”).

Każdy z dostarczonych z pakietem przykładów w języku PHP (patrz pasek narzędzi „Sample Files”) demonstruje różne możliwości edytora. Jeden z nich pozwala na przykład wybrać pasek narzędzi, inny język i tak dalej. Każdy ze składników da się dodać do dowolnej strony, możesz więc zbudować dostosowany do swoich potrzeb edytor, umożliwiając wybranie każdej z opcji.

Wybrałem jeden z plików przykładowych i wykorzystałem go jako punkt wyjścia do tworzenia własnej konfiguracji. Wybrany przeze mnie plik wyglądał na najłatwiejszy do zaadaptowania. W danym przypadku nie chciałem, aby użytkownicy mieli możliwość manipulowania przy konfiguracji samego edytora, ponieważ jestem zwolennikiem jak najprostszych rozwiązań. Wyciąłem więc cały dodatkowy kod. Wyłączyłem także automatyczne wykrywanie języka. Ponieważ serwis WWW znajduje się w Niemczech, ustawiłem domyślny język niemiecki:

```
$oFCKeditor->Config >
['AutoDetectLanguage'] = false ;
$oFCKeditor->Config >
['DefaultLanguage'] = 'de' ;
```

Jeżeli konfigurujesz FCKeditor dla potrzeb swojego środowiska, wybierz przykład, który najlepiej pasuje do danej sytuacji i w razie potrzeby odpowiednio dostosuj ustawienia.

Wczytywanie tekstu do edytora

Pierwszym etapem integracji programu FCKeditor z istniejącą stroną WWW jest określenie sposobu wczytywania tekstu do edytora. Atrybut *Value* zawiera tekst, który pojawia się w oknie edytora - tekst ten można zatem zdefiniować po prostu podając wartość atrybutu *Value*:

```
$oFCKeditor->Value = >
'Twoj tekst' ;
```

Można tu podać pusty ciąg znaków - uruchomi się wówczas zupełnie pusta strona, ale jedną z najlepszych funkcji FCKeditora jest możliwość edytowania istniejących stron. Musimy więc znaleźć sposób na wstawienie zawartości istniejącego pliku do edytora. Najprostsza metoda to wykorzystanie funkcji PHP *file_get_contents()*:

```
$oFCKeditor->Value = >
file_get_contents >
("ściezka_do_pliku") ;
```

Uwaga: Funkcja *file_get_contents()* pozwala wczytać adres URL lub załadować plik bezpośrednio. Strona wskazywana przez adres URL może zawierać polecenia SSI, pływające ramki i dowolną liczbę innych elementów. Plik, który pojawi się w przeglądarce, może więc nie być dokładnie tym plikiem, który naprawdę chcesz edytować. Moim zdaniem najbezpieczniej jest unikać podawania adresów URL. Przed przekazaniem nazwy pliku funkcji *file_get_contents()* upewniam się, że wczytuje ona plik lokalny.

Nawet przy wykorzystaniu tej metody, zawartość strony jest w zasadzie statyczna. Potrzebny jest nam sposób na dynamiczne przekazywanie nazwy pliku do wczytania. Można to z łatwością osiągnąć, przekazując nazwę pliku jako część zapytania:

```
http://linux.local/FCKeditor >
/jimmoedit.php?filename= >
index.html
```

Następnie w skrypcie edytora (w naszym przykładzie jest to *jimmoedit.php*) odczytujemy zmienną *filename*:

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

Należy przy tym zachować ostrożność, da się bowiem utworzyć zapytanie, które zrobi znacznie więcej, niż tylko załadowanie pliku na serwer WWW. Złośliwy użytkownik mógł

Ustalanie położenia programu

Choć instrukcja zaleca rozpakowanie pakietu do podkatalogu *FCKeditor* w katalogu głównym dokumentów na serwerze, program można w zasadzie umieścić w dowolnym miejscu. Aby zmienić jego położenie, należy ustawić własność *BasePath* obiektu edytora. Na przykład w pliku *jimmoedit.php* znajduje się wpis:

```
$oFCKeditor->BasePath = §§
„/FCKeditor/” ;
```

Uwaga: wcześniej utworzyliśmy już nową instancję edytora, tu definiujemy więc atrybut *BasePath* dla tej właśnie instancji (ponieważ bieżącą instancją obiektu edytora jest *\$oFCKeditor*).

Jeżeli przyjrzesz się komentarzom w przykładzie, zobaczysz, że własność *BasePath* określana jest na podstawie katalogu *_samples*. Należy ją określić w tym miejscu lub w pliku tworzącym klasę edytora. W przypadku PHP będzie to plik *fckeditor.php* (oprócz utworzenia obiektu *FCKeditor*, plik *fckeditor.php* odpowiada także za stworzenie kodu HTML, definiującego ramkę pływającą, w której znajduje się edytor, pasek narzędzi itd.). W funkcji konstruktora dla klasy edytora można ustawić atrybut *BasePath* (oraz parę innych przydatnych własności). W tym przykładzie, ponieważ ścieżka jest bezpośrednio podana w konstruktorze, należy wykomentować lub usunąć dodatkowy kod z pliku *jimmoedit.php*.

Następnie trzeba zmienić ścieżkę tak, aby wskazywała położenie właściwego kodu edytora. Definiuje się to na początku pliku *jimmoedit.php* przy użyciu ścieżki względnej:

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

Ponieważ plik edytora znajduje się teraz w katalogu głównym programu (tym samym, w którym znajduje się plik *fckeditor.php*), trzeba usunąć ścieżkę względną.

by na przykład odczytać plik *.htaccess* albo dowolny inny plik z Twojego systemu. Przed wczytaniem pliku należy więc zawsze sprawdzić, czy wartość wpisywana do pola *\$filename* jest poprawna.

Zmuszanie użytkowników do wpisywania zapytania ręcznie wymaga od nich najprawdopodobniej zbyt wiele pracy, musimy więc pomyśleć o prostszym sposobie przekazywania pliku. Zależy to od tego, jak dana strona jest generowana, kto ma prawo do edytowania stron i tak dalej. W administrowanym przeze mnie serwisie strony tworzone są za pomocą PHP, mogą więc sprawdzić, czy dany użytkownik się loguje. Jeżeli tak, odsyłacz

„Edit this page” zawiera pełny adres URL edytora i ścieżkę względną do właściwego pliku.

W innym serwisie jest to nieco bardziej skomplikowane, ponieważ nie korzystam tam z PHP ani innych języków skryptowych, które pomogłyby mi w sprawdzeniu, czy użytkownik jest zalogowany. Rozwiązaniem jest umieszczenie edytora w zabezpieczonym hasłem katalogu na serwerze i wykorzystanie standardowego uwierzytelniania serwera Apache. W następujący sposób mogę określić nazwę zalogowanego użytkownika:

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

Po sprawdzeniu, że użytkownik jest zalogowany, sprawdzam, do jakiej należy grupy, od tego zależy bowiem, jakie pliki może edytować. Na przykład osoba należąca do grupy *volleyball* (siatkówka) może edytować tylko strony z katalogu o tej samej nazwie. Ponieważ nazwy katalogów są takie same jak nazwy grup, łatwo określić, który katalog jest właściwy. Mogę później użyć różnych funkcji PHP do przeszukania katalogu i utworzenia (w HTML) listy dostępnych plików, generując odpowiednie odsyłacze, przekazujące nazwy plików do edytora. Kluczową zaletą takiego podejścia jest możliwość odfiltrowania wszelkich plików, do których użytkownicy nie powinni mieć dostępu lub których nie powinni edytować.



Rysunek 2: Konfiguracja tabeli w HTML.

Korzystanie z edytora

Sam edytor zachowuje się tak samo jak każdy inny edytor tekstu. Na przykład zaznaczając fragment tekstu i klikając odpowiedni przycisk można zmienić czcionkę na pogrubioną lub kursywę, zmienić rozmiar tekstu itd. Nie dotyczy to tylko standardowych znaczników - FCKeditor pozwala wstawiać do znaczników różne style. Program ma nawet tryb edycji kodu źródłowego, umożliwiając pisanie bezpośrednio w języku HTML.

Funkcją, która naprawdę zrobiła na mnie wrażenie, jest edytor tabel. Po kliknięciu odpowiedniego przycisku wyświetlane jest okno wyskakujące, zaprezentowane na Rysunku 2. Podobnie jak w innych edytorach HTML, można określać rozmiar tabeli (zarówno liczbę komórek, jak i wymiary w pikselach), jej obramowanie, odstępy między komórkami itd.

Po wstawieniu tabeli można modyfikować wiele jej własności. Jeżeli zaznaczysz tabelę klikając ją lewym klawiszem myszy, możesz przesunąć jej krawędzie lub narożniki, aby zmienić jej rozmiar. Kliknięcie tabeli prawym klawiszem umożliwia wstawianie lub usuwanie wierszy, kolumn i pojedynczych komórek. Można także dzielić wiersze lub kolumny, a nawet zmieniać podstawowe własności całej tabeli. Inne funkcje umożliwiają tworzenie list, dodawanie odsyłaczy i obrazków i zmianę koloru tła.

Przeglądanie plików na serwerze

FCKeditor udostępnia mechanizm służący do przeglądania plików na serwerze. Odbywa się to przy pomocy „łączników” (connectors) dla róż-

nych języków. Łącznik i powiązany z nim kod dla języka PHP znajduje się w katalogu *editor/filemanager/browser/default/connectors/php*.

W pliku *fckconfig.js* należy określić, z którego łącznika ma korzystać przeglądarka. Należy ustalić dwie wartości: *FCKConfig.LinkBrowserURL* i *FCKConfig.ImageBrowserURL*. Domyślnie zmienne te wskazują na łącznik dla ASP, trzeba więc odkomentować łącznik dla PHP.

Domyślnie FCKeditor szuka plików w różnych katalogach, zależnie od ich typu. Możesz na przykład chcieć utworzyć inny katalog dla plików, a inny dla grafiki. Aby ścieżka nie zależała od typu zasobów, trzeba zmodyfikować plik *connectors/php/io.php*. Najpierw w funkcji *GetUrlFromPath()* zmien wiersz

```
return $GLOBALS >  
["UserFilePath"] >  
. $resourceType . $folderPath ;
```

na:

```
return $GLOBALS >  
["UserFilePath"] >  
. $folderPath ;
```

Czy w Twojej firmie jest INTERNET ?

TAK !

Masz z nim wieczne problemy ?

TAK !

Martwisz się o brak przestrzeni na dysku ?

TAK !

Twoja strona www jest przestarzała ?

TAK !

MY rozwiążemy Twoje problemy

**TANIE I SZYBKIE WDROŻENIA
NA PLATFORMIE LINUX/UNIX**

WWW.BBCREATION.PL

694-238-193, 501-101-206

**SERWERY LINUX/UNIX,
STRONY WWW, WDROŻENIA,
SIECI, INTERNET**

Następnie w funkcji *ServerMapFolder()* zmien wiersz

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

na

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

Później musisz zmienić ścieżkę do plików użytkowników. Robi się to w pliku *connector.php*. Na początku pliku znajdziesz odwołanie, które wygląda następująco:

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

Zmień położenie na dowolnie wybrany katalog, podając ścieżkę względem katalogu DocumentRoot na serwerze.

Zapisywanie zmienionego pliku

Po zakończeniu edycji pliku kliknij przycisk *Submit* – wprowadzony tekst zostanie wysłany pod określony adres (podobnie jak w przypadku dowolnego formularza internetowego). Domyślnym adresem (action) jest plik *sampleposteddata.php*, który nazwałem po prostu *posteddata.php*.

Jeżeli przyjrzymy się bliżej przykładowemu plikowi, do którego wysyłana jest zawartość formularza, zorientujemy się, że jego działanie sprowadza się do pobrania zawartości pola wejściowego z formularza i wyświetlenia jej. Nie ma odwołań do oryginalnego pliku - po prostu dlatego, że zadaniem przykładów nie było odczytywanie tekstu z plików. Aby więc umożliwić skryptom zapisanie pliku, musimy podać mu jego nazwę. Zrobiłem to, dodając do formularza ukrytą zmienną:

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

Wówczas na stronie wysyłania podana jest informacja, który plik został zmieniony.

Należy zauważyć, że przykładowy skrypt wysyłający to tylko baza, na podstawie której możesz stworzyć swój własny mechanizm zapisywania pliku. Poniższy wiersz w przykładowym pliku usuwa wszystkie doda-

ne do formularza odwrócone ukośniki i konwertuje znaczniki HTML na encje tego języka:

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

Oznacza to, że wszystkie znaczniki tracą swoje szczególne znaczenie. Na przykład znak mniejszości (<) przekształcany jest na *<*. Dzieje się tak po to, aby można było zobaczyć wygenerowany kod HTML. W przeciwnym przypadku kod ten jest interpretowany przez przeglądarkę.

Choć dobrze nadaje się to do celów demonstracyjnych i testowych, w rzeczywistości nie widzę większego sensu w wyświetlaniu kodu HTML, zwłaszcza jeśli mamy do czynienia z mało zaawansowanymi technicznie użytkownikami. Możesz natomiast zrobić to, co ja: pod formularzem utworzyć formularz z podglądem, zawierający jeden przycisk, służący do zapisywania pliku.

Pamiętaj, że po zapisaniu pliku jego starsza wersja nie jest już dostępna. Aby zapobiec nieodwracalnemu zniszczeniu, w każdym katalogu utworzyłem podkatalog *archive*. Przed zachowaniem pliku, kopiuję do tego podkatalogu plik oryginalny, dodając do nazwy pliku czas jego zapisania. Jeżeli więc ktoś zepsuje coś w pliku, można z łatwością odtworzyć oryginał.

Należy także uwzględnić środowisko, w którym wyświetlany będzie dany plik. W przypadkach, gdy jest on wczytywany bezpośrednio, użytkownik mógłby dodać polecenie SSI albo kod PHP, aby zdobyć dostęp do nieprzeznaczonych dla niego plików. Przed zapisaniem kodu należy go więc przeanalizować.

W tym momencie zapisanie pliku jest bardzo proste. Nazwa pliku znajduje się w zmiennej *\$filename*, a jego zawartość – w zmiennej *\$FCKeditor1*. Do zapisania pliku należy wykonać odpowiednią funkcję PHP.

Konfiguracja edytora

Wygląd i funkcje edytora można dostosować do określonego serwisu WWW. Można na przykład użyć własnego arkusza stylów. Domyślnie edytor korzysta z arkusza *_samples/sample.css*, do którego pliki przykładowe odwołują się za pomocą ścieżki względnej (*./sample.css*).

Wydawało mi się na przykład, że domyślny pasek narzędzi ze wszystkimi tymi

wspianymi funkcjami to trochę za dużo dla moich użytkowników. Szukaliśmy przecież jak najprostszego sposobu edytowania plików online, trzeba więc było jakoś ograniczyć dostępne opcje. FCKeditor udostępnia domyślnie dwa różne zestawy narzędzi: domyślny (*Default*) i podstawowy (*Basic*). W zestawie podstawowym znajduje się zaledwie sześć opcji – za mało jak na nasze potrzeby. Potrzebowaliśmy rozwiązania pośredniego.

Na szczęście FCKeditor pozwala w bardzo prosty sposób zmienić listę dostępnych narzędzi. Paski narzędzi zdefiniowane są w pliku *fckconfig.js*:

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

Najprostszą metodą utworzenia własnego paska narzędzi jest skopiowanie istniejącego paska, nadanie mu nowej nazwy i dodanie lub usunięcie narzędzi odpowiednio do potrzeb. W nawiasach kwadratowych znajduje się lista funkcji, udostępnianych na danym pasku. Dodając nową parę nawiasów kwadratowych, można wstawić separator między zestawami narzędzi (pamiętaj o oddzieleniu ich przecinkami). Aby dodać nieco mniejszy separator, wystarczy wpisać znak *'! wewnątrz* zestawu narzędzi.

Podsumowanie

FCKeditor ma wprawdzie znacznie więcej funkcji, ale informacje zawarte w niniejszym tekście powinny wystarczyć do rozpoczęcia pracy z tym programem. Przeglądając różne pliki konfiguracyjne znajdziesz wiele opcji, umożliwiających dostosowanie aplikacji do własnych potrzeb. ■

Dodatkowe informacje

[1] Strona programu FCKeditor: www.fckeditor.net

Dodatkowe informacje

James Mohr odpowiada za monitorowanie kilku centrów danych dla działającej w Co-burgu w Niemczech firmy, dostarczającej rozwiązania informatyczne dla przedsiębiorstw. Prowadzi stronę WWW z podręcznikiem do Linuksa, <http://www.linux-tutorial.info>, i jest autorem kilku książek i wielu artykułów na różne tematy.