

## Personalizzare WordPress Per i Tuoi Clienti - Parte 2

Nel secondo articolo continuiamo a vedere alcune best practice per personalizzare WordPress per i clienti. Se vi siete persi [la prima puntata la trovate qui](#).

Queste che presento sono alcune delle pratiche che ritengo più valide e comuni, ma l'invito a condividere anche le vostre personalizzazioni è sempre valido.

### Personalizzare le pagine di errore di WordPress

A quanti è capitato di vedere la famosa pagina con scritto “**Error establishing a database connection**”? O peggio ancora vedere la pagina di installazione di WordPress in un sito che dovrebbe essere pienamente operativo?



The screenshot shows the WordPress installation page. At the top, there is the WordPress logo and the word "WORDPRESS" in blue. Below that, the heading "Salve" is followed by a horizontal line. The text reads: "Benvenuti nella famosa procedura dei cinque minuti per l'installazione di WordPress! Potreste voler scorrere la [documentazione in Leggimi](#). Altrimenti, indicare le seguenti informazioni e sarete sulla strada per utilizzare la più estensibile e potente piattaforma di editoria personale del mondo." Below this, the heading "Servono delle informazioni" is followed by another horizontal line. The text reads: "Prima di iniziare, abbiamo bisogno di alcune informazioni. Non preoccuparsi, si potrà sempre cambiarle in seguito." At the bottom, there is a label "Titolo sito" followed by a text input field.

C'è un modo molto semplice per **personalizzare queste pagine di errore di default di**

**WordPress** e consiste nell'inserire due file PHP all'interno della cartella **wp-content** della vostra installazione.

Il primo file che andremo ad aggiungere va chiamato **db-error.php** e rappresenta la pagina che WordPress visualizzerà in caso di errori legati al database MySQL. Il più frequente potrebbe essere, per esempio, il superamento del limite sulle query orarie impostato dal vostro fornitore di hosting.

Ecco come può tornare utile mostrare un messaggio personalizzato, invece del poco professionale avviso di WordPress.

All'interno del file **db-error.php** potrete inserire quello che volete, trattatelo come se fosse una **normalissima pagina HTML**. L'unica cosa sui cui dovete prestare massima attenzione è di **non usare le funzioni di WordPress**. Potreste essere infatti tentati di chiamare funzioni come *get\_header* o *get\_footer*, ma se vi trovate in questa pagina significa che il database non funziona correttamente e non è quindi consigliabile rischiare di chiamare funzioni che potrebbero utilizzarlo.

Vediamo un esempio con una struttura di base per il vostro file:

La parte più "complessa" sta nelle prime 3 righe di PHP, dove inviamo un codice di stato **HTTP 503** per dichiarare lo stato corretto del server. Questo potrebbe aiutare gli spider dei motori di ricerca a capire il motivo per cui non riescono a raggiungere il sito (e non perdere quindi le sudate posizioni nelle SERP).

Inviemo inoltre una mail all'amministratore del sito in modo che possa intervenire in tempi rapidi per risolvere il problema.

Il resto dipende dalla vostra fantasia! :)

Io, in alcuni casi, proponevo in questa pagina il widget Like Box di Facebook, nella speranza di non perdere un potenziale cliente fornendogli una modalità di contatto alternativa.

L'altro file che andremo ad aggiungere nella cartella wp-content è *maintenance.php*, che verrà utilizzato da WordPress quando il sito viene messo in modalità manutenzione. Questa modalità si attiva, ad esempio, quando è in corso un aggiornamento del core di WordPress, ma può essere attivata anche manualmente (tramite plugin o inserendo un file chiamato *.maintenance* nella root del vostro spazio web). Invece di vedere l'anonimo messaggio: "*Momentaneamente non disponibile per manutenzione. Riprovare fra un minuto.*" vedremo la nostra pagina di manutenzione personalizzata.

La struttura interna del file è praticamente uguale a quella precedente, ma cambia il messaggio per l'utente e non verrà inviata nessuna mail all'amministratore (che con tutta probabilità è già a

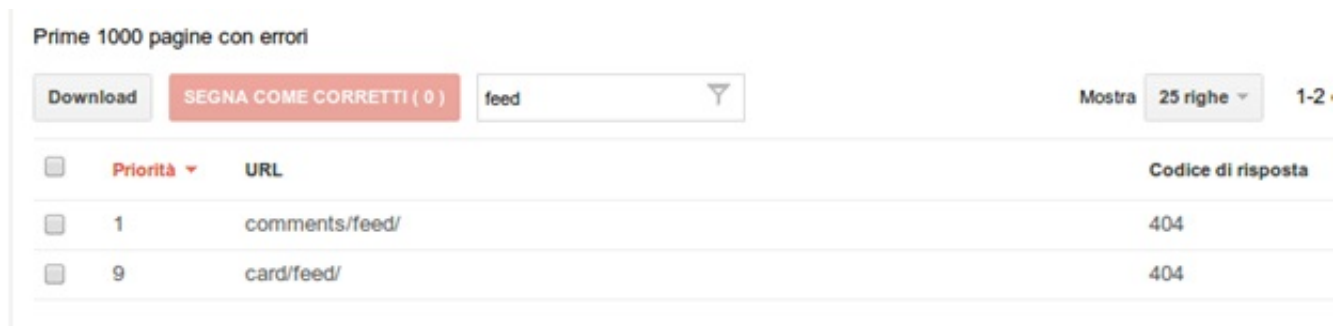
conoscenza della situazione). Ve la riporto ugualmente:

I file *db-error.php* e *maintenance.php* non vengono sovrascritti in caso di aggiornamento del sistema. Vi consiglio di creare 2 modelli da inserire in tutti i siti WordPress che realizzate.

## Disabilitare i Feed Rss

Quando si usa WordPress come CMS, i feed RSS possono diventare una seccatura. Se il sito è composto da poche pagine “statiche”, i feed non hanno molto senso, ma WordPress (che nasce come piattaforma per il blogging) produce in automatico i feed RSS per il sito.

Potreste quindi trovare, all’interno del vostro *Webmaster Tools*, errori 404 come quello che segue:



Prime 1000 pagine con errori

Download **SEGNA COME CORRETTI ( 0 )** feed  Mostra 25 righe 1-2

<input type="checkbox"/>	Priorità ▼	URL	Codice di risposta
<input type="checkbox"/>	1	comments/feed/	404
<input type="checkbox"/>	9	card/feed/	404

Per evitarli è sufficiente **disabilitare tutti i feed RSS generati da WordPress**, aggiungendo queste righe al solito file *functions.php* del vostro tema:

## “Brandizzare” WordPress

Se volete aggiungere un tocco personale alle pagine dell’amministrazione potete modificare e personalizzare diverse parti delle vostre pagine.

### ***Il footer***

Aggiungendo i crediti, i vostri clienti si ricorderanno di voi. :)

Anche la pagina di login può essere personalizzata con 3 semplici operazioni. I codici PHP vanno sempre inseriti nel file *functions.php* del vostro tema.

### ***Il logo del login***

Pratica a mio avviso migliore dell'utilizzo di plugin che fanno la stessa cosa. Come regola generale io cerco di utilizzare il minor numero di plugin possibile, visto che (estremizzando) possiamo dire che ogni plugin che viene aggiunto comporta un utilizzo maggiore delle risorse (che si traduce in maggiore lentezza del sito).

### ***Il mini logo all'interno dell'amministrazione***

"*admin\_head*" è un *hook* che permette di inserire codice nella sezione *head* delle pagine dell'amministrazione di WordPress.

Se ne avete bisogno, all'interno di queste funzioni, potete inoltre usare le classiche funzioni di WP, come quella per recuperare il percorso alla cartella del tema in uso `get_bloginfo('template_directory')` qualora voleste scrivere del codice più generico e riusabile.

### ***Il titolo e l'url a cui punta il logo del login***

Se non includete questi 2 snippet di codice il titolo riporterà "WordPress", mentre l'URL di destinazione sarà *wordpress.org*.

## **Disabilitare gli aggiornamenti automatici**

Nella versione 3.7 è stata aggiunta una funzionalità che permette al core di *autoaggiornarsi* senza attendere le azioni degli amministratori. L'aggiornamento automatico viene effettuato solo per le release minori, passerà quindi dalla 3.7.2 alla 3.7.3, ma non dalla 3.7.3 alla 3.8.

In alcuni casi, specie quando avete personalizzato sostanzialmente WordPress, può essere utile **disabilitare gli aggiornamenti automatici** in modo da avere il controllo assoluto e decidere in autonomia quando e se procedere con l'aggiornamento.

Per farlo inserite nel vostro file *wp-config.php* la seguente riga di codice:

## **Conclusioni**

Nella "prossima puntata" vedremo come semplificare l'interfaccia di amministrazione, quindi continuate a seguire **Your Inspiration Web!**

Oggi abbiamo visto una carrellata di possibili **personalizzazioni da fare su Wordpress per i vostri clienti**. Voi usate qualche altro accorgimento? Ne parliamo nei commenti? Magari estendiamo insieme questo articolo. Vi aspetto!

**PERSONALIZZARE WORDPRESS PER I TUOI CLIENTI: INDICE LEZIONI**

- 1) [Personalizzare WordPress per i tuoi clienti — Parte 1](#)
- 2) [Personalizzare WordPress per i tuoi clienti — Parte 2](#)
- 3) [WordPress per i tuoi clienti: semplificare l'interfaccia con il plugin Adminimize – Parte 1](#)
- 4) [WordPress per i tuoi clienti: semplificare l'interfaccia con il plugin Adminimize – Parte 2](#)
- 5) [WordPress per i tuoi clienti: semplificare l'interfaccia con il plugin Adminimize – Parte 3](#)