Come sviluppare un widget complesso (2/3)

Nel precedenete tutorial abbiamo iniziato a dare forma al widget che stiamo sviluppando. Le due tabs sono pronte per ricevere i contenuti che dovremo estrarre dal database. Quindi iniziamo subito a scoprire come si interroga il database di WordPress.

$wpdb: le chiavi di casa

La gestione del database di WordPress é basata sulla libreria ezSQL; tutti gli strumenti che servono (metodi e proprietà) li troviamo nell'oggetto $wpdb.
Sebbene a livello teorico potremmo utilizzare le funzioni native di php (ogni volta che WordPress carica una pagina stabilisce anche ovviamente la connessione al db) per lanciare le nostre query al database, **questa procedura é vivamente sconsigliata**. Vedremo subito perché.

Implementare il metodo _getMostComment

Iniziamo dunque dal metodo più semplice da implementare cioé *_getMostComment* che restituirà gli articoli più commentati.

Si tratta di una semplice query sulla tabella *wp_posts* nella quale selezioneremo il titolo del post (post_title), il suo permalink (guid) ed il campo dove viene registrato il numero di commenti (comment_count). Dovremo poi fare una distinzione per evitare che la query tenga conto anche delle pagine (il *post_type* dovra essere "post"). In seguito dovremo ordinare il risultato in base al campo *comment_count* ed infine limiteremo i risultati a quanto contenuto nell'opzione *num_most_comment*.
Tutto questo in SQL diventa così

Fatto? Bene, hai commesso il primo imperdonabile errore. Infatti chi dice che il prefisso delle tabelle debba essere *wp_* ?

- Prima dell'installazione é possibile configurare un prefisso diverso da quello standard (wp_).
- In un installazione network di WordPress, come abbiamo visto, le tabelle dei sottositi hanno questa forma: [prefisso][id_del_sottosito][nome_tabella]

In questi casi, il nostro widget non funzionerebbe.

Per ovviare a questo inconveniente é necessario utilizzare la proprietà *prefix* dell'oggetto *$wpdb* che contiene il giusto prefisso delle tabelle.

Dunque al posto di

wp_posts

Scriveremo

La proprietà *$prefix* verrà utilizzata soprattutto quando abbiamo a che fare con tabelle non native. Ad esempio se il plugin che si sta sviluppando necessita di tabelle personalizzate, dovremo indicare il giusto prefisso **già a partire dalla creazione della tabella**.

Per le tabelle native invece si può disporre di una forma semplificata; ad esempio il nome esatto della tabella dei posts sarà contenuto nella proprietà *$posts*.

Dunque torniamo al metodo *_getMostComment*. Inizia a dichiararlo passando come argomento *$instance*.

Ora portiamo all'interno del metodo l'istanza *$wpdb* e scriviamo la query in modo corretto, come abbiamo visto

Il modo migliore, in questo caso, per recuperare il risultato della query, é quello di utilizzare il metodo *get_results()* dell'oggetto *$wpdb*. Questo metodo restituisce i risultati in forma di oggetti che potremo scorrere con una semplice ciclo foreach.

Ed ecco il metodo *_getMostComment()* completo

Implementare il metodo _getLastPost()

Questo metodo é molto simile a quello che abbiamo visto in precedenza, é solo la query ad essere un po' più complessa. Dovremo infatti fare una join tra la tabella *posts* e la tabella *users* per poter recuperare anche il nome dell'autore dell'articolo (che nella tabella dei post é espresso come riferimento relazionale all'id dell'autore).

Come prima cosa dovremo selezionare le due tabelle:

In seguito dovremo allineare il record del post con il corrispondente record della tabella *users* e richiedere che vengano selezionati unicamente gli articoli.

In seguito l’ordine di selezione, che deve essere per data

Ed infine la limitazione dei risultati

Ed ora riscrivi la query utilizzando le best practices che abbiamo visto in precedenza

Ed ecco il metodo completo

Come vedi, ho utilizzato la comoda funzione *mysql2date()* in grado di trasformare il timestamp di mysql nel formato che passeremo come primo argomento (i parametri sono quelli della funzione date() di php.

Ed ora non resta che decommentare l’esecuzione dei metodi appena sviluppati nel metodo widget per renderlo completamente operativo.

Conclusione

In questo secondo tutorial abbiamo visto un’introduzone alla manipolazione del database di WordPress. La cosa più importante da ricordare é senz’altro **quanto sia maledettamente importante l’utilizzo corretto dell’oggetto *$wpdb***.

Nel prossimo ed ultimo tutorial di questa serie tratteremo l’argomento della localizzazione, ovvero della traduzione in più linguaggi del plugin.