Easy Form: Un piccolo fw per i tuoi form

E se ti dicessi che puoi creare un form di registrazione o di contatto o di quello che vuoi con validazione lato server, validazione lato client e tante altre amenità in due minuti?

Easy Form

Easy Form é un piccolo framework altamente configurabile ed estendibile che ho sviluppato partendo da un presupposto. Il 90% dei form che svilupppiamo sono moduli di contatto o di registrazione o di login e hanno quasi sempre le stesse caratteristiche.
Generalizzando e standardizzando queste caratteristiche ho ottenuto un framework che, lo dico subito, non va bene per tutto, ma certamente per la maggior parte dei casi.
Easy Form mette a disposizione diverse API per diversi livelli di competenza ed integra alcuni plugin che vedremo.
Intanto inizia a scaricare il framework, poi ci divertiremo un po'.

Warm Up

Nella cartella di easy form creiamo un nuovo documento e chiamalo *test.php*.
Inizieremo ora ad esplorare le potenzialità di easy form. Come primo passo, dobbiamo includere la libreria di base, il file denominato *easy.form.class.php.*

In seguito creiamo un'istanza di questa classe:

Al costruttore di classe può essere passata una grande quantità di parametri di configurazione in forma di array associativo (nome parametro => valore).
Vediamo nel dettaglio alcuni dei parametri che è possibile utilizzare:

- **method**: Il metodo di invio di dati. Il valore di default è "*post*".
- **action:** La pagina alla quale punta il form. Il valore di default è la stessa pagina (*PHP_SELF*)
- **prefix:** Il prefisso per id, nomi, ecc... Se volessimo due form sulla stessa pagina, dovremo creare due istanze ed avere dei prefissi diversi. Valore di default: *ef-*
- **html5:** Da settare a *true* se si desidera sfruttare le caratteristiche per i form implementate da html5. Il valore di default è *false*.
- **inline:** Mostra le label e i campi allineati sulla stessa riga. Se settato su *false* mostra le label sopra i campi. Il valore di default è *true*.

- **language:** La lingua per mostrare i vari messaggi di errore e di servizio. I file di linguaggio sono dei semplici file di testo contenuti nella cartella languages. Il valore di default è *it*.
- **ajax:** Di default é impostato su *false*. Se volessimo inviare i dati del form con chiamata asincrona, dovremo passare come parametro il percorso e nome del file al quale passare i dati per l'elaborazione. Vedremo un esempio più avanti.

Ci sono altri parametri che possono essere configurati ma li vedremo tra poco.

Dunque, se volessi ad esempio passare i dati in *get* ed utilizzare le caratteristiche di html5, dovremmo istanziare in questo modo:

Per i nostri esempi, per il momento, utilizzeremo i parametri di default. Quindi istanziamo senza passare parametri.

Have errors?

Subito dopo la creazione dell'istanza utilizzeremo il metodo *_have_errors()* che restituisce *false* se la validazione lato server é andata a buon fine o *true* in caso contrario (eccezione nel caso di login - anche perché la validazione non serve - dove bisognerà utilizzare la proprietà *login_error*; si veda *sample3.php* contenuto nel pacchetto). Dunque possiamo procedere in questo modo:

Qualora non desideriamo reindirizzare ad un'altra pagina una volta eseguite le operazioni necessarie é possibile utilizzare il metodo *_success()* che passa come parametro la frase che si vuole visualizzare nell'area di notifica del form.

Ora é possibile iniziare a scrivere il markup della pagina. Una cosa molto semplice. A questo punto il nostro documento si presenta così:

Il foglio di stile

Easy form dispone di un foglio di stile di default (*css/ef-base.css*) che é da intendere come esempio per mostrare l'utilizzo delle varie classi. Puoi modificarlo, crearne uno nuovo oppure integrare le classi nel foglio di stile del tuo sito.
Puoi caricare un foglio di stile per easy form utilizzando il metodo *_loadStyle()*, che passa come parametro il percorso del foglio di stile.
Inserisci quindi nell'header del documento questo codice:

API super-rapide

Easy form dispone di tre metodi che corrispondono ad altrettanti form predefiniti e sono:

- _simpleContactForm() (esempio)
- _simpleRegistrationForm() (esempio)
- _simpleLoginForm (esempio)

Questi tre metodi passano due parametri:

- Il titolo del form (label > legend)
- Il value del pulsante di submit

Quindi se volessimo creare un form di contatto potremmo fare così:

Come puoi vedere negli esempi, easy form esegue la validazione lato server in modo automatico. Le espressioni regolari utilizzate per la validazione si trovano in *lib/regExp.php* e sei libero di modificarle secondo le tue esigenze.

L'utilizzo di questi tre metodi ha il vantaggio dell'estrema rapidità, ma lo svantaggio di non poter intervenire minimamente sulla struttura del form (i campi sono quelli, in quell'ordine, con quelle label).
Vedremo più avanti come personalizzare il form utilizando le API avanzate, ma prima diamo uno sguardo su come implementare la validazione del form lato client.

Come implementare la validazione lato client?

La validazione lato client é basata su jQuery. Dunque o il tuo sito già include la libreria jQuery, oppure bisognerà includerla. Per farlo basterà inserire nell'header del documento il metodo _loadjQuery(), in questo modo:

E per implementare il plugin di validazione, basterà eseguire il metodo *_loadScripts()* nell'header del documento (dopo l'inclusione di jQuery), in questo modo:

Ora la validazione tramite javascript é attiva (per qualsiasi tipo di API utilizzata) come in questo esempio. Le espressioni regolari utilizzate sono sempre quelle contenute nel file *lib/regExp.php*. Easy Form funzionerà perfettamente anche con javascript disattivato, quindi conviene includere il plugin di validazione in ogni caso.

Come implementare le API custom

Con Easy Form è possibile anche implementare dei form personalizzati. Per farlo utilizzeremo il metodo *_customForm()* che passa tre parametri.

- Il titolo del form
- Il testo del bottone di invio
- I campi che intendiamo inserire in forma di array associativo *tipo_di_campo => label*

I tipi di campo disponibili sono i seguenti:

- name
- email
- password
- password2
- url
- phone
- message

Proviamo subito ad implementare un form di registrazione con il metodo *_customForm()*.

Otterremo questo risultato.

Come vedi, di default, tutti i campi sono obbligatori. Per renderli facoltativi, bisogna passare nelle opzioni del costruttore *tipo_campo => false*.
Se ad esempio vuoi che il numero di telefono ed il sito web siano facoltativi, dovrai passare questi parametri al costruttore:

Come vedi, ora non ci sono più gli asterischi su questi campi e sia la validazione lato server che quella lato client non segnalano errori in caso di campo vuoto.

Questo metodo ha una limitazione. I vari tipi di campo possono comparire una sola volta. Non sarà quindi possibile utilizzare phone per il numero di telefono e poi phone ancora per il numero di fax. Questo creerebbe delle ambiguità sugli id e sui nomi, rendendo di fatto inutilizzabile il form. Quindi ogni tipo di campo al massimo una volta.

Extra plugin

In Easy Form ho integrato due plugin. Il primo é il “misuratore di forza di una password” che ho presentato in questo articolo. Per attivarlo non dovrai fare altro che passare al costruttore il

parametro password-test impostato a *true* per ottenere questo risultato.
Il controllo viene applicato sul campo password (che naturalmente deve essere presente nel form).
Ti ricordo che è un plugin molto esigente. Per avere una buona valutazione dovrai avere maiuscole, minuscole, caratteri speciali e numeri.

Il secondo plugin che ho integrato è il limitatore di testo per i campi textarea che ho presentato in questo articolo.
Per attivarlo dovrai passare al costruttore il parametro *message-limit* e come valore il numero massimo di caratteri ammessi per ottenere questo risultato.
Il controllo viene applicato sul campo message (che naturalmente deve essere presente nel form).

Questi due plugin necessitano di *jQuery* e che il metodo *_loadScripts()* sia stato invocato nell'header del documento.

Come implementare ajax su Easy Form

Se desideri passare i dati di un form sviluppato con Easy Form tramite una chiamata asincrona, dovrai passare al costruttore il parametro *ajax*, valore: il percorso al file che riceve ed elabora i dati.
Nella cartella *lib* ho messo un file denominato *ajaxSample.php* per testare le chiamate ajax. Quello che fa é semplicemente stampare a video il contenuto dell'array *$_REQUEST*.
Prova dunque a passare questa opzione al costruttore:

E guarda il risultato.

Quello che Easy Form farà in caso di riuscita della chiamata è contenuto nel file *js/success.js* che praticamente è la funzione di callback della chiamata.
In questo caso fa semplicemente sparire il form e mostrare i dati inviati da ajaxSample.php.
Naturalmente puoi scrivere la funzione che desideri tenendo presente che *myform* si riferisce all'elemento *form* e *data* contiene i dati di risposta della pagina di elaborazione.

Come estendere Easy Form

Come ho detto, Easy Form è fatto per essere usato alla velocità della luce. API semplici, tutto pronto, al prezzo di alcune limitazioni una delle quali é: i campi disponibili sono quelli e non se ne possono aggiungere altri. Il motivo principale é che sarebbe poi impossibile gestire la validazione che è tanto comoda così com'è.
Però... c'é un però.
I campi vincolati (select, radio, check ed alcuni di html5 come range) non necessitano di validazione quindi li possiamo aggiungere senza problemi.
Per farlo dovremo creare dei nuovi tipi di campo alla libreria di Easy Form; in pratica dovremo estendere la classe.

Vediamo ad esempio come creare un form di contatto con nome email messaggio e, prima del campo nome la scelta tra Signor o Signora tramite una select.
Iniziamo a creare il file *easy.form.ext.php* nella cartella di Easy Form.

Includiamo la libreria di base e dichiariamo un'estensione (*formWithSelect*) della classe easy_form:

Ora dichiariamo il costruttore che andrà a sovrascrivere il costruttore della classe parent:

Ed ora il metodo che restituirà la select che ci serve. E' importante per l'associazione *campo -> label* che il campo abbia *id [prefisso][nome_del_metodo_senza_underscore]*. Il prefisso lo troviamo nell'array opt alla voce prefix.
Il nome del metodo deve essere preceduto dall'underscore.

Ora, se vogliamo essere pignoli, possiamo anche sviluppare un metodo che, in caso la pagina venga ricaricata, sappia mantenere il valore della select.

Utilizziamo ora questo metodo all'interno delle option ed otteniamo l'estensione definitiva:

Ora, per ottenere il form con *select-nome-email-messaggio*, procederemo in questo modo.

Includiamo ed istanziamo l'estensione che abbiamo appena scritto, ed in seguito creiamo un custom form, tenendo presente che ora abbiamo a disposizione anche il tipo *myselect*.

Ed otterremo questo risultato.

Conclusione

Come hai potuto vedere Easy Form rende il processo di sviluppo di un form un'operazione molto semplice e veloce. La validazione lato server e lato client è già implementata ed è possibile disporre di plugin supplementari ed del suporto ajax in modo estremamente semplificato.

Come ho detto si può fare molto ma ovviamente non tutto. Easy Form si riferisce infatti a delle esigenze molto generali. Se il form che devi sviluppare ha una struttura particolare, è inutile cercare di adattare questo framework a martellate.

Infine segnalo che con il rilascio della versione stabile di standardLib, Easy Form è stato integrato

come plugin in grado di interagire con i modelli per la registrazione e l'autenticazione utente, rendendo lo sviluppo di queste procedure ancora più semplice e veloce.