

## jQuery Mobile: La gestione avanzata delle liste



Continuiamo la serie di tutorial dedicati a **jQuery Mobile** con questo articolo nel quale svilupperemo una piccola applicazione con tanto di database e chiamate asincrone. Lo scenario sarà l'ottimizzazione per dispositivi mobili della [pagina degli autori](#) di YIW. Il risultato finale é visualizzabile su [questa pagina](#) di esempio (per semplificazione ho considerato solo tre autori).

Ora mettiamoci al lavoro.

### Creazione del database

Iniziamo con il creare un nuovo database che chiameremo *mobile* ed al suo interno la tabella autori che comprenderà i seguenti campi:

- id
- nome
- descrizione
- immagine
- sito
- ruolo

Popoliamo la tabella autori con i seguenti dati:

## Creare il file di connessione

Creiamo ora il file *db\_config.php* che conterrà i parametri di connessione al nostro database:

## La classe per la gestione dell'applicazione

Creiamo un nuovo file denominato *authorList.class.php*. La classe che svilupperemo in questo file dovrà fornire i metodi per **estrarre la lista degli autori dal database e mostrare i dettagli di ogni singolo autore**.

Iniziamo ad includere il file con i parametri di connessione al database e a dichiarare la classe *authorList*:

Come vedi abbiamo anche dichiarato due proprietà; una privata (*conn*) che conterrà la risorsa di connessione ed una pubblica (*details*) che conterrà l'oggetto con i dettagli dell'autore.

Ora dichiareremo il metodo per la connessione al database che invocheremo nel costruttore di classe, in questo modo:

A questo punto lasciamo in sospeso questo file ed iniziamo a sviluppare la pagina principale.

## Sviluppare la pagina principale

Creiamo un nuovo file e chiamiamolo *autori.php*. Iniziamo adesso con l'includere le librerie necessarie:

Nel body inseriamo l'header della pagina:

Ora dovremo inserire il contenuto che nella fattispecie é la lista degli autori. Andremo ovviamente a leggere i dati che ci servono dal database, ma prima vediamo quali sono le convenzioni del markup per ottenere una lista come quella che vogliamo (ovvero con l'immagine dell'autore).

Abbiamo visto nell'[articolo precedente](#) che le liste devono avere l'attributo *data-role="listview"*.

Anche nel nostro caso dovremo procedere in questo modo; per quello che riguarda l'immagine basterà inserirla nell'elemento di lista. **jQuery Mobile** provvederà a ridimensionarla e a posizionarla correttamente.

Vediamo un esempio di come va dichiarata la nostra lista (lo vediamo in modo statico, poi svilupperemo il metodo che la produrrà in modo dinamico).

Otterremo questo risultato:

Chiaramente a noi serve che la lista sia costruita in base alle informazioni contenute nel database. E' giunto il momento di tornare sul file *authorList.class.php*.

## Sviluppare il metodo per la visualizzazione della lista autori

Il metodo *showList()* dovrà semplicemente leggere le informazioni di tutti gli autori e creare dinamicamente la lista con il formato che abbiamo visto precedentemente.

Nota una cosa importante: ad ogni link assegniamo un id che praticamente corrisponde all'id dell'autore. **Questo servirà in seguito per avere un riferimento univoco da inviare al server nella chiamata ajax.**

Già che siamo su questo file, sviluppiamo anche l'altro metodo, quello che fornirà i dettagli dell'autore scelto.

Come vedi selezioniamo tutti i dati dell'autore il cui id é uguale all'id passato nella variabile GET. In seguito inseriamo l'oggetto risultante nella proprietà *details*.

Alla fine del documento istanziamo la classe, in questo modo:

Ed ecco la classe completa

Ora andiamo ad inserire la lista dinamica nella pagina principale (*autori.php*) in questo modo:

A questo punto la pagina principale dovrebbe essere funzionante.

## Sviluppare la pagina di supporto

Sviluppiamo ora la pagina di supporto (*result.php*), quella alla quale sarà indirizzata la richiesta ajax e che **restituirà i dettagli dell'autore scelto**.

Iniziamo con l'includere la classe *authorList* ed invocare il metodo *showAuthor()*.

In seguito l'header della pagina

E infine il contenuto della pagina che creeremo dinamicamente grazie all'oggetto *details* che contiene tutte le informazioni necessarie.

Ecco la pagina *result.php*

## Implementare la chiamata ajax

Ora non ci resta che far arrivare alla pagina di dettaglio *result.php* l'id dell'autore che é stato scelto (all'evento click); iniziamo a rilevarlo.

Ora che abbiamo l'id dell'autore che é stato scelto possiamo eseguire la chiamata ajax. Lo faremo tramite un metodo specifico di **jQuery Mobile** ovvero *changePage()*.

Grazie a questo metodo possiamo caricare il contenuto richiesto. **jQuery Mobile** provvederà ad inserirlo nella pagina utilizzando lo stesso modello che abbiamo visto nel [primo tutorial](#).

Vediamo come implementare questo metodo:

Come vedi é piuttosto semplice:

- il primo parametro é rappresentato da una serie di valori (url, type e data come in una classica chiamata ajax);
- il secondo parametro indica il tipo di transizione; in questo caso é inutile dal momento che *slide* é la transizione di default.

Ancora un ultimo accorgimento: Traduciamo il bottone di ritorno (back) in italiano.

Ed ecco la pagina *autori.php* completa:

## jQuery Mobile: Conclusione

In questo tutorial abbiamo visto un'applicazione concreta di **jQuery Mobile**. Il risultato é certamente interessante é può naturalmente essere personalizzato senza alcun limite estendendo o modificando il foglio di stile.

Un ultimo appunto importante. Avremmo potuto anche fare diversamente. Ad esempio, utilizzando il modello che abbiamo visto nel [primo tutorial](#), avremmo potuto caricare la lista ed i dettagli nello stesso documento e collegarli con un sistema di link interni. Sebbene funzionerebbe benissimo, questo sarebbe un errore. Ottimizzare per i dispositivi mobili significa anche prestare attenzione alla quantità di dati trasferiti, che ovviamente devono essere il meno possibile. La soluzione che abbiamo visto, in questo senso, é la più razionale. **I dettagli vengono caricati solo quando é richiesto.**

Il prossimo tutorial sarà più semplice ma non meno interessante. Vedremo infatti l'eccellente supporto fornito da **jQuery Mobile** allo sviluppo di form.