Koha, JavaScript e jQuery

Stefano Bargioni, Pontificia Università della Santa Croce, Koha Gruppo Italiano

2015-01-21; ultima modifica 2015-01-22

Come apportare modifiche o aggiungere funzionalità alle pagine di Koha utilizzando alcune preferenze di sistema

> N.B. Quanto segue è già spiegato molto essenzialmente, in inglese, in
> http://wiki.koha-community.org/wiki/JQuery_Library
> dove vengono riportati numerosi script pronti da utilizzare o da cui imparare come farne altri. E dove contribuire con i propri, volendo.

1. La potenza dell'interfaccia web

Koha lavora completamente tramite browser web. Questo implica che tutte le sue pagine sono codice HTML, CSS e JavaScript leggibile, e su cui si può intervenire una volta che la pagina è stata caricata dal browser. Per poter effettuare modifiche, bisogna far giungere al browser altro codice JavaScript, scritto da noi e aggiunto a Koha stesso (vedi par. 2) , in modo che Koha lo unisca alla pagina o pagine su cui ci interessa intervenire.

Il codice che scriveremo potrà interagire con gli elementi della pagina, modificando testi, posizioni degli oggetti, colori, ecc., e anche aggiungere funzionalità come il recupero di informazione da fonti esterne.

2. Le preferenze di sistema

Anzitutto va tenuto presente che Koha fornisce le due preferenze di sistema OPACUserJS e IntranetUserJS proprio per questo scopo. Sono aree di testo in cui scrivere il codice JavaScript da aggiungere a Koha.

È importante notare che, trattandosi di configurazioni di Koha, non verranno perdute in seguito a aggiornamenti di Koha stesso. Tuttavia va tenuto presente che Koha potrebbe cambiare i nomi o le classi degli elementi della pagina web (par. 3) con cui individuare i punti della pagina su cui intervenire.

All'interno delle suddette preferenze, va aggiunto obbligatoriamente:

```
$(document).ready(function(){
      il nostro codice JavaScript va qui
});
```

`$(document).ready` serve per indicare al browser che il nostro codice JavaScript va eseguito solo dopo che tutta la pagina sarà caricata dal browser.

3. Individuare i punti della pagina su cui intervenire: la console web

Spesso si vuole cambiare un testo in un altro, o modificarne la posizione, ecc. Come si individua il punto su cui intervenire? I browser moderni offrono la voce "Analizza elemento" o "Inspect element" nel menù contestuale che appare cliccando con il tasto destro sul punto che ci interessa.

In genere si apre una finestra -la console web, composta di più sezioni- che mostra il codice HTML relativo all'elemento della pagina a cui agganceremo l'intervento. Spesso questo elemento ha un **id**, o un **class**. che ci permette di richiamarlo e quindi di manipolarlo.

Nella sezione Console si possono dare dei comandi. È la zona della console web che si usa per preparare e provare il nostro codice JavaScript. È anche la zona dove vengono mostrati errori eventualmente causati dal codice JavaScript, nostro o altrui.

4. jQuery

Per semplificare il lavoro, ci viene incontro una libreria JavaScript che Koha stesso include nelle sue pagine web. Si tratta di jQuery (http://www.jquery.com), che ha tra le caratteristiche fondamentali quella di comportarsi allo stesso modo per ogni browser, e quella di facilitare l'individuazione degli elementi della pagina, tramite i "selettori".

È molto utile imparare i selettori di jQuery, tramite i tutorial disponibili in rete.

Il selettore più semplice è quello che utilizza l'**id**: `$('#id')`. Questo comando restituisce un solo elemento, dato che l'id deve essere univoco. Se si usa invece `$('.class')` si ottiene un gruppo di elementi su cui intervenire, che condividono la stessa **class**, cioè appartengono a un gruppo.

Normalmente, a un selettore si applicano delle azioni, che si applicano a tutti gli elementi individuati dal selettore. Il metodo per farlo è scrivere dopo il selettore che cosa si intende fare:

`$('#id').css('text-weight','bold')` mette in neretto l'elemento

`$('#id').text('nuovo testo')` sostituisce il testo

`$('#id').val('nuovo valore')` cambia il valore di un campo

`$('.class').hide()` nasconde un gruppo di elementi; è come cancellarli.

Le operazioni disponibili sono numerose, e sono elencate sui tutorial di jQuery. Possono essere impilate o concatenate:

`$('#id').css('text-weight','bold').text('mio testo')` mette in neretto l'elemento e sostituisce il testo.

5. La pagina su cui lavorare

Un identificatore o una classe possono essere comuni a più pagine di Koha. Non sempre interessa che il nostro codice intervenga su più pagine di Koha, anzi. Per limitare il nostro codice JavaScript a una determinata pagina, conviene circondarlo da un "if" del tipo:

```
if (document.location.href.indexOf('...')>-1) {
      il nostro codice JavaScript va qui
}
```

dove al posto dei puntini si dovrà mettere qualche elemento caratteristico dell'URL della pagina. Esempio: '`opac-search.pl`' se si intende lavorare sulla pagina della ricerca avanzata.

6. Aggiungere informazione da siti esterni, o da Koha stesso

Sono ormai numerosi i siti esterni che offrono informazione bibliografica o di altro tipo utile per gli

OPAC o per il lavoro dello staff, soprattutto della catalogazione. In genere si tratta di grandi biblioteche, di Wikipedia o DBpedia, di OCLC, della Library of Congress, ecc. Molti sono gratuiti. Spesso offrono le cosiddetteAPI, tramite le quali permettono di eseguire interrogazioni a cui rispondono inviando dati in codice XML o JSON.

jQuery facilita molto il lancio dell'interrogazione e il trattamento della risposta. Va costruito l'URL del sito da interrogare, con i parametri che normalmente dipendono dai dati della nostra pagina. Poi si lancia l'interrogazione e si manipolano i dati della risposta per inserirli nella pagina web di Koha da arricchire.

Un esempio generico:

```
var url = '...';
$.getJSON(url,function(risposta_in_json) {
     $('#id').text(parte della risposta_in_json)
});
```

Per completezza, andrebbe anche controllata l'eventuale risposta di errore.

Un paio di esempi concreti:

http://wiki.koha-community.org/wiki/JQuery_Library#Add_iDreamBooks.com_Readometer_to_OPAC
per aggiungere il "Readometer" di iDreamBooks.com nell'OPAC, oppure

http://wiki.koha-community.org/wiki/JQuery_Library#Add_VIAF_autosuggest_for_new_NAME_authorities

per l'aggiunta dell'autocompletamento con dati del VIAF alla pagina della catalogazione di un authority record di tipo Personal Name.

7. Quando il codice di una preferenza di sistema diventa molto lungo

Per evitare di avere troppo codice JavaScript dentro IntranetUserJS o OPACUserJS, si potrebbe caricare un file JavaScript da salvare sul server di Koha e in cui mettere tutto il codice con cui arricchiremo Koha. Per esempio, nella preferenza di sistema IntranetNav mettere:

```
<script src="/myIntranet.js"></script>
```

e nel file myIntranet.js, che deve essere salvato sul server Koha nella directory intranet/htdocs, si metterà il nostro codice, diviso per pagine su cui interviene:

```
$(document).ready(function(){
    if (document.location.href.indexOf('pagina1.pl')>-1) {
     il nostro codice JavaScript va qui
    }
    if (document.location.href.indexOf('pagina2.pl')>-1) {
     altro nostro codice JavaScript va qui
    }
})
```

Analogamente per l'OPAC, nella preferenza di sistema opacheader mettere:

```
<script src="/myOpac.js"></script>
```

e nel file myOpac.js, che deve essere posizionato dentro il server Koha nella directory opac/htdocs, si metterà tutto il nostro codice, diviso per pagine su cui interviene:

```
$(document).ready(function(){
    if (document.location.href.indexOf('pagina1.pl')>-1) {
     il nostro codice JavaScript va qui
    }
    if (document.location.href.indexOf('pagina2.pl')>-1) {
     altro nostro codice JavaScript va qui
    }
})
```

7. La cache del browser

Va tenuto presente che la cache del browser conserva spesso la vecchia versione di un file JavaScript. Per fare la prova di una modifica che abbiamo apportato, specialmente nel caso del par. 6, bisogna svuotare la cache del browser, o fare un super-reload. Con Firefox e Google Chrome, usare maiusc-ctrl-R (su Mac: maiusc-cmd-R).

File originale kohaJavascriptJquery.odt