

MultiSelector ReactJS

Introduzione

Il componente React da realizzare può essere descritto in prima analisi come un selettore multiplo di oggetti eterogenei; con il termine *eterogenei* si intende il fatto che ogni istanza del componente può consentire la selezione di tipologie diverse di oggetti (utenti, gruppi, articoli, prodotti, ...) ma all'interno della stessa istanza la tipologia sarà presumibilmente la stessa.

Il componente dovrà essere realizzato con ReactJS per la parte frontend e con Java 8 per quella backend; non è richiesta una connessione al database (pertanto i dati possono anche essere cablati nel codice).

Funzionamento

La schermata iniziale del widget presenta un conteggio degli elementi (che all'inizio è ovviamente 0) abbinato ad un messaggio di testo personalizzabile *Elenco ruoli vuoto*.

Dopodiché viene visualizzato un titolo personalizzabile (*Associa ruoli*) che rimane fisso all'interno del componente.

Infine viene visualizzato il campo di input *Filtra nome ruolo* attraverso il quale avviare la ricerca degli oggetti.

Il pulsante *Crea gruppo* non fa parte del componente e pertanto può essere ignorato.

Iniziando a digitare caratteri nel campo di input *Filtra nome ruolo* il componente deve invocare un servizio REST per ottenere un elenco di risultati in modalità *autocomplete* (vale a dire che mentre l'utente digita i caratteri, il componente presente un primo elenco dei risultati che si affina man mano che l'utente continua a digitare); l'elenco dei risultati è un array JSON con un formato che vedremo più avanti.

Per ogni risultato selezionato nell'autocomplete, viene visualizzato in alto un box relativo all'elemento selezionato (ad es. *Nome ruolo 1*) ed il messaggio di conteggio cambia in conformità al numero di elementi via via selezionati; ciascun box è inoltre dotato di un pulsante di cancellazione per rimuovere l'elemento (il messaggio di conteggio cambierà di conseguenza).

Nel momento in cui gli elementi selezionati superano un certo valore, al posto del box viene visualizzato un elemento grafico che indica il numero di elementi che *non* sono correntemente mostrati; ad esempio +2 significa che ci sono altri 2 elementi selezionati ma non visualizzati; questo nuovo elemento grafico è inoltre dotato di un'icona con 3 puntini.

Come detto sopra, nel momento in cui gli elementi selezionati superano un certo valore, viene visualizzato un nuovo elemento grafico dotato di un'icona con 3 puntini.

Cliccando sull'icona la lista viene espansa e vengono mostrati tutti gli elementi selezionati.

Un volta che lista viene espansa, non c'è modo di tornare nella visualizzazione compatta.

Configurazione del widget

Il componente dovrà prevedere i seguenti parametri di configurazione.

| Nome | Tipo | Default | Obbligatorio | Descrizione |
|----------------------|--------|------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| autocompleteKeywords | String | 'keywords' | no | Nome del parametro della URL inviato dall'autocomplete al servizio REST di ricerca. |
| autocompleteLabel | String | 'label' | no | Nome del campo, del singolo oggetto JSON restituito dall'autocomplete, che rappresenta la label da visualizzare a video |
| autocompleteURL | String | ' ' | sì | URL del servizio REST di ricerca; i risultati sono rappresentati da un array JSON i cui elementi sono oggetti JSON aventi unicamente 2 campi: uno per la label ed uno per il valore univoco |
| autocompleteValue | String | 'value' | no | Nome del campo, del singolo oggetto JSON restituito dall'autocomplete, che rappresenta il valore univoco (tipicamente la chiave primaria) |

| | | | | |
|------------------|--------|----------------------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| emptyListMessage | String | ' ' | no | Messaggio (già tradotto) da visualizzare in caso di lista delle entità vuota Ad esempio: <i>Elenco ruoli vuoto</i> |
| entries | Array | [] | no | Lista delle entità selezionate con cui prevalorizzare eventualmente il widget; deve essere un array JSON i cui campi rispettano i parametri autoCompleteLabel ed autoCompleteValue |
| hiddenInputId | String | 'multiSelectorHiddenInput' | no | Identificativo del nodo del DOM che rappresenta il campo di input nascosto in cui vengono memorizzati tutti i valori degli oggetti selezionati ai fini del submit del form; i valori sono separati da virgola |
| maxEntries | Number | 0 | no | Numero massimo di elementi selezionati visualizzabili; superato il limite viene visualizzata un'icona con 3 puntini al posto degli elementi mancanti |
| maxResults | Number | 10 | no | Numero massimo di risultati restituiti dall'autocomplete |
| minQueryLength | Number | 3 | no | Numero minimo di caratteri da digitare nell'autocomplete per avviare la ricerca |
| placeholder | String | ' ' | no | Messaggio (già tradotto) da utilizzare come placeholder del campo di input dell'autocomplete Ad esempio: <i>Filtra nome ruolo</i> |
| title | String | ' ' | no | Messaggio (già tradotto) da utilizzare come titolo del componente Ad esempio: <i>Associa ruoli</i> |
| viewListMessage | String | ' ' | no | Messaggio (già tradotto) da utilizzare per visualizzare il messaggio in caso di lista delle entità contenente uno o più elementi Ad esempio: <i>Visualizza elenco ruoli</i> |