

Node.js: usare l'autocomplete di jQuery UI in ExpressJS

In questo tutorial vedremo come implementare l'autocomplete sui campi dei form usando jQuery UI in ExpressJS.

jQuery UI prevede che il codice lato server restituisca un array JSON di oggetti aventi le proprietà `value` e `label`. La prima è il valore reale da associare al risultato (ad esempio un ID) mentre la seconda è il testo da visualizzare nel menu a discesa.

In Node.js possiamo semplicemente effettuare una query con l'operatore `$regex` in MongoDB e restituire i risultati nel formato desiderato.

```
'use strict';

const express = require('express');
const router = express.Router();
const Product = require('../models/product');

router.get('/', (req, res, next) => {
  res.render('index');
});

router.get('/autocomplete', async (req, res, next) => {
  const s = req.query.s.trim();
  const results = [];

  try {
```

```

        const products = await Product.find({ title:
{ $regex: s, $options: 'i' } } ).limit(10);
        products.forEach(product => {
            let { id, title } = product;
            results.push({ value: id, label: title
});
        });

    } catch(err) {
        console.log(err);
    }
    res.json(results);
});

module.exports = router;

```

Il parametro s è il termine di ricerca inviato dal form. Con jQuery UI dobbiamo ora scrivere il seguente codice.

```

"use strict";

$(function() {

    $( "#s" ).autocomplete({
        source: function ( request, response ) {
            $.ajax({
                url: "/autocomplete",
                type: 'GET',
                dataType: "json",
                data: {
                    s: request.term
                },
                success: function ( data ) {
                    response( data );
                }
            });
        }
    });
});

```

```
        }
    });
},
select: function ( event, ui ) {
    $( "#s" ).val( ui.item.label );
    return false;
}
});
});
```

L'evento `select` ci permette di inserire il testo selezionato dal menu a discesa nel corrispondente campo del form. In questo callback abbiamo anche accesso alla proprietà `value` usando la sintassi `ui.item.value`.