

Richieste POST

Il prossimo passo è quello di inserire un nuovo elemento nella lista dei **todos**.

L'utente deve poter inserire la descrizione (campo **text**), mentre non ha alcun bisogno di vedere il campo **id** che verrà assegnato dal server. Per quando riguarda **done**, il valore iniziale sarà **false**.

Per inserire valori tramite una pagina Web creiamo un form:

```
<!-- Nota: Il form fa partire una richiesta POST verso
      /todos. Nel corpo della richiesta (body) sara'
      presente un campo chiamato 'text' -->
<form action="/todos" method="POST">
  Todo: <input type="text" name="text">
  <input type="submit" value="Crea nuovo ToDo">
</form>
```

Gestione della richiesta POST

Il server deve avere un punto d'ingresso per gestire questo tipo di richieste:

```
app.post('/todos', function (req, res) {
  // gestione della richiesta

  // il corpo della richiesta dovrebbe
  // essere memorizzato in req.body
  console.log(req.body);

  // purtroppo pero' questa stampa non
  // andra' a buon fine. Per poter
  // gestire il corpo di una richiesta
  // POST e' necessario unaltro modulo:
  // body-parser

  // Infine, torna alla home page
  res.redirect('/');
});
```

body-parser

Per gestire il corpo delle richieste POST è necessario un ulteriore modulo: **body-parser**. Va installato:

```
$ npm install body-parser
```

Importato e configurato:

```
const bodyParser = require('body-parser');

// configurazione tratta dalla documentazione del pacchetto
app.use( bodyParser.urlencoded({ extended: false }) );
```

Infine possiamo utilizzarlo:

```
app.post('/todos', function (req, res) {
  // il corpo della richiesta e'
  // memorizzato in req.body
  console.log(req.body);

  // crea un nuovo todo
  const todo = {
    id: nextId,
    text: req.body.text,
    done: false
  };
  // aggiornando nextId
  nextId++;

  // e lo aggiunge alla lista
  todos.push(todo);

  // Infine, torna alla home page
  res.redirect('/');
});
```