

2 Template

Una volta che abbiamo definito i dati, bisogna trovare un modo per visualizzarli nella pagina.

EJS è uno dei tanti strumenti che permettono di creare template, cioè una sorta di prestampati all'interno dei quali andare ad inserire i dati che di volta in volta vogliamo visualizzare.

Installazione e configurazione

EJS va installato nella directory della nostra applicazione:

```
$ npm install ejs
```

E va configurato nella nostra app (notare che si fa riferimento ad una directory **views** da creare nella directory di lavoro):

```
app.set('views', path.join(__dirname, 'views'));  
app.set('view engine', 'ejs');
```

Passare i dati al client

Immaginando di aver creato la sottodirectory **views** e che al suo interno sia stato creato il file **index.ejs** (vedi sezione successiva), per passare la lista **todos** si utilizza la funzione **render()**:

```
const todos = [ ... ];  
  
app.get('/', function (req, res) {  
  // il nome del file ejs va passato senza percorso e  
  // senza estensione.  
  // Il secondo parametro e' un oggetto che indica  
  // come far riferimento nel client (il file ejs)  
  // ai dati. In particolare stiamo dicendo che  
  // la lista todos (dopo i due punti) sara' indicata  
  // con il nome di 'todos' (prima dei due punti).  
  res.render('index', {  
    todos: todos  
  });  
});
```

Il file EJS

All'interno della sottodirectory **views** inseriremo il file **index.ejs**. Si tratta di un normale file HTML in cui sono aggiunte delle parti che specificano come visualizzare i dati:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <title>ToDo App</title>
  </head>

  <body>
    <h1>ToDo App</h1>

    <ul>
      <!-- Qui i dati si chiamano todos
      perche' cosi' abbiamo indicato
      nella chiamata a render() -->
      <% todos.forEach( function (todo) { %>
        <li>
          <%= todo.text %> - <%= todo.done %>
        </li>
      <% }); %>
    </ul>
  </body>
</html>
```