
Persistencia del modelo utilizando Mongoose y MongoDB

Testing con Jasmine

Jasmine es un framework de desarrollo, guiado por el comportamiento DGC (*behavior-driven development*), para testear código Javascript. El DGC no nos compete ahora, pero podrás encontrar mucha información en wikipedia, si te interesa. Directamente vamos a usarlo principalmente para testear.

En este link <https://jasmine.github.io/> podrás encontrar toda la documentación de Jasmine.

Verás que es muy simple y claro, y una vez que lo instalamos, contamos con todo lo necesario para comenzar a trabajar.

Para agregarlo a nuestro proyecto como dependencia del ambiente de desarrollo, únicamente debemos:

Primero, instalarlo globalmente para poder ejecutarlo directamente desde la terminal.

1) `npm install -g jasmine`

Luego, lo agregamos como dependencia desarrollo.

2) `npm install --save-dev jasmine`

Luego, debemos inicializar el módulo haciendo:

3) `node node_modules/Jasmine/bin/Jasmine init`

Y finalmente, agregamos una tarea en el `package.json`, en la sección de `scripts`:

4) `"scripts": {"test": "jasmine"}`

De esta forma, si ejecutamos `npm test`, se corren todos los tests que tengamos en nuestro proyecto.

Luego de inicializar Jasmine, verás que se crea una carpeta nueva llamada *spec*, que va a contener todos los tests, specs dentro de la terminología DGC. En el path *spec/support/jasmine.json* tenemos el archivo de configuración de Jasmine. En principio no hace falta que lo modifiquemos.

Presenta el siguiente contenido que es un objeto JSON.

```
{
  // Spec directory path relative to the current working dir when
  // jasmine is executed.
  "spec_dir": "spec",
  // Array of filepaths (and globs) relative to spec_dir to include
  // and exclude
  "spec_files": [
    "**/*[sS]pec.js",
    "!**/*nospec.js"
  ],
  // Array of filepaths (and globs) relative to spec_dir to include
  // before jasmine specs
  "helpers": [
    "helpers/**/*.js"
  ],
  // Stop execution of a spec after the first expectation failure in
  // it
  "stopSpecOnExpectationFailure": false,
  // Run specs in semi-random order
  "random": false
}
```

No lo vamos a analizar en detalle, pero a *grosso modo*, vemos que especifica la raíz de la carpeta donde se encontrarán todos los tests y demás configuraciones de la ejecución de ellos; como ser, si queremos que se ejecuten en orden aleatorio o secuencial, o si se frena la ejecución de los tests al primer test fallido.

Para ejecutar los tests, hacemos *npm test* para ejecutar todos los specs de la carpeta *spec*, o bien podemos hacer *jasmine spec/xxx.spec.js* para ejecutar un único archivo de testing (*xxx.spec.js* en este caso).

En este link podrás ver toda la documentación de Jasmine para NodeJS:

<https://jasmine.github.io/setup/nodejs.html>.

No copiar ni publicar