Generadores (práctica)

Tiempo estimado: 25min

El objeto de esta práctica es asentar y consolidar los conceptos presentados en la parte teórica de la lección.

Al finalizarla, el estudiante:

- Habrá instalado y usado el generador express-generator.
- Habrá instalado y usado el generador justo-generator-express.

Objetivos

El objetivo de la práctica es crear la estructura inicial de una aplicación Express mediante un generador con el objeto de facilitar el arranque del proyecto. En el caso del generador de Justo, además, mostrar cómo crear componentes de proyecto con el generador.

Generador express-generator

Primero, vamos a probar el generador oficial de Express. El más sencillo de los dos.

Instalación del generador

Comencemos instalando el generador:

1. Abrir una consola.
2. Instalar el generador globalmente mediante npm:

   ```
   > npm install -g express-generator
   ```
3. Probar el acceso al comando express:

   ```
   > express -V
   ```

Generación de la estructura inicial del proyecto

Ahora, vamos a crear un proyecto de aplicación Express con el generador, con Handlebars como motor de plantillas:

1. Ir a la consola.
2. Crear el directorio de la práctica e ir a él.
3. Mostrar la ayuda del comando:

   ```
   > express -h
   ```
4. Generar el proyecto:

   ```
   > express --hbs
   ```
5. Echar un vistazo detallado a los archivos creados por el generador. Tómese su tiempo.
6. Instalar dependencias:

   ```
   > npm install
   ```
7. Arrancar la aplicación:

   ```
   > npm start
   ```
8. Abrir el navegador.
9. Ir a localhost:3000.

Generador justo-generator-express

En este punto, vamos a probar el generador Express de Justo. La idea es que el estudiante pueda comparar ambos y decidirse por aquel que más le convenza.

Instalación del generador

Recordemos que Justo es una herramienta de automatización que proporciona varias funcionalidades entre las que se encuentran los generadores. Por un lado, hay que instalar justo-cli, que proporciona el comando justo; y por otro lado, el generador de Express, justo-generator-express.

1. Abrir una nueva consola.
2. Instalar Justo y el generador:

   ```
   > npm install -g justo-cli justo-generator-express
   ```
3. Comprobar que se tiene acceso al comando justo:

   ```
   > justo -h
   ```
4. Comprobar que se tiene acceso al generador:

   ```
   > justo -g express help
   ```

Generación de la estructura inicial del proyecto

Ahora, vamos a crear la estructura inicial de un proyecto Express:

1. Ir a la nueva consola.
2. Crear un nuevo directorio de práctica e ir a él.

   Mantendremos el anterior proyecto para poder comprobar posteriormente ambos.
3. Generar la estructura de directorios inicial del proyecto Express:

   ```
   > justo -g express
   ```
   El generador le solicitará información sobre el proyecto. Con parte de ella, rellenará el archivo package.json. Otra la utilizará para configurar componentes en la pila de *middleware* en el archivo app.js.

   El único componente de *middleware* que todavía no se ha presentado, pero que se hace en el siguiente curso de Express, es Helmet. Este componente no hace más que configurar ciertos aspectos de la configuración con objeto de hacerla más segura y menos propensa a ataques. Se recomienda encarecidamente su instalación.
4. Echar un vistazo detallado a los archivos creados por el generador. Tómese su tiempo. Es más, compare el proyecto creado por ambos generadores para ver cuál le parece más completo.
5. Instalar dependencias:

   ```
   > npm install
   ```

Arranque de la aplicación

El generador de Justo crea el proyecto bajo la especificación ES2015 de JavaScript. Por lo que hay que compilar la aplicación a ES5 para que funcione en la mayor parte de las versiones de Node. Por suerte, se puede utilizar Justo para compilar la aplicación. Veamos cómo.

1. Ir a la consola.
2. Liste las tareas automatizadas del proyecto:

   ```
   > justo -c
   ```
3. Compilar la aplicación mediante Babel con la tarea build del proyecto:

   ```
   > justo build
   ```
   Esta tarea lo que hace es compilar el proyecto y generar un paquete de distribución en el directorio dist.
4. Abrir una nueva consola.
5. Ir al directorio del proyecto.

6. Ir al directorio dist/es5/nodejs/ del proyecto.
7. Entrar en el directorio de la aplicación que hay en el directorio actual. Su nombre dependerá del nombre usado por el estudiante para crear el directorio de la práctica
8. Listar el contenido del directorio:

   ```
   > dir
   ```

   Ésta es la versión distribuible del proyecto. Lo que deberíamos instalar en una máquina de producción. No tiene ningún componente instalado. Sólo el código fuente compilado bajo la especificación ES5. Nosotros la vamos a usar para probar nuestra aplicación durante el desarrollo.
9. Instalar dependencias:

   ```
   > npm install
   ```
10. Arrancar la aplicación en modo desarrollo:

    ```
    > npm run start-dev
    ```
11. Abrir el navegador.
12. Solicitar localhost:8080.

A partir de este momento, trabajaremos sobre el directorio del proyecto, no del de distribución. Cuando deseemos probar algo, bastará con ejecutar la tarea build, la cual compilará y dejará el código compilado en el directorio de distribución. El cual se ejecuta bajo nodemon, por lo que los cambios se aplicarán automáticamente. Sólo tenemos que recordar que si instalamos algún componente de software, habrá que abrir una nueva consola, ir al directorio de distribución y solicitar mediante npm update que se instale para que así nodemon pueda trabajar correctamente. nodemon no instala dependencias. Eso debemos hacerlo manualmente.

Creación de rutas

En este punto, vamos a ver cómo crear encaminadores y rutas mediante el generador de Justo.

1. Ir a la consola del proyecto.
2. Solicitar la creación del encaminador band mediante el generador:

   ```
   > justo -g express router
   ```

   Responda como sigue:

   - No indique ninguna carpeta especial, lo que añadirá el archivo del encaminador en la carpeta routes. Para ello, seleccione <none>.
   - Indique como nombre del encaminador band.
   - Indique que se presente la plantilla band/index cuando se solicite la página índice raíz.

   Observe qué ha hecho el generador. Por un lado, ha creado el archivo routes/band.js. El que contiene la definición del encaminador del componente band. Tendremos algo parecido a:

   ```
   //imports
   import express from "express";

   //router
   export const router = express.Router();

   //routes
   router.get("/", function(req, res) {
     const app = req.app;
     res.render("band/index", {});
   });
   ```

   Por otro lado, ha registrado el nuevo encaminador en el archivo routes/map.js. Por favor, échele un vistazo a los dos archivos.

   Es raro actualizar manualmente el contenido del archivo routes/map.js si usamos el generador. En cambio, la modificación de los archivos de los encaminadores es muy común, tanto mediante el

generador como manualmente.

3. Generar una ruta GET para la ruta parametrizada /:name en encaminador band:

```
> justo -g express route
```

Responda como sigue:

- No indique ninguna carpeta especial. Para ello, seleccione /. La barra hace referencia al directorio routes.
- Seleccione el encaminador band, el que está asociado al archivo routes/band.js.
- Indique como ruta :name.
- Indique como vista a presentar band/info.
- Seleccione el método get.

En este caso, lo que hace el generador es crear una ruta en el archivo routes/band.js.

4. Editar el archivo routes/band.js.
5. Echarle un vistazo y observar la ruta recien creada.
6. Añadir la siguiente constante de datos al archivo, por ejemplo, después de la sección de importaciones:

```
//data
const BANDS = {
  "baustelle": {
    name: "Baustelle",
    origin: "Siena, Italy",
    genres: ["indie"]
  },
  "perturbazione": {
    name: "Perturbazione",
    origin: "Rivoli, Italy",
    genres: ["indie"]
  }
};
```

7. Ir al controlador de peticiones /:name y modificarlo como sigue:

```
router.get("/:name", function(req, res) {
  res.render("band/info", {scope: BANDS[req.params.name]});
});
```

8. Guardar cambios.

Creación de plantillas

Ya tenemos creada el encaminador y una ruta de ejemplo. Vamos a ver cómo crear una plantilla con el generador:

1. Generar la plantilla Handlebars band/info.hbs:

```
> justo -g express hbs view
```

Responda como sigue:

- Indique la carpeta views/band/, para añadir en ella todas las vistas relacionadas con las bandas. Para ello, primero, seleccione <other> y, a continuación, indique band.
- Indique info como nombre de vista. No indique la extensión .hbs, sólo su nombre.
- Responda N a la pregunta Is a form?

  Cuando se responde Y, el generador añadirá un elemento HTML <form> inicial a la vista.

Lo que hace el generador es muy sencillo. Primero, comprueba si la carpeta que le hemos indicado existe. Si no existe, la crea. Finalmente, crea la vista.

2. Editar el archivo views/band/info.hbs.
3. Modificar la plantilla:


```
<p>
  Name: {{scope.name}}<br>
  Origin: {{scope.origin}}<br>
  Genres: {{scope.genres}}
</p>
```

4. Guardar cambios.
5. Ir a la consola.
6. Construir el paquete de distribución:

```
> justo build
```

7. Ir a la consola de nodemon y comprobar que ha detectado los cambios generados por justo build.
8. Ir al navegador.
9. Solicitar localhost:8080/band/baustelle.
10. Solicitar localhost:8080/band/perturbazione.

Se recomienda encarecidamente organizar las aplicaciones Express en carpetas. Por lo general, cada una de ellas representa un componente. En el caso de Express, cuando se organiza el código fuente de las vistas y rutas en componentes, para cada componente se crea:

- Un archivo encaminador en la carpeta /routes, el cual además hay que tenerlo registrado en el archivo /routes/map.js. Ambas cosas se pueden hacer mediante el comando router del generador.

  Observe que a cada componente se le asocia su propio archivo encaminador en la raíz de la carpeta routes. Es raro crear una carpeta para un determinado componente, pero cada organización es un mundo.

- Una carpeta específica en el directorio /views. Si la carpeta no existe cuando se crea la primera plantilla mediante el comando hbs view del generador, el propio generador la creará.

