

Introducción a Justo.js (práctica)

Tiempo estimado: 10min

El objeto de esta práctica es afianzar, reforzar y consolidar los conocimientos teóricos presentados en la lección.

Al finalizarla, el estudiante:

- Habrá instalado **Justo.js**.
- Habrá creado los archivos requeridos por un proyecto **Justo**, tanto integrado como independiente.

Objetivos

El objetivo de la práctica es, por un lado, instalar **Justo.js** y, por otro lado, usar el generador de **Justo** para crear proyectos integrados.

Requisitos

Lo primero es instalar las dependencias. La única dependencia que tiene **Justo.js** es **Node.js** y su administrador de paquetes **NPM**. Se debe tener instalada la versión **7** o superior, la cual se puede descargar de su sitio web oficial, nodejs.org.

Instalación de Justo.js

Una vez instalado **Node.js** y **NPM**, podemos pasar a instalar **Justo.js**. Para ello, usaremos el comando **npm**:

1. Abrir una consola de **PowerShell** o terminal *shell* en **Linux**.
De ahora en adelante, nos referiremos a ella como simplemente *consola*.
2. Instalar **justo-cli** globalmente, mediante **npm**:

```
$ npm install -g justo-cli
```
3. Comprobar que tenemos acceso a la utilidad **justo**:

```
$ justo -h
```

Creación de un módulo de Justo

Una vez instalada la utilidad **justo**, vamos a ver cómo crear los archivos **Justo.js** y **package.json**.

Recordemos que existe dos tipos de proyectos donde podemos usar **Justo**: un **proyecto independiente** (*standalone project*) o **módulo** (*module*), autónomo, dedicado exclusivamente a tareas de **Justo** como, por ejemplo, la instalación o configuración de software. Mientras que un **proyecto integrado** (*embedded project*) es aquel que se integra en otro, dedicándose la parte de **Justo** a automatizar ciertas tareas del proyecto principal.

Independientemente de cómo usemos **Justo** en el proyecto, la manera más fácil de crear los archivos requeridos, es mediante algún tipo de generador. Uno de ellos es **justo-generator-justo**:

1. Ir a la consola.
2. Instalar globalmente el generador **justo-generator-justo**:

```
$ npm install -g justo-generator-justo
```
3. Comprobar que el generador se ha instalado correctamente. Para ello, mostrar su ayuda:

```
$ justo -g justo help
```


Para invocar un generador, hay que utilizar la opción **-g** de **justo** y el nombre del generador sin el prefijo **justo-generator-**.

4. Crear el directorio de la práctica e ir a él.
5. Crear el directorio del proyecto de módulo mediante el generador:

```
$ justo -g justo module
```

El generador recolecta información sobre el proyecto **Justo** a crear, antes de crear sus archivos. Para crear un módulo, tal como veremos en una lección específica, hay que utilizar el comando **module** del generador. A continuación, responda a las preguntas que le sean formuladas. No se preocupe, puede inventárselas, pero sea creativo.

Una vez recolectada la información, el generador creará los archivos teniendo en cuenta lo respondido por el usuario.

6. Listar el contenido del directorio.

En proyectos independientes, el generador crea otros archivos, generalmente, muy útiles como, por ejemplo, **.editorconfig**, **.gitignore**, etc.

7. Mostrar el contenido de los archivos:

```
$ cat package.json
$ cat Justo.js
$ cat Justo.dev.js
```

El archivo **Justo.dev.js** se utiliza como catálogo de tareas, similar a **Justo.js**, pero de desarrollo. Ya lo veremos en la lección de módulos. Por ahora, eche un vistazo detallado a los archivos.

En la lección siguiente, comenzaremos a trabajar con tareas y su registro en el archivo **Justo.js**.

Creación de un proyecto integrado de Justo

Ahora, vamos a suponer que usaremos **Justo** para automatizar determinadas tareas de otro proyecto. En este caso, el proceso es similar al anterior, usaremos el generador **justo-generator-justo**, pero no usaremos ninguno de sus comandos, lo invocaremos sin más:

1. Ir a la consola.
2. Vaciar el contenido del directorio de la práctica.
3. Invocar el generador:

```
$ justo -g justo
```

4. Listar el contenido del directorio.

Observe que algunos archivos, generados en la sección anterior, no se han creado en este tipo de proyecto. Generalmente, cuando se usa **Justo** en otro proyecto, primero, se crea la estructura del principal y después los archivos de **Justo**. Sólo hay que recordar que si el proyecto principal ya ha creado el archivo **package.json**, habrá que indicarle al generador que *no* lo genere, porque si lo hace, sobrescribirá su contenido. En este caso concreto, habrá que añadir manualmente el paquete **justo** como dependencia de desarrollo en la propiedad **devDependencies** del archivo **package.json**.

También hay que decir que otros generadores como, por ejemplo, **justo-generator-express**, **justo-generator-horizon**, **justo-generator-node** y **justo-generator-react**, ya crean los archivos de **Justo**. Por lo que *no* habrá que utilizar el generador **justo-generator-justo**.

5. Mostrar el contenido de los archivos **package.json** y **Justo.js**.
6. Cerrar la consola.

En la siguiente lección, empezaremos a trabajar con **Justo**, creando tareas e invocándolas. Por ahora, nos quedamos en su proceso de instalación y la utilización del generador **justo-generator-justo**.