MANUAL EXPRESS

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para:

* Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
* Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
* Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
* Añadir procesamiento de peticiones "middleware"\* adicional en cualquier punto dentro de la tubería de manejo de la petición.

*\* Middleware es software que se sitúa entre un sistema operativo y las aplicaciones que se ejecutan en él.*

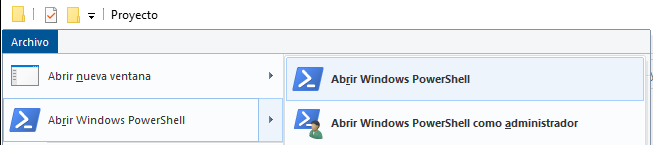
Los frameworks web frecuentemente se refieren a sí mismos como "dogmáticos" o "no dogmáticos"

Express es no dogmático, transigente. Puedes insertar casi cualquier middleware compatible que te guste dentro de la cadena de manejo de la petición, en casi cualquier orden que te apetezca. Puedes estructurar la app en un fichero o múltiples ficheros y usar cualquier estructura de directorios. ¡Algunas veces puedes sentir que tienes demasiadas opciones!

INSTALACIÓN

Para poder utilizar Express, es necesario instalar antes varias cosas, en primer lugar, el Node.js, se puede hacer desde su página web: <https://nodejs.org/en/download/>.

Una vez se ha instalado el Node.js, debemos crear la carpeta en la que almacenaremos el proyecto y abrir la consola de comandos en dicha carpeta



Utilice el comando npm init para crear un archivo package.json para la aplicación.

$ npm init

Este mandato solicita varios elementos como, por ejemplo, el nombre y la versión de la aplicación. Por ahora, sólo tiene que pulsar INTRO para aceptar los valores predeterminados para la mayoría de ellos, con la siguiente excepción:

entry point: (index.js)

Especifique app.js o el nombre que desee para el archivo principal. Si desea que sea index.js, pulse INTRO para aceptar el nombre de archivo predeterminado recomendado.

A continuación, instale Express en el directorio myapp tras crearlo y guárdelo en la lista de dependencias. Por ejemplo:

$ npm install express --save

Para instalar Express temporalmente y no añadirlo a la lista de dependencias, omita la opción --save:

$ npm install express

FUNCIONAMIENTO BÁSICO

Cuando se recibe una petición, la aplicación determina cuál es la acción adecuada correspondiente, de acuerdo a la estructura de la URL y a la información (opcional) indicada en la petición con los métodos POST o GET.

**Esto se tiene que hacer en un archivo .js, el cual se puede modificar con varias aplicaciones, tales como “Notepad++”, “Sublime Text” o directamente el block de notas, lo que prefiráis.**

Primero consideremos el tradicional ejemplo de ¡Hola Mundo! (se comentará cada parte a continuación).

var express = require('express');

var app = express();

app.get('/', function(req, res) {

res.send('Hola Mundo!');

});

app.listen(3000, function() {

console.log('Aplicación ejemplo, escuchando el puerto 3000!');

});

Las primeras dos líneas incluyen (mediante la orden require()) el módulo de Express **(Explicación más adelante)** y crean una aplicación de Express. Este elemento se denomina comúnmente app, y posee métodos para el enrutamiento de las peticiones HTTP, configuración del 'middleware', y visualización de las vistas de HTML, etc.

Las líneas que siguen en el código (las tres líneas que comienzan con app.get()) muestran una definición de ruta que se llamará cuando se reciba una petición HTTP GET con una dirección ('/') relativa al directorio raíz. La función 'callback' coge una petición y una respuesta como argumentos, y ejecuta un send () en la respuesta, para enviar la cadena de caracteres: "Hola Mundo!". En resumidas cuentas, las líneas de código que se escriben entre la selección de módulos y aplicaciones y las líneas que definen el localhost están ahí para especificar las funciones que se llevarán a cabo en el archivo .js, ya sea mostrar el mensaje “Hola mundo” o llevarnos a una página web.

El bloque final de código, define y crea el servidor, escuchando el puerto 3000 e imprime un comentario en la consola. Cuando se está ejecutando el servidor, es posible ir hasta la dirección localhost:3000 en un navegador, y ver como el servidor de este ejemplo devuelve el mensaje de respuesta.

Para poner en marcha el servidor es necesario acceder a la consola de comandos:

node index.js

(index es el nombre del archivo, suele ser ese por costumbre)

Una vez se pone en marcha el servidor, se puede acceder desde cualquier navegador introduciendo su dirección en la barra de búsqueda:

http://localhost:3000/

MODULOS

Un módulo es una librería o archivo JavaScript que puede ser importado dentro de otro código utilizando la función require() de Node. Por sí mismo, Express es un módulo, como lo son el middleware y las librerías de bases de datos que se utilizan en las aplicaciones Express.

Así es como se puede acceder a las propiedades y funciones del Express.

var express = require('express');

var app = express();