MANUAL EXPRESS

https://github.com/expressjs/express

¿Qué es 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.

A pesar de que *Express* es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. Hay librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos `POST`, cabeceras de seguridad y *muchos* más. Puedes encontrar una lista de paquetes middleware mantenida por el equipo de Express en Express Middleware (junto con una lista de algunos de los paquetes más populares de terceros).

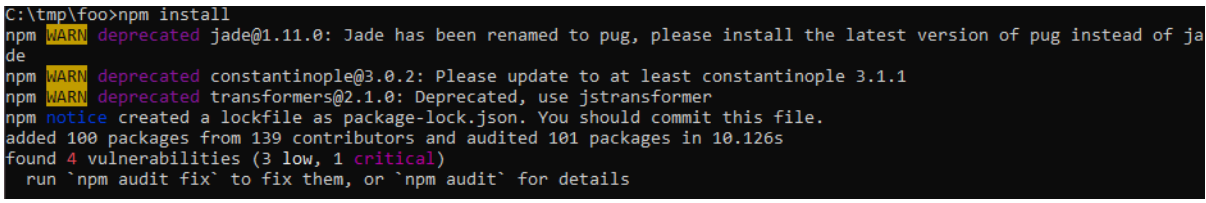
Inicio:

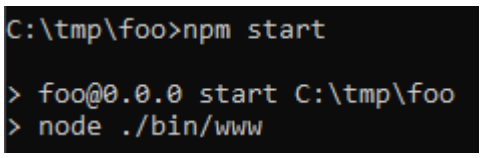
- Instalación del ejecutable:
  - npm install -g express-generator@4
  -

```
C:\Users\IN1DAM-B>npm install -g express-generator@4
npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (No
te that the API surface has changed to use Promises in 1.x.)
C:\Users\IN1DAM-B\AppData\Roaming\npm\express -> C:\Users\IN1DAM-B\AppData\Roaming\npm\node_modules\express-generator
\bin\express-cli.js
+ express-generator@4.16.1
added 10 packages from 13 contributors in 2.256s
```

- Crear la app:
  - express /tmp/foo && cd /tmp/foo
  -

```
C:\Users\IN1DAM-B>express /tmp/foo && cd /tmp/foo

  warning: the default view engine will not be jade in future releases
  warning: use `--view=jade' or `--help' for additional options


   create : \tmp\foo\
   create : \tmp\foo\public\
   create : \tmp\foo\public\javascripts\
   create : \tmp\foo\public\images\
   create : \tmp\foo\public\stylesheets\
   create : \tmp\foo\public\stylesheets\style.css
   create : \tmp\foo\routes\
   create : \tmp\foo\routes\index.js
   create : \tmp\foo\routes\users.js
   create : \tmp\foo\views\
   create : \tmp\foo\views\error.jade
   create : \tmp\foo\views\index.jade
   create : \tmp\foo\views\layout.jade
   create : \tmp\foo\app.js
   create : \tmp\foo\package.json
   create : \tmp\foo\bin\
   create : \tmp\foo\bin\www

   change directory:
     > cd /tmp/foo

   install dependencies:
     > npm install

   run the app:
     > SET DEBUG=foo:* & npm start
```

- Instalación de las dependencias:
    - npm install
    - 

- Iniciar el servidor:
    - npm start
    - 

ESTRUCTURA DEL JS PARA INICIAR UN SERVIDOR

```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

Métodos de ruta

- app.get
- app.post
- app.put
- app.delete

```
// GET method route
app.get('/', function (req, res) {
  res.send('GET request to the homepage');
});

// POST method route
app.post('/', function (req, res) {
  res.send('POST request to the homepage');
});
```

PRUEBA:

Hola mundo:

```
PS C:\Users\IN1DAM-B\Desktop\myapp> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (myapp)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\IN1DAM-B\Desktop\myapp\package.json:

{
  "name": "myapp",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
PS C:\Users\IN1DAM-B\Desktop\myapp> npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN myapp@1.0.0 No description
npm WARN myapp@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 5.267s
found 0 vulnerabilities

PS C:\Users\IN1DAM-B\Desktop\myapp> npm install express
npm WARN myapp@1.0.0 No description
npm WARN myapp@1.0.0 No repository field.

+ express@4.17.1
updated 1 package and audited 50 packages in 1.759s
found 0 vulnerabilities
```

luego ejecutamos el comando node app.js y entramos en localhost