

EXPRESS.JS

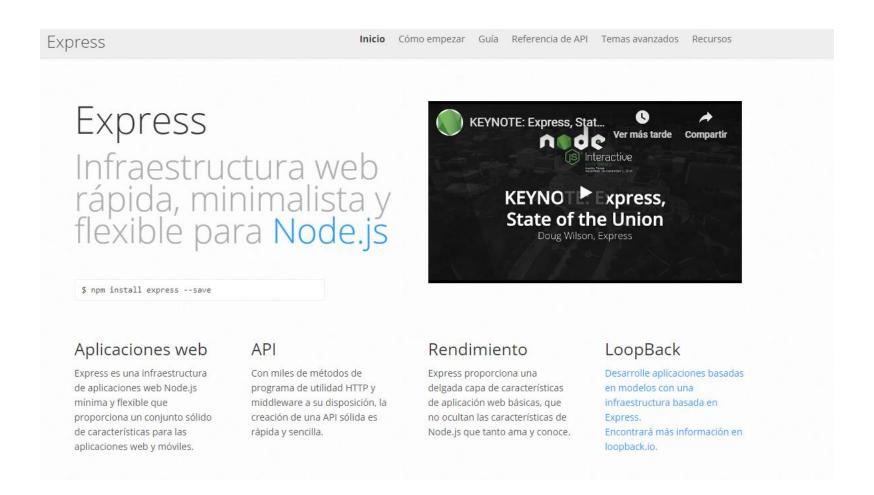


Contenido

- Motores de plantillas
- Middlewares de terceros



EXPRESS





EXPRESS

- Creamos un proyecto con archivo app.js y generamos package.json
- Instalamos express: npm install express --save
- De la página oficial bajamos el ejemplo "hola mundo":
 - https://expressjs.com/es/starter/hello-world.html



Utilización de motores de plantilla con Express

- Para que Express pueda representar archivos de plantilla, deben establecerse los siguientes valores de aplicación:
 - views, el directorio donde se encuentran los archivos de plantilla. Ejemplo:

```
app.set('views', './views')
```

- view engine, el motor de plantilla que se utiliza. Ejemplo: app.set('view engine', 'pug')
- Instalar el paquete npm de motor de plantilla correspondiente:: \$ npm install pug --save
- Crear un archivo de plantilla Pug index.pug en el directorio views, con el siguiente contenido:

```
html
head
title= title
body
h1= message
```

Crear una ruta para representar el archivo index.pug:

```
app.get('/', function (req, res) {
  res.render('index', { title: 'Hey', message: 'Hello
  there!'}); });
```



Middlewares de terceros

Middleware	Descripción
body-parser	Parsea el cuerpo de la petición entrante: req.body property.
compression	Comprime el cuerpo de la respuesta HTTP
cookie-parser	Parsea la cabecera "Cookie" y genera la propiedad req.cookies. Permite manejar cookies firmadas
cookie-sesión	Almacena todos los datos de sesión de cliente en una cookie
express-debug	Añade a la aplicación un separador con información sobre las variables de plantilla (locals), la sesión actual, datos de solicitud útiles, etc.
express-session	Almacena el id de session en la cookie y el resto de información en el servidor
express-simple-cdn:	Utiliza una CDN (Red de entrega de contenido) para los activos estáticos, con soporte de varios hosts (por ejemplo, cdn1.host.com, cdn2.host.com).
helmet:	Protege las aplicaciones estableciendo varias cabeceras HTTP.
method-override	Anula las cabeceras que no permiten ciertos métodos HTTP (PUT, DELETE)
morgan: anteriormente logger	Genera un log de la aplicación
Passport	Autentica la aplicación mediante diferentes estrategias
serve-static:	Sirve contenido estático



Servicio de contenido estático con Express

- Se utiliza el middleware express.static para servir archivos estáticos como, por ejemplo, imágenes, archivos CSS y archivos JavaScript
- Se pasa el nombre del directorio que contiene los activos estáticos a la función de middleware express.static para empezar directamente el servicio de los archivos. Por ejemplo: Cargar ficheros de carpeta /public

```
app.use(express.static('public'));
```

Así se cargaría:

```
http://localhost:3000/images/kitten.jpg
http://localhost:3000/css/style.css
http://localhost:3000/js/app.js
http://localhost:3000/images/bg.png
http://localhost:3000/hello.html
```



Referencias

- Libros:
 - Getting MEAN with Mongo, Express, Angular, and Node, Manning Publications, early 2015.
 - Express.js in Action, Manning Publications, early 2015.
 - Mastering Web Application Development with Express, Packt Publishing,
 September 2014.
 - Web Development with Node and Express, O'Reilly Media, July 2014.
 - Node.js in Action, Manning Publications, October 2013.
 - Express Web Application Development, Packt Publishing, June 2013.