

# Express

Express es un framework para programar servidores web y sus APIs REST con Node.JS

Para iniciar una nueva aplicación de express, instalad express con npm y probad el siguiente ejemplo como script principal:

```
var express = require('express');
var http = require('http');

var app = express();

app.get('/', function(req, res) {
  res.send("Hello world!");
});

http.createServer(app).listen(8080);
```

Si poneis la url `http://localhost:8080/` en el navegador, veréis el mensaje: Hello World!

Simple, no? Ahora veremos como podemos extender este servidor.

Las dos principales herramientas que nos ofrece el módulo Express son el **middleware** y el **routing**.

## Middleware

El middleware son las funciones que se ejecutan durante una request a nuestro servidor.

Porqué son útiles?

- Hacer logs de las requests
- Autenticación
- Parsear headers y body de la request
- Cargar objetos de la base de datos

Ejemplo:

```

var bodyParser = require('body-parser');
var winston = require('winston');
var expressWinston = require('express-winston');

// app.use es una función que sirve para especificar middleware a ejecutar
// durante la petición. Recibe como parámetro una función, el middleware
// En este caso le pasamos una función que parsea el body de la request
// interpretandolo en formato json
app.use(bodyParser.json());

// En este caso, este middleware loguea por consola todas las peticiones que
// lleguen
app.use(expressWinston.logger({
  transports: [
    new winston.transports.Console({
      colorize: true
    })
  ],
  meta: false,
  msg: "HTTP {{req.method}} {{req.url}} {{res.statusCode}}
{{res.responseTime}}ms",
  colorStatus: true
}));

// El middleware es una función que recibe tres parametros
// Los dos primeros son la request y la response de la petición
// El tercero es una función que se llama para que se sigan ejecutando
// los siguientes middlewares
// Si un middleware no ejecuta next debe responder la petición,
// sino esta se quedará colgada
app.use(function(req, res, next) {
  console.log("Hola yo solo escribo por pantalla");
  next();
});

```

## Routing

Routing se refiere a la definición de los puntos de llegada (caminos de url) de nuestra API y como se responden las peticiones de los clientes.

Una ruta es una combinación de una URI, un tipo de petición HTTP y una o más funciones de tratamiento de las peticiones.

Un ejemplo de función a ejecutar cuando una request con el verbo (i.e tipo de petición HTTP) se hace en el camino '/', es decir cuando abrimos el navegador con la url:  
<http://localhost:8080/>

```

app.get('/', function(req, res, next) {
  res.status(200).send('Hi from express!')
});

```

Por ejemplo, si queremos crear un nuevo usuario (registrar), crearíamos la siguiente ruta:

```
app.post('/users', function(req, res, next) {  
  var userData = req.body;  
  // Aquí pondríamos el código necesario para guardarlo en la bd  
  ...  
  res.status(200).send(userData);  
});
```

Para más info, puedes visitar estos enlaces:

- <http://expressjs.com/guide/using-middleware.html>
- <http://expressjs.com/guide/routing.html>
- <https://github.com/bithavoc/express-winston>
- <https://github.com/winstonjs/winston>