

Cómo escribir a un log el registro de tu aplicación nodejs con express

Normalmente el archivo **app.js** de tu aplicación express dispondrá de un error handler por defecto que suele ser como éste:

```
// error handler
app.use(function (err, req, res, next) {
// set locals, only providing error in development
res.locals.message = err.message;
res.locals.error = req.app.get('env') === 'development' ? err
: {};

// render the error page
res.status(err.status || 500);
res.render('error');
});
```

Podemos crear un log personalizado con datos diversos. Vamos a poner un ejemplo usando **morgan** para obtener el **access-log** de lo que estamos haciendo y los **errores de la aplicación** simultáneamente. Este log no es recomendable para una aplicación en producción, asegúrate de que lo eliminas o lo limitas cuando termines.

En primer lugar nos situamos en la raíz de la aplicación e instalamos morgan:

```
cd nodejs/app/ ; npm install morgan --save
```

Después sólo tenemos que añadir las siguientes líneas , junto con el handler modificado (sólo una línea). A partir de este momento encontraremos un log

bastante detallado en el archivo node.log

```
//////////////////////////////////// Configuración para Morgan y creación
del stream node.log
loggerHTTP = require('morgan');
var fs = require('fs'); var util = require('util');
var log_file = fs.createWriteStream(__dirname + '/node.log',
{flags : 'a'});
app.use(loggerHTTP({stream: log_file}));
app.use(loggerHTTP('dev'));
////////////////////////////////////

// error handler
app.use(function (err, req, res, next) {
// set locals, only providing error in development
res.locals.message = err.message;
res.locals.error = req.app.get('env') === 'development' ? err
: {};

// Escribimos el error
log_file.write(err.stack)

// render the error page
res.status(err.status || 500);
res.render('error');
});
```