

REGISTRO DE EVENTOS (PRÁCTICA)

Tiempo estimado: 15min

El objeto de esta práctica es asentar y consolidar los conceptos presentados en la parte teórica de la lección.

Al finalizarla, el estudiante:

- Habrá configurado varios registros de eventos.

objetivos

En esta práctica, vamos a crear una aplicación **Express** que tenga configurado tres registros de eventos: el archivo **error.log**, el archivo **access.log** y el registro de eventos de la consola para ayudarnos durante el desarrollo.

creación del proyecto

Vamos a crear el proyecto con el que trabajar:

1. Abrir una consola.
2. Crear el directorio de la práctica.
3. Ir al directorio de la práctica.
4. Crear el archivo **package.json** con el siguiente contenido:

```
{
  "name": "express-app",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "start": "node ./app.js"
  },
  "dependencies": {
    "express": "*"
  }
}
```

5. Crear el archivo **app.js** con el siguiente contenido:

```
"use strict";

//imports
const express = require("express");
const http = require("http");

//app
const app = express();

//config app
app.get("*", function(req, res) {
  res.send("Página solicitada: " + req.originalUrl);
});

//listen
http.createServer(app).listen(8080, function() {
  console.log("Listening...");
});
```

6. Instalar dependencias:

```
> npm install
```

7. Iniciar la aplicación **Express**:

- > npm start
- 8. Abrir el navegador.
- 9. Ir a <http://localhost:8080>.
- 10. Ir a <http://localhost:8080/the-national>.

CONFIGURACIÓN DE REGISTROS DE EVENTOS

Vamos a configurar un registro de eventos a la consola:

1. Editar el archivo `package.json`.
2. Añadir la dependencia `morgan` al archivo `package.json`:

```
"dependencies": {  
  "express": "*",  
  "morgan": "*"   
}
```

3. Guardar cambios.
4. Ir a la consola.
5. Detener la aplicación.
6. Instalar la nueva dependencia:

```
> npm update
```

7. Editar el archivo `app.js`.
8. Importar el módulo `morgan` como `logger`:

```
//imports  
const express = require("express");  
const http = require("http");  
const logger = require("morgan");
```

9. Añadir el siguiente registro al comienzo de la pila de *middleware*, antes de `app.get()`:

```
app.use(logger("dev"));
```

10. Guardar cambios.
11. Ir a la consola.
12. Iniciar la aplicación.
13. Ir al navegador.
14. Ir a <http://localhost:8080>.
15. Ir a <http://localhost:8080/nada-surf>.
16. Ir a la consola.
17. Comprobar que aparecen las entradas del registro para las peticiones realizadas.

Archivo de registro `ACCESS.LOG`

Ahora, vamos a configurar un registro adicional, el archivo `access.log` para que registre los recursos solicitados por los usuarios:

1. Crear el directorio `logs` en la aplicación.
2. Editar el archivo `app.js`.
3. Importar los módulos `path` y `fs`:

```
//imports  
const express = require("express");  
const http = require("http");  
const logger = require("morgan");  
const path = require("path");  
const fs = require("fs");
```

4. Añadir debajo del registro anterior, lo siguiente:

```
app.use(logger("combined", {
  stream: fs.createWriteStream(path.join(__dirname, "logs/access.log")),
  flags: "a"
}));
```

5. Guardar cambios.
6. Ir a la consola.
7. Reiniciar la aplicación.
8. Ir al navegador.
9. Ir a <http://localhost:8080>.
10. Ir a <http://localhost:8080/mumford-and-sons>.
11. Ir a la consola.
12. Comprobar que aparecen las entradas de registro en formato **dev**.
13. Editar el archivo **logs/access.log** y comprobar que aparecen las entradas en formato **combined**.

Archivo de registro error.log

Ahora, vamos a generar el archivo **error.log**. Actualmente, **Morgan** devuelve siempre una función de *middleware* normal, no es posible obtener funciones de control de errores para registrar errores generados en la aplicación. Pero se puede simular fácilmente. Veamos cómo.

1. Editar el archivo **app.js**.
2. Modificar la proposición **app.get()** como sigue:
3. Añadir *al final* de la pila de *middleware* la siguiente función de control de errores:

```
app.use(function(err, req, res, next) {
  logger(":date :url => " + err.toString(), {
    stream: fs.createWriteStream(path.join(__dirname, "logs/error.log")),
    flags: "a"
  })(req, res, function() {
    next(err);
  });
});
```

Hacerlo debajo de la proposición **app.get()**.

4. Guardar cambios.
5. Ir a la consola.
6. Reiniciar la aplicación.
7. Ir al navegador.
8. Ir a <http://localhost:8080/echo-and-the-bunnymen>.
9. Ir a <http://localhost:8080/jesus-and-mary-chain>.
10. Ir a <http://localhost:8080/error>.
11. Ir a la consola.
12. Comprobar que han aparecido las entradas de registro.
13. Editar el archivo **logs/access.log** y comprobar que se dispone de las entradas de acceso, sin mostrarse ningún error.

14. Editar el archivo `logs/error.log` y comprobar que aparece la entrada de error.