***APUNTES DWES***

Tabla de contenido

[NODE.JS 2](#_Toc160717605)

[AJAX 2](#_Toc160717606)

[FETCH 3](#_Toc160717607)

[EXPRESS 4](#_Toc160717608)

[-GET 4](#_Toc160717609)

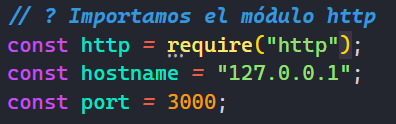
[-USE 5](#_Toc160717610)

[-Middlewares 5](#_Toc160717611)

[-Require JSON 5](#_Toc160717612)

# NODE.JS

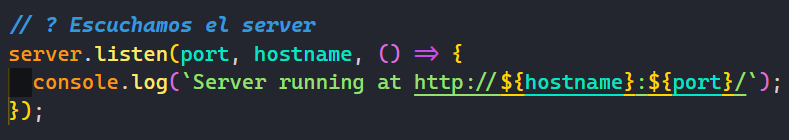
1. Para crear un server con Node.JS debemos usar el módulo http.



1. Luego deberemos crear el servidor.



1. Escucharemos el servidor



# AJAX

Para crear una petición con Ajax deberemos hacer lo siguiente:

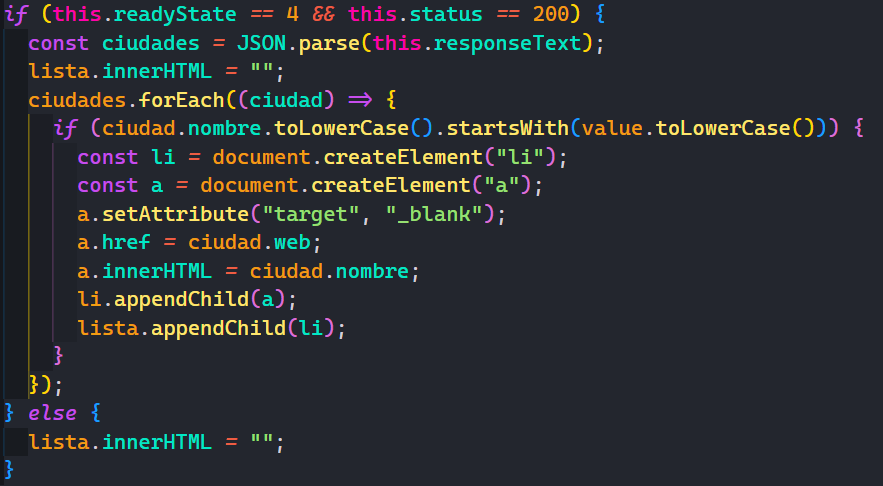
1. Creamos el objeto de la request



1. Hacemos la función para realizar la petición



1. Comprobamos el estado de nuestra petición para después usar la respuesta como nosotros queramos



1. Después abriremos la petición con open(‘{método}’,’{ruta}, true’)



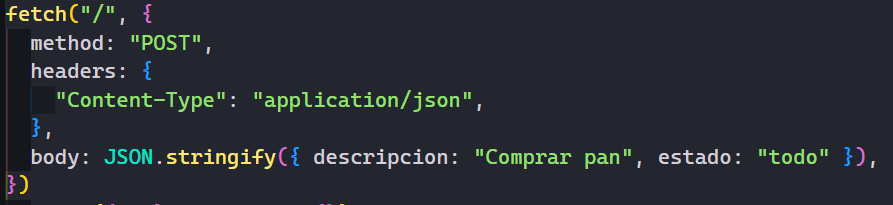
1. Enviamos la petición con send



# FETCH

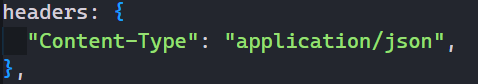
Para poder hacer peticiones con fetch deberemos hacer lo siguiente:

1. Primero pondremos la ruta y el tipo de petición que queremos realizar. En caso de ser un GET con poner la ruta es suficiente.





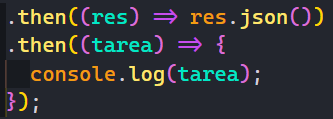
1. En caso de no ser GET, definiremos las cabeceras con headers



1. Después le pasaremos el contenido de la petición con body



1. Manejamos la respuesta como queramos con .then()

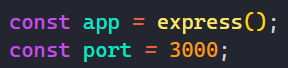


# EXPRESS

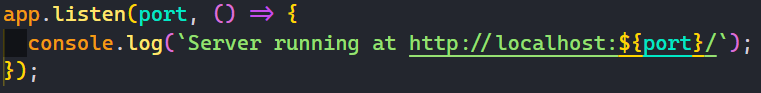
1. Crearemos el proyecto con npm init -y
2. Instalaremos express usando npm install express
3. En el package.json añadiremos "type":"module"
4. En nuestro server.js/app.js importaremos express



1. Crearemos la app de express y el puerto que usaremos



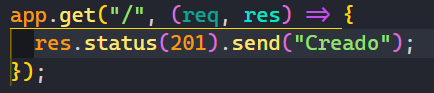
1. Pondremos a escuchar para saber si nuestro server funciona en el puerto elegido

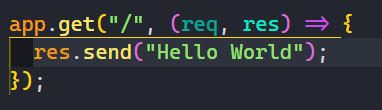


## -GET

Define un manejador de ruta para la ruta '/{nombre}. Cuando un cliente hace una solicitud GET a '/{nombre}, la función proporcionada se ejecuta.

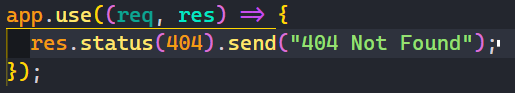
Disponemos de dos objetos req, sería el objeto de la request y res, el objeto de la response.

Usaremos el .status para enviar el estado de una respuesta, res.send() envía automáticamente un estado 200.



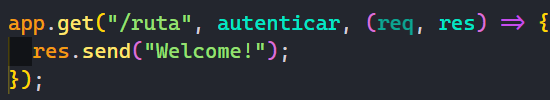
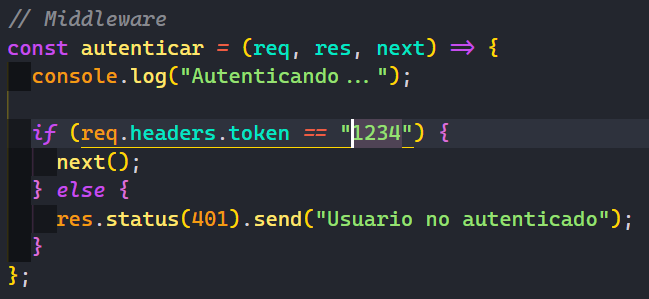
## -USE

También disponemos de un ***use*** que nos permitirá redirigir las malas urls a un 404 o donde nosotros elijamos. Deberemos ponerlo al final de todo nuestro código porque sino el resto de request solo llegarán hasta aquí.



## -Middlewares

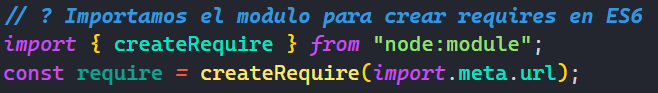
Los middlewares son funciones que podemos ejecutar durante el proceso de la solicitud y se colocan entre la ruta y el (req, res, next). Usa el next para pasar al siguiente middleware o lo que haya después.



## -Require JSON

Para poder importar JSON en ES6 podemos crear un require(), para ello deberemos hacer lo siguiente:

1. Primero importaremos el createRequire de node:module



1. Importamos el archivo que queramos con Require(“nombreArchivo”)

