

## Ejercicio 1

1. Crea un servidor web en Node. Usa `console.log()` para mostrar el objeto `request` completo. Identifica dos propiedades que te resulten conocidas.

ivan@debian-ivan: ~

```
'max-age=0',
'Upgrade-Insecure-Requests',
'1',
'User-Agent',
'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/69.0.3497.100 Safari/537.36',
'Accept',
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apn
g,*/*;q=0.8',
'Accept-Encoding',
'gzip, deflate',
'Accept-Language',
'es-ES;q=0.9' ],
trailers: {},
rawTrailers: [],
aborted: false,
upgrade: false,
url: '/',
method: 'GET',
statusCode: null,
statusMessage: null,
client:
  Socket {
    connecting: false,
    _hadError: false,
    _handle:
```

```
highWaterMark: 16384,
buffer: BufferList { length: 0 },
length: 0,
pipes: null,
pipesCount: 0,
flowing: true,
ended: false,
endEmitted: false,
reading: true,
sync: false,
needReadable: true,
emittedReadable: false,
readableListening: false,
resumeScheduled: false,
emitClose: false,
destroyed: false,
defaultEncoding: 'utf8',
awaitDrain: 0,
readingMore: false,
decoder: null,
encoding: null },
readable: true,
_events:
  { end: [Array],
    drain: [Array],
```

## Código

ivan@debian-ivan: ~

```
GNU nano 2.7.4 Fichero: modulo.js

var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  console.log(req);
  res.end();
}).listen(8080);
```

## Ejercicio 2

Modifica tu código para escribir la URL actual como respuesta, es decir, al visitar por ejemplo `http://tu_ip:3000/nacktschnecke`, deberías obtener una página web que contenga la palabra `nacktschnecke`. Esta palabra deber estar dentro de una etiqueta `h1`

ivan@debian-ivan: ~

```
GNU nano 2.7.4 Fichero: modulo.js

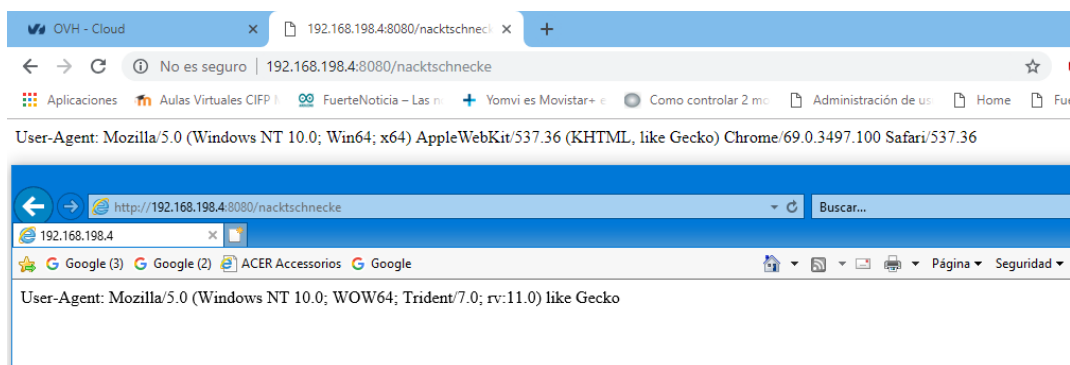
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("<h1>" + req.url + "</h1>");
  res.end();
}).listen(8080);
```



## Ejercicio 3

Por último, muestra como respuesta en una página web la versión del navegador (user-agent) que ha hecho la solicitud. Pista: ten en cuenta que el objeto `request` tiene una propiedad `headers`.

**(Se muestra Chrome e Internet Explorer respectivamente)**



## Código

ivan@debian-ivan: ~

```
GNU nano 2.7.4 Fichero: modulo.js

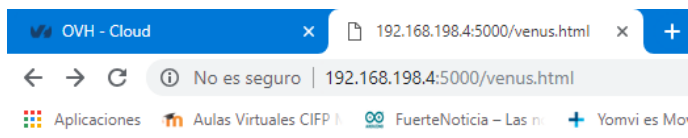
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write('User-Agent: ' + req.headers['user-agent']);
  res.end();
}).listen(8080);
```

## Ejercicio 2. Enrutador básico

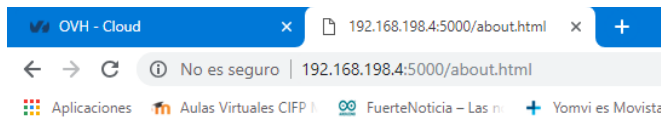
Vamos a crear un sitio web con las siguientes características:

- Nuestro servidor responderá a dos rutas: `http://tu_ip:5000/venus` y `http://tu_ip:5000/about`. En el primer caso mostrará entre etiquetas `<h1>` el texto "Página sobre el planeta Venus". En el segundo caso mostrará entre etiquetas `<h1>` el texto "Página about".
- Si se introduce una ruta diferente el servidor responderá con un código de estado 404 y con la página correspondiente. Por ejemplo "error 404 file not found". En este último caso comprueba que puedes ver el código de error en el navegador (cabeceras HTTP en modo desarrollador).

Nota. Más adelante en este curso veremos cómo hacer un enrutador de una manera sencilla.



## Pagina sobre el planeta venus



## Pagina about

### Código 404

