

REDIRECCIONAMIENTO HTTP

(PRÁCTICA)

Tiempo estimado: 5min

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

Al finalizarla, el estudiante:

- Habrá llevado a cabo redireccionamientos.

objetivos

El objetivo de la práctica es crear una aplicación **Express** que remita de un recurso a otro.

creación del proyecto

Para comenzar, vamos a crear el proyecto de la aplicación.

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",
    "start-dev": "./node_modules/.bin/nodemon ./app.js"
  },
  "dependencies": {
    "express": "*"
  },
  "devDependencies": {
    "nodemon": "*"
  }
}
```

5. Crear el archivo **index.html**:

```
<!doctype html>

<html>
  <head>
    <title>Express app</title>
  </head>

  <body>
    <p>¡Hola Mundo!</p>
  </body>
</html>
```

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

```
"use strict";

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

```
//app
const app = express();

//rutas
app.get("/", function(req, res) {
  res.sendFile("index.html", {root: __dirname});
});

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

7. Instalar dependencias:

```
> npm install
```

8. Iniciar la aplicación **Express**:

```
> npm run start-dev
```

9. Abrir el navegador.

10. Ir a <http://localhost:8080>.

creación de redireccionamiento

Veamos cómo redirigir de un recurso a otro:

1. Editar el archivo **app.js**.

2. Añadir los siguientes controladores de ruta:

```
app.get("/de", function(req, res) {
  res.redirect("/a");
});

app.get("/a", function(req, res) {
  res.set("Content-Type", "text/plain");
  res.send("Sí, Express es fácil\n");
});
```

3. Ir al navegador.

4. Solicitar <http://localhost:8080/de>.

Observe que el navegador al recibir mensajes de redireccionamiento, los sigue de manera implícita. Así, ayuda al usuario a no tener que hacerlo manualmente. De ahí, que muestre el contenido del recurso `/a`.

5. Abrir una consola.

6. Solicitar <http://localhost:8080/de> mediante **cURL**:

```
> curl http://localhost:8080/de -v
```

En este caso, **cURL** no sigue el redireccionamiento. Pero si lo deseamos, podemos indicarle que lo haga mediante la opción **-L**:

```
> curl http://localhost:8080/de -vL
```