

OBJETO RESPUESTA (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 redactado mensajes de respuesta **HTTP**.
- Habrá enviado archivos mediante **sendFile()**.
- Habrá enviado el contenido de recursos en bloques mediante **write()** y **end()**.
- Habrá añadido campos de cabecera a la respuesta.

objetivos

En esta práctica, vamos a crear una aplicación **Express** que responda a peticiones **HTTP** remitidas por clientes.

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"
  },
  "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();
const index = "<!doctype html>\n" +
  "<html>\n" +
  "<head><title>Express app</title></head>\n" +
  "<body><p>¡Hola Mundo!</p></body>\n" +
  "</html>\n";

//config app
app.get("*", function(req, res) { res.send(index); });

//listen
```

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

6. Instalar dependencias:

```
> npm install
```

7. Iniciar la aplicación **Express**:

```
> npm start
```

8. Abrir una nueva consola.

9. Solicitar la página de inicio de la aplicación:

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

remisión de Archivos de disco

Para comenzar, vamos a ver cómo enviar archivos de disco:

1. Crear la carpeta data en el directorio del proyecto.

2. Crear el archivo phoenix.json en la carpeta data:

```
{
  "name": "Phoenix",
  "origin": "Versailles, France",
  "year": 1999
}
```

3. Crear el archivo dawes en la carpeta data:

```
{
  "name": "Dawes",
  "origin": "Los Angeles, CA",
  "year": 2009
}
```

Observe que el archivo no tiene extensión.

4. Editar el archivo app.json.

5. Importar el módulo **path**:

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

6. Modificar app.get() para que cuando el usuario solicite dawes, la aplicación responda con el archivo data/dawes; cuando solicite phoenix, con data/phoenix.json; en cualquier otro caso, con el contenido de la constante index:

```
app.get("/*", function(req, res) {
  if (req.path == "/dawes") {
    res.sendFile(path.join(__dirname, "data/dawes"));
  } else if (req.path == "/phoenix") {
    res.sendFile(path.join(__dirname, "data/phoenix.json"));
  } else {
    res.send(index);
  }
});
```

7. Guardar cambios.

8. Reiniciar la aplicación.

9. Solicitar mediante **cURL** localhost:8080/phoenix:

```
> curl -v localhost:8080/phoenix
```

Observe que cuando se remite un archivo con extensión **.json**, **sendFile()** añade automáticamente el campo de cabecera **Content-Type** a **application/json**.

El método `sendFile()` analiza la extensión del archivo y, si la reconoce, añade automáticamente el tipo de contenido correspondiente. En nuestro caso, reconoce la extensión `.json` y añade el tipo de contenido correspondiente. Lo mismo ocurre con imágenes y otros tipos de archivo.

10. Solicitar mediante `cURL` `localhost:8080/dawes`.

Ahora, el archivo `dawes` no tiene extensión. En este caso, `sendFile()` usa como tipo de contenido predeterminado `application/octet-stream`. Si necesitamos que sea `application/json`, sin modificar la extensión del archivo, debemos fijar explícitamente el valor del campo de cabecera:

```
if (req.path == "/dawes") {
  res.set("Content-Type", "application/json");
  res.sendFile(path.join(__dirname, "data/dawes"));
}
```

O bien:

```
if (req.path == "/dawes") {
  res.sendFile("data/dawes", {
    root: __dirname,
    headers: {
      "Content-Type": "application/json"
    }
  });
}
```

11. Solicitar mediante `cURL` `localhost:8080/otro`.
12. Editar el archivo `app.js`.
13. Fijar el campo de cabecera `Content-Type` para `dawes` a `application/json`:

```
app.get("*", function(req, res) {
  if (req.path == "/dawes") {
    res.sendFile("data/dawes", {
      root: __dirname,
      headers: {
        "Content-Type": "application/json"
      }
    });
  } else if (req.path == "/phoenix") {
    res.sendFile(path.join(__dirname, "data/phoenix.json"));
  } else {
    res.send(index);
  }
});
```

14. Guardar cambios.
15. Reiniciar la aplicación.
16. Solicitar mediante `cURL` `localhost:8080/dawes`.
17. Comprobar que ahora el campo de cabecera `Content-Type` remitido es `application/json`.

REMISIÓN DE CONTENIDO EN BLOQUES

Ahora, vamos a ver cómo enviar un contenido en bloques mediante los métodos `write()` y `end()`. Recordemos que `end()` indica que la respuesta ha finalizado y ya no deseamos añadir nada más. A partir de ese momento, cualquier intento de añadidura de algo a la respuesta, desembocará en un error.

1. Editar el archivo `app.js`.
2. Enviar la información de Snow Patrol en bloques:

```
app.get("*", function(req, res) {
  if (req.path == "/dawes") {
    res.sendFile("data/dawes", {
      root: __dirname,
      headers: {
        "Content-Type": "application/json"
      }
    });
  }
});
```

```

    });
    } else if (req.path == "/phoenix") {
      res.sendFile(path.join(__dirname, "data/phoenix.json"));
    } else if (req.path == "/snow-patrol") {
      res.set("Content-Type", "application/json");
      res.write('{ "name": "Snow Patrol", "origin": "Dundee, Scotland",'});
      res.write('"year": 1993 }\\n')
      res.end();
    } else {
      res.send(index);
    }
  });
});

```

3. Guardar cambios.
4. Reiniciar la aplicación.
5. Solicitar mediante **cURL** localhost:8080/snow-patrol.
6. Editar el archivo **app.js**.
7. Indicar el campo de cabecera después del método **end()**:

```

app.get("*", function(req, res) {
  if (req.path == "/daves") {
    res.sendFile("data/daves", {
      root: __dirname,
      headers: {
        "Content-Type": "application/json"
      }
    });
  } else if (req.path == "/phoenix") {
    res.sendFile(path.join(__dirname, "data/phoenix.json"));
  } else if (req.path == "/snow-patrol") {
    res.write('{ "name": "Snow Patrol", "origin": "Dundee, Scotland",'});
    res.write('"year": 1993 }\\n')
    res.end();
    res.set("Content-Type", "application/json");
  } else {
    res.send(index);
  }
});

```

8. Guardar cambios.
9. Reiniciar la aplicación.
10. Solicitar mediante **cURL** localhost:8080/snow-patrol.
11. Ir a la consola donde se arranca la aplicación.
12. Comprobar el mensaje de error que ha aparecido.