

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

Al finalizarla, el estudiante:

- Habrá creado rutas a nivel de aplicación o encaminador.
- Habrá creado encaminadores secundarios.
- Habrá creado un módulo de rutas.

objetivos

En esta práctica, vamos a crear una aplicación **Express** organizada mediante rutas.

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...");
});
```

Observe una primera definición de ruta. La cual se adapta a todos los recursos de la aplicación: *****.

7. Instalar dependencias:

```
> npm install
```

8. Iniciar la aplicación **Express**:

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

9. Abrir el navegador.

10. Solicitar la página de inicio, localhost:8080.

definición de rutas de aplicación

En primera instancia, vamos a definir rutas específicas para los siguientes recursos `/`, `/band/dinosaur-jr` y `/band/sisters-of-mercy`:

1. Editar el archivo **app.js**.
2. Suprimir la ruta definida, *****.
3. Definir las siguientes rutas estáticas:

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

app.get("/band/dinosaur-jr", function(req, res) {
  res.json({
    name: "Dinosaur Jr.",
    origin: "Amherst, Massachusetts, United States",
    year: 1984
  });
});

app.get("/band/sisters-of-mercy", function(req, res) {
  res.json({
    name: "Sisters of Mercy",
    origin: "Leeds, West Yorkshire, England",
    year: 1977
  });
});
```

4. Guardar cambios.
5. Ir al navegador.
6. Solicitar localhost:8080.
7. Solicitar localhost:8080/band/dinosaur-jr.
8. Solicitar localhost:8080/band/sisters-of-mercy.
9. Solicitar localhost:8080/no-existo.

Como el recurso no tiene ninguna ruta asociada, ni estática ni dinámicamente, **Express** devuelve un mensaje **404 Not Found**.

10. Editar el archivo **app.js**.

11. Reemplazar las rutas que cuelgan de **/band/** por una única dinámica:

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

app.get("/band/*", function(req, res) {
  const BANDS = {
    "/band/dinosaur-jr": {
      name: "Dinosaur Jr.",
      origin: "Amherst, Massachusetts, United States",
      year: 1984
    },
    "/band/sisters-of-mercy": {
      name: "Sisters of Mercy",
      origin: "Leeds, West Yorkshire, England",
      year: 1977
    }
  }

  res.json(BANDS[req.path]);
});
```

12. Guardar cambios.

13. Solicitar localhost:8080.

14. Solicitar localhost:8080/band/dinosaur-jr.

15. Solicitar localhost:8080/band/sisters-of-mercy.

16. Solicitar localhost:8080/no-existo.

17. Editar el archivo **app.js**.

18. Reemplazar la ruta dinámica por una parametrizada para obtener el nombre de la banda mediante un parámetro:

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

app.get("/band/:name", function(req, res) {
  const BANDS = {
    "dinosaur-jr": {
      name: "Dinosaur Jr.",
      origin: "Amherst, Massachusetts, United States",
      year: 1984
    },
    "sisters-of-mercy": {
      name: "Sisters of Mercy",
      origin: "Leeds, West Yorkshire, England",
      year: 1977
    }
  }

  res.json(BANDS[req.params.name]);
});
```

19. Guardar cambios.

20. Solicitar localhost:8080/band/dinosaur-jr.

21. Solicitar localhost:8080/band/sisters-of-mercy.

A continuación, vamos a suponer que tenemos un componente que se encarga de atender todas las peticiones **HTTP** cuyo *path* cuelgue de `/band/`. Para ello, definiremos las rutas en un módulo aparte, `routes/band.js`.

1. Crear el directorio `routes`.
2. Añadir el archivo `band.js` al directorio `routes`:

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

//api
const router = express.Router();
module.exports.router = router;

//data
const BANDS = {
  "dinosaur-jr": {
    name: "Dinosaur Jr.",
    origin: "Amherst, Massachusetts, United States",
    year: 1984
  },
  "sisters-of-mercy": {
    name: "Sisters of Mercy",
    origin: "Leeds, West Yorkshire, England",
    year: 1977
  }
};

//routes
router.get("/", function(req, res) {
  res.set("Content-Type", "text/plain");
  for (let band in BANDS) res.write(band + "\n");
  res.end();
});

router.get("/:name", function(req, res) {
  res.json(BANDS[req.params.name]);
});
```

3. Editar el archivo `app.js`.
4. Establecer el encaminador del componente para el punto de montaje `band`:

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

app.use("/band", require("../routes/band").router);
```

5. Guardar cambios.
6. Solicitar `localhost:8080`.
7. Solicitar `localhost:8080/band/dinosaur-jr`.
8. Solicitar `localhost:8080/band/sisters-of-mercy`.
9. Solicitar `localhost:8080/no-existo`.