

FORMULARIOS HTML

(PRÁCTICA)

Tiempo estimado: 10min

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

Al finalizarla, el estudiante:

- Habrá configurado **body-parser**.
- Habrá accedido al contenido remitido por el usuario mediante un formulario.

objetivos

El objetivo de la práctica es crear una aplicación **Express** que procese un formulario mediante el método **POST** y el tipo de contenido **application/x-www-form-urlencoded**.

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": "*",
    "serve-static": "*"
  },
  "devDependencies": {
    "nodemon": "*"
  }
}
```

5. Crear la carpeta pública **public**.
6. Crear el archivo **index.html** en la carpeta pública:

```
<!doctype html>

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

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

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

```

"use strict";

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

//app
const app = express();

//middleware
app.use(require("serve-static")(path.join(__dirname, "public")));

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

```

8. Instalar dependencias:

```
> npm install
```

9. Iniciar la aplicación **Express**:

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

10. Abrir el navegador.

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

CONFIGURACIÓN de body-parser

Veamos cómo usar **body-parser**:

1. Editar el archivo **package.json**.

2. Añadir **body-parser** a las dependencias de la aplicación:

```

"dependencies": {
  "express": "*",
  "body-parser": "*",
  "serve-static": "*"
}

```

3. Guardar cambios.

4. Abrir una nueva consola.

5. Ir al directorio de la práctica.

6. Instalar las nuevas dependencias:

```
> npm update
```

7. Crear el archivo **public/form.html**:

```

<!doctype html>

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

  <body>
    <form action="/form" method="post" enctype="application/x-www-form-urlencoded">
      <label>Nombre de usuario: <input type="text" name="username"></label><br>
      <label>Contraseña: <input type="password" name="password"></label><br>
      <button type="submit">Enviar</button>
    </form>
  </body>
</html>

```

8. Editar el archivo **app.js**.

9. Añadir el analizador de cuerpo **urlencoded** a la pila de *middleware*:

```
//middleware
app.use(require("serve-static")(path.join(__dirname, "public")));
app.use(require("body-parser").urlencoded({extended: false}));
```

10. Añadir el controlador de la ruta `/form`, el cual se encargará de procesar los datos remitidos por el usuario mediante el formulario:

```
//rutas
app.post("/form", function(req, res) {
  res.json(req.body);
});
```

11. Guardar cambios.
12. Ir al navegador.
13. Solicitar <http://localhost:8080/form.html>.
14. Rellenar el formulario y hacer clic en el botón Enviar.

restricción de body-parser a ruta particular

En la sección anterior, configuramos `body-parser` para que analizase todas las peticiones `HTTP` recibidas por la aplicación. Ahora, vamos a configurarlo para que sólo analice las peticiones del recurso `/form`:

1. Editar el archivo `app.js`.
2. Suprimir el registro de `body-parser` en la pila de `middleware`.
3. Modificar el controlador `POST` de la ruta `/form` para que antes ejecute `body-parser`:

```
//rutas
app.post(
  "/form",
  require("body-parser").urlencoded({extended: false}),
  function(req, res) { res.json(req.body); }
);
```

4. Guardar cambios.
5. Ir al navegador.
6. Solicitar <http://localhost:8080/form.html>.
7. Rellenar el formulario y hacer clic en el botón Enviar.