

Express

Express es un marco de aplicaciones web para Node.js que proporciona una forma rápida y sencilla de crear aplicaciones web. Express proporciona una API simple y flexible para manejar solicitudes HTTP y se integra bien con una amplia gama de módulos de Node.js.

Router en Express

En Express, el enrutador (router) es un objeto que te permite separar tus rutas y controladores en módulos separados. Esto hace que tu código sea más fácil de mantener y escalar, ya que puedes organizar tus rutas y controladores en archivos separados y luego importarlos en tu aplicación principal.

Para utilizar el enrutador en Express, primero debes crear un nuevo objeto de enrutador utilizando el método Router de Express:

JavaScript

```
const express = require('express');  
  
const router = express.Router();
```

Una vez que hayas creado un objeto de enrutador, puedes agregar rutas y controladores a él utilizando los métodos HTTP correspondientes (como get, post, put, delete, etc.) y las funciones de controlador de ruta correspondientes.

Ejemplo de uso del enrutador en Express

JavaScript

```
const express = require('express');
const router = express.Router();

router.get('/', (req, res) => {
  res.send('Lista de usuarios');
});

router.get('/:id', (req, res) => {
  const { id } = req.params;
  res.send(`Detalles del usuario ${id}`);
});

router.post('/', (req, res) => {
  const { nombre, correo } = req.body;
  res.send(`Usuario ${nombre} creado con éxito`);
});

module.exports = router;
```

En este ejemplo, hemos creado un archivo usuarios.js que contiene nuestras rutas y controladores para las operaciones de usuario. Hemos creado un objeto de enrutador utilizando el método Router de Express y hemos agregado tres rutas utilizando los métodos get y post del enrutador.

Luego, en nuestro archivo principal de la aplicación, podemos importar el enrutador y usarlo como middleware en nuestra aplicación:

JavaScript

```
// app.js

const express = require('express');

const usuariosRouter = require('./usuarios');

const app = express();

app.use('/usuarios', usuariosRouter);

app.listen(3000, () => {

  console.log('Servidor iniciado en el puerto 3000');

});
```

En este ejemplo, hemos importado nuestro enrutador de usuarios y lo hemos utilizado como middleware en nuestra aplicación principal utilizando el método `use` de Express. Hemos especificado que todas las rutas del enrutador de usuarios deben tener el prefijo `/usuarios`.

Conclusión

El enrutador en Express es una forma conveniente de organizar tus rutas y controladores en módulos separados. Esto puede hacer que tu código sea más fácil de mantener y escalar a medida que tu aplicación crece. Espero que esta breve explicación te haya sido útil. Si tienes alguna pregunta o necesitas más información, no dudes en preguntar.

Middleware - Express

En Express, el middleware es una función que se ejecuta antes de que se maneje una solicitud. El middleware puede realizar tareas como la autenticación, el registro de solicitudes y la validación de datos.

Express proporciona middleware incorporado, como el middleware de análisis de cuerpo (body-parser) y el middleware de sesión (session). También puedes crear tu propio middleware personalizado.

Ejemplo de middleware en Express

Aquí hay un ejemplo de cómo agregar un middleware personalizado a una aplicación Express:

JavaScript

```
const express = require('express');
const app = express();

const myMiddleware = (req, res, next) => {
  console.log('Middleware personalizado ejecutado');
  next();
};

app.use(myMiddleware);

app.get('/', (req, res) => {
  res.send('¡Bienvenido a mi aplicación!');
});

app.listen(3000, () => {
  console.log('Servidor iniciado en el puerto 3000');
});
```

En este ejemplo, hemos creado un middleware personalizado llamado myMiddleware. Este middleware registra un mensaje en la consola cada vez que se maneja una solicitud HTTP. Luego, hemos utilizado el método use de Express para agregar el middleware a nuestra aplicación. Finalmente, hemos definido una ruta para la página de inicio utilizando el método get de Express.

Middleware de análisis de cuerpo en Express

El middleware de análisis de cuerpo (body-parser) es un middleware incorporado en Express que se utiliza para analizar el cuerpo de una solicitud HTTP. El cuerpo de una solicitud puede contener datos como JSON, texto o datos de formulario.

Aquí hay un ejemplo de cómo utilizar el middleware de análisis de cuerpo en Express:

JavaScript

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

app.use(bodyParser.json());

app.post('/api/usuarios', (req, res) => {
  const { nombre, correo } = req.body;
  // Aquí puedes hacer lo que quieras con los datos del cuerpo de
  la solicitud
  res.send(`¡Hola ${nombre}! Tu correo es ${correo}`);
});

app.listen(3000, () => {
  console.log('Servidor iniciado en el puerto 3000');
});
```

En este ejemplo, hemos utilizado el middleware de análisis de cuerpo para analizar el cuerpo de una solicitud HTTP que se envía a la ruta /api/usuarios. Luego, hemos extraído el nombre y el correo electrónico del cuerpo de la solicitud y los hemos utilizado para enviar una respuesta al cliente.