

Express-Basic Routing

Routing is a fundamental concept in Express.js that enables you to handle HTTP requests and map them to specific routes. Here's an overview of basic routing in Express:

Route Types

There are two types of routes in Express:

1. **HTTP Methods:** Represented by verbs like `GET`, `POST`, `PUT`, etc.
2. **Paths:** Represented by strings, used to identify the endpoint.

Creating Routes

To create a route, you can use the `app` object (which is an instance of Express) and specify the HTTP method and path:

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

// GET /hello
app.get('/hello', (req, res) => {
  res.send('Hello World!');
});

// POST /users
app.post('/users', (req, res) => {
  // Handle user creation logic here
});
```

In this example:

- `GET /hello` maps to the `/hello` path and responds with a "Hello World!" message when an HTTP GET request is made.
- `POST /users` maps to the `/users` path and will handle user creation logic.

Example Use Case

Let's create a simple API that allows users to perform CRUD (Create, Read, Update, Delete) operations on books:

```

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

// Create route for getting all books
app.get('/books', (req, res) => {
  const books = [
    { id: 1, title: 'Book 1' },
    { id: 2, title: 'Book 2' }
  ];
  res.json(books);
});

// Create route for creating a new book
app.post('/books', (req, res) => {
  const newBook = req.body;
  // Save the new book to the database
  console.log(newBook);
  res.status(201).send('Book created successfully!');
});

// Start the server
const port = 3000;
app.listen(port, () => {
  console.log(`Server listening on port ${port}`);
});

```

In this example:

- `GET /books` returns a JSON array of all books.
- `POST /books` creates a new book and saves it to the database.

Route Parameters

You can also define routes with parameters using the following syntax:

```

app.get('/users/:id', (req, res) => {
  const userId = req.params.id;
  // Handle user retrieval logic here
});

```

In this example:

- `GET /users/123` will pass `123` as the value of `userId`.

This is a basic overview of routing in Express.js. You can explore more advanced concepts like route middleware, error handling, and parameter validation to build robust APIs.

