ONE PAGE SUMMARY

IT & Software › Backend Development › Express.js

# Introduction to Express.js

Express.js: Simplifying Node.js Web Application Development

## DEFINITION

> Express.js is a minimal and flexible Node.js web application framework that provides robust features for building web and mobile applications. It simplifies the process of creating server-side applications with a variety of tools and middleware.

## KEY CONCEPTS

### Middleware

Middleware functions are functions that have access to the request and response objects, allowing for processing of requests before reaching the final route handler.

### Routing

Routing refers to defining the endpoints of the application, allowing the server to respond to various HTTP requests at specific URLs.

### Request and Response

Express.js handles HTTP requests and responses, allowing developers to easily manage data sent to and from the client.

### Static Files

Express can serve static files, such as images and stylesheets, directly from a specified directory, enhancing the user experience.

### Error Handling

Express provides a systematic way to handle errors, ensuring that the application can gracefully manage unexpected issues.

## EXAMPLES

→ app.get('/home', (req, res) => res.send('Welcome to Home!'))

→ app.use(express.static('public'))

→ app.post('/submit', (req, res) => res.send('Form submitted!'))

## MEMORY TIPS

★ Remember 'M' for Middleware: Think of it as the middle layer between request and response.

★ R for Routing: Picture a map where routes lead to different destinations in your app.

★ Use 'S' for Static: Static files are like the furniture in your house, always there and unchanging.

## COMMON MISTAKES

✗ Forgetting to use 'next()' in middleware functions, causing requests to hang.

✗ Not defining routes correctly, leading to 404 errors.

✗ Neglecting error handling, which can cause crashes or unhelpful responses.

## QUICK RECAP

Express.js is a powerful framework for building web applications in Node.js. It simplifies routing, middleware management, and error handling, making development faster and more efficient.