

Express.js Developer Cheatsheet

What is Express.js?

Express.js is a minimalist web framework for Node.js that simplifies building APIs, web servers, and middleware-based applications.

Feature	Description
Lightweight & Fast	Built directly on Node's HTTP module with minimal overhead.
Middleware Support	Easily extendable via middleware functions.
Routing System	Clean syntax for handling GET, POST, PUT, DELETE routes.
Integration	Compatible with databases, security tools, and third-party libraries.
Ecosystem	Rich middleware ecosystem (morgan, helmet, cors, etc.).

Routing Example

```
import express from 'express';
const app = express();
app.get('/', (req, res) => res.send('Welcome!'));
app.post('/users', (req, res) => res.json({ message: 'User created' }));
app.listen(3000, () => console.log('Server running on port 3000'));
```

Middleware Usage

Middleware functions have access to req, res, and next().

Common middleware:

- express.json() - Parse JSON bodies
- helmet - Secure HTTP headers
- morgan - Log HTTP requests
- cors - Handle cross-origin requests

Error Handling

Define middleware with four arguments:

```
app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).json({ message: 'Something went wrong!' });
});
```

Router Example

```
// routes/userRoutes.js
import { Router } from 'express';
const router = Router();
router.get('/', (req, res) => res.send('All Users'));
router.post('/', (req, res) => res.send('Add User'));
export default router;
```

Recommended Folder Structure

```
express-app/
  └── src/
    ├── app.js
    └── routes/
      ├── index.js
      └── userRoutes.js
    └── controllers/
    └── middlewares/
```

```
─ └─ models/
    └─ utils/
        └─ .env
    └─ package.json
    └─ README.md
```

Best Practices

- Use Helmet and CORS for security.
- Avoid blocking operations in routes.
- Validate inputs with express-validator.
- Log requests using Morgan.
- Separate controllers, routes, and middleware.
- Handle async errors gracefully with try/catch.

Common Commands

```
npm init -y
npm install express
npm install dotenv cors helmet morgan
npm run dev
```

Summary

Express.js is the most popular Node.js web framework for building APIs. It is fast, modular, and extensible with middleware — ideal for enterprise-grade applications and microservices.