

What is Express.js?

Express.js is a **minimal and flexible web application framework for Node.js** that simplifies building web servers and APIs.

□ In Simple Words:

Express is like the "brain" of a web server. Instead of writing everything from scratch in Node.js, Express gives you shortcuts and tools to:

- Handle HTTP requests (GET, POST, etc.)
 - Serve HTML pages or JSON data
 - Route users to different pages
 - Use middleware (like logging, authentication, etc.)
-

□ Built on Node.js

Node.js lets you run JavaScript outside the browser (on a server).

Express.js runs on Node.js and makes server-side development easier and faster.

🔗 Example: A Simple Express Server

```
js
CopyEdit
const express = require('express');
const app = express();
const PORT = 3000;

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

app.listen(PORT, () => {
  console.log(`Server is running at http://localhost:${PORT}`);
});
```

□ Why Use Express?

- □ Easy to learn
- □ Tons of middleware available
- □ Perfect for APIs and websites
- □ Used by companies like Uber, PayPal, and IBM

🔗 Step-by-Step Guide to Creating Pages in Express:-

1. Go to your project folder in the terminal:

```
cd d:\express-js
```

2. Install Express:

Make sure you run this command inside your project folder:

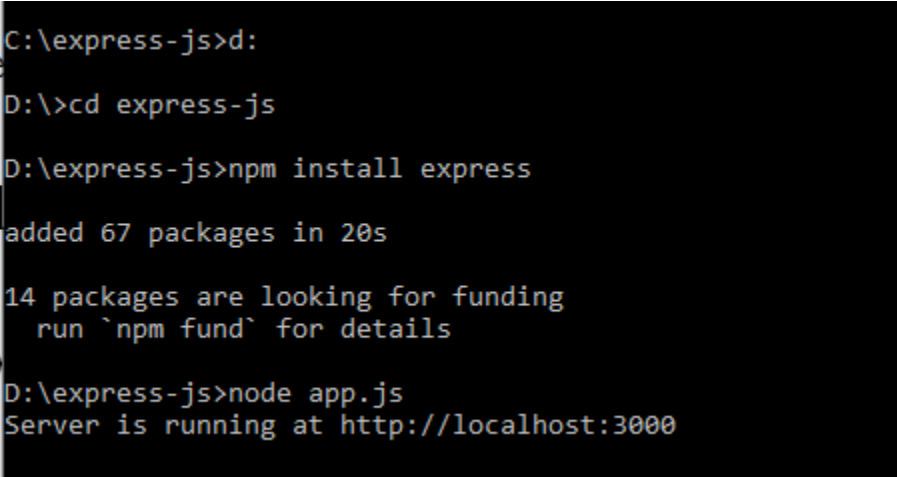
```
npm install express
```

That will create a `node_modules` folder and install the Express package there.

3. Then run your app again:

```
node app.js
```

see following example as shown below :-



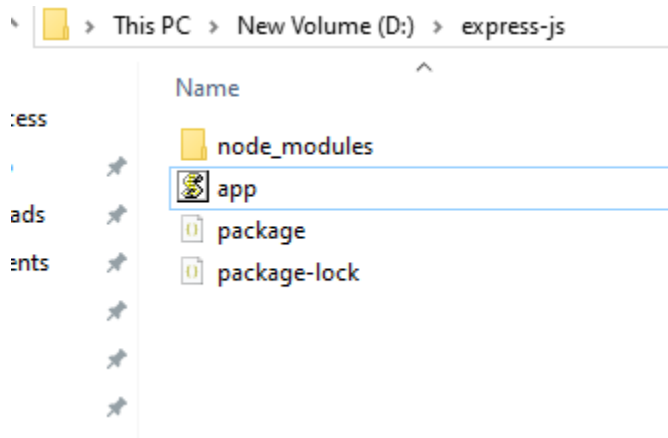
```
C:\express-js>d:
D:\>cd express-js
D:\express-js>npm install express
added 67 packages in 20s
14 packages are looking for funding
  run `npm fund` for details
D:\express-js>node app.js
Server is running at http://localhost:3000
```

□ **Optional: Check your project structure**

You should see something like this in `d:\express-js`:

```
d:\express-js
├── app.js
├── package.json
├── node_modules
│   └── express
```

As shown below :-



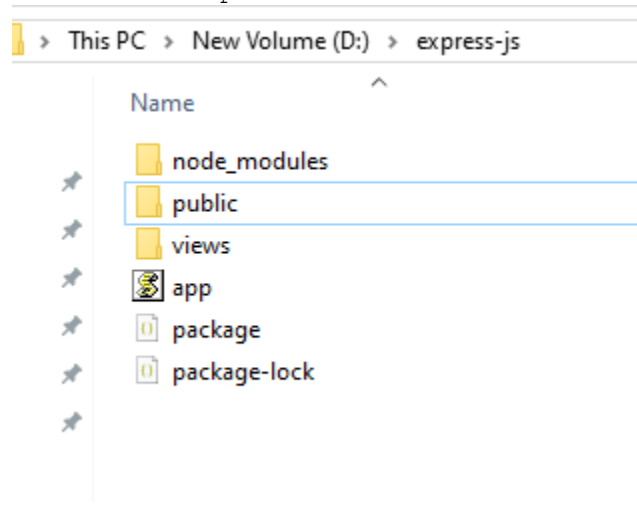
🔗 [Step-by-Step Guide to Creating Pages in Express](#)

□ **1. Project Structure Example**

Let's aim for a simple structure like this:

```
express-js/
├── app.js
├── package.json
├── public/
│   └── styles.css      (optional CSS)
├── views/
│   ├── index.html    (Home Page)
│   ├── about.html    (About Page)
│   └── contact.html  (Contact Page)
```

As shown example:-



2. Set Up Express to Serve Static Files and Pages

Update your `app.js` like this:

```
const express = require('express');
const path = require('path');
const app = express();
const PORT = 3000;

// Serve static files like CSS
app.use(express.static(path.join(__dirname, 'public')));

// Routes
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'views', 'index.html'));
});

app.get('/about', (req, res) => {
  res.sendFile(path.join(__dirname, 'views', 'about.html'));
});

app.get('/contact', (req, res) => {
  res.sendFile(path.join(__dirname, 'views', 'contact.html'));
});

app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
```

3. Create Basic HTML Pages in /views Folder

views/index.html

```
<!DOCTYPE html>

<html>

<head>

  <title>Home</title>

  <link rel="stylesheet" href="/styles.css">

</head>

<body>

  <h1>Home Page</h1>

  <nav>

    <a href="/">Home</a> |

    <a href="/about">About</a> |

    <a href="/contact">Contact</a>

  </nav>

  <p>Welcome to the home page!</p>

</body>

</html>
```

views/about.html

```
html
CopyEdit
<!DOCTYPE html>
<html>
<head>
  <title>About</title>
  <link rel="stylesheet" href="/styles.css">

</head>
<body>
  <h1>About Page</h1>
  <nav>
    <a href="/">Home</a> |
    <a href="/about">About</a> |
```

```
    <a href="/contact">Contact</a>
  </nav>
  <p>This is the about page.</p>
</body>
</html>
```

views/contact.html:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Contact</title>
  <link rel="stylesheet" href="/styles.css">

</head>
<body>
  <h1>Contact Page</h1>
  <nav>
    <a href="/">Home</a> |
    <a href="/about">About</a> |
    <a href="/contact">Contact</a>
  </nav>
  <p>Get in touch with us here.</p>
</body>
</html>
```

Step-by-Step: Add CSS to Your Express Pages

□ 1. Create a `public/` folder

Inside your project folder (`d:\express-js`), create a folder named `public`.

```
body {
  font-family: Arial, sans-serif;
  background-color: #f2f2f2;
  margin: 0;
  padding: 0;
}
```

```
nav {
  background-color: #333;
```

```
padding: 10px;  
text-align: center;  
}
```

```
nav a {  
color: white;  
margin: 0 10px;  
text-decoration: none;  
font-weight: bold;  
}
```

```
nav a:hover {  
text-decoration: underline;  
}
```

```
h1 {  
color: #333;  
text-align: center;  
margin-top: 40px;  
}
```

```
p {  
text-align: center;  
font-size: 18px;  
}
```

□ 4. Run Your App Again

From the terminal:

```
node app.js
```

Now visit:

- <http://localhost:3000/> → Home
 - <http://localhost:3000/about> → About
 - <http://localhost:3000/contact> → Contact
-