

Using a Framework

This lesson teaches how to use a framework in order to create a web server in JavaScript.

WE'LL COVER THE FOLLOWING ^

- About Frameworks
- Choosing a Framework
- Installing Express

It's time to put your Node.js knowledge into practice and create a real-world web server in JavaScript. This is often called *back-end programming*.

You will build exactly the server that was used in the previous chapters dealing with client-side web development. To test your server code, you can go back to code examples from chapters 22 and 23, and only change the start of the server URL from `https://thejsway-server.herokuapp.com` to your own server URL (which would be `http://localhost:3000` if your server runs on your local machine).

We saw in the previous chapter that Node.js is a platform for building JavaScript applications outside the browser. as such, Node is well suited for creating *web servers* in JavaScript.

As a reminder, a web server is a machine built specially to publish resources on the Web.

About Frameworks

It's entirely possible to build a web server from scratch with Node, but we'll

It's entirely possible to build a web server from scratch with Node, but we'll take a different approach and use a framework for it.

In computer programming, a *framework* provides a standard way to design and structure an application. It typically takes care of many low-level details so that the developer can concentrate on high-level, business-related tasks.

Choosing a Framework

Among the many possible frameworks for creating a web server in JavaScript, we'll use one of the most well-known: *Express*. To paraphrase its [web site](#), Express is “a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications”.

In other words, Express provides a foundation on which you can easily and quickly build a web server.

Installing Express

The Express framework is available as an `npm` package and its installation is straightforward. First, you'll need an existing Node application with a `package.json` file it. Run the following command in a terminal open in your application folder to install Express as a dependency.

```
npm install express
```

As an alternative, you can directly add Express as a dependency in your `package.json` file and run the `npm install` command.

```
"dependencies": {  
  "express": "^4.15.3"  
},
```

