



# NodeJS

## INTRODUCTION TO EXPRESS

In the previous lessons, you got up and running with Node. You learned how to serve up a multi-page website and read/write data from files on the server. You probably found that while it wasn't too difficult, the code was verbose and a little complicated. However, you can easily see how complicated and hard to maintain things would get if you decided to start adding more features. [Express](#) is a tiny and simple framework that makes the things you did in the previous lessons even quicker and simpler, and does so in a way that makes projects even more easily extensible.

In this unit, we are going to be following the [express tutorial](#) on MDN. We'll go one lesson at a time, occasionally supplementing with a little deeper explanation or side projects to help you deepen your understanding of the material. There is a *ton* of information there, so be sure to take your time and READ EVERYTHING. The blue "notes" that are scattered throughout the tutorial often link to articles or other tutorials that will definitely improve your understanding of the content. Don't be lazy!

### Learning Outcomes

By the end of this lesson, you should be able to do the following:

#### Express web framework

- Describe Express and Node's main benefits.
- Describe the relationship between Node and Express.
- Explain what a module is and how Express fits in.
- Import and create modules.
- Describe asynchronous APIs.
- Describe and create route handlers.
- Describe and use middleware.
- Describe error handling in Express.
- Describe what the main parts of an Express app might look like.

#### Setting up a Node development environment

- Describe Express development environment.
- Import Express into an application using NPM.
- Create and run applications using the Express application generator tool.
- Set up a development environment for Express on your computer.



## Assignment

1. Read the [introductory lesson](#) on the MDN website. It is long but its main purpose is to show you the various bits and pieces that you'll be learning in the rest of the tutorial. If you want you can code along a bit, but most of the examples aren't really intended for you to follow 100%. *DO* read the entire thing! Some of it will be a review, but that's OK! Follow the links that they give and at least look at them. They are a vital part of this lesson and will often direct you to the relevant portion of the official express docs (which are quite good)! You'll want to be somewhat familiar with them when it comes time to do your own projects.
2. If you've gotten this far, you should already have node and npm set up and running smoothly. The second lesson on MDN's website walks you through setting them up and goes on to talk about a couple of ways to install and get started with Express. Read the [second half](#) of that article.
3. Once you're all set up, take a look at the simple Node site you created in our first project. Rewrite it using express! You should be able to do this with just one app.js file and a few `app.get()`s.

## Additional Resources

This section contains helpful links to other content. It isn't required, so consider it supplemental for if you need to dive deeper into something.

- The book [Express in Action](#) by Evan M. Hahn is an incredible resource for Express beginners.
- [This](#) video crash course provides you with all the basic concepts.

[View Course](#)

[Login to track progress](#)

[Next Lesson](#)

 [Improve this lesson on GitHub](#)



# Have a question?

Chat with our friendly Odin community in our Discord chatrooms!

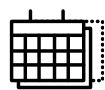
[Open Discord](#)

Are you interested in accelerating your web development learning experience?

[Get started](#)



[Thinkful](#)



5-6 months



Job Guarantee



1-on-1 Mentorship



[THE ODIN PROJECT](#)



[About](#)

[FAQ](#)

[Blog](#)

[Success Stories](#)

[Contribute](#)

[Terms of Use](#)

