



Express web framework (Node.js/JavaScript)

Express is a popular unopinionated web framework, written in JavaScript and hosted within the Node.js runtime environment. This module explains some of the key benefits of the framework, how to set up your development environment and how to perform common web development and deployment tasks.

Prerequisites

Before starting this module you will need to understand what server-side web programming and web frameworks are, ideally by reading the topics in our [Server-side website programming first steps](#) module. A general knowledge of programming concepts and [JavaScript](#) is highly recommended, but not essential to understanding the core concepts.

Note: This website has many useful resources for learning JavaScript *in the context of client-side development*: [JavaScript](#), [JavaScript Guide](#), [JavaScript Basics](#), [JavaScript \(learning\)](#). The core JavaScript language and concepts are the same for server-side development on Node.js and this material will be relevant. Node.js offers [additional APIs](#) for supporting functionality that is useful in browserless environments (e.g., to create HTTP servers and access the file system), but does not support JavaScript APIs for working with the browser and DOM.

This guide will provide some information about working with Node.js and Express, and there are numerous other excellent resources on the Internet and in books — some of these linked from [How do I get started with Node.js](#) (StackOverflow) and [What are the best resources for learning Node.js?](#) (Quora).

Guides

[Express/Node introduction](#)

In this first Express article we answer the questions "What is Node?" and "What is Express?" and give you an overview of what makes the Express web framework special. We'll outline the main features and show you some of the main building blocks of an Express application (although at this point you won't yet have a development

environment in which to test it).

[Setting up a Node \(Express\) development environment](#)

Now that you know what Express is for, we'll show you how to set up and test a Node/Express development environment on Windows, Linux (Ubuntu), and Mac OS X. Whatever common operating system you are using, this article should give you what you need to be able to start developing Express apps.

[Express Tutorial: The Local Library website](#)

The first article in our practical tutorial series explains what you'll learn and provides an overview of the "local library" example website we'll be working through and evolving in subsequent articles.

[Express Tutorial Part 2: Creating a skeleton website](#)

This article shows how you can create a "skeleton" website project, which you can then go on to populate with site-specific routes, templates/views, and databases.

[Express Tutorial Part 3: Using a Database \(with Mongoose\)](#)

This article briefly introduces databases for Node/Express. It then goes on to show how we can use [Mongoose](#) to provide database access for the *LocalLibrary* website. It explains how object schema and models are declared, the main field types, and basic validation. It also briefly shows a few of the main ways you can access model data.

[Express Tutorial Part 4: Routes and controllers](#)

In this tutorial we'll set up routes (URL handling code) with "dummy" handler functions for all the resource endpoints that we'll eventually need in the *LocalLibrary* website. On completion, we'll have a modular structure for our route handling code, that we can extend with real handler functions in the following articles. We'll also have a really good understanding of how to create modular routes using Express.

[Express Tutorial Part 5: Displaying library data](#)

We're now ready to add the pages that display the *LocalLibrary* website books and other data. The pages will include a home page that shows how many records we have of each model type and list and detail pages for all of our models. Along the way, we'll gain practical experience in getting records from the database and using templates.

[Express Tutorial Part 6: Working with forms](#)

In this tutorial we'll show you how to work with [HTML Forms](#) in Express, using Pug, and in particular how to write forms to create, update, and delete documents from the database.

[Express Tutorial Part 7: Deploying to production](#)

Now you've created an awesome *LocalLibrary* website, you're going to want to install it on a public web server so that it can be accessed by library staff and members over the Internet. This article provides an overview of how you might go about finding a host to deploy your website, and what you need to do in order to get your site ready for production.

Adding more tutorials

All existing tutorials are listed above, but if you would like to extend this module, some other interesting topics to cover include:

- Using sessions.
- User authentication.
- User authorization and permissions.
- Testing an Express web application.
- Web security for Express web applications.

An assessment for the module would also make a wonderful addition!

Last modified: Mar 1, 2021, [by MDN contributors](#)

Change your language

English (US) ▼

Change language