

Introduction to Node.js

Course Summary

Description

Introduction to Node.js teaching the fundamental concepts involved in writing server-side applications using Node.js. Students will learn about Node.js best practices and conventions while working with popular libraries, frameworks, and back-end services to write a variety of different applications.

Topics

- Node.js concepts and conventions
- Understanding callbacks and asynchronous code
- Finding and installing modules with NPM
- Creating, using, and publishing modules
- Listening for and handling events
- Working with streams
- Promises
- Web APIs
- Using Express

Audience

This course is for professional JavaScript programmers and back-end web developers.

Prerequisites

Before taking this course, students should have an intermediate to advanced level of knowledge with JavaScript and server-side development. Specifically, they should be comfortable working with JavaScript functions, closures, callbacks, and objects.

Duration

Two days

Introduction to Node.js

Course Outline

I. Intro to Node.js

- A. What is Node.js?
- B. What is it good for?
- C. History of Node.js
- D. Who Uses Node?
- E. How Does Node.js Work?
- F. How is Node.js Different?
- G. What is Node.js Made Of?
- H. Blocking vs. Non-blocking
- I. Your First Node Program
- J. Running a Node.js program

Lab: Intro to Node

Lab: First Look at Asynchronous Code

II. Node Modules

- A. CommonJS Example
- B. Using Modules
- C. Accessing Modules with require()
- D. Node's Core Modules
- E. The http Module
- F. Lab 03: A Simple Node.js Server
- G. The fs module
- H. Buffer Objects
- I. Modularizing Your Code
- J. Returning Values from Modules
- K. Using a Local Module

Lab: Creating Modules

III. Events and Streams

- A. Non-blocking with Events
- B. EventEmitter
- C. Emitting and listening
- D. EventEmitter Patterns
 - 1. return value
 - 2. extend EventEmitter
- E. Node Stream Objects
- F. Types of Streams
- G. Using Readable Streams
- H. Using Writable Streams
- I. Lab 05: Working with Streams
- J. Lab 06: Piping Between Streams
- K. The process Object
- L. Command Line Arguments

Lab: The process Object

IV. Promises

- A. What are Promises?
- B. Promises vs. Event Listeners
- C. Why Use Promises
- D. Demo: Callback vs. Promise
- E. Using Promises

Lab: Promises

- F. Promises with Bluebird

V. Node on the Web

- A. Making an HTTP Request

Lab: Getting Data with HTTP

Lab: Making a Bot

Lab: Making a Hello World Bot

VI. Testing in Node.js

- A. The assert Module
 - B. Using assert
- Lab: Testing with assert**
- C. Mocha and should.js
 - D. Testing with Mocha and should.js

VII. Using Express

- A. What is Express?
- B. Getting Started with Express
- C. Routing with Express
- D. Serving Static Content with Express

Lab: Make an Express Server