ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Introduction to Development with Node.js

Course Summary

Description

The Node.js is an open-source, cross-platform, runtime environment that allows JavaScript to run outside of the browser. Developers can create desktop and server applications using many provided libraries including those for work with the file system and HTTP. The community provides a multitude of tools and code that can help to create applications quickly using best practices.

This course combines lecture and hands-on exercises to get students up to speed quickly. Students will work with core modules of Node.js, and explore popular modules such as Express. During the course students will develop a web application that allows records to be Created, Read, Updated, and Deleted (CRUD). The course can be customized for different data stores, or for those who want more in-depth programming experience.

Objectives

By the end of this course, students will be able to:

- Understand the overall architecture and features of Node.js
- Install and setup an environment using Node and NPM
- Build a Node.js server application
- Work with some of the most common core modules
- Create a module in a TDD manner using Mocha and chai testing tools
- Practice with common Express.js middleware
- Build a CRUD application using REST API servers

Topics

- Reviewing Modern JavaScript Concepts
- Getting Started with Node
- The Node Package Manager (NPM)
- Node Fundamentals
- File Access with Node
- Database Access with Node
- Creating a Simple HTTP Server to receive requests
- Getting Started with the Express Module
- Session Management in Node
- Validation, CSRF and Database Relations
- Unit Testing Node

Audience

This course is designed for experienced JavaScript developers who wish to understand Node JS and to execute JavaScript outside of the browser.

Prerequisites

Before taking this course, programming experience with HTML, CSS, and JavaScript is required. Knowledge of REST based web services is helpful, but not required.

Duration

Three days

ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Introduction to Development with Node.js

Course Outline

I. Reviewing Modern JavaScript Concepts
- A. Distinguishing between EcmaScript and JavaScript
- B. Using ES6 syntax
- C. Practicing with arrow functions
- D. Using promises
- E. Leveraging JavaScript modules
  1. Using the CommonJS interface for modules
  2. Using the require with native modules
  3. Using the require with relative files

II. Getting Started with Node
- A. Overview of Node.js
  1. Background and overview of Node.js
  2. Features of Node.js
  3. The Node.js Runtime
- B. Environment Setup
  1. Helpful Tools and Editors
  2. Installation of Node
  3. Installing build tools
- C. Using the REPL Terminal
- D. Resources
  1. API Reference Documentation

III. The Node Package Manager (NPM)
- A. Background of NPM and its usage
  1. The meaning and usage of packages
  2. Searching repositories
  3. Installing Modules using NPM client
  4. Global vs. local package installation
  5. Listing installed modules
  6. Uninstalling a module
  7. Updating a module
- B. Creating and using package.json files
  1. Understanding how to use semver
  2. When to use --save and --save-dev

IV. Node Fundamentals
- A. Understanding the Event Loop
  1. Blocking Code Example
  2. Non-Blocking Code Example
  3. Concurrency support with events and callbacks
- B. The role of callbacks in Node.js
  1. Achieving sequential behavior with asynchronous calls
- C. The EventEmitter pattern and methods
  1. Binding callbacks to events
  2. Using emit to fire events
- D. Improving asynchronous calls using Promises
  1. Using Bluebird for Promises

V. File Access with Node
- A. Using Node.js global objects
  1. Get relative path information
  2. Accessing current file information
  3. Setting intervals and timeouts for callbacks
- B. Use the fs module to
  1. Open files
  2. Write files
  3. View directories
  4. Create directories
  5. Synchronous vs. asynchronous I/O
  6. Path and directory operations
- C. Working with Streams
  1. Writing and Reading from Stream Resources
  2. Piping and Chaining Streams
- D. Using buffers for binary data

VI. Database Access with Node
- A. Overview of course database
- B. Installing and managing database
- C. Using Knex as a Query Builder
- D. Creating a database and tables with Node
- E. CRUD with Node
- F. Creating a Migration for managing database changes

Introduction to Development with Node.js

Course Outline (cont'd)

VII. Creating a Simple HTTP Server to receive requests
- A. Reviewing the HTTP protocol
- B. Discussing HTTP Request methods
- C. Reviewing Response Codes
- D. Creating a HTTP basic server
- E. Installing and using nodemon for faster development

VIII. Getting Started with the Express Module
- A. Creating the course CRUD application using Node
- B. Configuring Express for our web server
- C. Creating Routing files
- D. Using Pug (formerly Jade) for templating
- E. Using express.static to access static assets
- F. Using middleware to add error handling

IX. Session Management in Node
- A. Creating a Login Route
- B. Authenticating Login with database
- C. Allowing User Registration
- D. Sharing State of User
- E. Logging out to end the Session

X. Validation, CSRF and Database Relations
- A. Adding validation to forms
- B. Addressing Cross-Site Request Forgery (CSRF)
- C. Using relational data with foreign keys

XI. Unit Testing Node
- A. Using the Mocha testing framework
- B. Creating assertions with Chai
- C. Creating test doubles with Sinon
- D. Using Istanbul for code coverage