Node.js Application Development

Duration: 4 Days *(Face-to-Face & Remote-Live)*, or 28 Hours *(On-Demand)*

Price: $2095 *(Face-to-Face & Remote-Live)*, or $895 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more information.

Delivery Options: Attend face-to-face in the classroom, remote-live or via on-demand training.

Description

This hands on Node.js course will teach front end designers and server side developers how to rapidly create, maintain, and deploy Node.js applications. This course includes thorough coverage of the Node.js architecture, the V8 engine and the Node Ecosystem. The course takes a practical approach to creating and organizing Node.js applications using functions and modules as part of the Node.js organizational structure. This course will provide attendees with a comprehensive understanding of Node.js core modules and how to procure and install packages using npm. Attendees will learn to identify and correct problems through unit testing and exception handling.

Attendees will learn how to use Node.js to create Command Line Interface applications for system administration and process management, use Node.js networking modules to communicate with TCP/IP clients and servers such as HTTP servers, and create / consume REST (Representational State Transfer) data services. Attendees will learn how to leverage frameworks such as Express to rapidly build Web Applications. Attendees will learn how to use Node.js to connect to NoSQL databases such as MongoDB to store, retrieve, and manipulate data (i.e., achieve data persistence).

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Prerequisites

Knowledge of JavaScript equivalent to attending the JavaScript Programming course. Attendees should also have familiarity with a Windows, Linux, or OS X command line interface as well as a basic understanding of network protocols such as HTTP.

Course Overview

About Node.js

- Server Side JavaScript
- The Event Loop
- Asynchronous Execution
- `npm`
- Node.js Architecture
- Node.js Core Modules
- Node.js Projects and Applications
- Node.js Command Line Arguments

Node.js Global Environment

- Timers
- Global Objects
- The `process` Object
- Buffers and Buffer Manipulation

ES6 Enhancements

- Arrow Functions
- `let` and `const` Keywords
- Enhanced Object Literals
- Map and Set Collections
- `WeakMap` and `WeakSet` Collections
- String Template Literals

Installation and Configuration

- Node.js
- `npm`
- Express
- MongoDB
- Mongoose

Programming Fundamentals with Node

- Using Modules to Organize Code
  - Creating and Reusing Code with Modules
- Asynchronous Programming Fundamentals
- Using Callbacks to Handle Events
- Using `EventEmitters` to Handle Repeating Events
- Type Testing with the `util` Module
- The Node.js Module Loading System
- Module Caching

Unit Testing and Debugging with Node.js

- Implementing Unit Tests with the `Assert` Module
- Test Driven Development with Nodeunit
- Behavioral Testing with Mocha
- Facilitating Debugging with the `Console` Module
- Working with the Node.js Debug Utility

System and Data Interaction With Node

- Interacting with the System Using the `os` Module
- Working with the File System Using the `fs` and `path` Modules
- Managing Data Flow with Pipes and Streams
- Synchronous vs. Asynchronous File System Interaction
- Using Buffers to Read and Manipulate Binary Data
- Handling and Transforming File Path Information with the `path` Module
- Spawning and Managing Processes

Data Persistence with Node.js

- Data Persistence Mechanisms
- MongoDB Capabilities
- Connecting to MongoDB
- Using the `fs` Module to Read and Write Files Synchronously and Asynchronously

using the `process` Module

Web Application and Network Programming

- Using Express to Rapidly Build and Deploy Web Applications
- REST-Based Web Service Fundamentals
- Creating HTTP Servers and Clients Using the `http` Module
  - Sending HTTP Requests and Receiving HTTP Responses
- Making DNS Requests with the `dns` Module
- Creating TCP Clients and Servers with the `net` Module

Creating Modules

- Planning Modules
- Working with the `package.json` File
- Working with Dependencies
- Understanding Semantic Versioning
- Testing Modules

Process Management with the `process` Module

- Executing External Applications
- Attaching and Detaching Child Processes
- Facilitating Inter-Process Communication between Node.js Processes
- Making Node.js Applications Executable

Software Skills Training, Inc.
6 Hemlock Drive
Chelmsford, MA 01824
978.250.4983
www.software-skills-training.com

Copyright © 2021 Software Skills Training, Inc.