2018

Web Development with Node.js

Seikyung Jung
Bridgewater State University, sjung@bridgew.edu

WEB DEVELOPMENT WITH NODE.JS [*]

TUTORIAL PRESENTATION

Seikyung Jung
Computer Science Department
Bridgewater State University
131 Summer Street, Bridgewater, MA 02325
508 531-2222
sjung@bridgew.edu

This tutorial demonstrates how to teach a Web development course by building web applications with Node.js and Express then deploying to Heroku, a cloud hosting service. This tutorial assumes some familiarity with HTML and JavaScript, but no prior experience with Node.js is necessary. The tutorial covers all necessary setup and step-by-step instructions to build a sample web application with these technologies. The tutorial concludes by describing how to incorporate Node.js, PostgreSQL, Git and Heroku in a web development course, and my experiences with using it for the past two years

TUTORIAL

Web development has advanced rapidly over the past two decades, with an abundance of competing technologies for faculty, students and industry to choose among, study and try. Although the web benefits from these advances, faculty who teach web development may agonize over the growing number of choices of what to cover in a short semester. Essentials such as HTML, CSS and JavaScript typically consume enough of a web development course that coverage of the server-side (i.e., yet another language) may get neglected, despite its critical role in web applications.

This tutorial addresses these concerns by introducing web development with Node.js, a modern JavaScript server-side framework featuring an event-driven, asynchronous API consistent conceptually with standardized JavaScript APIs found in web browsers. Using Node.js has the pedagogical advantage of limiting the number of languages and new concepts to introduce to students in a semester, thereby freeing time to focus their attention on developing and deploying web applications.

The tutorial introduces participants to the concepts, terminology and implications of event-driven programming and asynchronous I/O. Event-driven programming decouples changes to object state (events) from actions (listeners), typically by registering (subscribing) listeners with the objects publishing events of interest, as is the case in programming graphical user interfaces. Asynchronous I/O means that a program can do other things (thereby avoiding idling) while slow I/O operations complete, resulting in increased throughput and reduced latency compared to synchronous (blocking) I/O. Web browsers feature standardized, event-driven, asynchronous APIs (i.e., DOM and AJAX) to enable responsive user interfaces. Likewise, Node.js features an event-driven, asynchronous server-side API conceptually consistent with APIs found in web browsers to enable high throughput, low-latency web applications.

The tutorial then motivates using Node.js in a web development course. Node.js features an easy to use package manager (npm) that simplifies building web applications with third-party components, such as Express, Angular, React and more. Express's API organizes web applications into the model-view-controller (MVC) architecture, thereby decoupling routes (the controller) from logic (the model) and decoupling presentation (views) from logic either through templates or by serving model state as JSON asynchronously to the web browser. I describe how exposing students to these design choices in a web development course encourages good habits that may suit them well in later courses, particularly software engineering. I also discuss the pedagogical disadvantages of teaching older server-side technologies such as PHP and CGI, as well as similar MVC web frameworks written in other languages (such as Django and Ruby on Rails).

Using their laptops, participants first install Node.js and these software packages as I discuss their purposes in turn: Heroku CLI, PostgreSQL, and Git. Public-facing web applications do not run on laptops; they run on servers, hence the need for deployment. Heroku is a cloud web-hosting platform that offers free hosting and supports multiple web frameworks, including Node.js. Participants will create their free accounts on Heroku's web site and then install the Heroku CLI (a command-line interface) to complete Heroku setup. PostgreSQL is an open source relational database hosted on Heroku's servers, and is offered free of charge. Heroku uses Git, the distributed version control system, to handle deployment of web application updates to their servers. With this suite of software installed, participants have all necessary tools to complete the tutorial.

Participants then complete a detailed step-by-step hands-on tutorial for building a simple web application written in Node.js, featuring access to a database behind an authentication layer (provided by the passport.js, express-session and bcryptjs packages). Given the limited time, some familiarity with HTML and JavaScript is assumed, but no prior experience with Node.js is necessary. The tutorial exercise exposes participants to the similarities between Node.js, programming in the browser, and the details of Node.js's architecture. Participants will use a PostgreSQL database and Git to deploy to Heroku.

The tutorial concludes by describing how to incorporate Node.js, PostgreSQL, Git and Heroku in a web development course, and my experiences with using it for the past two years, showcasing a portfolio of student projects using these technologies.

PRESENTER

Seikyung Jung is an Associate Professor in the Computer Science Department at Bridgewater State University.