Vocabulary

Client-Side -- refers to everything in a web application that is displayed or takes place on the client (end user device). This includes what the user sees, such as text, images, and the rest of the UI, along with any actions that an application performs within the user's browser.

App -- Something that has functionality not a static collection of content.

Client -- a desktop computer or workstation that is capable of obtaining information and applications from a server.

Server -- a computer program or device that provides a service to another computer program and its user, also known as the client.

Browser -- a computer program with a graphical user interface for displaying and navigating between web pages.

Framework -- a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful.

Middleware -- (In this context) A pluggable Node module that can handle web requests to varying degrees.

Thread -- A set of instructions that a computer's brain (CPU) can follow to get things done. It's like a to-do list for the computer, telling it what tasks to perform.

The Java Script Revolution

Javascript is a powerful, flexible , and elegant language. Many people are just now starting to take JavaScript seriously, even though the language as we know it has been around since 1996. The creator of Node Ryan Dahl saw potential in JavaScript for a server-side language.

Introducing Express

Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi page and hybrid web applications.

Minimal

Following the philosophy of “less is more,” express provides the minimal layer between your brain and the server in a way that is robust and filled with features.

Flexible

Express provides you a very minimal framework, and you can add in different parts of express functionality as needed, replacing whatever doesn’t meet your needs. Express always wants you to add what you need when you need it.

Web Application Framework

A website is a web application and a web page is a web application. But a web application can become: it can provide functionality to other web applications.

Single-page web applications

Instead of a website requiring a network request every time the user navigates to a different page, a single-page web application downloads the entire site to the client’s browser. After that initial download, navigation is faster because there is little or no communication with the server.

Multipage and hybrid web applications

Multipage web applications are a more traditional approach to websites. Each page on a website is provided by a separate request to the server. Multi Page web applications are not better than single page web applications or vice versa.

A Brief History of Express

The creator of Express TJ Holowaychuk describes Express as a web framework inspired by Sinatra. It also was deeply intertwined with connect, a “plugin” library for Node.

Upgrading to Express 4.0

As an open source project that continues to be developed and maintained, upgrading to Express 4.0 is easy if you come from Express 3.0. Changes introduced in this need version ensue:

- Connect has been removed from Express so with the exception of the static middleware, you will need to install the appropriate packages. Connect has been moving some of its middleware into their own packages for easy accessibility.

- body-parser is now its own package, which no longer includes the multipart middleware. Making it much safer to use.

- You no longer have to link Express routers into your application.

- *App.configure* was removed.

Node: A New Kind of Web Server

Node's approach to web servers is very minimal. Node is very easy to set up and configure too. Not to say that setting up Node is always an easy task but configuration options are simpler and more straightforward.

Another of Node's quirks is that node is single threaded. Single threading vastly simplifies the business of writing web apps, and if you need the performance of a multithreaded app, you can simply spin up more instances of Node, and you will effectively have the performance benefits of multithreading. It comes with a cost, though: They require considerable expertise to set up and tune to achieve a good performance.

Google V8 compiles JavaScript to native machine code; it does so transparently that from the perspective of a user it is being interpreted instead of compiled. Another pro for Node is that it is an incredible platform independent making it perfect for collaboration.

The Node Ecosystem

Node is the software that enables JavaScript to run on the server, uncoupled from a browser, which in turn allows frameworks written in JavaScript to be used. The advent of Node development has revitalized a new approach to database storage jocosely named "NoSQL." which is just "document databases" or "key/value pair databases."

There are acronyms describing the "stack" of multiple pieces of technology that can build a functional website. LAMP, for example, is a combination of Linux, Apache, MySQL, and PHP. MEAN is a combination of Mongo, Express, Angular, and Node. Which is catchy but ultimately not very useful.

The two other components that are usually essential for web app development are a database server and a templating engine. A templating engine provides the ability to seamlessly combine code and markup output. For the purpose of this book “JavaScript Stack” will be the name used to refer to Node, Express, and MongoDB

Licensing

One of the beauties of the Node ecosystem is the vast array of packages available to you. However, each of those packages carries its own licensing, and worse, each package may depend on other packages, meaning that understanding the licensing of the various parts of the app you’ve written can be tricky. Good news is; one of the most popular licenses for Node packages is MIT license which is painfully permissive although we should never assume every package is MIT licensed. Other Licenses include:

GNU General Public License GPL
: A very popular open source license that has been cleverly crafted to keep software free. Funny thing is that using GPL-licensed code in one’s project makes the project also GPL licensed.

Apache 2.0
: This license allows you to use a different license for your project, including a closed source license. You must, however, include notice of components that use the Apache 2.0 license.

Berkeley Software Distribution (BSD)
: This license allows you to use whatever license you wish for your project, as long as you include notice of the BSD-licensed components.

Dual Licensed
: Allow the software to be used in both GPL projects and projects with more permissive licensing. (For a component to be used in GPL software, the component must be GPL licensed. When writing a package we should be good and pick up a license and document it correctly.

IN A NUTSHELL

- Javascript is a powerful, flexible , and elegant language.
- Express is a minimal and flexible node.js web application framework. It is minimal, flexible, and adds a lot of functionalities as a framework to single and multi page web applications.

- TJ Holowaychuk was the creator of Express.
- Upgrading to Express 4.0 is easy and suggested.
- Node is relatively easy to set up.
- Node is also single threaded and allows for multithreaded functionality if having the expertise.
- Javascript is compiled by Google V8 but does it so fast an user would feel like it's interpreted.
- Licensing is very important to collaborate with other programmers. The most popular licenses in node are: GNU General Public License GPL, Apache 2.0, Berkeley Software Distribution (BSD) or even a Dual license.