What Is Node.js?

On November 8, 2009, Ryan Dahl delivered a speech to jsconf.eu, a conference on JavaScript, Introducing Node.js to the JavaScript community. He had become frustrated that concurrency (Doing more than one thing at once) is difficult in many programming languages and often leads to poor performance. He wanted to make it easier to write networked software that is fast, Supports many users, and uses memory efficiently. So he created Node.js.

Around the same time that Ryan Dahl was exploring ways to approach the problem, companies like Apple and Google started to heavily invest in browser technologies
.
With Google relying on browsers for the delivery of products like Gmail, Google's engineers created V8, a JavaScript engine for the Google Chrome web browser. This is a highly optimized piece of software designed specifically for the Web. Google wanted to encourage the development and adoption of V8, so it was open sourced under a Berkley Software Distribution (BSD) license.

Ryan Dahl decided to use the V8 engine to create a JavaScript server-side environment. This made sense for many reasons:

- The V8 engine is extremely fast.
- V8 is focused on the web, so it is proficient at things like Hypertext Transfer Protocol (HTTP), Domain Name System (DNS), and Transmission Control Protocol (TCP).
- JavaScript is a well-known language on the web, making it accessible to most developers.

At its core, Node.js is an event-driven server-side JavaScript environment. This means that you can create server-side applications with JavaScript in the same way that you can with languages like PHP, Ruby, and Python. It is particularly focused on networks and creating software that interacts with networks.

What You Can Do with Node.js

Node.js is a programming platform, so you are limited only by your imagination and programming skills. You can create small scripts to do things on a file system or large-scale web applications to run entire businesses. Because of the way Node.js is designed, **it lends itself well to multiplayer games, real-time systems, networked software, and applications that have thousands of concurrent users.**

Some companies using Node.js are

- LinkedIn
- eBay
- Yahoo !
- Microsoft

Some examples of applications that you can create with Node.js are

- Real-time multiplayer games
- Web-based chat clients
- Mashups that combine data sources from around the web
- Single-page browser applications
- JSON-based APIs

BY THE WAY

What Is Server-Side JavaScript?

Most web developers are familiar with using JavaScript to manipulate and interact with the content of web pages in a browser. This is commonly known as client-side JavaScript, because it happens in the browser or client. Server-side JavaScript occurs on the server before the page is sent to the browser. It is the same language, though, of course!

Verify installation: Executing a File

Create a js file named **main.js** on your machine (Windows or Linux) having the following code.

```
/* Hello, World! program in node.js */

console.log("Hello, World!")
```

Now execute main.js file using Node.js interpreter to see the result:

```
$ node main.js
```

If everything is fine with your installation, this should produce the following result:

```
Hello, World!
```

Hello World Server

```
var http = require('http');
http.createServer(function (req, res) {
res.writeHead(200, {'Content-Type': 'text/plain'});
res.end('Hello World\n');
}).listen(3000, "127.0.0.1");
console.log('Server running at http://127.0.0.1:3000/');
```

3. Save the file to your computer as server.js.

4. Run the program from the terminal:
node server.js

You should see Server running at http://127.0.0.1:3000 . This means that the server has started.

5. Open a web browser and visit http://127.0.0.1:3000. If you see Hello World, you successfully created your first Node.js program

6. To stop the server run, go back to the terminal and press Ctrl+C. This kills the Node.js process and stops the server.

Features of Node.js

Following are few of the important features which are making Node.js as the first choice of software architects.

- **Aynchronous and Event Driven** All APIs of Node.js library are aynchronous that is non-blocking. It essentially means a Node.js based server never waits for a API to return data. Server moves to next API after calling it and a notification mechanism of Events of Node.js helps server to get response from the previous API call.

- **Very Fast** Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

- **Single Threaded but highly Scalable** - Node.js uses a single threaded model with event looping. Event mechanism helps server to respond in a non-bloking ways and makes server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and same program can services much larger number of requests than traditional server like Apache HTTP Server.

- **No Buffering** - Node.js applications never buffer any data. These applications simply output the data in chunks.

- **License** - Node.js is released under the MIT license.

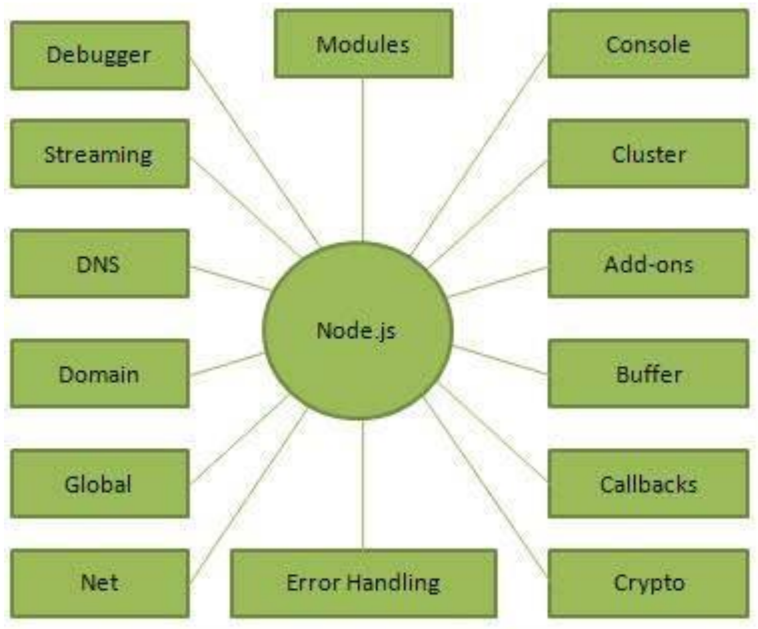
Who Uses Node.js?

Following is the link on github wiki containing an exhaustive list of projects, application and companies which are using Node.js. This list include eBay, General Electric, GoDaddy, Microsoft, PayPal, Uber, Wikipins, Yahoo!, Yammer and the list continues.

- Projects, Applications, and Companies Using Node

Concepts

The following diagram depicts some important parts of Node.js.



Where to Use Node.js?

Following are the areas where Node.js is proving itself a perfect technology partner.

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Realtime Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications