

Full Stack Development With Node.js

DANIEL SEONG



What is full stack and Node.js?

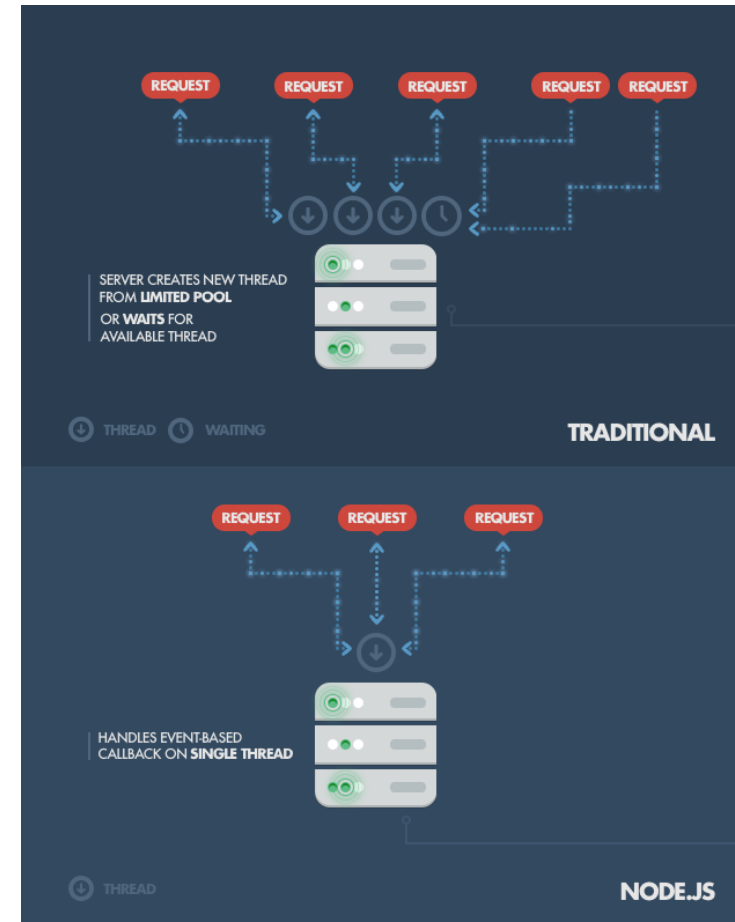
- Both backend and frontend development
 - Backend: Uses server-side code to connect to database
 - Frontend: Uses client-side code to display to client
 - Full stack connects the two!
- Node.js
 - Open source platform for developing server side applications

Why use Node.js?

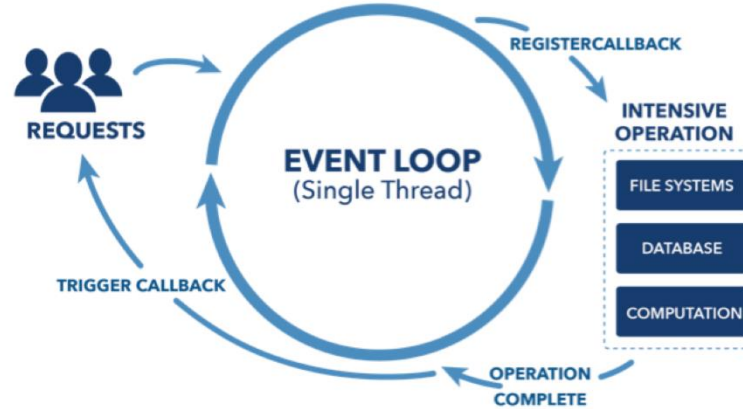
- JavaScript uniformity
 - Client-side vs Server-side development
 - Unifies language and data format (JSON)
- Fast, scalable network applications
 - Single-thread vs multiple concurrent threads
 - 2MB per thread, 8GB RAM gives 4,000 connections vs over 1M connections!
 - Does this with non-blocking with event loop
 - Example: messaging app that supports over 10,000 concurrent connections
- Node Package Manager (NPM)

```
{
  "empid": "SJ011MS",
  "personal": {
    "name": "Smith Jones",
    "gender": "Male",
    "age": 28,
    "address": {
      "streetaddress": "7 24th Street",
      "city": "New York",
      "state": "NY",
      "postalcode": "10038"
    }
  },
  "profile": {
    "designation": "Deputy General",
    "department": "Finance"
  }
}
```

www.kodingmadesimple.com



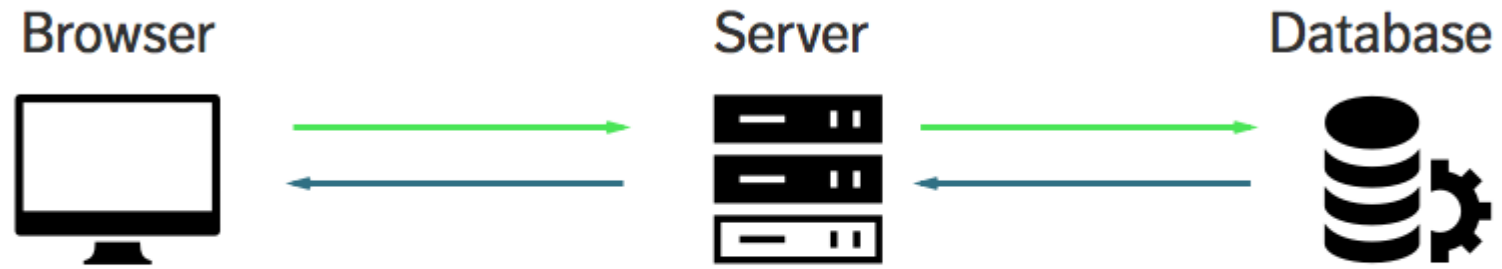
Event Loop



What we will be using

- Express
 - Framework that provides robust set of features for web applications
 - Idea of middleware
 - Allows to dynamically render HTML based on arguments passed to templates
- Dust
 - JavaScript templating engine
 - Way to build client side HTML pages with server-side arguments
- MySQL
 - Can use MongoDB to complete JavaScript uniformity and increased flexibility

Schema



Common functions

- `require()`
 - Load libraries and modules
- `app.use([path,] callback [, callback...])`
 - Mounts specified middleware function at specified path
- `app.set(name, value)`
 - Assigns setting “name” to “value”
- `app.get(path, callback [, callback ...])`
 - Routes HTTP GET requests to specified path
 - Think of GET requests as “getting” data from server
- `res.render(view, [locals], callback)`
 - Returns rendered HTML of a view via callback
- Reference: <https://expressjs.com/en/api.html>

Recap

- Node.js
 - Platform that allows you to make fast, scalable web applications
- Data
 - You know how to get data from user, store in a database, retrieve it, and display to user
- This is just an introduction...
 - Explore npm – some really cool libraries out there!
 - Explore different frameworks
 - Try building your own web application!