Event Loop

Introduction:

The event loop is a fundamental concept of Node.js. It opens the door to understanding Node's asynchronous processes and non-blocking I/O. It outlines the mechanisms that make Node a successful, powerful, and popular modern framework. This tutorial is useful for Node.js developers who want a deeper understanding of what's happening under the hood of every application, and those who want to take full control of every step of its running cycle.

Event Loop in Node Js:

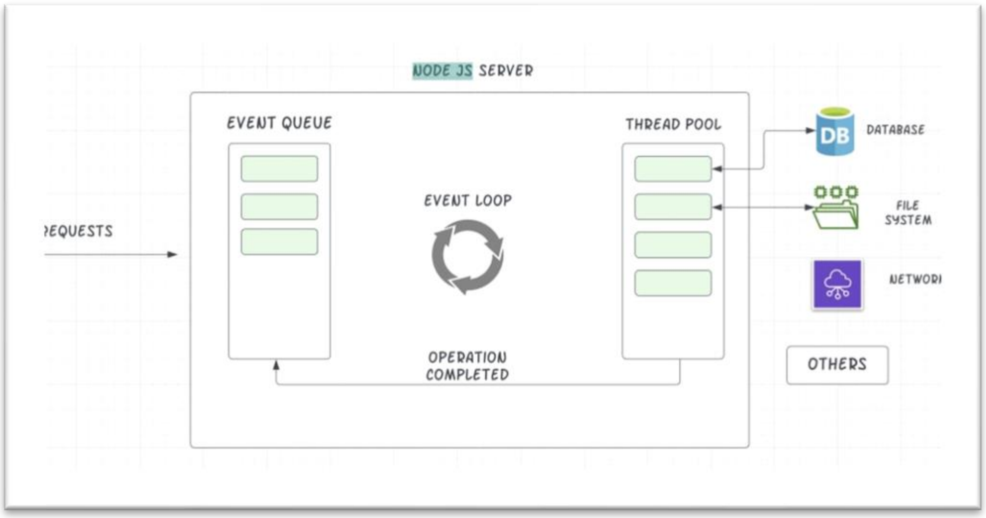

Figure 1, Event loop in Node Js

The Node Server consist of following parts:

1. **Event Queue -** On completion of the Thread Pool a callback function is issued and sent to the event queue. When call stack is empty the event goes through the event queue and sends callback to the call stack.
2. **Thread Pool -** The thread pool is composed of 4 threads which delegates operations that are too heavy for the event loop. I/O operations, Opening and closing connections, set Timeouts are the example of such operations.

3. Event loop in Node Js has different phases which has **FIFO** queue of callbacks to execute. When event loop enters a given phase it operates callbacks in that phase queue until the queue has been exhausted and maximum number of callbacks has executed and then moves to next phase.

<u>Components of Event Loop:</u>

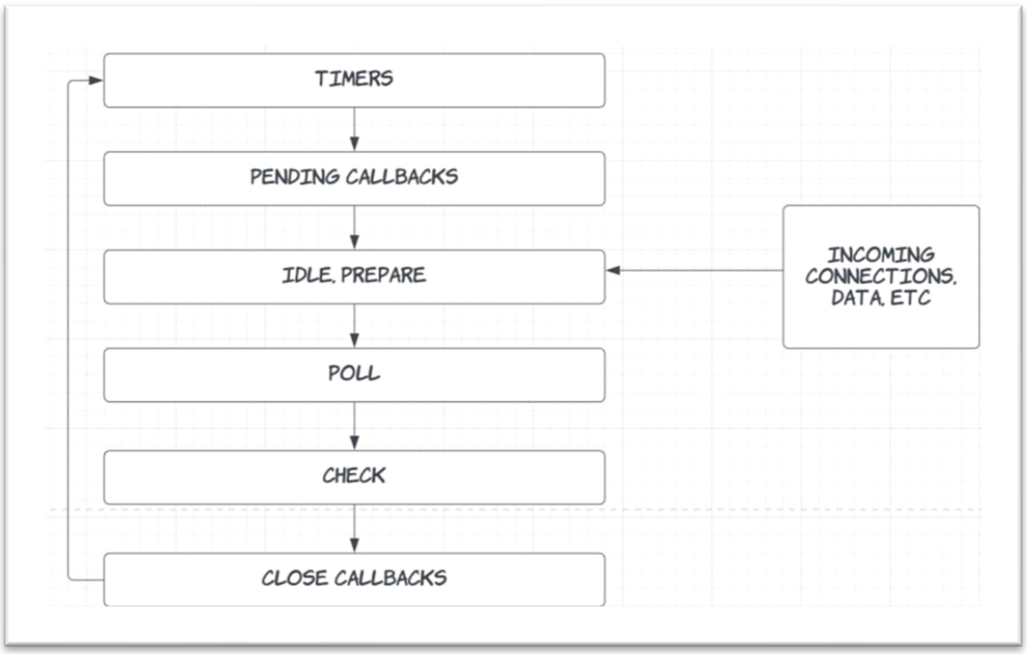

Figure 2, Components of Event Loop

The Event Loop is an endless loop in which waits for the tasks, executes them and then sleeps until it receives more tasks. The event loop executes tasks from queue only when stack is empty. It processes the oldest task first and allows us to use callbacks and promises.

References:

https://heynode.com/tutorial/how-event-loop-works-nodejs/

https://dev.to/jasmin/difference-between-the-event-loop-in-browser-and-node-js-1113