

Node.js Event Loop

Topics : [Node.js](#)

Written on [December 05, 2022](#)

Node.js Event Loop

Node.js is a single-threaded event-driven platform that is capable of running non-blocking, asynchronously programming.

it can support concurrency via the concept of event and callbacks.

The event loop allows Node.js to perform non-blocking I/O operations despite the fact that JavaScript is single-threaded.

Node.js Event Loop that listens for events, and then triggers a callback function when one of those events is detected.

The event loop executes tasks from the event queue only when the call stack is empty.

Syntax

```
var events = require('events'); // import events module to handle events

var EventEmitter = new events.EventEmitter(); //Create an EventEmitter object

eventEmitter.on('eventName', eventHandler); // create event

eventEmitter.emit('eventName'); // fire event
```

Example :

```
// Import events module
var events = require('events');

// Create an EventEmitter object
var EventEmitter = new events.EventEmitter();

// Create an event handler as follows
var firstHandler = function connected() {
```

```
console.log('Entered in first loop successfully.');
```

```
// Fire the secondLoop event
eventEmitter.emit('secondLoop');
}
```

```
// Bind the connection event with the handler
eventEmitter.on('firstloop', firstHandler);
```

```
// Bind the secondEvent event with the anonymous function
eventEmitter.on('secondLoop', function(){
  console.log('Entered in second loop successfully.');
```

```
});
```

```
// Fire the firstloop event
eventEmitter.emit('firstloop');
```

Output :

```
Entered in first loop successfully.
Entered in second loop successfully.
```