

# Async Hooks

In Node.js



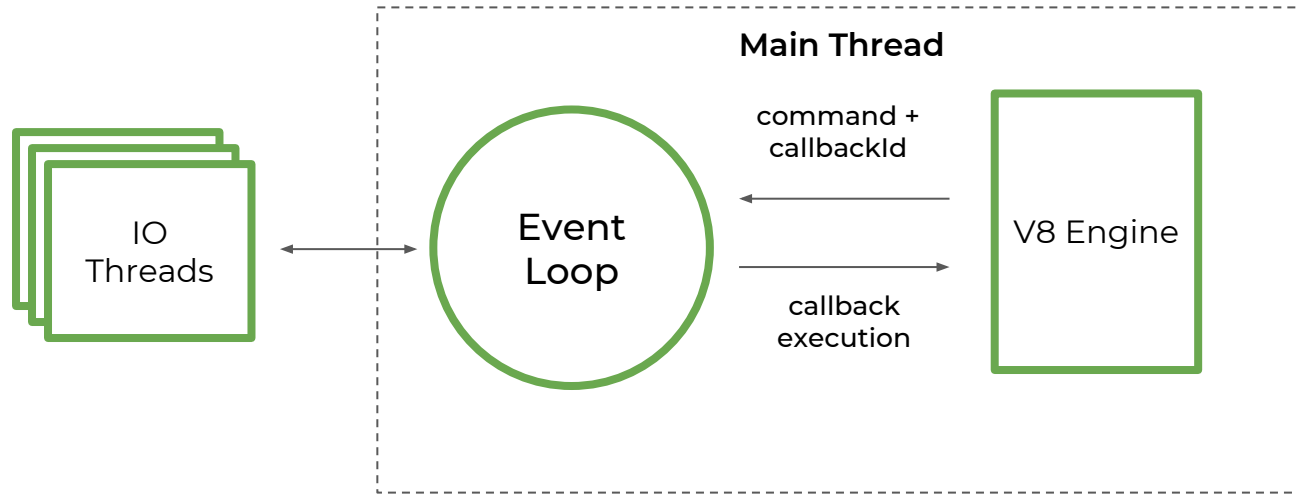
# Background

Node.js is event-loop-based single-thread system

Parallel processing is done through asynchronous calls

Callbacks (or now Promises) is used to notify job status

# Event Loop



# Test

```
1
2  Promise.resolve(true).then(() => {
3      console.log(1);
4      setImmediate(()=> {
5          console.log(2);
6      });
7  }).then(() => {
8      console.log(3);
9  });
10 console.log(4);
11
```

Output?

# Test

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/
```

4

1

3

2

## Test 2

```
1
2 let i = 1;
3 setImmediate(() => {
4   setImmediate(() => {
5     i = 0;
6   })
7   i++;
8 });
9 console.log(i);
10 setImmediate(() => {
11   console.log(i);
12 });
13 setTimeout(() => {
14   console.log(i);
15 }, 100);
```

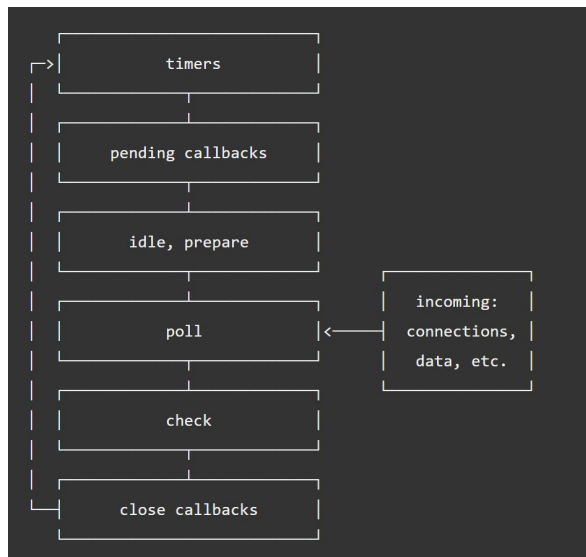
Output?

# Test 2

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/Users/rizqi  
1  
2  
0  
.
```

# Event Loop Phases

- **timers:** this phase executes callbacks scheduled by `setTimeout()` and `setInterval()`.
- **pending callbacks:** executes I/O callbacks deferred to the next loop iteration.
- **idle, prepare:** only used internally.
- **poll:** retrieve new I/O events; execute I/O related callbacks (almost all with the exception of close callbacks, the ones scheduled by timers, and `setImmediate()`); node will block here when appropriate.
- **check:** `setImmediate()` callbacks are invoked here.
- **close callbacks:** some close callbacks, e.g. `socket.on('close', ...)`.





# Synchronous Error

```
1
2  function foo() {
3    |   bar();
4  }
5
6  function bar() {
7    |   foobar();
8  }
9
10 function foobar() {
11 |   throw new Error("Oops!");
12 }
13
14 foo();
```

# Synchronous Error

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/Users/rizqi/Development/asynchooks$ node example1.js
/mnt/c/Users/rizqi/Development/asynchooks/example1.js:11
  throw new Error("Oops!");
    ^
```

Error: Oops!

```
  at foobar (/mnt/c/Users/rizqi/Development/asynchooks/example1.js:11:11)
  at bar (/mnt/c/Users/rizqi/Development/asynchooks/example1.js:7:5)
  at foo (/mnt/c/Users/rizqi/Development/asynchooks/example1.js:3:5)
  at Object.<anonymous> (/mnt/c/Users/rizqi/Development/asynchooks/example1.js:14:1)
  at Module._compile (module.js:652:30)
  at Object.Module._extensions..js (module.js:663:10)
  at Module.load (module.js:565:32)
  at tryModuleLoad (module.js:505:12)
  at Function.Module._load (module.js:497:3)
  at Function.Module.runMain (module.js:693:10)
```

# Asynchronous Error

```
1
2  function foo() {
3    |   setImmediate(bar);
4  }
5
6  function bar() {
7    |   setImmediate(foobar);
8  }
9
10 function foobar() {
11   |   throw new Error("Oops!");
12 }
13
14 foo();
```

# Asynchronous Error

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/Users/rizqi/Development/asynchooks$ node example2
/mnt/c/Users/rizqi/Development/asynchooks/example2.js:11
  throw new Error("Oops!");
      ^
```

Error: Oops!

```
  at Immediate.foobar [as _onImmediate] (/mnt/c/Users/rizqi/Development/asynchooks/example2.js:11:11)
  at runCallback (timers.js:810:20)
  at tryOnImmediate (timers.js:768:5)
  at processImmediate [as _immediateCallback] (timers.js:745:5)
```

# Real World Example

```
2   let express = require("express");
3
4   let app = express();
5
6   app.get("/hello", (req, res) => {
7     |   setTimeout(() => {
8       |     res.end("ok");
9       |     // some error happens
10      |     throw new Error("Ooops!");
11      |   }, 1);
12   });
13
14   app.listen(8000, () => {
15     |   console.log("app started.");
16   });
```

# Real World Example

GET  Params

Authorization Headers Body Pre-request Script Tests [Cookies](#) [Code](#)

**TYPE**

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies Headers (4) Test Results Status: 200 OK Time: 765 ms Size: 124 B

```
1 pk
```

# Real World Example

```
6  app.get("/hello", (req, res) => {
7    |   setTimeout(() => {
8    |     |   errorGeneratingFunction(1);
9    |     |   res.end("ok");
10   |   }, 1);
11   });
12
13  app.get("/world", (req, res) => {
14    |   setTimeout(() => {
15    |     |   errorGeneratingFunction(2);
16    |     |   res.end("hap");
17    |   }, 1);
18   });
19
20  function errorGeneratingFunction(x) {
21    |   setImmediate(() => {
22    |     |   if(x == 1)
23    |     |     |   throw new Error("Ooops!");
24    |     |   })
25   }
```

# Real World Example

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/Users/rizqi/Development/asynchooks$ node example3.js
app started.
/mnt/c/Users/rizqi/Development/asynchooks/example3.js:23
  throw new Error("Ooops!");
      ^
```

```
Error: Ooops!
    at Immediate.setImmediate (/mnt/c/Users/rizqi/Development/asynchooks/example3.js:23:19)
    at runCallback (timers.js:693:18)
    at tryOnImmediate (timers.js:664:5)
    at processImmediate (timers.js:646:5)
```



# What is AsyncHook

```
var asyncHooks = require('async-hooks')
```



- `createHooks ( hooks )`
- `enable ( )`  $\implies$  start tracking 🔍
- `disable ( )`  $\implies$  stop tracking

source:

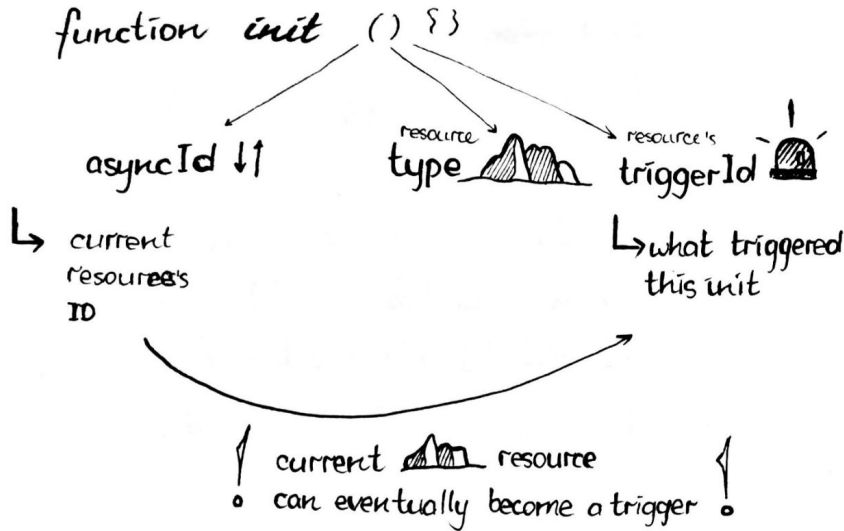
<https://medium.com/the-node-js-collection/async-hooks-in-node-js-illustrated-b7ce1344111f>

# AsyncHook Example

```
1  var fs = require("fs");
2  var http = require('http');
3  var asyncHooks = require('async_hooks');
4
5  function init (asyncId, type, triggerId) {
6    fs.writeFileSync(1, `${type} - ${asyncId}, ${triggerId}\n`);
7  }
8
9  var asyncHook = asyncHooks.createHook({init});
10 asyncHook.enable()
11
12 http.createServer(function (req, res) {
13   res.end('hello qts')
14 }).listen(8079);
```

# AsyncHook Callback

```
function init (asyncId, type, triggerId) {}
```



source:

<https://medium.com/the-node-js-collection/async-hooks-in-node-js-illustrated-b7ce1344111f>

# AsyncHook Example

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/Users/rizqi/Development/asynchooks$ node asynchook1.js
```

```
TCPSEVERWRAP - 5, 1  
TickObject - 6, 5  
TCPWRAP - 7, 5  
Timeout - 8, 7  
TIMERWRAP - 9, 7  
HTTPPARSER - 10, 7  
HTTPPARSER - 11, 7  
TickObject - 12, 7  
TCPWRAP - 13, 5  
Timeout - 14, 13  
HTTPPARSER - 15, 13  
HTTPPARSER - 16, 13  
TickObject - 17, 13  
TickObject - 18, 7  
TickObject - 19, 7  
TickObject - 20, 18  
TickObject - 21, 20  
Timeout - 22, 16  
TIMERWRAP - 23, 16  
TickObject - 24, 13  
TickObject - 25, 16  
TickObject - 26, 16  
TickObject - 27, 24  
Timeout - 28, 24  
TIMERWRAP - 29, 24  
TickObject - 30, 26  
TickObject - 31, 27  
TickObject - 32, 27  
TickObject - 33, 27  
TickObject - 34, 28
```

# Express Tracker Middleware

```
2  const {createHook, executionAsyncId} = require("async_hooks");
3
4  exports.createTracker = function() {
5    let contexts = {};
6    let asyncHook = createHook({
7      init: (asyncId, type, triggerId) => {
8        if (contexts[triggerId]) {
9          contexts[asyncId] = contexts[triggerId];
10        }
11      },
12      destroy: (asyncId) => {
13        delete contexts[asyncId];
14      }
15    });
16
17    asyncHook.enable();
18
19    process.on("uncaughtException", (error) => {
20      let id = executionAsyncId();
21      console.error(`(path=${contexts[id].path})\n`, error);
22    });
23
24    return function(req, res, next) {
25      let id = executionAsyncId();
26      contexts[id] = req;
27      next();
28    };
29  };
```

# Add middleware to Example

```
2 let express = require("express");
3 let tracker = require("./tracker");
4
5 let app = express();
6 app.use(tracker.createTracker());
7 app.get("/hello", (req, res) => {
8   |   setTimeout(() => {
9   |       |   errorGeneratingFunction(1);
10  |       |   res.end("ok");
11  |   }, 1);
12  });
13
14 app.get("/world", (req, res) => {
15   |   setTimeout(() => {
16   |       |   errorGeneratingFunction(2);
17   |       |   res.end("hap");
18   |   }, 1);
19  });
20
```

# Magic!

```
rizqme@DESKTOP-M6JUPK8:/mnt/c/Users/rizqi/Development/asynchooks$ node asynchook2.js
app started.
(path=/hello)
Error: Ooops!
  at Immediate.setImmediate (/mnt/c/Users/rizqi/Development/asynchooks/asynchook2.js:24:19)
  at runCallback (timers.js:693:18)
  at tryOnImmediate (timers.js:664:5)
  at processImmediate (timers.js:646:5)
□
```

# Thank you!

**Slides+Examples:** <https://github.com/rizqme/asynchook-example>

**Reference:** [https://nodejs.org/dist/latest-v10.x/docs/api/async\\_hooks.html](https://nodejs.org/dist/latest-v10.x/docs/api/async_hooks.html)

