

Debug Techniques in Node.js

Default v8 node-Debugger	2
Steps to debug using default v8 node-Debugger	2
Node-Inspector	3
Steps to debug using node-inspector	4
References	4
Angular Debugging	4

Default v8 node-Debugger

V8 node comes with a debugger, it has a built in client for this debugger to use it just add the debug argument when start running the script.

Steps to debug using default v8 node-Debugger

```
node debug <file>.js
```

it enters into the debug mode.like *debug*>

some basic commands to debug the script

- cont, c - Continue execution
- next, n - Step next
- step, s - Step in
- out, o - Step out
- pause - Pause running code (like pause button in Developer Tools)
- setBreakpoint(), sb() - Set breakpoint on current line
- clearBreakpoint, cb(...) - Clear breakpoint
-

Example:

<pre>test.js var sum = 0; for(i=0;i<10;i++) { sum = sum + i; console.log('Iteration:' + i +', sum:' + sum); }</pre>	<p><u>Debugging</u></p> <p>1. <u>Entering to the debug mode</u> <code>node debug test.js</code> < debugger listening on port 5858 connecting... ok break in scriptDebug.js:1 1 var sum = 0; 2 3 for(i=0;i<10;i++) {.....</p> <p>2. <u>continuing the execution</u> <code>node debug test.js</code> debug> cont < Iteration:0, sum:0 < Iteration:1, sum:1 < Iteration:2, sum:3 ----- -----</p>
---	---

```
< Iteration:9, sum:45  
program terminated
```

3. Set the Breakpoints

```
node debug test.js
```

```
debug> sb(5);
```

```
1 var sum = 0;  
2  
3 for(i=0;i<10;i++) {  
4 sum = sum + i;  
* 5 console.log('Iteration:' + i + ', sum:' +  
sum);  
6 }
```

4. watch the changes

```
node debug test.js
```

```
debug> watch('sum')
```

```
debug> cont
```

```
< Iteration:0, sum:0
```

```
break in scriptDebug.js:6
```

Watchers:

0: sum = 0

```
4 sum = sum + i;
```

```
5 console.log('Iteration:' + i + ', sum:' +
```

```
sum);
```

```
6 debugger;
```

```
7 }
```

```
8
```

```
debug> cont
```

```
break in scriptDebug.js:6
```

```
< Iteration:1, sum:1
```

Node-Inspector

Node inspector to have a visual aid to debug our script.

Steps to debug using node-inspector

Installing node-inspector:

```
npm install -g node-inspector
```

```
$ node-inspector
```

Node Inspector v0.7.3

Visit <http://127.0.0.1:8080/debug?port=5858> to start debugging.

With node inspector running in other terminal you have to start in debug mode your application with the following command:

```
$ node --debug-brk scriptDebug.js debugger listening on port 5858
```

The --debug-brk option is a node inspector option which starts to debug the script and stops it in the first line.

Note: if you installed node-inspector in ur project directory, then you can access as below
./node_modules/node-inspector/bin/inspector.js &

References

- <http://nodejs.org/api/debugger.html>
- <https://github.com/node-inspector/node-inspector>
- <http://blog.nodeknockout.com/post/34843655876/debugging-with-node-inspector>

Angular Debugging

- <http://ng-inspector.org/>
- <http://24days.in/umbraco/2014/debugging-angularjs/>
- <http://www.codelord.net/2015/07/07/my-angular-debugging-tips/>
- <http://blog.ionic.io/angularjs-console/>