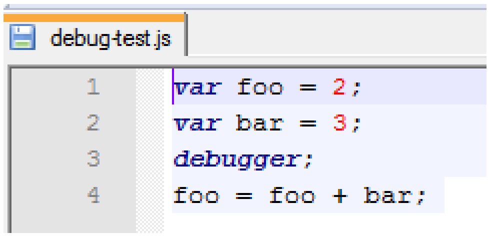
How to Debug in Nodejs

By

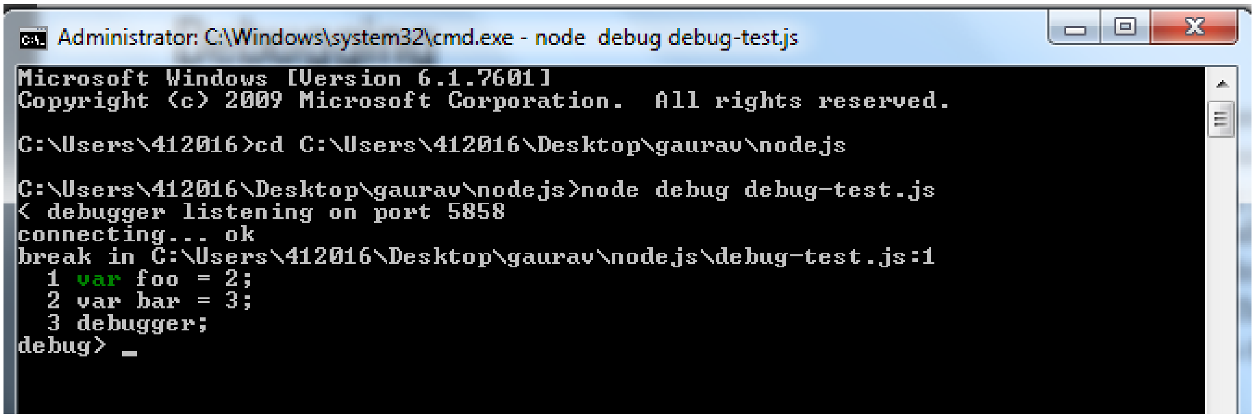
Gaurav Khurana

Lets create a file which we would like to debug



```
var foo = 2;
var bar = 3;
debugger;
foo = foo + bar;
```

On command line type following

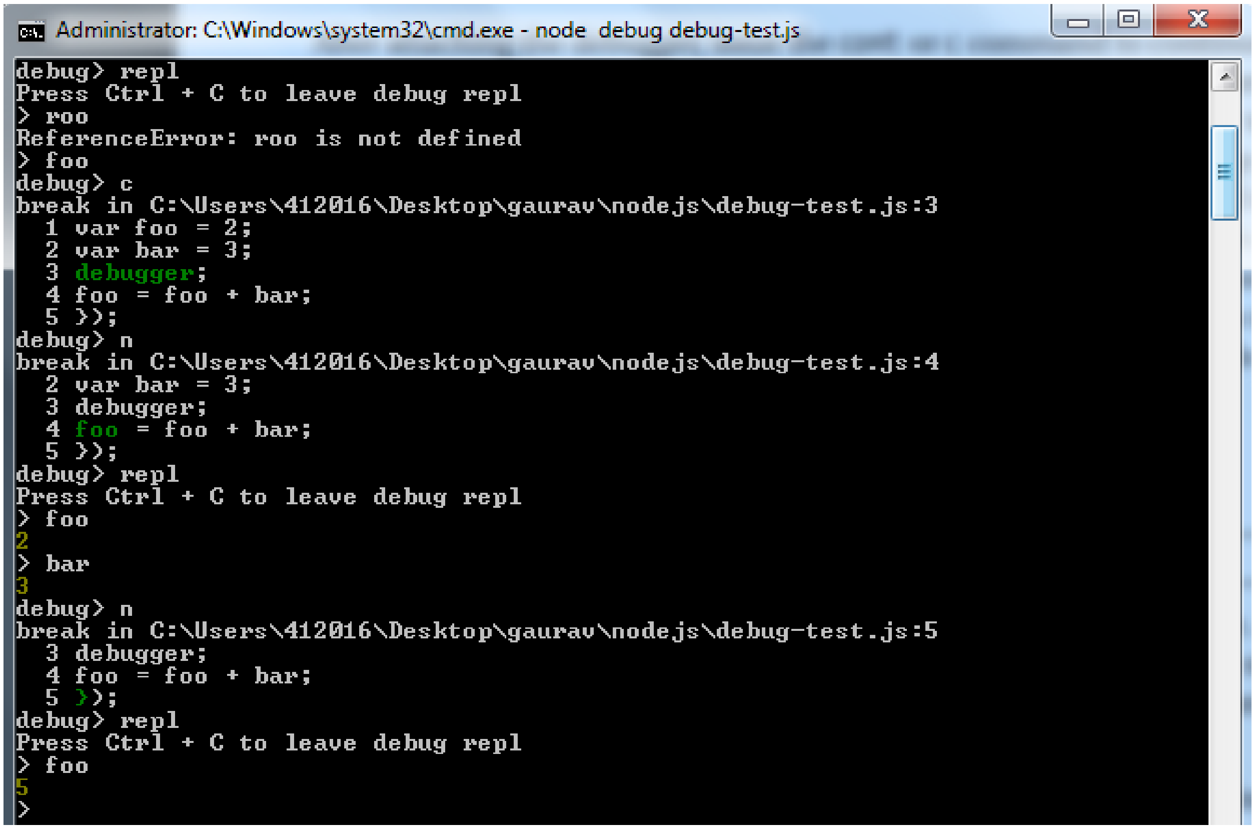

Administrator: C:\Windows\system32\cmd.exe - node debug debug-test.js

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\412016>cd C:\Users\412016\Desktop\gaurav\nodejs

C:\Users\412016\Desktop\gaurav\nodejs>node debug debug-test.js
< debugger listening on port 5858
connecting... ok
break in C:\Users\412016\Desktop\gaurav\nodejs\debug-test.js:1
  1 var foo = 2;
  2 var bar = 3;
  3 debugger;
debug>
```

Type con tot c to continue debugging

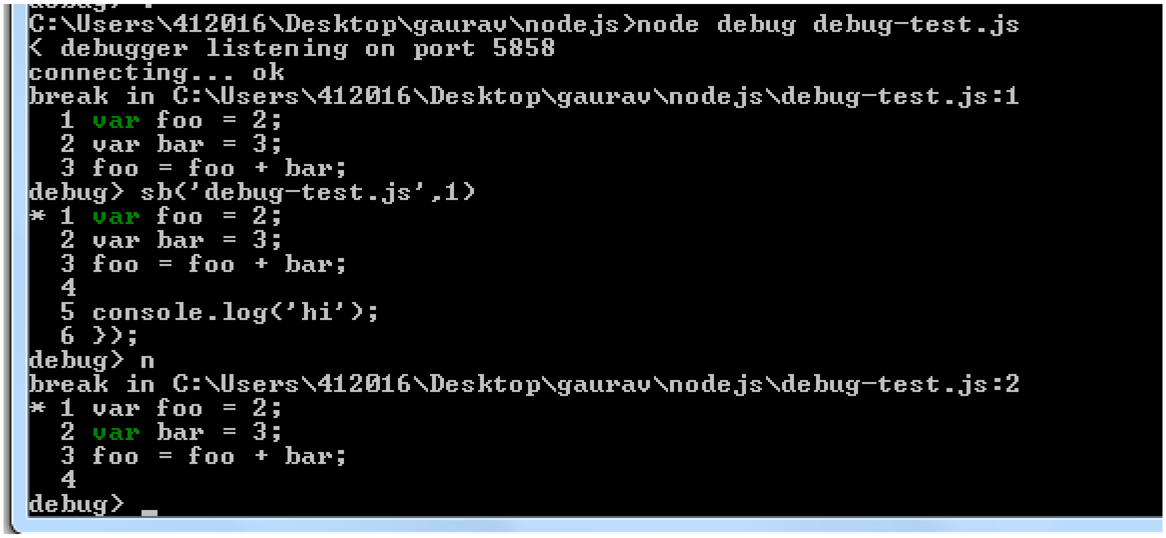
Type n to step to next line

Type repl any time to test variables and their values


Administrator: C:\Windows\system32\cmd.exe - node debug debug-test.js

```
debug> repl
Press Ctrl + C to leave debug repl
> roo
ReferenceError: roo is not defined
> foo
debug> c
break in C:\Users\412016\Desktop\gaurav\nodejs\debug-test.js:3
  1 var foo = 2;
  2 var bar = 3;
  3 debugger;
  4 foo = foo + bar;
  5 });
debug> n
break in C:\Users\412016\Desktop\gaurav\nodejs\debug-test.js:4
  2 var bar = 3;
  3 debugger;
  4 foo = foo + bar;
  5 });
debug> repl
Press Ctrl + C to leave debug repl
> foo
2
> bar
3
debug> n
break in C:\Users\412016\Desktop\gaurav\nodejs\debug-test.js:5
  3 debugger;
  4 foo = foo + bar;
  5 });
debug> repl
Press Ctrl + C to leave debug repl
> foo
5
>
```

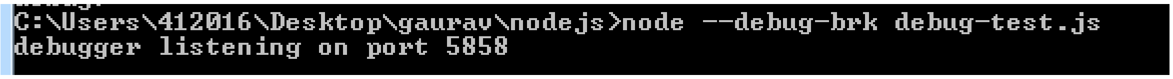
`setBreakpoint()` or `sb()`	Sets a breakpoint on the current line. As these are functions, you can also pass an argument to specify the line number on which to set the breakpoint. A breakpoint can be set on a line number in a specific file using the syntax `sb("script.js", line)`.
`clearBreakpoint()` or `cb()`	Clears a breakpoint on the current line. As when using `sb()`, you can pass arguments in to clear breakpoints on specific lines.
`backtrace` or `bt`	Prints the backtrace of the current execution frame.
`watch(expr)`	Adds the expression specified by `expr` to the watch list.
`unwatch(expr)`	Removes the expression specified by `expr` from the watch list.
`watchers`	Lists all watchers and their values.
`run`	Runs the script.
`restart`	Restarts the script.
`kill`	Kills the script.
`list(n)`	Displays source code with `n` line context (`n` lines before and `n` lines after the current line).
`scripts`	Lists all loaded scripts.
`version`	Displays the version of v8.



```
C:\Users\412016\Desktop\gaurav\nodejs>node debug debug-test.js
< debugger listening on port 5858
connecting... ok
break in C:\Users\412016\Desktop\gaurav\nodejs\debug-test.js:1
  1 var foo = 2;
  2 var bar = 3;
  3 foo = foo + bar;
debug> sb('debug-test.js',1)
* 1 var foo = 2;
  2 var bar = 3;
  3 foo = foo + bar;
  4
  5 console.log('hi');
  6 });
debug> n
break in C:\Users\412016\Desktop\gaurav\nodejs\debug-test.js:2
* 1 var foo = 2;
  2 var bar = 3;
  3 foo = foo + bar;
  4
debug> _
```

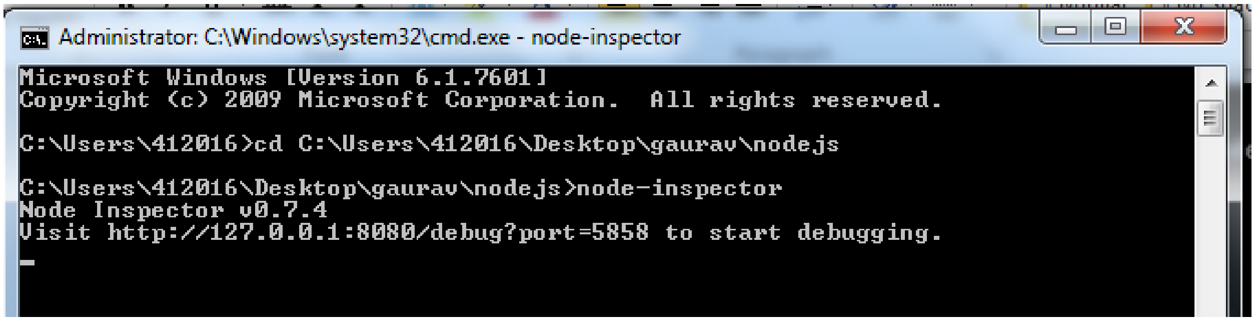
Real time debugging with chrome feeling

Install node-inspector



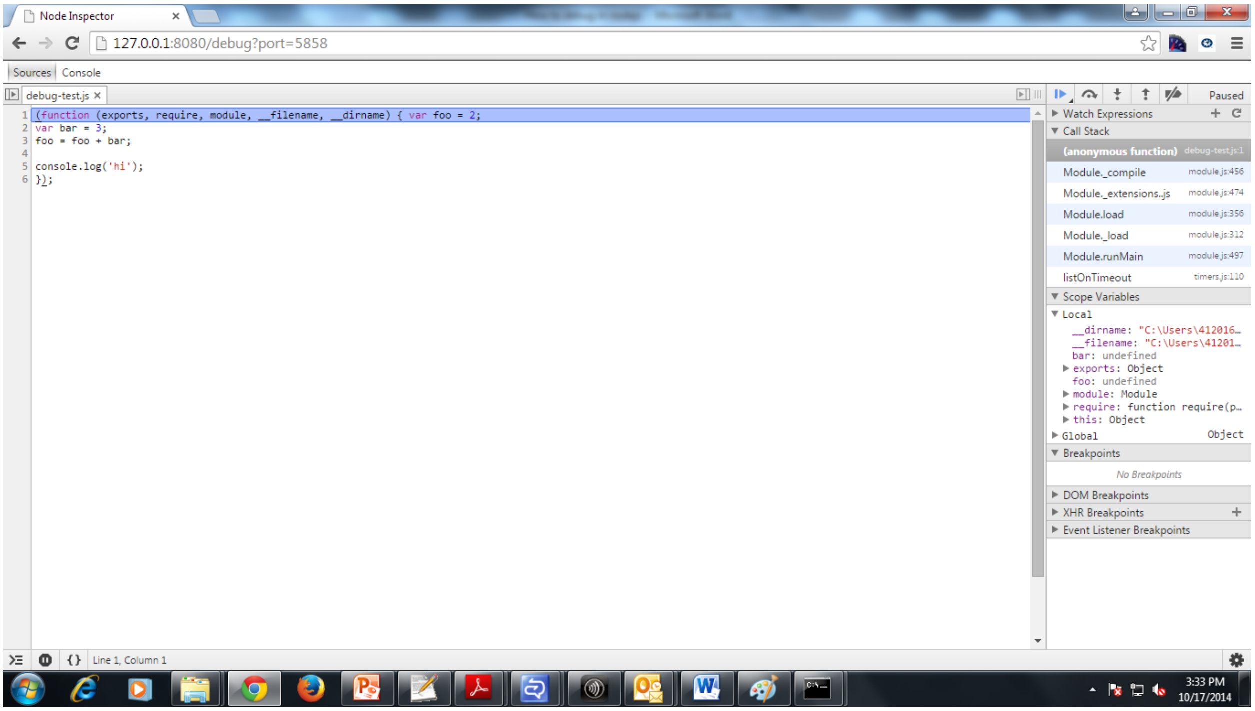
```
C:\Users\412016\Desktop\gaurav\nodejs>node --debug-brk debug-test.js
debugger listening on port 5858
```

Open another terminal and type



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\412016>cd C:\Users\412016\Desktop\gaurav\nodejs

C:\Users\412016\Desktop\gaurav\nodejs>node-inspector
Node Inspector v0.7.4
Visit http://127.0.0.1:8080/debug?port=5858 to start debugging.
_
```

Open this url in browser



```
(function (exports, require, module, __filename, __dirname) { var foo = 2;
var bar = 3;
foo = foo + bar;

console.log('hi');
});
```

(anonymous function)	debug-test.js:1
Module._compile	module.js:456
Module._extensions..js	module.js:474
Module.load	module.js:356
Module._load	module.js:312
Module.runMain	module.js:497
listOnTimeout	timers.js:110

```
Local
  __dirname: "C:\Users\412016…
  __filename: "C:\Users\41201…
  bar: undefined
  exports: Object
  foo: undefined
  module: Module
  require: function require(p…
  this: Object
Global                    Object
```