



Interactive

NORTH AMERICA

Austin, Texas
November 28–December 2, 2016



Quit writing everything yourself

Evan Lucas, Help.com

<https://github.com/evanlucas>

<https://twitter.com/evanhLucas>

About



- Node Engineer at Help.com
- Node.js Core Technical Committee (CTC) and release team member

Input Validation



- HTTP request, response, querystring param, and header validation
- Function argument validation
- Message format validation

How can we automate this?



- Validate an object's properties for different types
- Enforce how something is written (configuration vs code)



escodegen



“Escodegen ([escodegen](#)) is an [ECMAScript](#) (also popularly known as [JavaScript](#)) code generator from [Mozilla's Parser API AST](#).”

<https://github.com/estools/escodegen>

Abstract Syntax Tree (AST)

A screenshot of a web browser window displaying an Abstract Syntax Tree (AST) for the expression `1 + 4 + 5`. The browser interface includes a title bar with the URL `ast.evanlucas.com`, a toolbar with various icons, and a dark-themed code editor area. The code is presented in a monospaced font with line numbers on the left. The AST structure is represented as a JSON object.

```
1 4 + 5
1 | Node {
2 |   type: 'Program',
3 |   start: 0,
4 |   end: 5,
5 |   body:
6 |     [ Node {
7 |       type: 'ExpressionStatement',
8 |       start: 0,
9 |       end: 5,
10 |      expression:
11 |        Node {
12 |          type: 'BinaryExpression',
13 |          start: 0,
14 |          end: 5,
15 |          left: Node { type: 'Literal', start: 0, end: 1, value: 4, raw: '4' },
16 |          operator: '+',
17 |          right: Node { type: 'Literal', start: 4, end: 5, value: 5, raw: '5' } } ] ,
18 |      sourceType: 'script' }
```



How can we make this easier?



npm install @helpdotcom/build-ast

<https://github.com/helpdotcom/build-ast>

This is what we want



```
1 'use strict'  
2  
3 module.exports = function validate(opts, cb) {  
4   if (typeof opts.name !== 'string') {  
5     const er = new TypeError('name must be a string')  
6     return setImmediate(cb, er)  
7   }  
8  
9   setImmediate(cb, null, opts)  
10 }
```

@helpdotcom/build-ast



```
1 'use strict'  
2  
3 const B = require('@helpdotcom/build-ast')  
4 const ast = B.ast  
5  
6 const b = B().use('strict')
```

Example



```
8 // `typeof opts.name`  
9 const left = B.typeof('opts.name')  
10  
11 // `string`  
12 const right = B.string('string')  
13  
14 // `typeof opts.name !== 'string'`  
15 const check = B.notEquals(left, right)
```

Example



```
17 // new TypeError('name must be a string')  
18 const error = B.TypeError('name must be a string')  
19  
20 // if (typeof opts.name !== 'string') {  
21 //   const er = new TypeError('name must be a string')  
22 //   return setImmediate(cb, er)  
23 // }  
24 const errorBlock = B()  
25   .declare('const', 'er', error)  
26   .returns(  
27     B.callFunction('setImmediate', [  
28       B.id('cb')  
29       , B.id('er')  
30     ])  
31   ).build()
```

Example



```
32
33 const block = B()
34 block.if(check, B.block(errorBlock))
35 block.returns(
36   B.callFunction('setImmediate', [
37     B.id('cb')
38   , B.ast.literal(null)
39   , B.id('opts')
40   ])
41 )
42
43 b.module(B.function('validate', ['opts', 'cb'], block.build()))
```

Example



```
45 const toCode = require('escodegen').generate
46 console.log(toCode(b.program())))
```

Example - Output



```
1 'use strict'  
2  
3 module.exports = function validate(opts, cb) {  
4   if (typeof opts.name !== 'string') {  
5     const er = new TypeError('name must be a string')  
6     return setImmediate(cb, er)  
7   }  
8   return setImmediate(cb, null, opts)  
9 }
```

Thanks!



@evanhLucas



evanlucas