

Use Babel AST in ESLINT

Programming Experience 2019
Software Architecture Group

Ulrike Bath, Maximilian Götz
06.02.2019

Our Task

```

29     lively.lastScrollLeft = document.scrollingElement.scrollLeft;
30
31     ! 'charCode' is assigned a value but never used.
32     const { char, ctrl, shiftKey, altKey, keyCode, charCode } = keyInfo(evt);
33
34     const keyHandlers = [

```

Problem: ESLint uses its own Parser, which does not know about Babel Syntax/Language extensions

Goal: Make it compatible with our “JavaScript”

```

25     lively.addEventListener("Graffle", document, "selectionchange",
26         (evt) => { this.onSelectionHide(evt) } )
27     Parsing error: Unexpected token : ", document, "selectionchange",
28     ((evt) => { this.onSelectionChange(evt) } )::debounce(600) )
29     this.keysDown = {}
30 }

```

Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart 2

Our Result

=> It works!

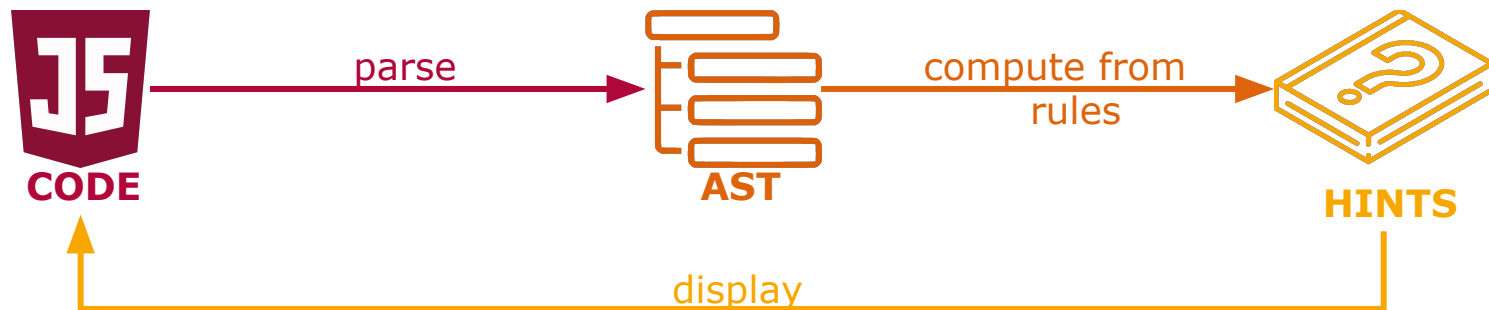
```
26      (evt) => { this.onSelectionHide(evt) } )  
27      lively.addListener("Graffle", document, "selectionchange",  
28      ((evt) => { this.onSelectionChange(evt) })::debounce(600) )  
29      this.keysDown = {}  
30  }
```

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **3**

How does ESLint work?

- supports current standards of ECMAScript



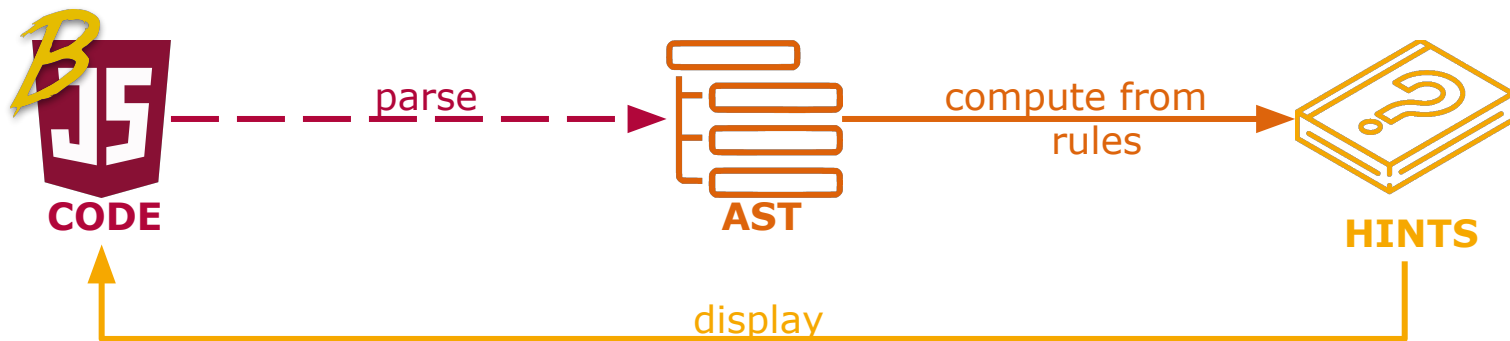
Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart 4

The Problem

Babel code cannot be parsed correctly

- Javascript* \neq Javascript
- experimental features like binding operator

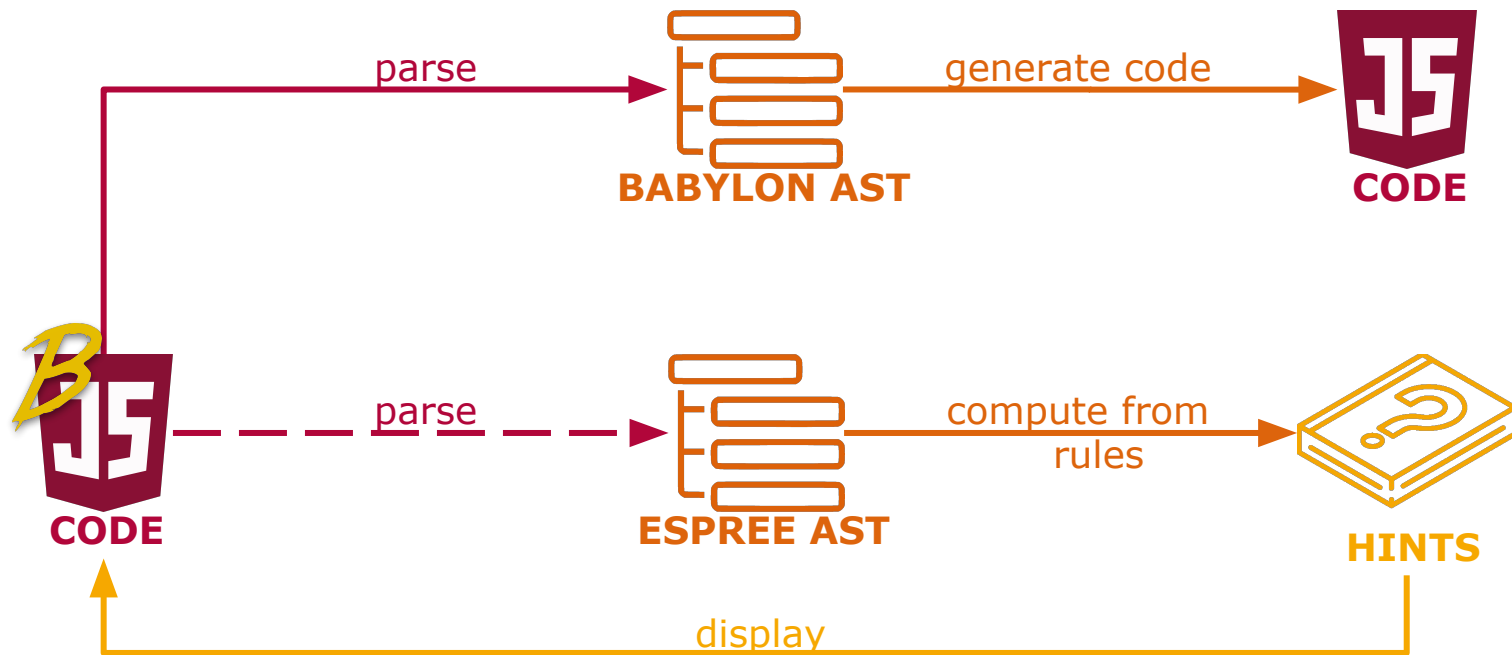


**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **5**

How does Babel work?

- Babel compiles code down to ECMAScript

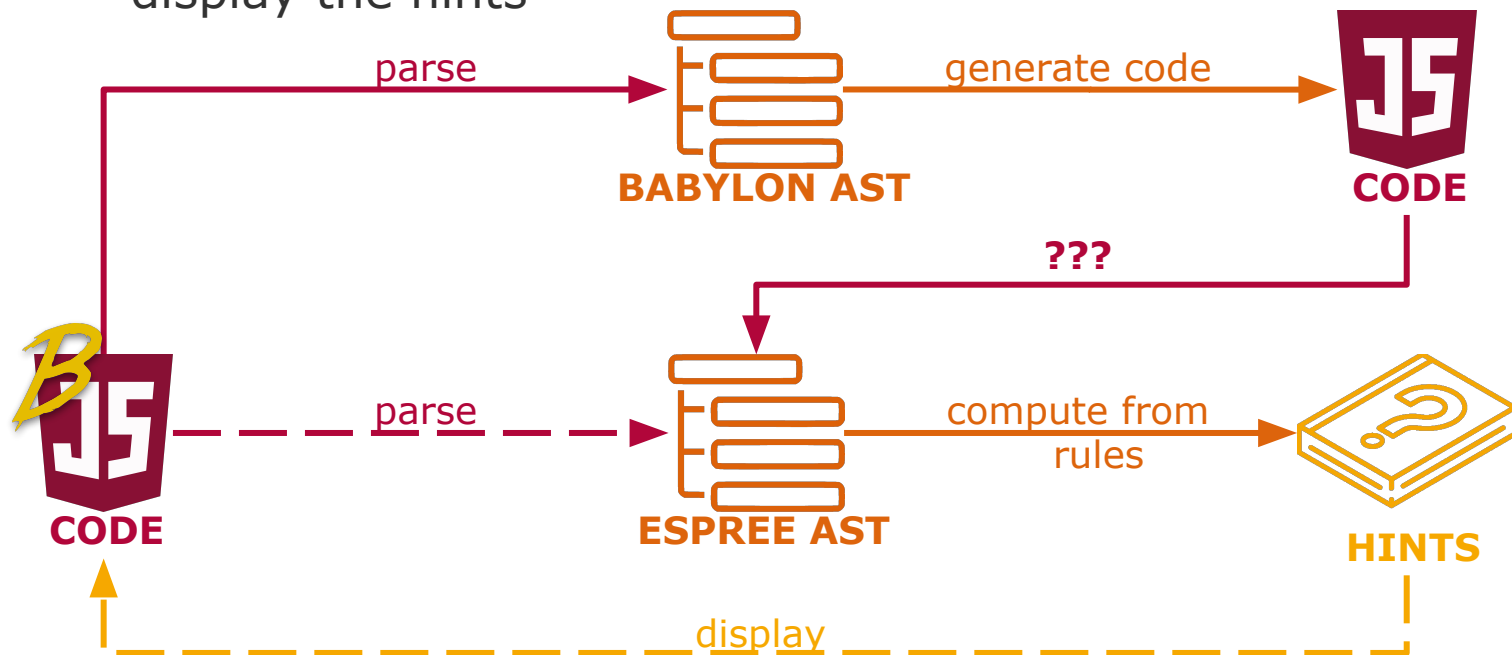


Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart 6

Naive Approach

- But we would lose the line information necessary to display the hints

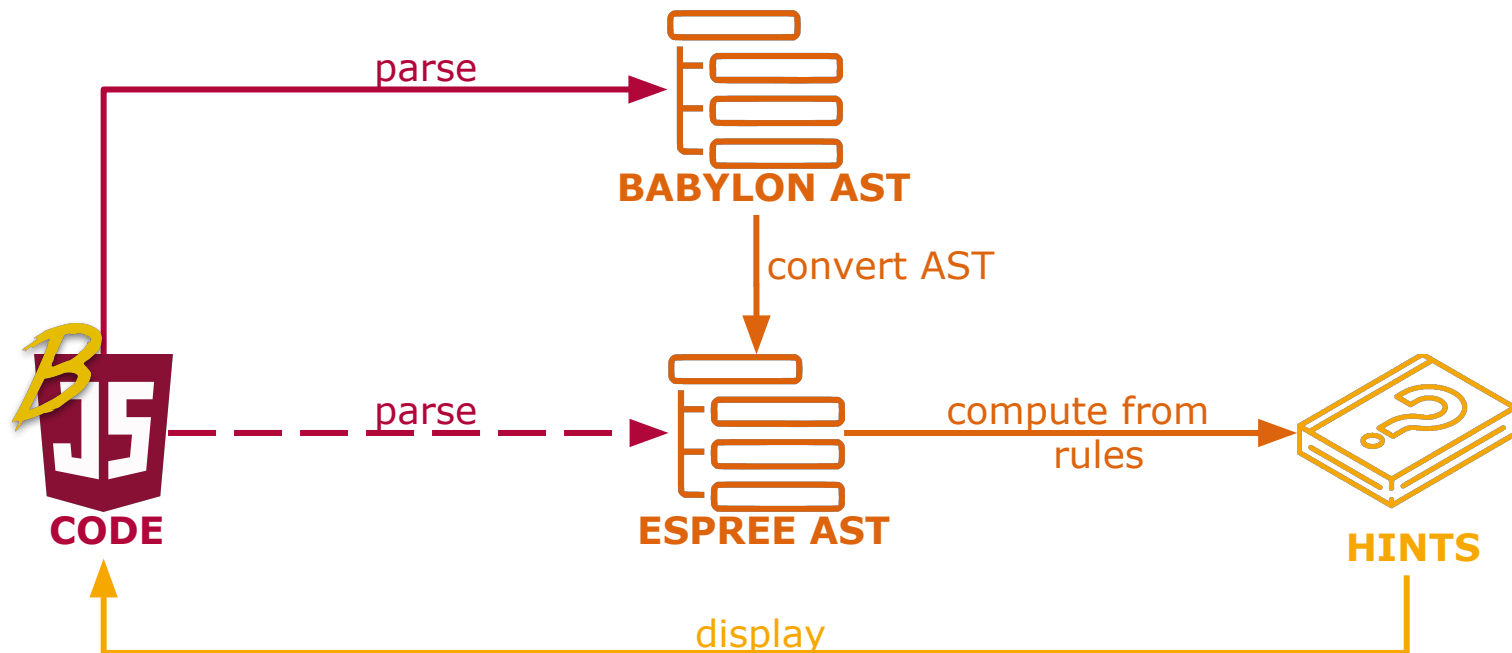


Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart 7

Our Approach

- But what is the difference between these ASTs?



Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **8**

Different AST Structures

ESLint: Espree AST

```
Program {  
  type: "Program"  
  start: 0  
  end: 476  
  + range: [2 elements]  
  - body: [  
    + VariableDeclaration {type, start, end, range, declarations,  
    + FunctionDeclaration {type, start, end, range, id, ... +5}  
  ]  
  sourceType: "module"  
}
```

Babel: Babylon AST

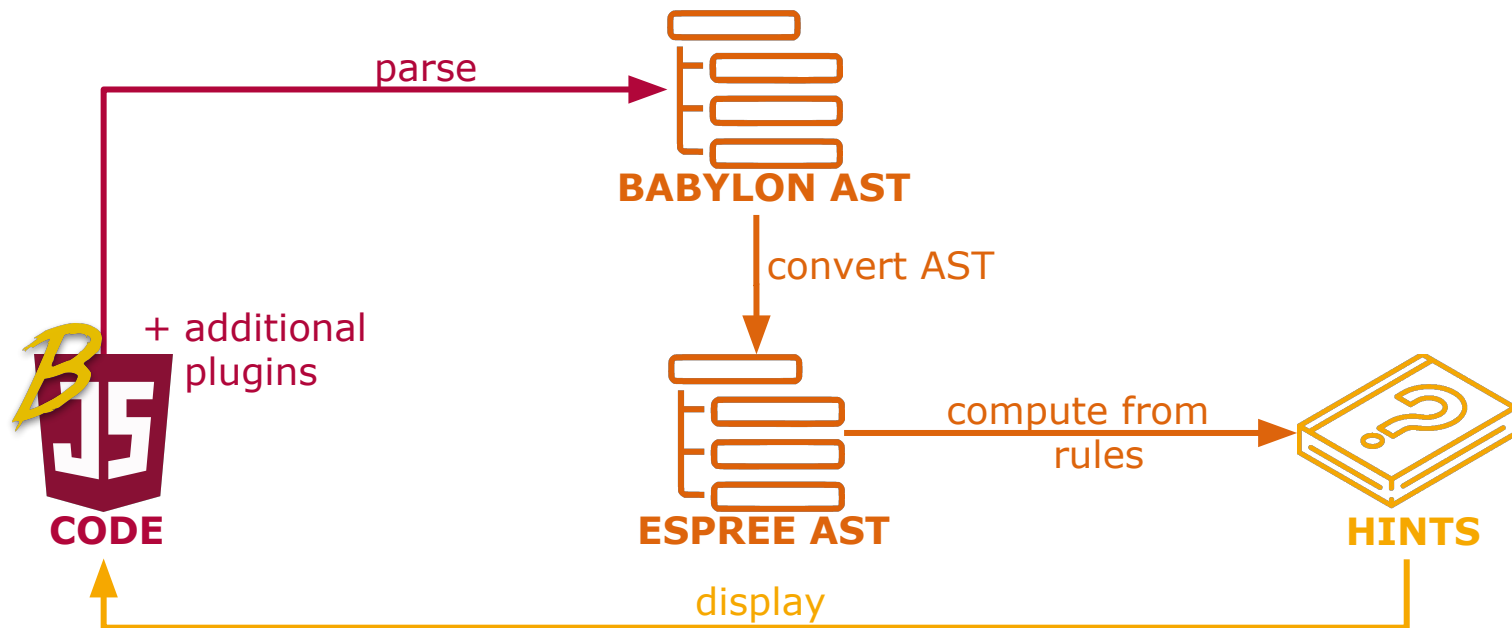
```
File {  
  type: "File"  
  start: 0  
  end: 476  
  + loc: {start, end}  
  - program: Program {  
    type: "Program"  
    start: 0  
    end: 476  
    + loc: {start, end}  
    sourceType: "module"  
    - body: [  
      + VariableDeclaration {type, start, end, loc, declarations, ... +2}  
      + FunctionDeclaration {type, start, end, loc, id, ... +5}
```

Our Idea: Get ESLint to use the babel parser

1. Parse code in
Lively4

2. Convert AST

3. Customize parser
in ESLint



**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **10**

Challenges

1. Browserify our converter & import it in Lively4
2. Unsupported Babel version in Lively4
3. How to hand over our custom parser to ESLint

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **11**

Challenges

Browserify and import our converter

- browserify allows us to use *require* in the browser
- we can load the converter as a single bundle
- dependency concerns are all taken care of

```

1 // data.js
2 module.exports = {
3   firstName: 'Jon',
4   familyName: 'Snow'
5 }
6
1 // user.js
2 var data = require('./data');
3 module.exports = {
4   getName : function() {
5     return data.firstName + ' ' + data.familyName;
6   }
7 }

```



**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **12**



Browserify and import our converter

```
browserify babelParser.js  
-o build/babelParser_bundle.js  
--standalone babelParser
```

```
7  
8 import "src/external/eslint/babelParser_bundle.js";  
9
```

Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **13**

Challenges

Unsupported Babel version

- current Babel version in Lively4 is not supported in babelToEsprez-converter
- additional attribute "range" cannot be added in parser options

```

start: 0
end: 476
+ range: [2 elements]
- body: [
  + VariableDeclaration {type, start, end, range, loc}
  + FunctionDeclaration {type, start, end, range, loc}
]
```

```

start: 0
end: 476
+ loc: {start, end}
- program: Program {
  type: "Program"
  start: 0
  end: 476
  + loc: {start, end}
  sourceType: "module"
  - body: [
    + VariableDeclaration {type, start, end, loc, range}
    + FunctionDeclaration {type, start, end, loc, range}
  ]
}
```

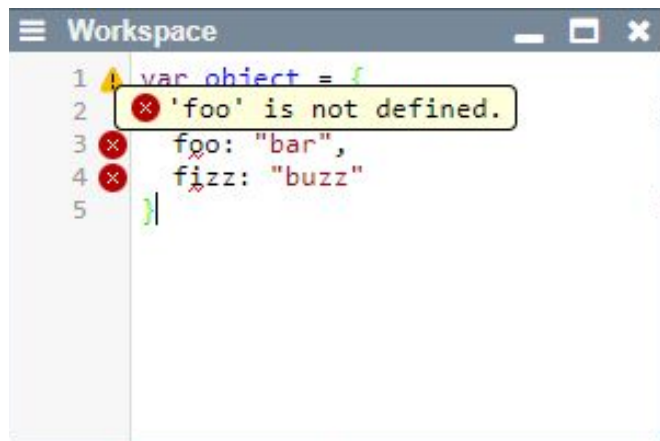
Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **14**

Challenges

Unsupported Babel version

- ESLint finally worked! But it finds false positives for object declaration
- => still differences in the AST, that are harder to fix



```

1  var object = {
2    'foo' is not defined.
3    foo: "bar",
4    fizz: "buzz"
5  }
  
```

```

Property {
  type: "Property"
  - key: Identifier = $node {
    type: "Identifier"
    name: "foo"
  }
  - value: Literal {
    type: "Literal"
    value: "bar"
  }
  kind: "init"
}
  
```

Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **15**

Challenges

How to hand over custom parser in ESLint

- Documentation only works for modules (npm)
- ESLint offers Node.js API documentation
- Manipulating ESLint-bundle to hand in a custom parser at run time

```
{  
  "parser": "./path/to/awesome-custom-parser.js"  
}
```

```
const Linter = require("eslint").Linter;  
const linter = new Linter();  
  
linter.defineParser("my-custom-parser", {  
  parse(code, options) {  
    // ...  
  }  
});
```

Use Babel AST in ESLint

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **16**

Evaluation

Is it noticeably slower?

- log current milliseconds before and after parsing:
`window.performance.now()`
- compare time differences between default and custom parser

No noticeable difference in time performance.

Longer loading time when opening Lively4.

	Average Validation Time	Average Module Loading time
Eslint Default Parser	12.9 ms	4 ms
Our Custom Parser	23.7 ms	8210 ms
Difference in seconds	0,0108 s	8,206 s

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **17**

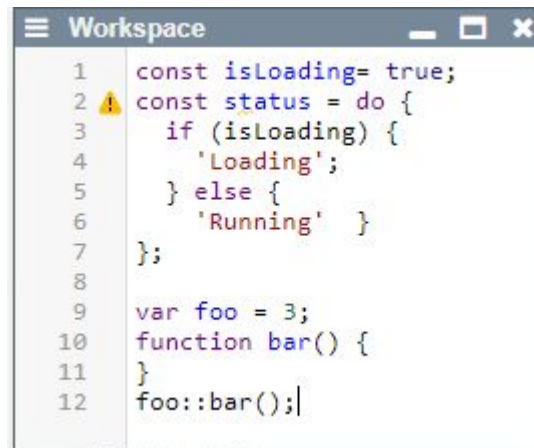
Evaluation

Does it not find false syntax errors because of not building the AST?



```

1  Parsing error: Unexpected token do
2  const status = do {
3    if (isLoading) {
4      'Loading';
5    } else {
6      'Running' }
7  };
8
9  var foo = 3;
10 function bar() {
11 }
12 foo::bar();
  
```

```

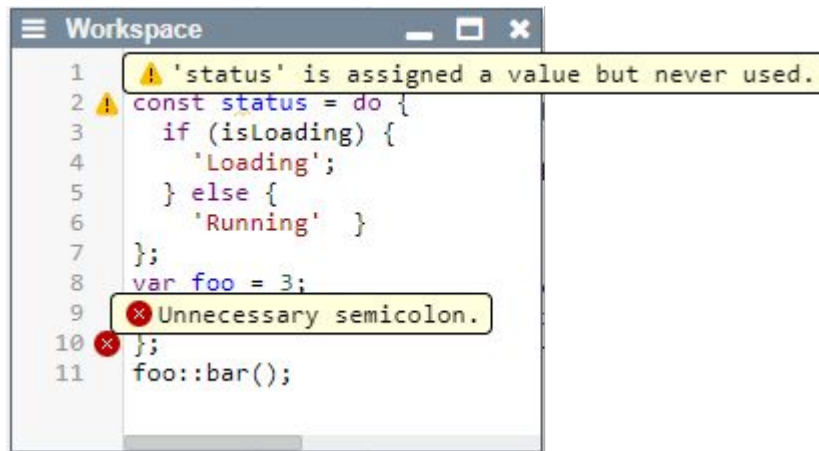
1  const isLoading= true;
2  const status = do {
3    if (isLoading) {
4      'Loading';
5    } else {
6      'Running' }
7  };
8
9  var foo = 3;
10 function bar() {
11 }
12 foo::bar();
  
```

Yes, the code is parsed correctly.

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **18**

Does it still works correctly?



The screenshot shows a code editor window titled 'Workspace' with the following JavaScript code:

```
1
2
3
4
5
6
7
8
9
10
11
```

```
const status = do {
  if (isLoading) {
    'Loading';
  } else {
    'Running' }
};
var foo = 3;
};
foo::bar();
```

Two ESLint error messages are displayed:

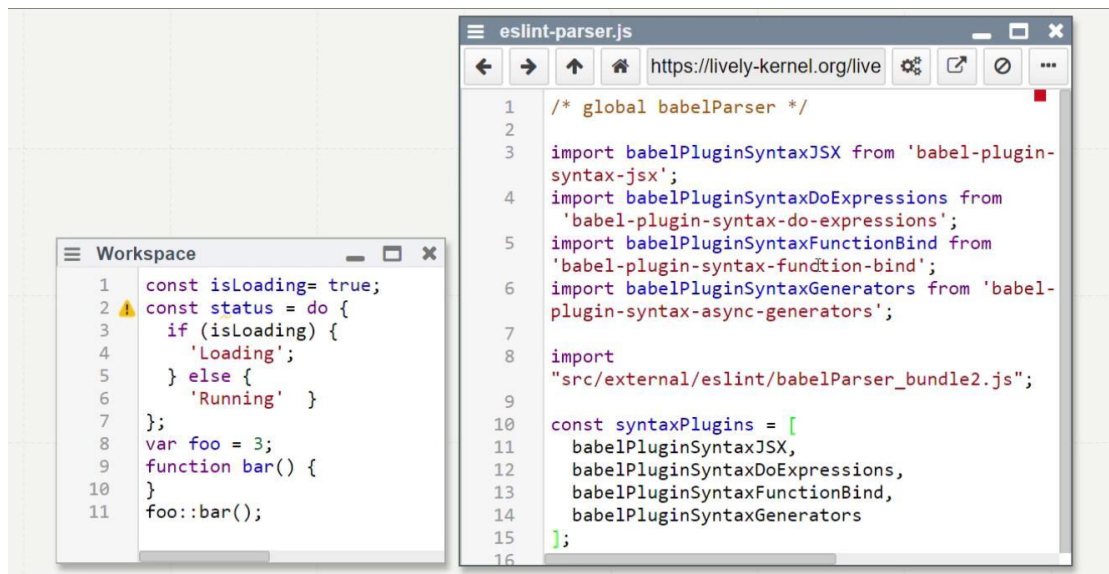
- Line 2: **'status' is assigned a value but never used.** (Warning icon)
- Line 10: **Unnecessary semicolon.** (Error icon)

Yes, it does!

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **19**

Is the parser customizable with the plugins?

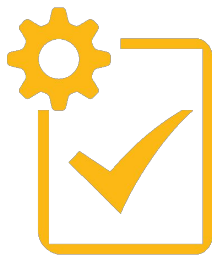


Yes, it is.

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart 20

Next Steps



Tests



Refactoring



Documentation



Performance
Analysis

**Use Babel AST in
ESLint**

Ulrike Bath,
Maximilian Goetz,
06.02.2019
Chart **21**

Thank you
for your attention!

Ulrike Bath, Maximilian Goetz
Use Babel AST in ESLint
Software Architecture Group