

MODULES, PACKAGES

AND CANDY *oh my!*

Hi I'm Francis



RECONBOT

- ▶ **Set the stage**
- ▶ **Explore the landscape**
- ▶ **What can you do to get free candy?**

THE STAGE

JOB INTERVIEW

I had a job interview and I've spent the last two years not really paying attention to new things.

WebPack is Cool!
Is Browserify still cool?

ES Modules are Cool!
Is Common JS still cool?

ES6 *itself* is *really* cool

ES2015 ~~ES6~~ itself is *really* cool

Babel

BABEL

DAABEL

BABEL

ES6 and ES7 proposals and specs
into
ES5

BABEL

~~ES6 and ES7~~ ECMA 2015-Next proposals and specs
into
ES5 (ES4 with polyfills)

BROWSER PACKAGING

commonJS with es6 -> browserify -> babelify
-> single file bundle of es5

NODE PACKAGING

commonJS with es6 -> browserify(--node) ->
babelify -> single file bundle of es5

NODE PACKAGING

commonJS with es6 -> babel -> many es5
files

This works for me for a while but I have a job interview.

**What are es6 modules?
Why should we use them?**

Exporting One thing

```
// CommonJS  
module.exports = 'a thing';  
// ES Modules  
export default 'a thing';
```

```
//
```

node-love is a good example

```
// CommonJS
var heart = require("node-love");
// '♥'

// ES Modules
import heart from "node-love";
// '♥'

//
```

Exporting Many Things

```
// CommonJS
module.exports = {
  oneThing: 'a thing',
  anotherThing: () => { console.log('a thing') },
  lastThing: 4
};
```

```
require('foo').oneThing; // 'a thing'
```

```
//
```


Don't do this

```
// ES Modules (equivalent to CommonJS export)
export default {
  oneThing: 'a thing',
  anotherThing: () => { console.log('a thing') },
  lastThing: 4
};

import foo from 'foo';
foo.oneThing // 'a thing'

//
```

Exporting Many Things

```
// ES Modules
export var oneThing = 'a thing';
export function anotherThing(){ console.log('a thing') };

var lastThing = 4;
export { lastThing };

import { oneThing } from 'foo'; // 'a thing'
```

```
//
```

Exporting Many Things

```
// ES Modules
export var oneThing = 'a thing';
export function anotherThing(){ console.log('a thing') };

var bar = 4;
export { bar as lastThing };

import { oneThing } from 'foo'; // 'a thing'
```

```
//
```

ES Modules Can Do Both!?

```
// foo.js  
export default 'bar';  
export var lastThing = '4';
```

```
// bar.js  
import { lastThing } from 'foo'; // 4  
import foo from 'foo'; // 'bar'  
import foo, { lastThing } from 'foo'; // bar and 4!
```

```
//
```

ES Modules Can Do Both

```
// underscore.js
export function flatten(array) { /* flatten the array */ };
export function zip(array) { /* zip the array? */ };
export default {
  flatten: flatten,
  zip: zip
};
```

```
// bar.js
import { flatten } from 'underscore';
import _ from 'underscore';
```

```
//
```

ES Modules Can Do Both

```
// underscore.js
export function flatten(array) { /* flatten the array */ };
export function zip(array) { /* zip the array? */ };
export default {
  flatten,
  zip
};
```

```
// bar.js
import { flatten } from 'underscore';
import _ from 'underscore';
```

```
//
```

MODULES ARE COOL AND ALL BUT WHO CARES?

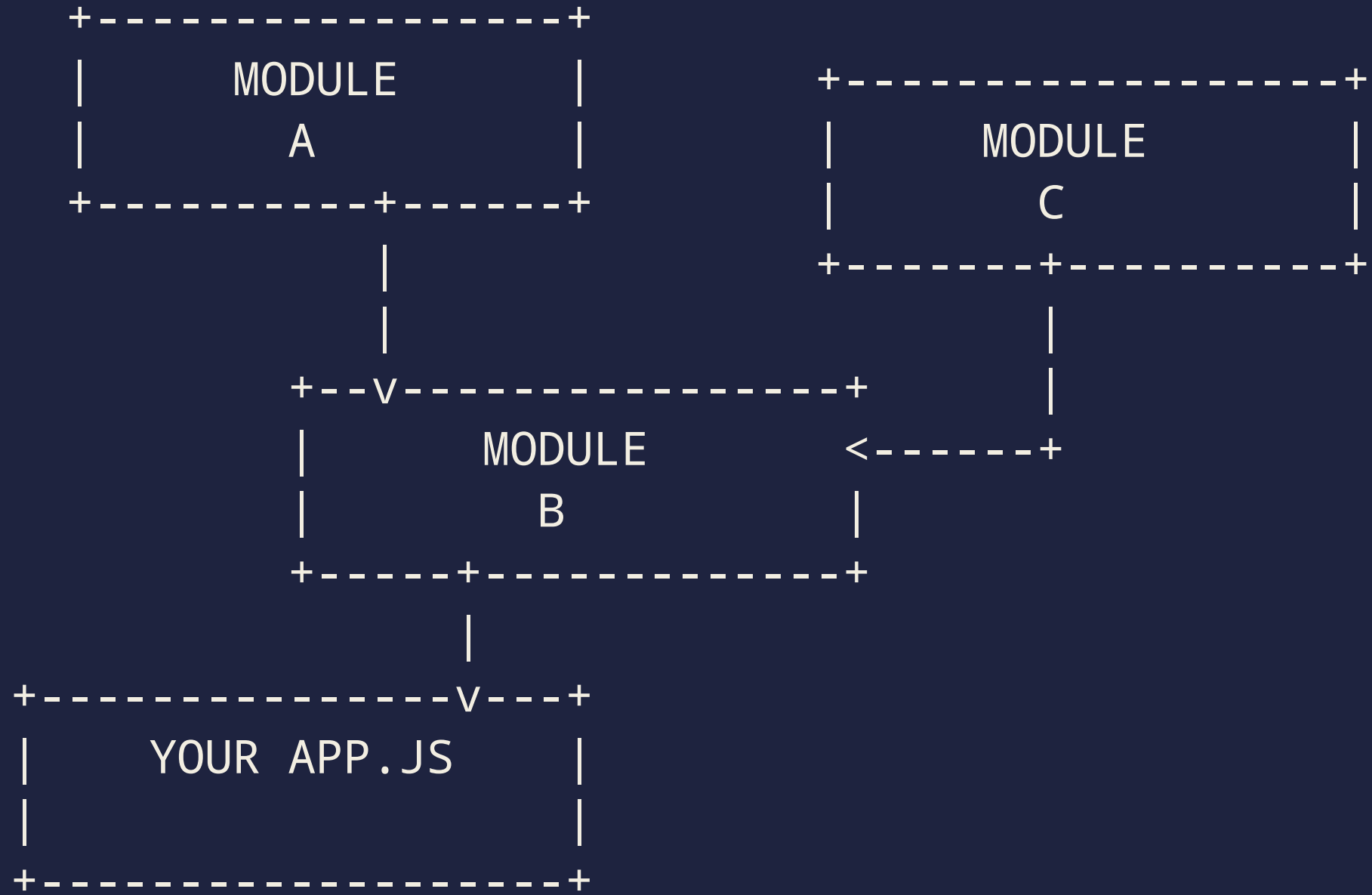
WE DO

Note CommonJS and ECMAScript 6 are only roughly similar. The latter has a flat structure, whereas the former is nested. Which style you prefer is a matter of taste, but the flat style has the advantage of being statically analyzable.

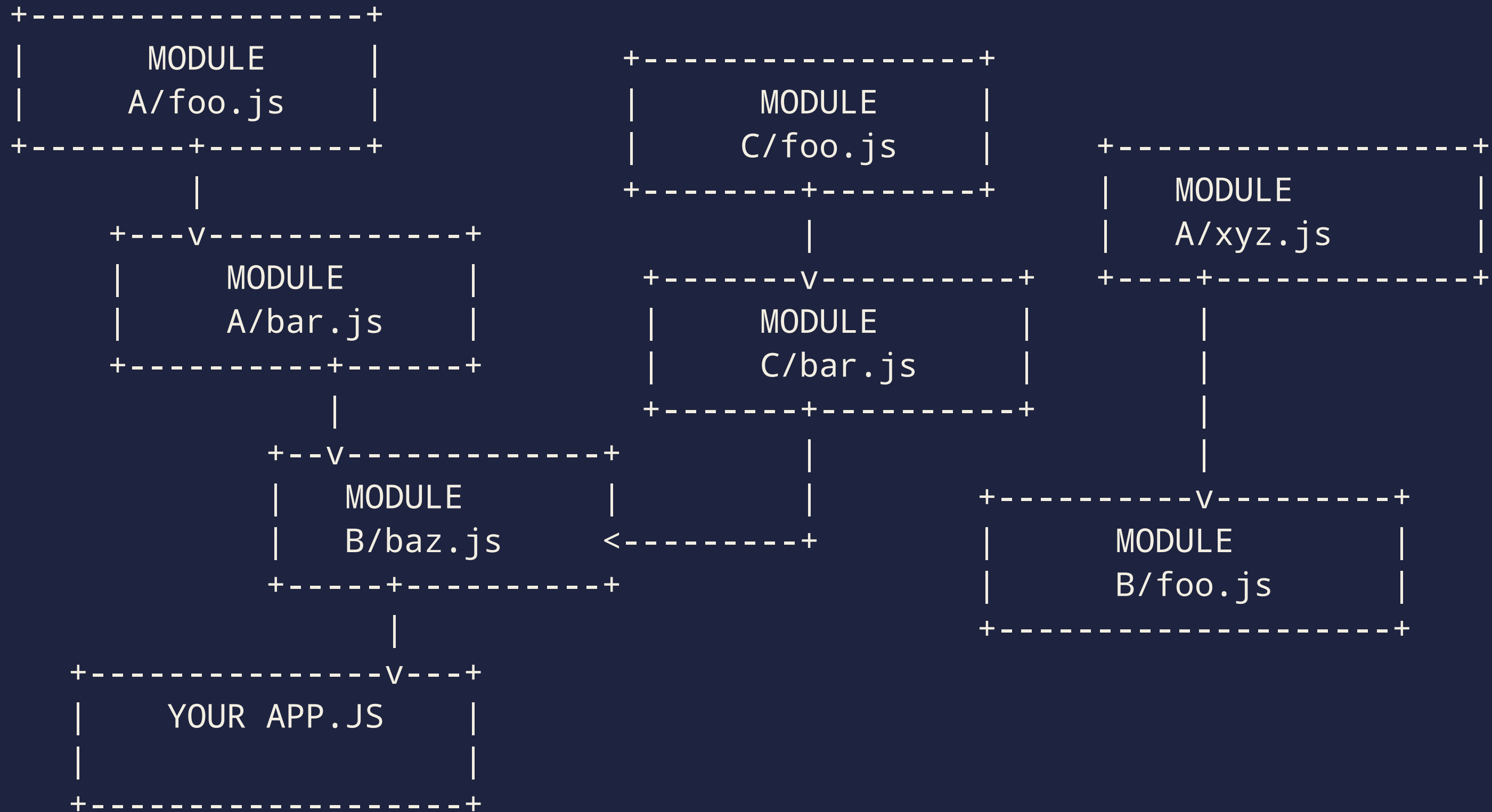
-- Dr. Axel Rauschmayer

STATICALLY ANALYZABLE

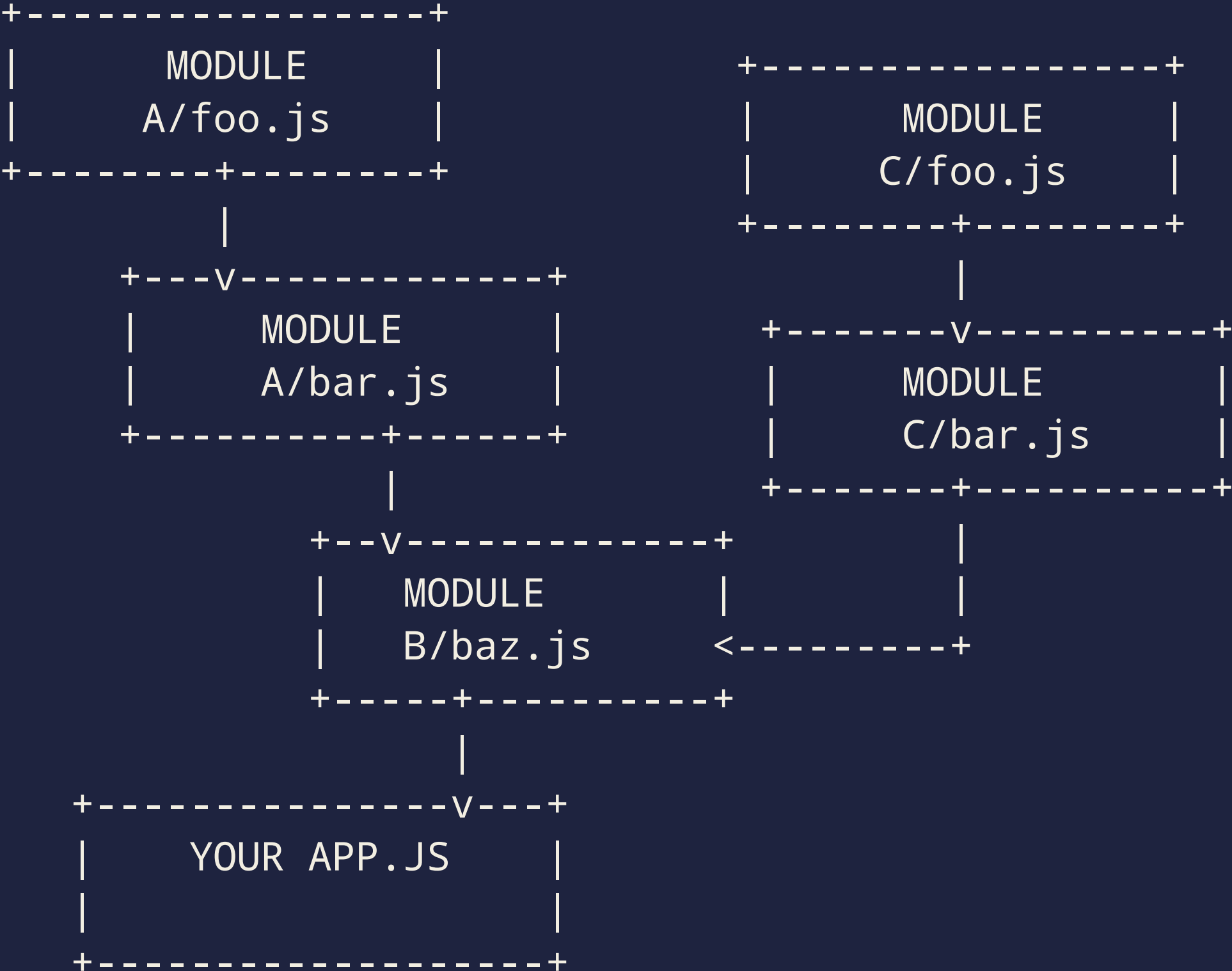
CommonJS Module Import Mapping



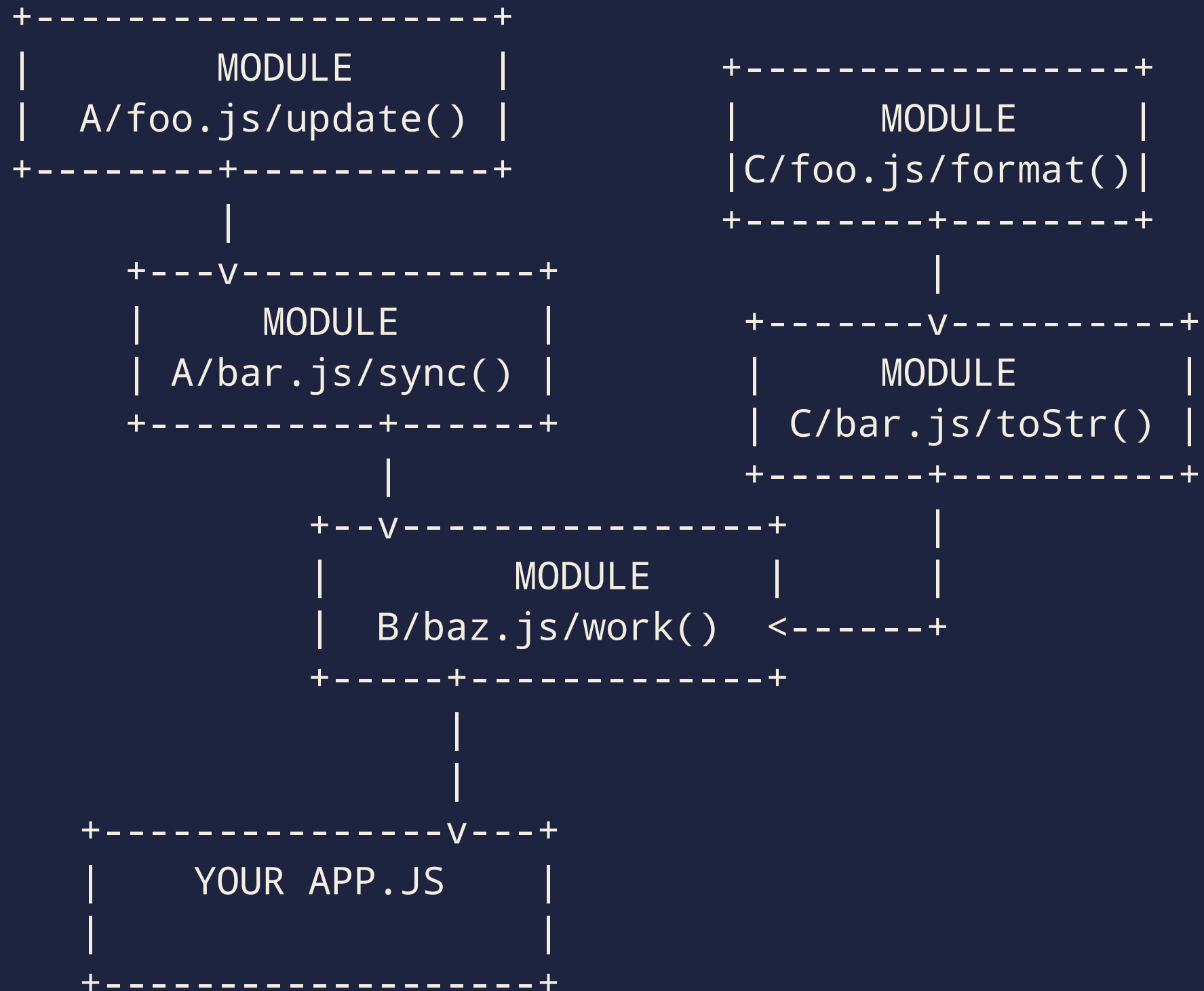
File Based Import Mapping



File Based Import Mapping



Named Export Based Import Mapping



ENTER ROLLUP

ROLLUP THE *TreeShaker*

ROLLUP THE *TreeShaker*

ES Modules + ES Module deps -> rollup/babel
-> "Shaken" CommonJS bundle

"Shaken" CommonJS bundle + CommonJS deps -> browserify
-> ES5 Bundle

THE LANDSCAPE






POUCHDB





The Database that Syncs!

Home Notifications Moments Messages Search Twitter

 Nolan Lawson @nolanlawson  Following

Browserify sadness: the more I modularize my code, the bigger it gets. 😞
github.com/pouchdb/pouchd...

 (#4454) - remove more stuff from PouchDB.utils by nolanla...
This removes nearly all the unnecessary stuff from utils.js. The remaining stuff is either dangerous to remove because of widespread (mis-)use (myself included here), or because it's in...
github.com

LIKES 3 

12:03 AM - 11 Oct 2015

Reply Retweet Like More

<https://twitter.com/nolanlawson/status/653058308643921920>



Nolan Lawson @nolanlawson · Oct 11

Browserify sadness: the more I modularize my code, the bigger it gets. 😞
[github.com/pouchdb/pouchd...](https://github.com/pouchdb/pouchdb/pouchdb)



[View summary](#)



Nolan Lawson @nolanlawson · Oct 11

Why is a monster file with 20 exports smaller than 20 files that you require() individually? Uglify even squashes the variable names...



Francis Gulotta @reconbot · Oct 11

@nolanlawson boiler plate require stuff



Francis Gulotta @reconbot · Oct 11

@nolanlawson it's part of what @Rich_Harris was talking about with
github.com/rollup/rollup



[View summary](#)





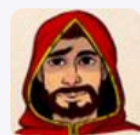
[HTTPS://GITHUB.COM/RICH-HARRIS/BROOKLYN-JS-ROLLUP-TALK](https://github.com/rich-harris/brooklyn-js-rollup-talk)



- ▶ **Green is live code**
- ▶ **Orange is boiler plate**
- ▶ **Purple is dead code**

[HTTPS://GITHUB.COM/NOLANLAWSON/ROLLUP-COMPARISON](https://github.com/nolanlawson/rollup-comparison)

Shows lack of overhead when requiring modules



Francis Gulotta @reconbot · Oct 11

@nolanlawson @Rich_Harris npm still didn't have a blessed way of mixing in es6 modules even as an alt source which makes me sad



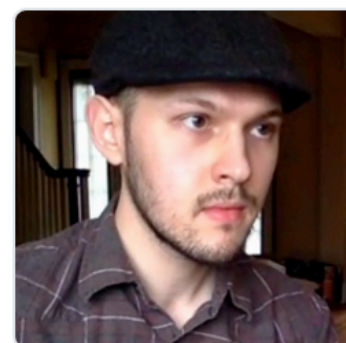
Nolan Lawson

@nolanlawson



Following

@reconbot @Rich_Harris Well, color me impressed: [github.com/nolanlawson/ro....](https://github.com/nolanlawson/rollup-comparison)
The rollup build is ~1/4 the size of either browserify or webpack.



nolanlawson/rollup-comparison

compare rollup to browserify/webpack. Contribute to rollup-comparison development by creating an account on GitHub.

github.com

RETWEETS

9

LIKES

31



1:52 AM - 11 Oct 2015



Reply to @nolanlawson @Rich_Harris

1,000,000 TWEETS LATER WE HAD A PR

<https://github.com/pouchdb/pouchdb/pull/4462>



<code>pouchdb-browserify.js</code>	380KB
<code>pouchdb-rollup.js</code>	364KB

%5 savings!



<code>pouchdb-browserify.min.js</code>	148KB
<code>pouchdb-rollup.min.js</code>	160KB

%8 loss!?

ROLLUP INLINES ALL FUNCTIONS

 **POUCHDB HAS `eval()` IN A FEW PLACES**

UGLIFY PROTECTS YOU FROM BREAKING THINGS WITH `eval()`



kzc commented 22 days ago

Your code calls `eval()` in certain scopes which disables mangling. Both `eval()` and `with` can have adverse effects when functions or variables are renamed. This problem is more apparent with Rollup as opposed to Browserify because Rollup evidently promotes all functions to the same scope.

If you use the following flags to acknowledge the use of `eval()` in your javascript you'll likely get the mangling results you are looking for:

```
uglifyjs -c -n 'eval=true'
```

but be warned that your `eval()`s may not necessarily work with mangled symbol names.

Admittedly it would be useful if `uglifyjs` would warn of mangling being disabled if `eval()` or `with` are encountered. Please feel free to open a ticket requesting that feature.

See "Mangler options" in the UglifyJS2 docs: <https://github.com/mishoo/UglifyJS2>



Rich-Harris commented 22 days ago

Ah-ha! Thanks, that's very interesting and helpful to know.

MOVED EVAL TO EXTERNAL MODULES

pouchdb-browserify.min.js	148KB
pouchdb-rollup.min.js	160KB
pouchdb-ext-rollup.min.js	144KB

4KB or 3% savings

rioted




How much space was in dead code?

How much space was in dead code in CommonJS deps?

How much space was in browserify overhead?

How much space was lost to `{objects}` in default exports?

#TheLEGOMovie



I have no idea what's going on... at all.

Goal is smaller bundle for libraries

Goal is smaller total bundle for apps

```
var PouchDB = require('./setup');
module.exports = PouchDB;

PouchDB.ajax = require('./deps/ajax/prequest');
PouchDB.utils = require('./utils');
PouchDB.Errors = require('./deps/errors');
PouchDB.replicate = require('./replicate').replicate;
PouchDB.sync = require('./sync');
PouchDB.version = require('./version');
var httpAdapter = require('./adapters/http');
PouchDB.adapter('http', httpAdapter);
PouchDB.adapter('https', httpAdapter);

PouchDB.plugin(require('./mapreduce'));

var adapters = require('./adapters');

Object.keys(adapters).forEach(function (adapterName) {
  PouchDB.adapter(adapterName, adapters[adapterName], true);
});
```

POUCHDB FEATURES (INCOMPLETE)

Multiple Storage engines (LevelDB, IndexedDB, WebSQL)

Database Replication with full support for 7 different servers!

Support for Views with CouchDB style map/reduce

```
import PouchDB from './setup';
export default PouchDB;

export ajax from './deps/ajax/prequest';
export utils from './utils';
export Errors from './deps/errors';
export sync from './sync';
export version from './version';
export { replicate } from './replicate';

import httpAdapter from './adapters/http'
export var http = PouchDB.adapter('http', httpAdapter);
export var https = PouchDB.adapter('http', httpAdapter);

import mapreduce from './mapreduce'
PouchDB.plugin(mapreduce);

import adapters from './adapters';

Object.keys(adapters).forEach(function (adapterName) {
  PouchDB.adapter(adapterName, adapters[adapterName], true);
});
```

```
// sweet-app.js
var PouchDb = require('pouchdb');
var db = new PouchDB('dbname');
db.put({
  _id: 'dave@gmail.com',
  name: 'David',
  age: 68
});
db.allDocs().then(console.log);
```

```
// sweeter-app.js
import PouchDb from 'pouchdb';
var db = new PouchDB('dbname');
db.put({
  _id: 'dave@gmail.com',
  name: 'David',
  age: 68
});
db.allDocs().then(console.log);
```

TOOLS KEEP EVOLVING

Babel 6!

Rollup now has a babel plugin!

`esnext:main` is gaining traction



Dan Abramov

@dan_abramov



Follow

Webpack 2 will include experimental support for dead code elimination based on ES6 imports: [github.com/webpack/webpac...](https://github.com/webpack/webpack)

RETWEETS

150

LIKES

182



7:09 PM - 21 Oct 2015



Reply to @dan_abramov



Dan Abramov

@dan_abramov



Follow

Webpack 2 will include experimental support for dead code elimination based on ES6 imports: [github.com/webpack/webpac...](https://github.com/webpack/webpack)

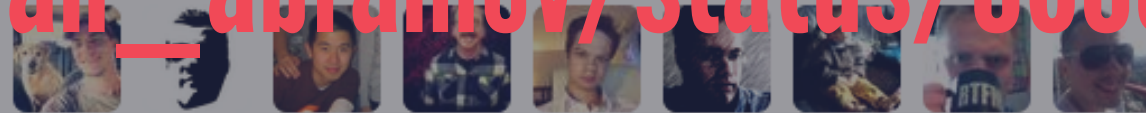
https://twitter.com/dan_abramov/status/656970508005736448

RETWEETS

150

LIKES

182



7:09 PM - 21 Oct 2015



Reply to @dan_abramov



Francis Gulotta

@reconbot

@npmjs what is npm? how closely does it care about commonjs? (re es6 modules)

11:20 AM - 11 Oct 2015



Reply to @npmjs



Forrest L Norvell @othiym23 · Oct 11

@reconbot @npmjs npm cares a great deal about there being a finished module loader spec and at least one browser implementation of it ;)



Francis Gulotta @reconbot · Oct 11

@othiym23 @npmjs I mean to say how closely is npm tied to commonjs. Were having a discussion about @Rich_Harris using "es6:main" for es6mods



Forrest L Norvell @othiym23 · Oct 11

@reconbot @Rich_Harris the basic philosophy of @npmjs is to put whatever you want into package.json, preferably with a . or a : in there



1



Francis Gulotta @reconbot · Oct 11

@othiym23 @Rich_Harris convention is good but I imagine es6 modules will become a mainstream npm concern eventually



IBM



Francis Gulotta

@reconbot

@npmjs what is npm? how closely does it care about commonjs? (re es6 modules)

11:20 AM - 11 Oct 2015

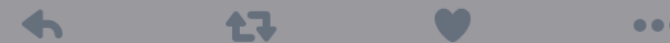


Reply to @npmjs



Forrest L Norvell @othiym23 · Oct 11

@reconbot @npmjs npm cares a great deal about there being a finished module loader spec and at least one browser implementation of it ;)



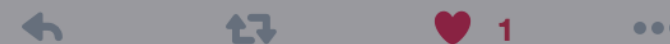
Francis Gulotta @reconbot · Oct 11

@othiym23 @npmjs I mean to say how closely is npm tied to commonjs. Were having a discussion about @Rich_Harris using "es6:main" for es6mods



Forrest L Norvell @othiym23 · Oct 11

@reconbot @Rich_Harris the basic philosophy of @npmjs is to put whatever you want into package.json, preferably with a . or a : in there



Francis Gulotta @reconbot · Oct 11

@othiym23 @Rich_Harris convention is good but I imagine es6 modules will become a mainstream npm concern eventually



<https://twitter.com/reconbot/status/653228811455209472>

IBM

Can I use

es modules





?



Settings

0 results found.

Why not [suggest this feature be added?](#)

 = Supported  = Not supported  = Partial support  = Support unknown

**Global usage share statistics based on data from [StatCounter GlobalStats](#) for October, 2015. See the [browser usage table](#) for usage by browser version.*

By [@Fyrd](#), design by [@Lensco](#). Location detection provided by [ipinfo.io](#).

Trouble with table colors/contrast? Try enabling [accessible colors](#).

Add ES6 modules #1869

New Issue

Open prettydiff opened this issue on Aug 31 · 2 comments



prettydiff commented on Aug 31

Please add ES6 modules. I am sure support is likely red across the board, but it would be handy to spell that out as a major missing feature is the browsers.

- <http://www.2ality.com/2014/09/es6-modules-final.html>
- <https://medium.com/ecmascript-2015/es6-modules-d5646d1b9a08>

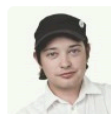


cvrebert commented on Aug 31

X-Ref: [kangax/compat-table#316](#)

Seems like they haven't been fully shipped in any browser:

- <http://bugzil.la/568953>
- <http://wkbug.com/147340>
- <https://code.google.com/p/v8/issues/detail?id=1569>
- <https://dev.modern.ie/platform/status/moduleses6/>



dstorey commented on Aug 31

Modules are "medium" on our (Edge's) priority list. So, we've not started development, but development is likely some time in future.

Fyrd added the **Support data suggestion** label on Sep 1

Labels

Support data sugg...

Milestone

No milestone

Assignee

No one assigned

Notifications

Subscribe

You're not receiving notifications from this thread.

4 participants





Rich Harris ✓

@Rich_Harris



Following

ES2015 modules are magic – if you're still writing CommonJS, you're just creating technical debt

Nolan Lawson @nolanlawson

@reconbot @Rich_Harris Well, color me impressed: [github.com/nolanlawson/ro...](https://github.com/nolanlawson/rollup)
The rollup build is ~1/4 the size of either browserify or webpack.

RETWEETS

24

LIKES

40



8:34 AM - 11 Oct 2015



Reply to @Rich_Harris



Rich Harris ✓

@Rich_Harris



Following

ES2015 modules are magic – if you're still writing CommonJS, you're just creating technical debt

Nolan Lawson @nolanlawson

@Rich_Harris Well, don't be impressed. [github.com/nolanlawson/ro...](#)
The Rollup build is 1/4 the size of either browserify or webpack.

RETWEETS

24

LIKES

40



8:34 AM - 11 Oct 2015



Reply to @Rich_Harris

https://twitter.com/Rich_Harris/status/653186863755304960



FREE CANDY

How much pouchdb do I really use? How big are the bundles?

Fun reading that helps understand things

"ECMAScript 6 modules: the final syntax" by Dr. Axel Rauschmayer
<http://www.2ality.com/2014/09/es6-modules-final.html>

My lovely graphs were made with
<http://asciiflow.com/>

Nolan's blog post on packaging libraries
<http://nolanlawson.com/2015/10/19/the-struggles-of-publishing-a-javascript-library/>

My name is Francis @reconbot



I write about the INTERNET I wish we had at

ROBOROOTER.COM

Ask me about great office space