

Revue de presse backbone.js 10 novembre 2015



Raphaël Lemaire @rlemaire

Backbone.js

On the road again



Jeremy Ashkenas

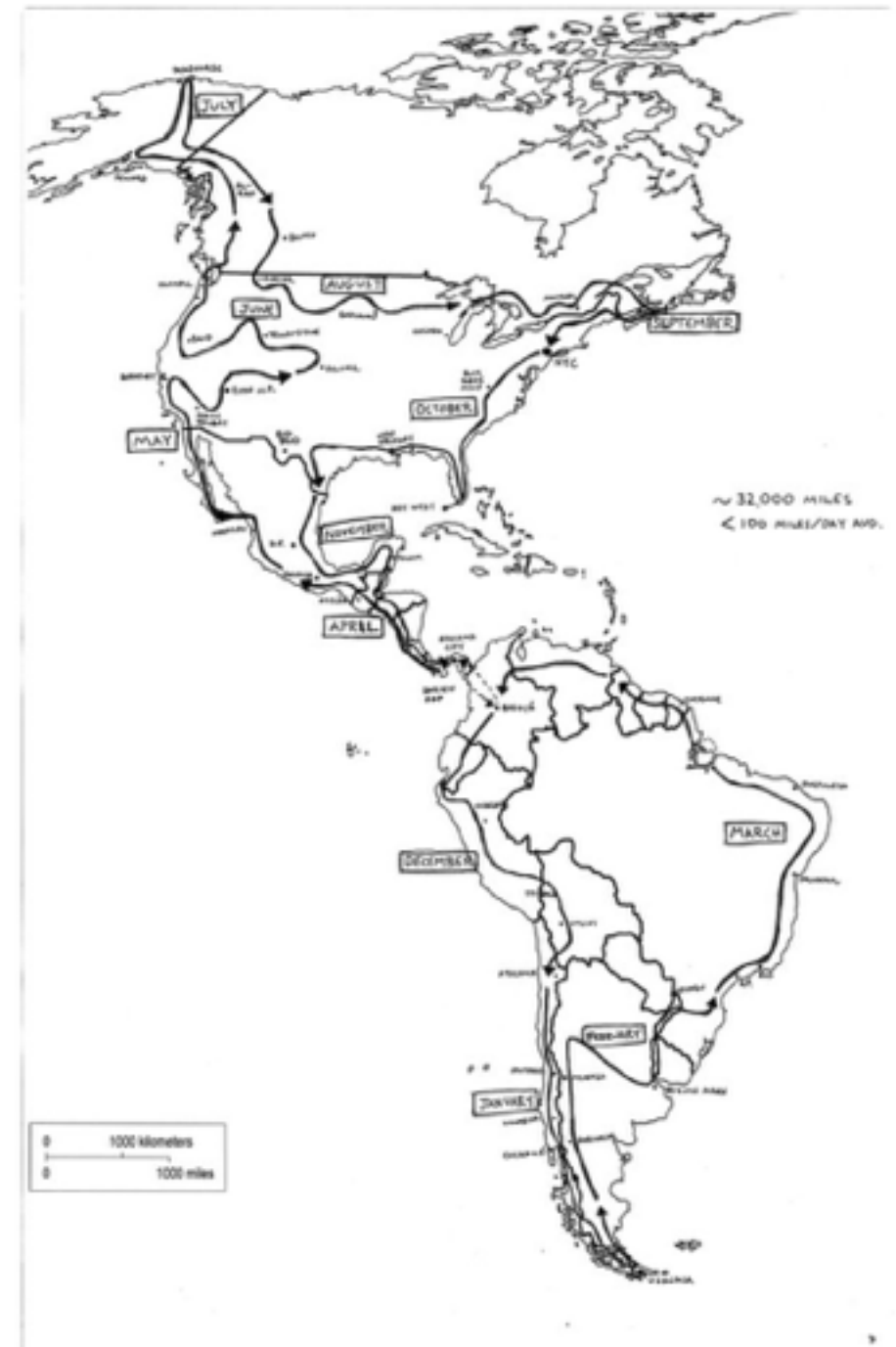
@jashkenas



Abonné

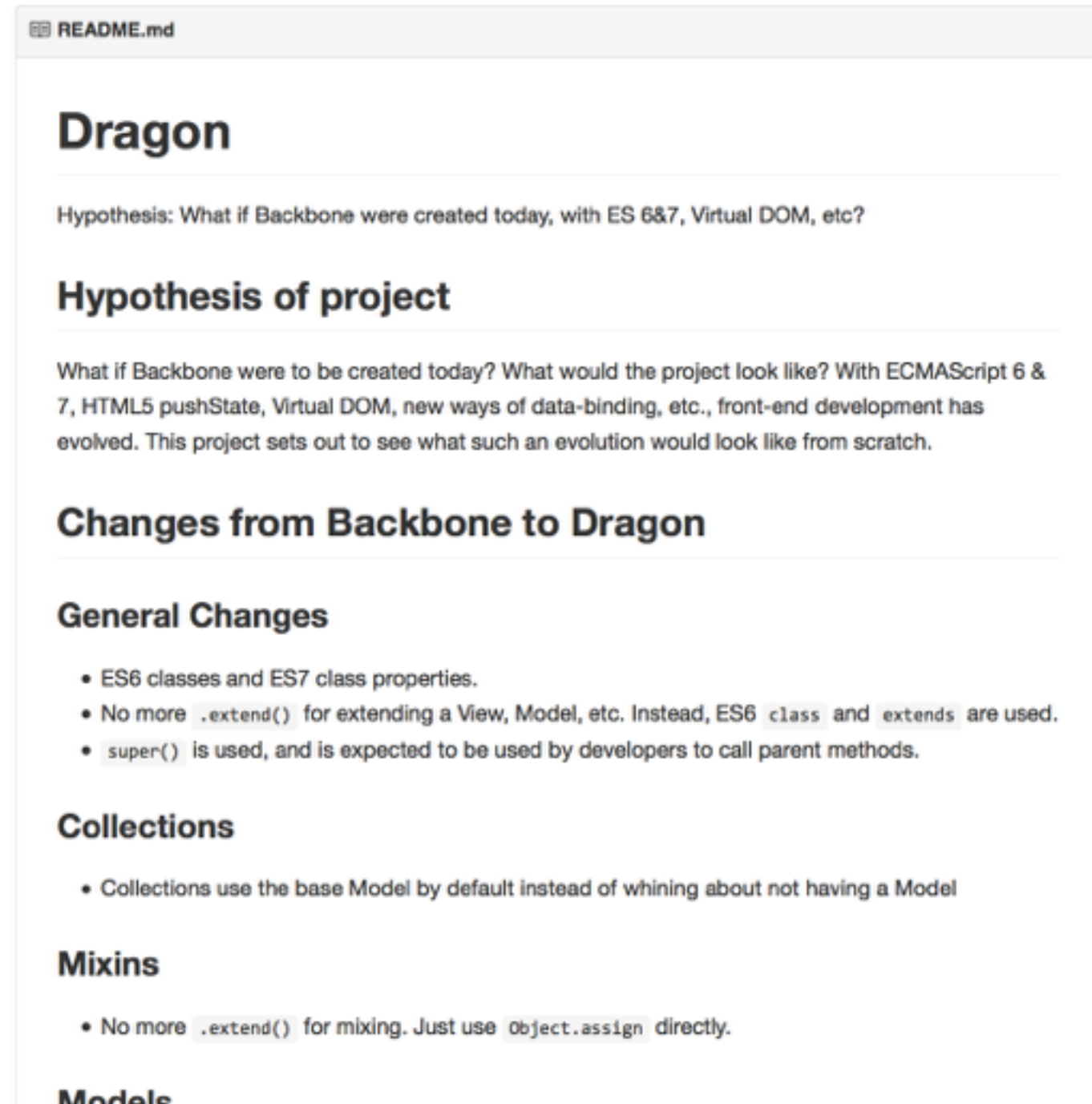
Dear friends, I'll be leaving NYC soon and hitting the road again. 35,000 miles, 20 countries, 1 year, 1 motorcycle.

[Voir la traduction](#)



What if backbone was written today ?

- <https://github.com/chrisabrams/dragon>
- Chris Abrams
- Utilisation des fonctionnalités ES6 et ES7 : classes, super, Object.assign, Object.observe, ...



Releases

- <http://bit.ly/1K6DDky> - Node 4 (inclusion de io.js, ES6 features...) supportée 30 mois
- <http://bit.ly/1NbzCgy> - Node 5 supportée 8 mois.
- <http://bit.ly/1LDhF5f> - Typescript 1.6 : inclusion de JSX (React), class expressions, classes et méthodes abstraites, intersection types...
- <http://bit.ly/1JeMp1f> - Ember 2.0
- <http://bit.ly/1iLehOe> - React v0.14
- <http://bit.ly/1RCUP1K> - Babel 6

ES6



ES6 In Depth Articles



[ES6 In Depth: The Future](#)

ES6 In Depth is a series on new features being added to the 6th Edition of the ECMAScript standard, ES6 for short. This is a survey of the major new features in ES6. This post covers:

Posted on August 21, 2015

- <https://hacks.mozilla.org/category/es6-in-depth/>
- Jason Orendorff
- Iterators and the for-of loop
- Generators
- Template strings
- Rest parameters and defaults
- Destructuring
- Arrow functions
- Symbols
- Collections
- Proxies
- Classes
- let and const
- Subclassing
- Modules

- <http://bit.ly/1MsZ19v> - A Study on Solving Callbacks with JavaScript Generators
- <http://bit.ly/1HAKEWn> - Good resources to learn, discover and explore ES6 in-depth
- <http://bit.ly/1kD6iDY> - ES7 Current proposals

```
Q.async(function*() {
  try {
    var post = yield db.hgetall('blog::post');
    var tags = post.tags.split(',');


    var taggedPosts = yield Q.all(tags.map(function(tag) {
      return db.hgetall('blog::tag::' + tag);
    }));

    // do something with post and taggedPosts
  }
  catch(e) {
    console.log(e);
  }

  client.quit();
})();
```

 **tdd / Learning and exploring ES6.md**
Created 10 days ago

Good resources to learn, discover and explore ES6 in-depth

 Learning and exploring ES6.md Raw

Learning

- [ES6 Katas](#) - Small, bite-size exercises to discover most aspects of ES6 features by doing interactive, online exercises. Pretty awesome.
- [Learn ES2015](#) - A great part of Babel's website that takes you through examples of all supported ES6+ features
- [ES6-Features.org](#) - Nice ES6 / ES5 comparisons of many ES6 language features
- [ES6 Features](#) - A single-page tour of code examples for just about every ES6 feature, by Luke Hoban


Exploring in-depth

- [ES6 in Depth](#) - A great series of articles on Mozilla Developer Network (MDN); also available [in French](#) through great translation efforts.
- [Exploring ES6](#) - Amazing free book by Axel Rauschmayer that goes into all the nitty-gritty with great example use cases.
- [ES6... and beyond!](#) - The latest, in-production chapter of Kyle Simpson's excellent [You Don't Know JS](#) series.

ECMAScript

Current Proposals

Proposals follow [this process document](#).

	Proposal	Champion	Stage
	Exponentiation Operator	Rick Waldron	3
	Array.prototype.includes	Domenic Denicola, Rick Waldron	3
	SIMD.JS - SIMD APIs + polyfill	John McCutchan, Peter Jensen, Dan Gohman, Daniel Ehrenberg	3
	Async Functions	Brian Terlson	3
	Object.observe	Adam Klein	2
	function.sent metaproperty	Allen Wirfs-Brock	2
	Object.values/Object.entries	Jordan Harband	2
	Rest/Spread Properties	Sebastian Markbage	2
	Trailing commas in function parameter lists	Jeff Morrison	2

Quizz ES6

<http://perfectionkills.com/javascript-quiz-es6/>

```
1. 1  (function(x, f = () => x) {  
    2      var x;  
    3      var y = x;  
    4      x = 2;  
    5      return [x, y, f()];  
    6  })(1)
```

gistfile1.js hosted with ♥ by GitHub

- ☐ [2, 1, 1]
- ☐ [2, undefined, 1]
- ☐ [2, 1, 2]
- ☐ [2, undefined, 2]

```
2. 1  (function() {  
    2      return [  
    3          (() => this.x).bind({ x: 'inner' })(),  
    4          (() => this.x)()  
    5      ]  
    6  }).call({ x: 'outer' });
```

- ☐ ['inner', 'outer']
- ☐ ['outer', 'outer']
- ☐ [undefined, undefined]
- ☐ Error

```
3. 1  let x, { x: y = 1 } = { x }; y;
```


Misc

	Heard of/Read About	Used a little	Feel Comfortable Using	Never Heard of
jQuery	1.4% (28)	7.2% (146)	91.4% (1,854)	0% (0)
Underscore	34.5% (700)	20% (406)	35.4% (718)	10.1% (204)
Backbone	60.8% (1,234)	21.1% (427)	16.2% (328)	1.9% (39)
AngularJS	36.9% (748)	33.5% (680)	29.1% (590)	0.5% (10)
Ember	78.7% (1,595)	13.1% (265)	4% (82)	4.2% (86)
React	61.8% (1,253)	19.9% (404)	14.3% (290)	4% (81)
Polymer	70.4% (1,429)	11.4% (231)	2.5% (50)	15.7% (318)
Knockout	65.2% (1,321)	9.7% (197)	6.1% (124)	19% (386)

- <http://bit.ly/1Lk4mGR> - The state of javascript tooling - 2015

If you don't, it's normal, you try to read the code abstracting the comment ;
it was slightly changed :

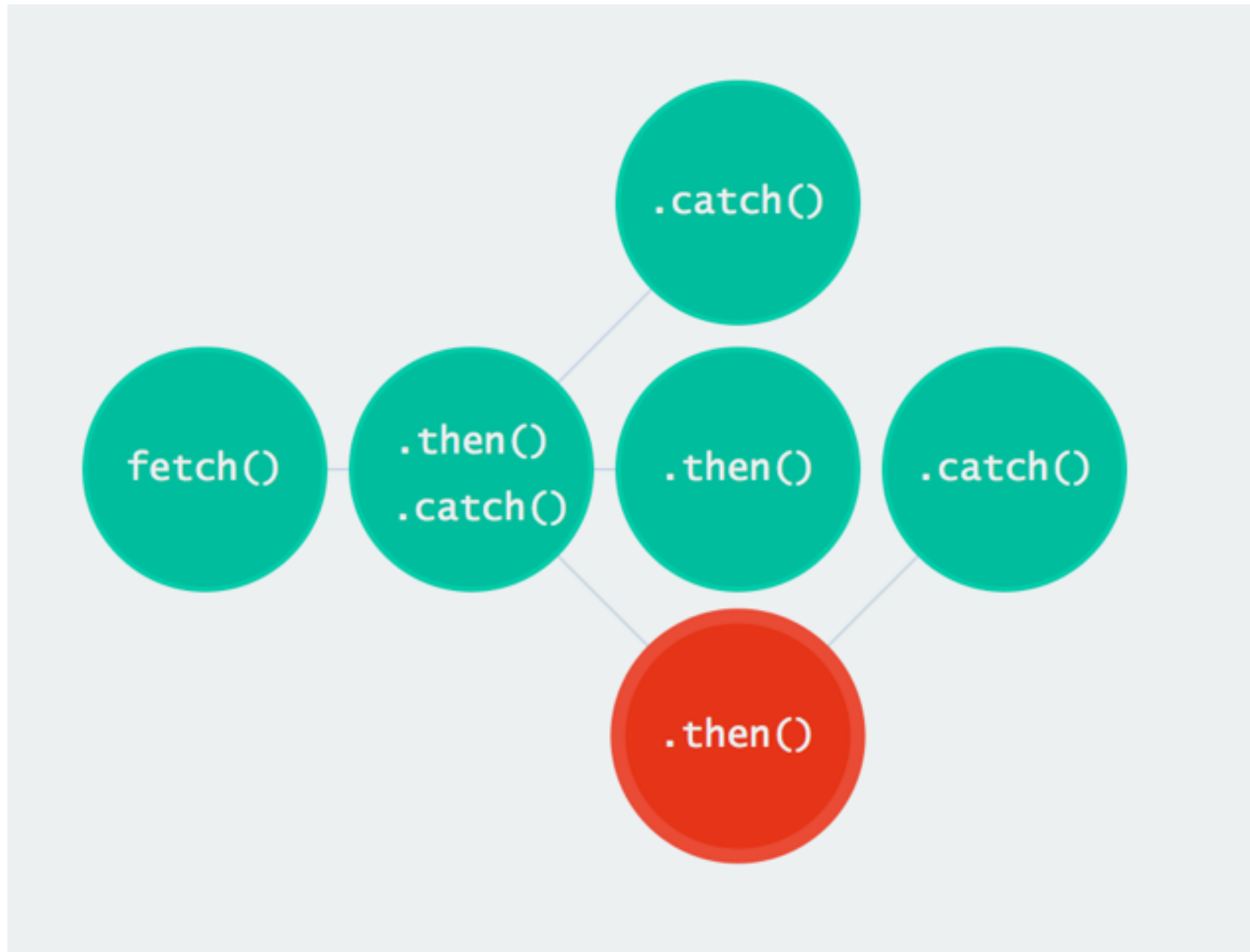
```
[ // Addition is one of the four elementary,
// mathematical operation of arithmetic arithmetic, with the other being subtraction, subtractions,
// multiplication multiplications and division. divisions. The addition of two whole numbers is the total
// amount of those quantity combined. For example in the picture on the right,
// there is a combination of three apples and two apples together making a total
// of 5 apples. This observation is equivalent to the mathematical expression
// "3 + 2 = 5"
// Besides counting fruit, addition can also represent combining other physical object.
```

What the ???

The small changes made the function body of `add()` growing over 600 character. v8 optimizer (crankshaft) inlines the functions whose body length, including the comments, is less than 600 characters.

- <http://bit.ly/1XXM7Q3> - Node : les commentaires influent sur la stratégie d'inlining

★ Top highli



Visualisation de promesses

<http://bevacqua.github.io/promisees/>





Merci.