

還給前端一片天空

by @tonytonyjan

先生不知何許人





tonytonyjan

大兜 @tonytonyjan

- 技術長 - Ziltag
- 發起人 - 五倍紅寶石
- 作者 - Ruby on Rails 自習手冊
- 作者 - jaro_winkler, exif, xdite gem, etc
- 講者 - ConFoo 2015
- 講者 - Ruby Kaigi 2014

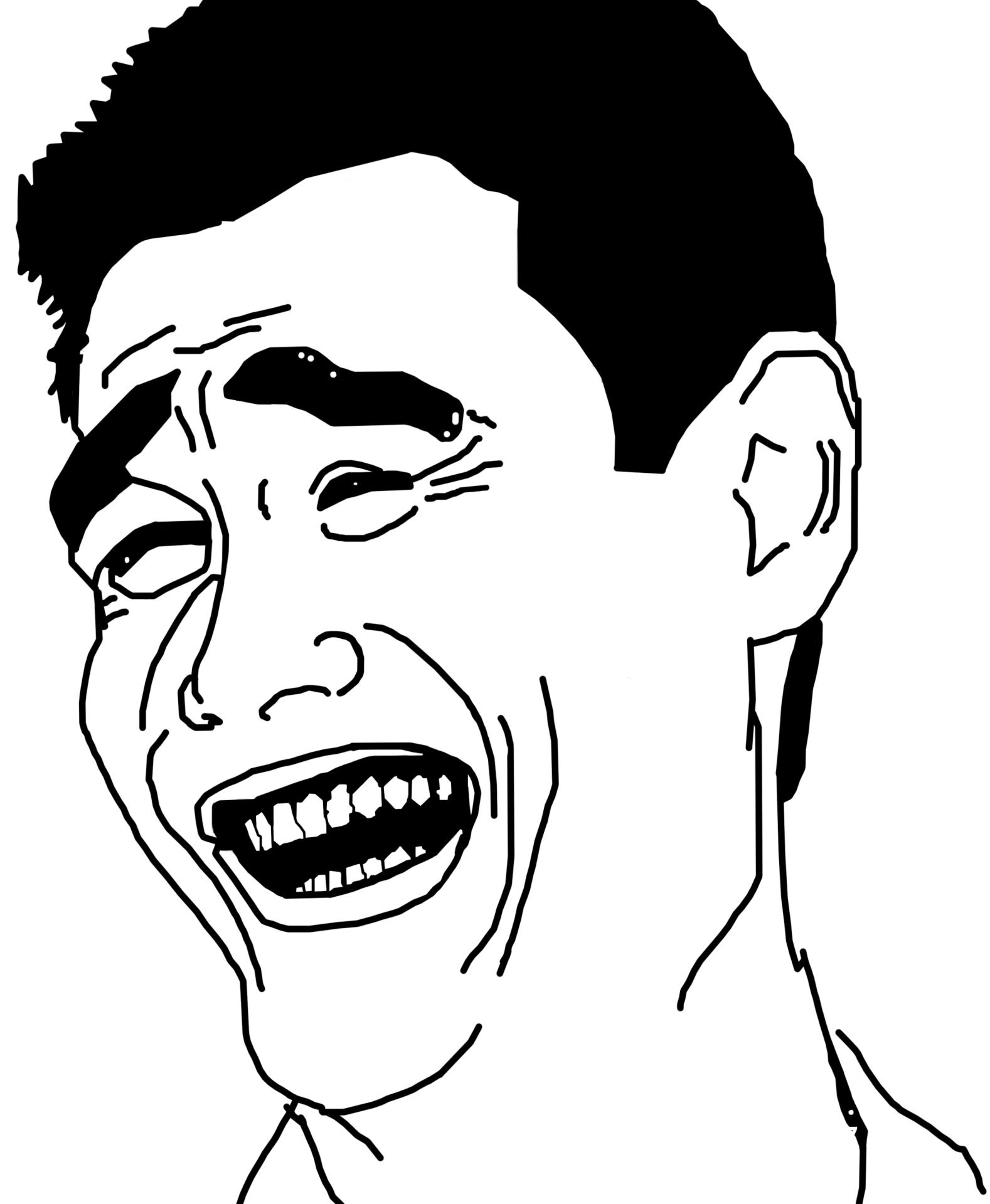


特長

- C
- Ruby
 - Rails
 - C Extension
- JavaScript
- CSS



AJAJ



講題大綱

還給前端一片天空

- 聽我講古
- 在 Rails 上遇到的前端問題
- 透過 JS 生態圈彌補 Rails 的不足

很久很久以前……

我們的黑暗時代 (1/4)

```
<head>
  <link rel="stylesheet" href="aaa.css">
  <link rel="stylesheet" href="bbb.css">
  <link rel="stylesheet" href="ccc.css">
  <script src="aaa.js"></script>
  <script src="bbb.js"></script>
  <script src="ccc.js"></script>
</head>
```

HTTP 請求太多

我們的黑暗時代 (2/4)

Concatination

ex. cat *.js

```
<head>
  <link rel="stylesheet" href="all.css">
  <script src="all.js"></script>
</head>
```

福山繁大

我們的黑暗時代 (3/4)

Compression

ex. YUI Compressor

```
<head>
  <link rel="stylesheet" href="all.min.css">
  <script src="all.min.js"></script>
</head>
```

深刻體驗學習快樂取

我們的黑暗時代 (4/4)

加上版本號防快取

```
<head>
  <link rel="stylesheet" href="all.min.css?v2">
  <script src="all.min.js?v3"></script>
</head>
```

難以 debug

「我們可以用 Source Map 嘎！」

犀利人要。
最終回

可是瑞凡

犀利人要
最終回

那時候還沒有 SOURCE MAP



在這個黑暗時代 (2011 以前)

前端網頁開發是如此麻煩



Rails 3.1

引進了 Assets Pipeline

Sprockets

Sprockets 做了些什麼 (1/4)

Transform

```
aaa.coffee ↗  ↗> aaa.js  
bbb.coffee +> (transformers) +> bbb.js  
ccc.coffee ↘  ↙> ccc.js
```

Sprockets 做了些什麼 (2/4)

Concat & Compress

```
aaa.js ↴  
bbb.js +> (concat) -> app.js → (compressor) -> app.min.js  
ccc.js ↴
```

Sprockets 做了些什麼 (3/4)

Digest

```
app.min.js → (digest) → app-{etag-string}.js
```

Sprockets 做了些什麼 (4/4)

Javascript Templating

```
<!-- templates/hello.jst.ejs -->
<div>Hello, <span><%= name %></span>!</div>

// application.js
//= require templates/hello
$("#hello").html(JST["templates/hello"]({ name: "Sam" }));
```



可是呢

在某些情況下

這讓有些事情變得複雜

你有可能……

- 喜歡 TypeScript 勝過 CoffeeScript
- 喜歡 LESS 勝過 SASS
- 想用 ReactJS 或 Polymer 取代 JST
- 喜歡用 bower 管理第三方的前端套件勝過 gem
- 想要使用 Mocha、Jasmine 寫測試

「一切只要交給 gem 就好啦！」

TypeScript -> typescript-rails

LESS -> less-rails

ReactJS -> react-rails

rails-assets.org

Gemfile

```
source 'https://rails-assets.org' do
  # gem "rails-assets-#{BOWER_PKG_NAME}"
  gem 'rails-assets-bootstrap'
  gem 'rails-assets-angular'
  gem 'rails-assets-leaflet'
end
```

萬一沒有 gem ?

可以儲存至 vendor/assets/

禹一 gem 太多？
bootstrap gem 有 3 個以上

官方 gem 優先
熱門 gem 其次

妥協的結果：

部分放 vendor/assets

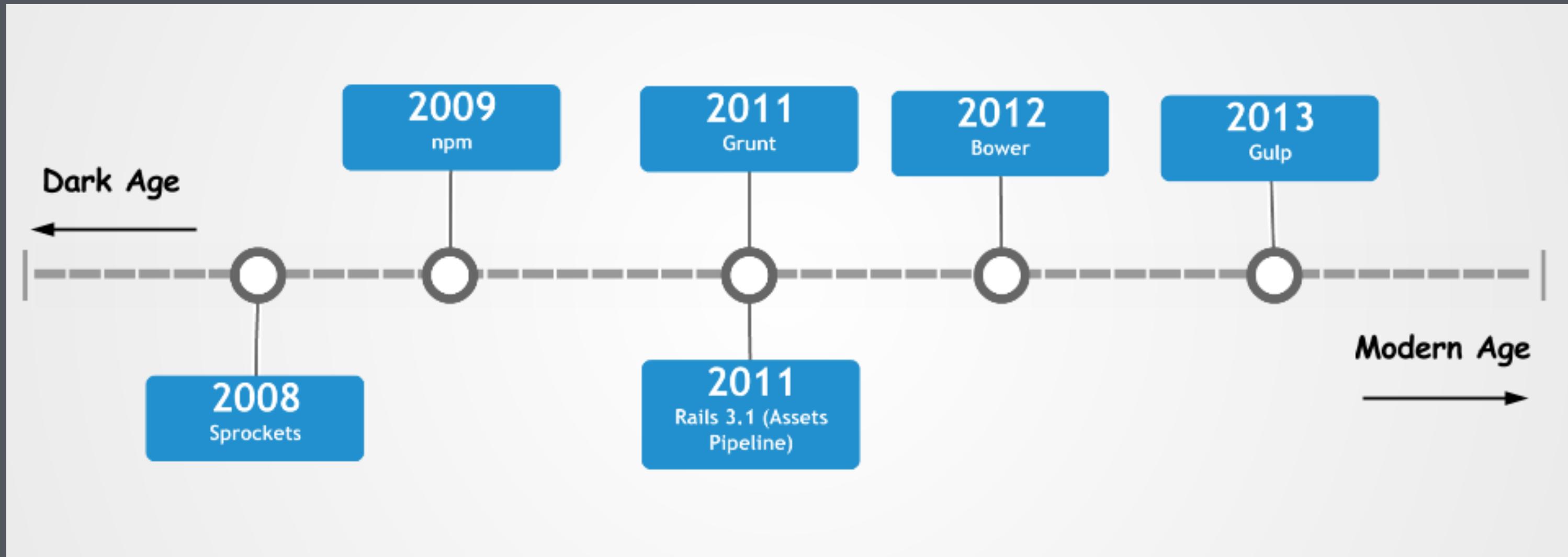
部分放 Gemfile

很糟糕米糕阿 · · · · ·

前端工程師的提問

- 所以我要學 JST 嗎？
- 我要學 sprockets manifest system 嗎？
- 我要怎麼安裝新的套件呢？
- 前端卡關應該求助 Ruby 社群還是 JS 社群？

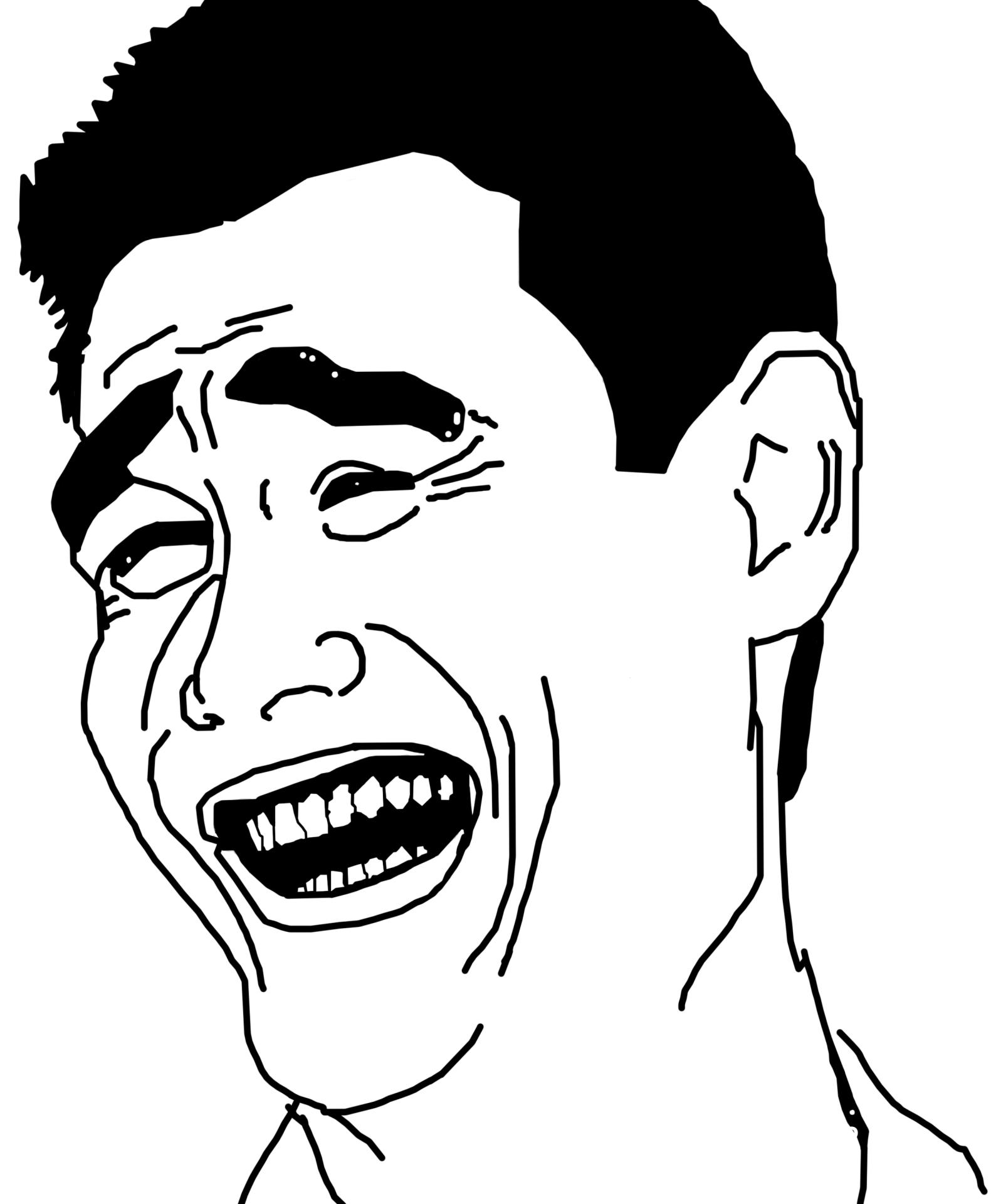
工具演進史



其實前端碼農們有自己的工具

Bower, Gulp, npm, Grunt, ...

可是在 Rails
通通不能用





Rails + Bower

你也許不知道

rails-assets.org

分手 不用 Gem 的作法

.bowerrc

```
{  
  "directory": "vendor/assets/components"  
}
```

或

```
{  
  "directory": "public/components"  
}
```

assets.rb

```
Rails.application.config.assets.paths.concat [  
  'vendor/assets/components'  
]
```

application.scss

```
@import 'material-design-lite/src/material-design-lite'
```

application.js

```
//= require jquery
//= require jquery_ujs
//= require turbolinks
//= require material-design-lite/material
```

乍看元大

不過有幾個問題……

- 不正確的 CSS 中的 assets 路徑
- 想使用前端世界的工具，例如：
 - Autoprefixer
 - BrowserSync
 - JSHint
 - etc

不正確的 assets 路徑

以 bootstrap 為例：

```
@font-face {  
  font-family: 'Glyphicons Halflings';  
  src: url('../fonts/glyphicons-halflings-regular.eot');  
}
```

bower install bootstrap-sass

當然，如果有官方提供的 sprockets 設定最好不過：

```
@import 'bootstrap-sass/.../bootstrap-sprockets';  
@import 'bootstrap-sass/.../bootstrap';
```

但不是每個 bower package 都有提供 sass 的版本。

至於想使用前端世界的工具
(Autoprefixer、BrowserSync、JSHint、etc)

前站是 Rails
生態圈的不足

他山之石
可以攻錯

Rails + Bower + gulp

看看別人怎麼做



EXTEND // FEB 18, 2015

24



Gulp on Rails: Replacing the Asset Pipeline

DAN TELLO

Senior Front-End
Developer

[Dan's profile](#)



THE GULP RAILS PIPELINE



Extend is...

a CODE & TECHNOLOGY
blog—our thirty foot view.
primarily written by/for
ruby, ios, and front-end
developers.

SEARCH

MORE VIGET
BLOGS

Advance
STRATEGY & MARKETING

Extend
CODE & TECHNOLOGY

Flourish
NEWS & CULTURE

Dan 的檔案結構

Source Assets

js modules, sass, svgs, images, etc.

✓ gulp

✓ assets

> icons

> images

> javascripts

> stylesheets

Public Assets

referenced in css or js

✓ public

✓ assets

> fonts

> images

> javascripts

> stylesheets

Manifest Files (if needed)

require gem assets + compiled files

✓ app

✓ assets

✓ javascripts

----- application.js

✓ stylesheets

----- application.css

可能有的問題

1. 各位的檔案必須這樣放
2. 多了一個叫做 `gulp_asset_path` 的 helper
3. `public` 有太多沒必要的中間產物



katryo

6 Presentations

[★ Star this Talk](#)

4 Stars

[Published in](#)

Technology

[Stats](#)

988 Views

Share

[Twitter, Facebook](#)[Embed](#)[Direct Link](#)[Download PDF](#)

Sprockets絶ちに挑戦した

@katryo



katryo 的工具包

- gulp-uglify
- gulp-css-minify
- gulp-concat
- gulp-rev
- gulp-rev-rails-manifest

可能有的問題

1. 完全取代了 sprockets，捨棄 gem 裡面的 assets
2. 複寫了 asset_path 方法以產生正確的 etag string
3. 使用 browserfy，巢狀相依會導致檔案變大

所以還能怎麼做呢？

在那之前

先看看問題是什麼

多想兩分鐘...
...

你 可 以 不 必
去 掉 Sprockets

為何不捨棄 sprockets

1. 有些 assets gem 需要 sprockets 的 directive processor
2. 內建的 compression、digestion 已經做得很好

之所以使用 bower、gulp、npm 等工具

是為解決 rails 前端生態圈的不足

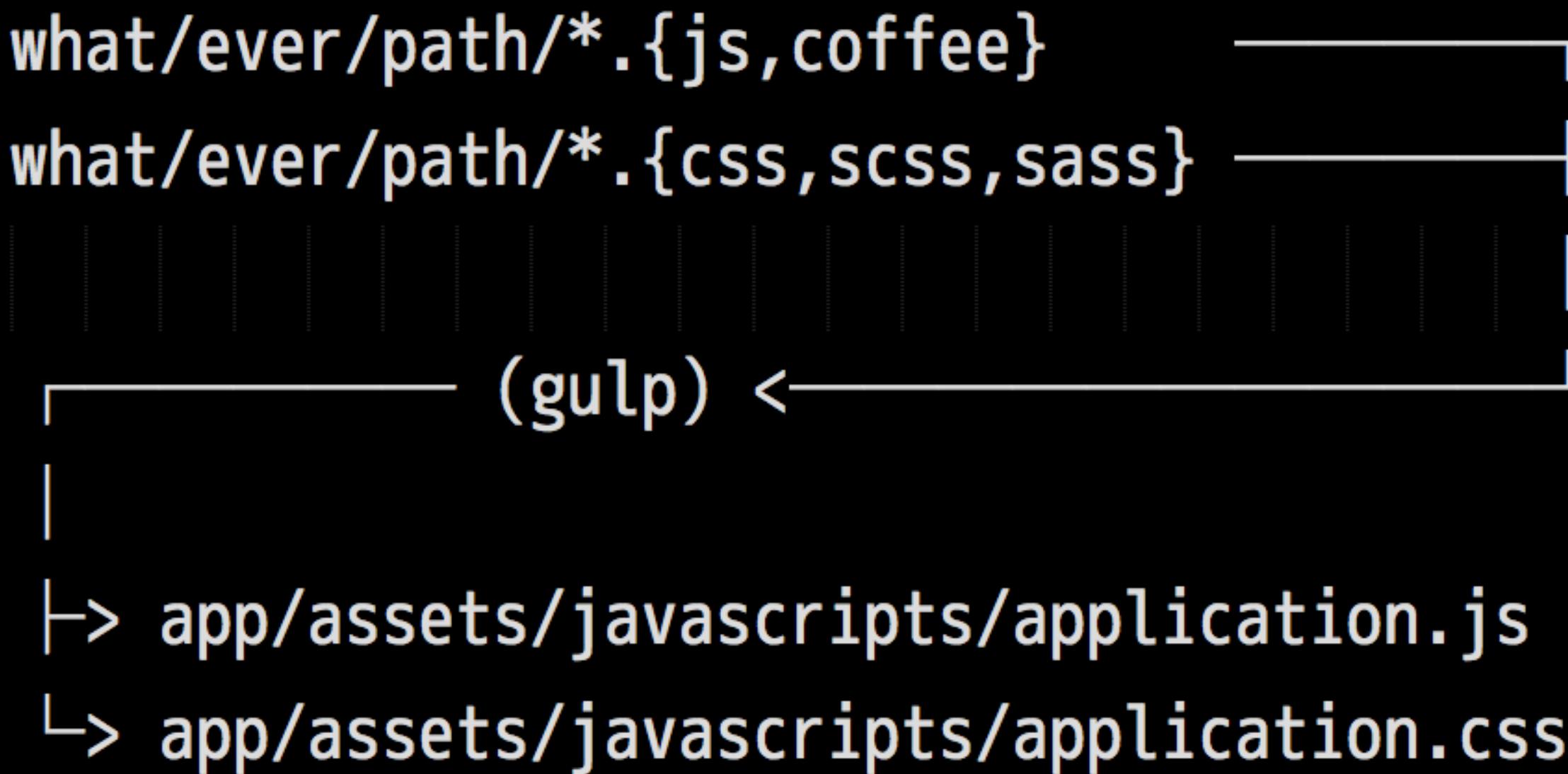
畢竟前端的 gem 再怎麼多

也不會比 JS 生態圈多

但也沒有必要拿掉
sprockets 完善的部分

我想是這樣啦……

```
what/ever/path/*.js,coffee  
what/ever/path/*.css,scss,sass  
|  
|  
|> app/assets/javascripts/application.js  
└> app/assets/javascripts/application.css
```



可能的好處有……

不用裝表 gem

不用整合

- gulp-rev
- gulp-uglify
- gulp-css-minify
- gulp-rev-rails-manifest

交給 sprockets 即可

要重視

前端的 code 前端工程師管

gulp application.js

```
gulp.task 'application.js', ->
  gulp.src [
    'app/assets/javascripts/application/manifest.js'
    'bower_components/bootstrap/dist/js/bootstrap.js'
    'bower_components/material-design-lite/material.js'
    'app/assets/javascripts/application/**/*.{coffee,js}'
  ]
  .pipe $.if /\.coffee$/, $.coffee()
  # 可以繼續 pipe 直到前端工程師滿意
  .pipe $.concat('application.js')
  .pipe gulp.dest 'app/assets/javascripts'
```

glup serve

```
browserSync = require('browser-sync').create()
```

```
gulp.task 'serve', ->
  browserSync.init
    proxy: 'localhost:3000'
```

「可是 font 路徑怎麼解決呢？」

一般 CSS

```
.logo {  
  background-image: url(logo.png)  
}
```

sprockets SCSS helper

```
.logo {  
  background-image: asset-url("logo.png")  
}
```

使用 ERB

```
.logo {  
  background-image: url("<%= asset_path 'logo.png' %>")  
}
```

這同時也是 JS 檔案中引用 assets 路徑的官方解法

處理 assets 路徑

config/initializers/assets.rb

```
Rails.application.config.assets.paths << 'vendor/assets/components'  
Rails.application.config.assets.paths.concat Dir['vendor/assets/components/**/*/fonts']  
Rails.application.config.assets.precompile += %w[*.eot *.woff2 *.woff *.ttf *.svg]
```

處理 assets 路徑

Rails.application.assets.paths

```
/.../hello/app/assets/images  
/.../hello/app/assets/javascripts  
/.../hello/app/assets/stylesheets  
/.../hello/vendor/assets/components  
/.../hello/vendor/assets/javascripts  
/.../hello/vendor/assets/stylesheets  
/.../gems/turbolinks-2.5.3/lib/assets/javascripts  
/.../gems/jquery-rails-4.0.4/vendor/assets/javascripts  
/.../gems/coffee-rails-4.1.0/lib/assets/javascripts  
/.../hello/vendor/assets/components  
/.../hello/vendor/assets/components/bootstrap-sass/assets/fonts    *  
/.../hello/vendor/assets/components/font-awesome-sass/assets/fonts *
```

處理 assets 路徑

路徑轉換

```
url(path/to/fonts/foo.svg) -> url("<%= asset_path 'foo.svg' %>")  
url(..../fonts/bar.eot) -> url("<%= asset_path 'bar.eot' %>")  
url(fonts/buz.woff2) -> url("<%= asset_path 'buz.woff2' %>")
```

處理 assets 路徑

看看 bootstrap-sass 怎麼支援 Mincer

```
@function twbs-font-path($path) {
  // do something like following
  // from "path/to/font.ext#suffix" to "<%- asset_path(path/to/font.ext)) + #suffix %>"
  // from "path/to/font.ext?#suffix" to "<%- asset_path(path/to/font.ext)) + ?#suffix %>"
  // or from "path/to/font.ext" just "<%- asset_path(path/to/font.ext)) %>"
  @return "<%- asset_path('#{$path}'.replace(/[#?].*$/, '')) + '#{$path}'.replace(/(^[^#?]*)([#?].*$/), '$2') %>";
}
```

處理 assets 路徑

gulp-replace

```
$ = require('gulp-load-plugins')()
gulp.task 'application.css', ->
  gulp.src [
    'app/assets/stylesheets/application.scss'
    'app/assets/stylesheets/application/**/*.{css,scss,sass}'
  ]
  .pipe $.concat 'application'
  .pipe $.sass
    includePaths: ['vendor/assets/components']
  .pipe $.replace /url\((([""])?[^\\1])*fonts\\/([^\\1]+)\\1\\)/g, "url('<%= asset_path '$2' %>')"
  .pipe $.rename 'application.css.erb'
  .pipe gulp.dest 'app/assets/stylesheets'
```

範例專案

https://github.com/tonytonyjan/rails_gulp_bower

感謝收穫