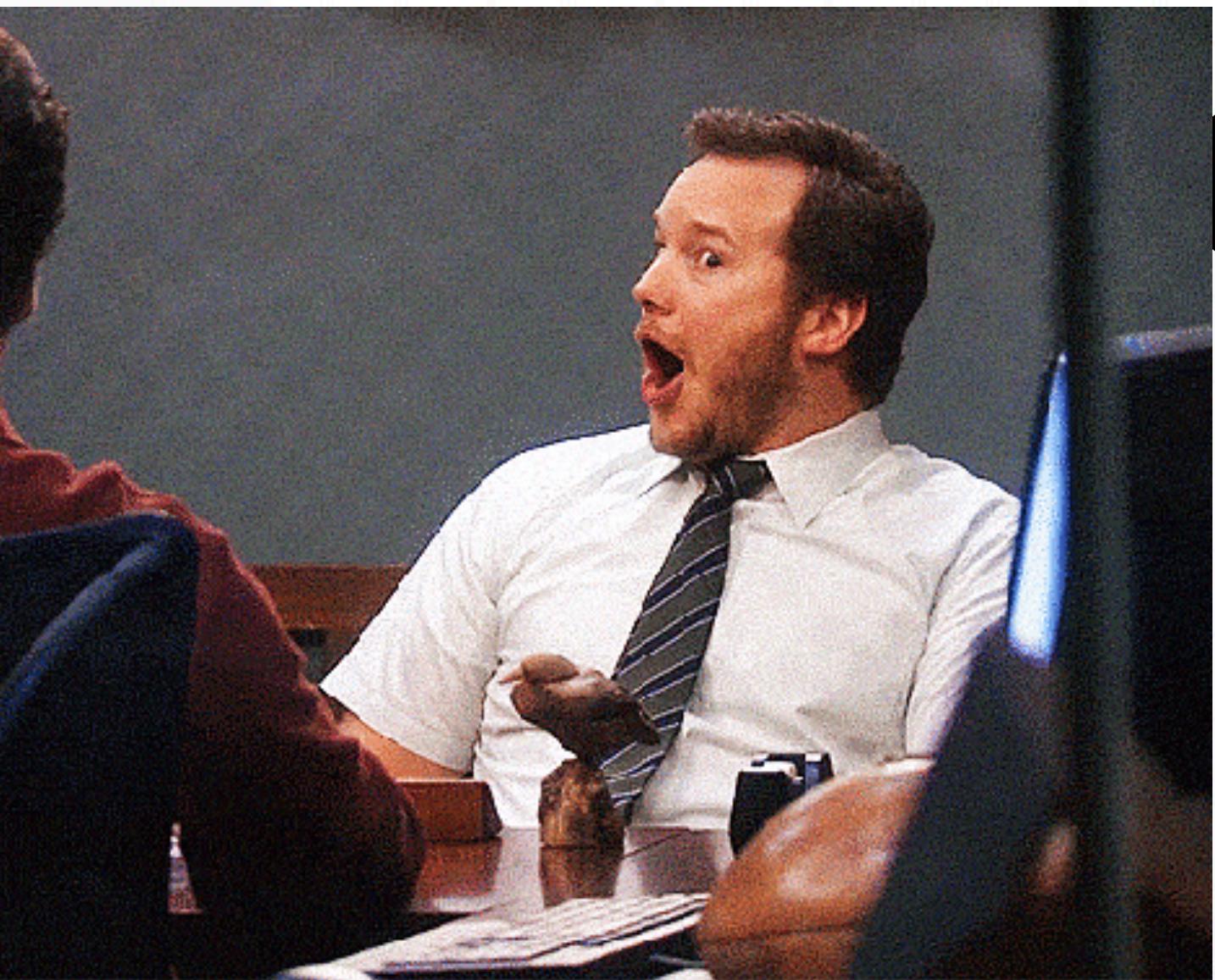
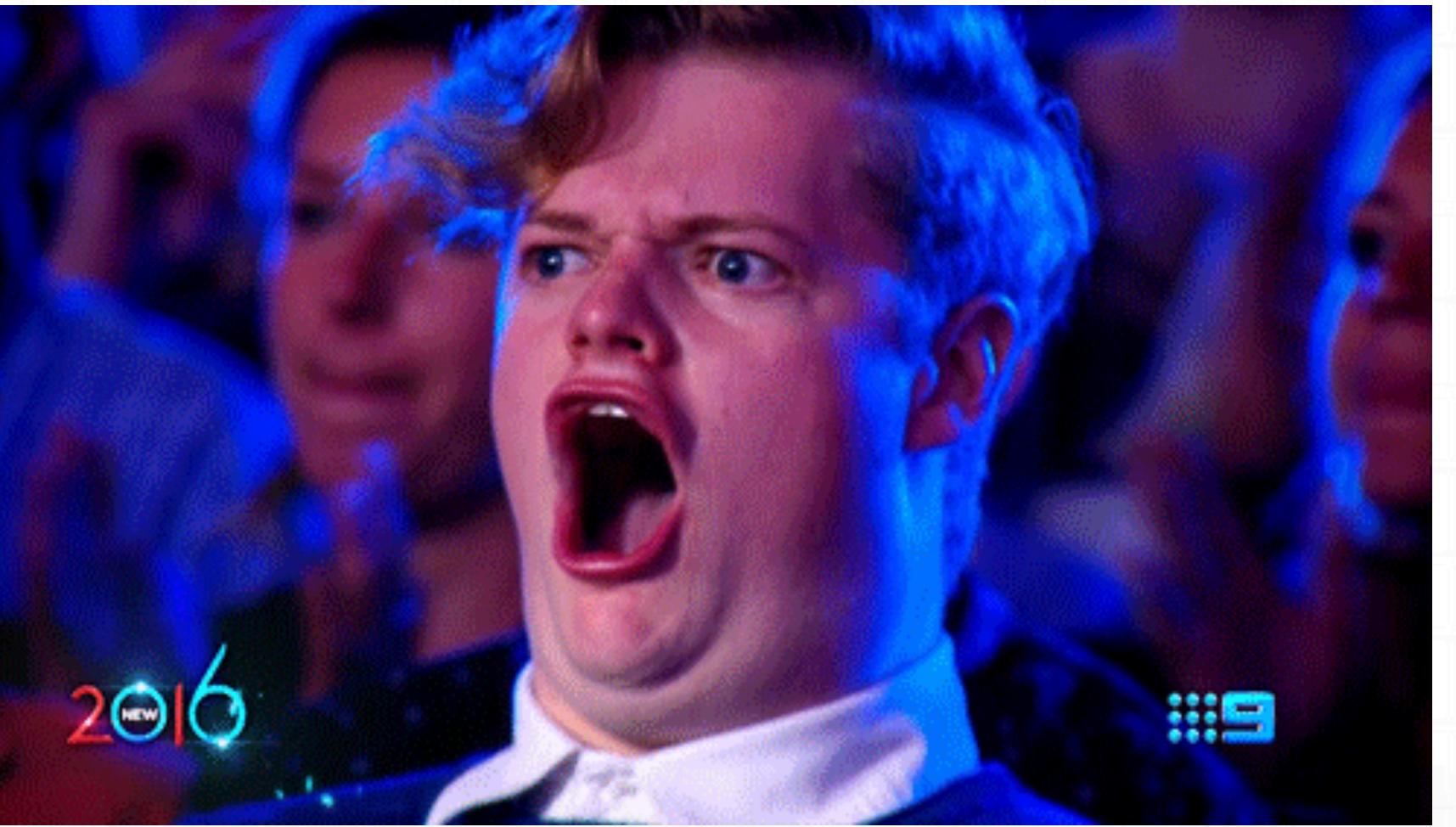


MY FIRST AND LAST RAILSCONF



MASTER THE RAILS ASSET PIPELINE: BEST PRACTICES FOR APPS & GEMS



CLAIMED



DISCLAIMER 2



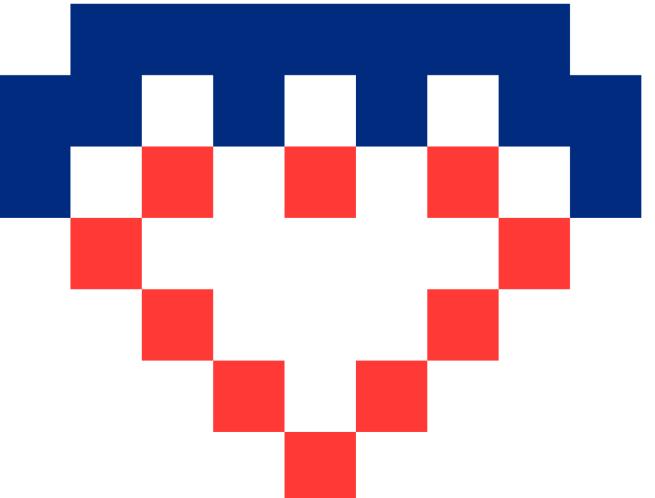
SUMMARY

- 1. What are assets?**
- 2. How was it in the ol' days**
- 3. Why do we have an asset pipeline**
- 4. Where are we now**
- 5. How can we leverage it to build effective Rails plugins**

ADRIAN MARIN

<https://www.linkedin.com/in/adrianthedev/>

- Doing development since before 2010
- Came to Rails in 2016
- Built a Rails engine in 2020 - <https://avo.cool> 
- Built Marksmith, a GitHub-styled markdown editor 
- I run a boutique Ruby conference in Bucharest called Friendly.rb



FRIENDLY RB

WHAT ARE ASSETS?

- **Images**
- **Javascript**
- **CSS**
- **Favicons**
- **JSON**
- **XML**

HOW DID WE DO IT IN THE OLD DAYS?

LOAD THINGS ONE BY ONE

- Load jQuery
- jQuery plugins
- Own plugins
- Vars.js
- main.js
- Others...

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/vendor/jquery-1.9.1.min.js"></script>');
<script src="components/fancybox/source/jquery.fancybox.js"></script>
<script src="js/plugins.js"></script>
<script src="js/main.js"></script>
```

```
<script src="{{URL::base()}}/js/plugins.js"></script>
<script src="{{URL::base()}}/js/vars.js"></script>
<script src="{{URL::base()}}/js/main.js"></script>
```

```
<script src='http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js'></script>
<script src='http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.2/jquery-ui.min.js'></script>
<script src='{{URL::base()}}/js/tinymce/tinymce.min.js'></script>
<script src='{{URL::base()}}/js/plugins.js'></script>
<script src='{{URL::base()}}/js/validation-{{Config::get("application.language")}}.js'></script>
<script src='{{URL::base()}}/js/script.js'></script>
```

WHY DO WE HAVE AN ASSET PIPELINE?

WHAT PROBLEMS DOES IT FIX?

1. Bundling (concatenation)

BUNDLING

```
gulp.task('icons', function () {  
  
  gulp.src([publicPath + 'svg/*.svg'])  
    .pipe(iconfont({  
      fontName: 'project-font',  
      appendCodepoints: true,  
    }))  
    .on('codepoints', function (codepoints, options) {  
      gulp.src(publicPath + 'css/font-template.css')  
        .pipe(consolidate('lodash', {  
          glyphs: codepoints,  
          fontName: 'project-font',  
          fontPath: publicPath + 'fonts/',  
          className: 'pf'  
        }))  
        .pipe(concat('project-font.css'))  
        .pipe(gulp.dest(publicPath + 'css/'));  
  
      // console.log(codepoints, options);  
      gulp.src(publicPath + 'style.less')  
        .pipe(gulp.dest(publicPath));  
    })  
    .pipe(gulp.dest(publicPath + 'fonts'));  
});
```

```
gulp.task('watch', ['browser-sync'], function () {  
  
  gulp.watch([  
    publicPath + "admin-assets/*.less",  
    publicPath + 'admin-assets/css/*.less'  
  ], ['adminStyles', function () {  
    browserSync.reload(["admin.min.css"]);  
  }]);  
  
  gulp.watch([  
    publicPath + "style.less",  
    publicPath + '/css/*.less'  
  ], ['styles', function () {  
    browserSync.reload(["style.min.css"]);  
  }]);  
  
  gulp.watch(['public/svg/*.svg'], ['icons']);  
  notify('Gulp watch started.');
```

```
gulp.task('scripts', function () {  
  
  var dependencies = require('./' + publicPath + "js/dependencies");  
  dependencies = dependencies.map(function (item) {  
    return publicPath + item;  
  });  
  
  console.log(dependencies);  
  return gulp.src(dependencies)  
    .pipe(concat('mainscript.js'))  
    .pipe(notify('Starting scripts task.'))  
    .pipe(gulp.dest(publicPath + 'js'))  
    .pipe(uglify())  
    .pipe(rename({suffix: '.min'}))  
    .pipe(gulp.dest(publicPath + 'js'))  
    .pipe(notify('Finished scripts task.'))
```

WHY DO WE HAVE AN ASSET PIPELINE?

WHAT PROBLEMS DOES IT FIX?

1. Bundling (concatenation)
2. Fingerprinting

FINGERPRINTING

```
// app/javascript/application.js  
console.log("I love Ruby on Rails") // becomes application-d3d22b9d.js
```

```
// app/javascript/application.js  
console.log("I love Ruby on Rails!!!") // becomes application-4b4da292.js
```

WHY DO WE HAVE AN ASSET PIPELINE?

WHAT PROBLEMS DOES IT FIX?

1. Bundling (concatenation)
2. Fingerprinting
3. Transpiling

TRANSPILING

FANCY TALK FOR THE MAP FUNCTION

- SCSS to CSS
- Add vendor prefixes
- CoffeeScript to JS
- minify scripts
- optimize images and assets

SPROCKETS TO THE RESCUE

Sprockets: Rack-based asset packaging

Sprockets is a Ruby library for compiling and serving web assets. It features declarative dependency management for JavaScript and CSS assets, as well as a powerful preprocessor pipeline that allows you to write assets in languages like CoffeeScript, Sass and SCSS.

	— Sprockets —
▼	autoload
»	babel.rb
»	closure.rb
»	coffee_script.rb
»	eco.rb
»	ejs.rb
»	jsminc.rb
»	sass.rb
»	sassc.rb
»	uglifyer.rb
»	yui.rb
»	zopfli.rb

WHERE ARE WE NOW?



DAVID HEINEMEIER HANSSON

February 11, 2022

Introducing Propshaft

- **HTTP 2 solved the must**
- **CSS became smarter**
- **JS became smarter and**

It's an exciting time in web development. After a decade's worth of front-end progress kept demanding ever more complicated setups, we're finally moving in the opposite direction. With simpler tools that are still able to hit those high-fidelity user interface notes, but at a sliver of the cost in complexity. The long expansion of enabling concepts is now at last being compressed for human comprehension. Hallelujah!

And like with expanding complexity, where one complication often leads to another, compressing complexity also cascades.

[Propshaft](#), a new asset pipeline library for Rails, is the result of such a cascading compression of complexity. Enabled by [the same trifecta](#) of HTTP/2, browser-run ES6, and import maps that powers [Rails 7](#), Propshaft is dramatically simpler than the Sprockets that went before it.

In our modern asset world, JavaScript and stylesheets are either [sent directly to the browser without preprocessing](#), or they're being preprocessed by standalone tools like [esbuild](#), [Dart Sass](#), and [Tailwind CSS](#). In both cases, our asset pipeline library needn't lift the burden.

PROPSHAFT

- Is mostly a drop-in replacement for Sprockets
- Does only 4 things
 - Fingerprinting
 - Dev server
 - Just a bit of transpiling
 - Adds a configurable load path

FRONT-END GOT EVEN BETTER FOR RAILS IN 2021

- **Importmaps**
- **jsbundling**
- **cssbundling**



DAVID HEINEMEIER HANSSON

September 6, 2021

Rails 7 will have three great answers to JavaScript in 2021+

Rails has been unapologetically full stack since the beginning. We've continuously sought to include ever-more default answers to all the major infrastructure questions posed by modern web development. From talking to a database, to sending and receiving emails, to connecting web sockets, to rendering HTML, to integrating with JavaScript. This full-stack strategy has been key to the success of Rails, but it also remains an enduring source of controversy. What's too much to include? What's not enough?

To consistently answer that evergreen question, we look to [The Rails Doctrine](#), and especially the third pillar of [The Menu Is Omakase](#). It's why we fret so much over the defaults, but also why the option to substitute is so crucial.

There's been no more fretting over the defaults, or a closer examination of the substitutes, than with the JavaScript part of the question over the years. And especially lately, as the ever-present churn and fundamental change has pushed new options into the limelight. But after much experimentation, I believe we now have a solid answer for Rails 7.

IMPORTMAPS

- Run JS directly in the browser
- No build step

```
<link rel="stylesheet" href="/assets/application-8b441ae0.css" data-turbo-track="reload" />
<script type="importmap" data-turbo-track="reload">{
  "imports": {
    "application": "/assets/application-98e8f90d.js",
    "@hotwired/turbo-rails": "/assets/turbo.min-3a2e143f.js",
    "@hotwired/stimulus": "/assets/stimulus.min-4b1e420e.js",
    "@hotwired/stimulus-loading": "/assets/stimulus-loading-1fc53fe7.js",
    "controllers/application": "/assets/controllers/application-3affb389.js",
    "controllers/hello_controller": "/assets/controllers/hello_controller-708796bd.js",
    "controllers": "/assets/controllers/index-ee64e1f1.js"
  }
}</script>
<link rel="modulepreload" href="/assets/application-98e8f90d.js">
<link rel="modulepreload" href="/assets/turbo.min-3a2e143f.js">
<link rel="modulepreload" href="/assets/stimulus.min-4b1e420e.js">
<link rel="modulepreload" href="/assets/stimulus-loading-1fc53fe7.js">
<link rel="modulepreload" href="/assets/controllers/application-3affb389.js">
<link rel="modulepreload" href="/assets/controllers/hello_controller-708796bd.js">
<link rel="modulepreload" href="/assets/controllers/index-ee64e1f1.js">
<script><!-->import "application"<!--&gt;&lt;/script&gt;</pre>
```

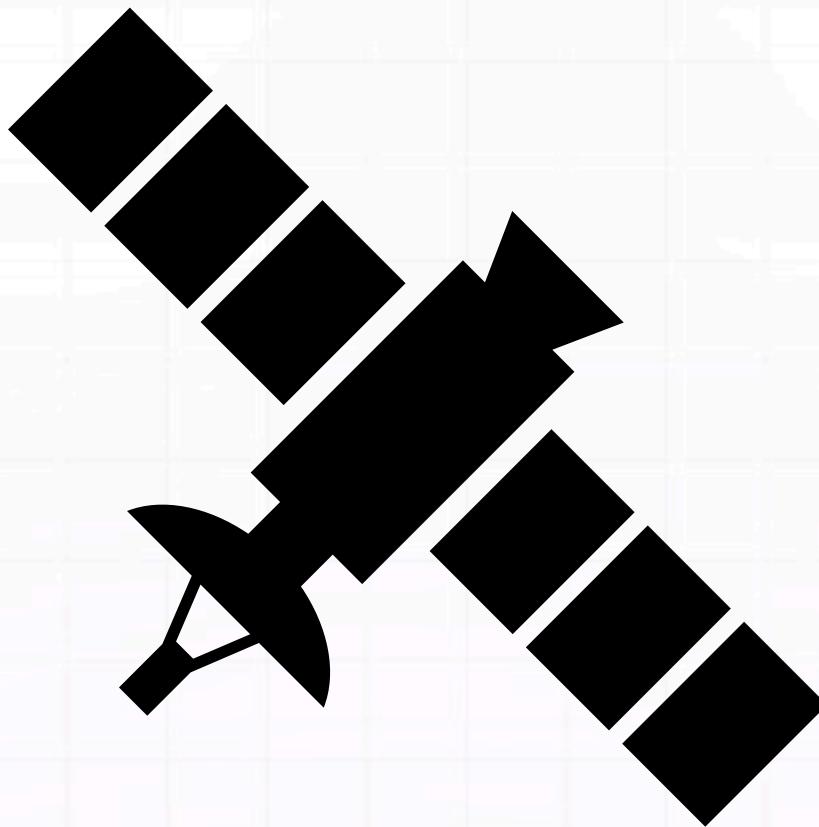
```
import "@hotwired/turbo-rails"
```

```
import "controllers"
```

```
import "confsmith"
```

IMPORTMAPS AND PROPSHAFT

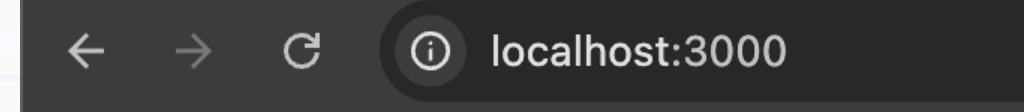
HOW DO THEY WORK TOGETHER?



Importmaps requests the assets

Propshaft sends them to the browser

```
<link rel="stylesheet" href="/assets/application-8b441ae0.css" data-turbo-track="reload" />
<script type="importmap" data-turbo-track="reload">{
"imports": {
  "application": "/assets/application-98e8f90d.js",
  "@hotwired/turbo-rails": "/assets/turbo.min-3a2e143f.js",
  "@hotwired/stimulus": "/assets/stimulus.min-4b1e420e.js",
  "@hotwired/stimulus-loading": "/assets/stimulus-loading-1fc53fe7.js",
  "controllers/application": "/assets/controllers/application-3affb389.js",
  "controllers/hello_controller": "/assets/controllers/hello_controller-708796bd.js",
  "controllers": "/assets/controllers/index-ee64e1f1.js"
}
}</script>
<link rel="modulepreload" href="/assets/application-98e8f90d.js">
<link rel="modulepreload" href="/assets/turbo.min-3a2e143f.js">
<link rel="modulepreload" href="/assets/stimulus.min-4b1e420e.js">
<link rel="modulepreload" href="/assets/stimulus-loading-1fc53fe7.js">
<link rel="modulepreload" href="/assets/controllers/application-3affb389.js">
<link rel="modulepreload" href="/assets/controllers/hello_controller-708796bd.js">
<link rel="modulepreload" href="/assets/controllers/index-ee64e1f1.js">
<script type="module">import "application"</script>
```



<https://radanskoric.com/articles/rails-assets-propshaft-importmaps>

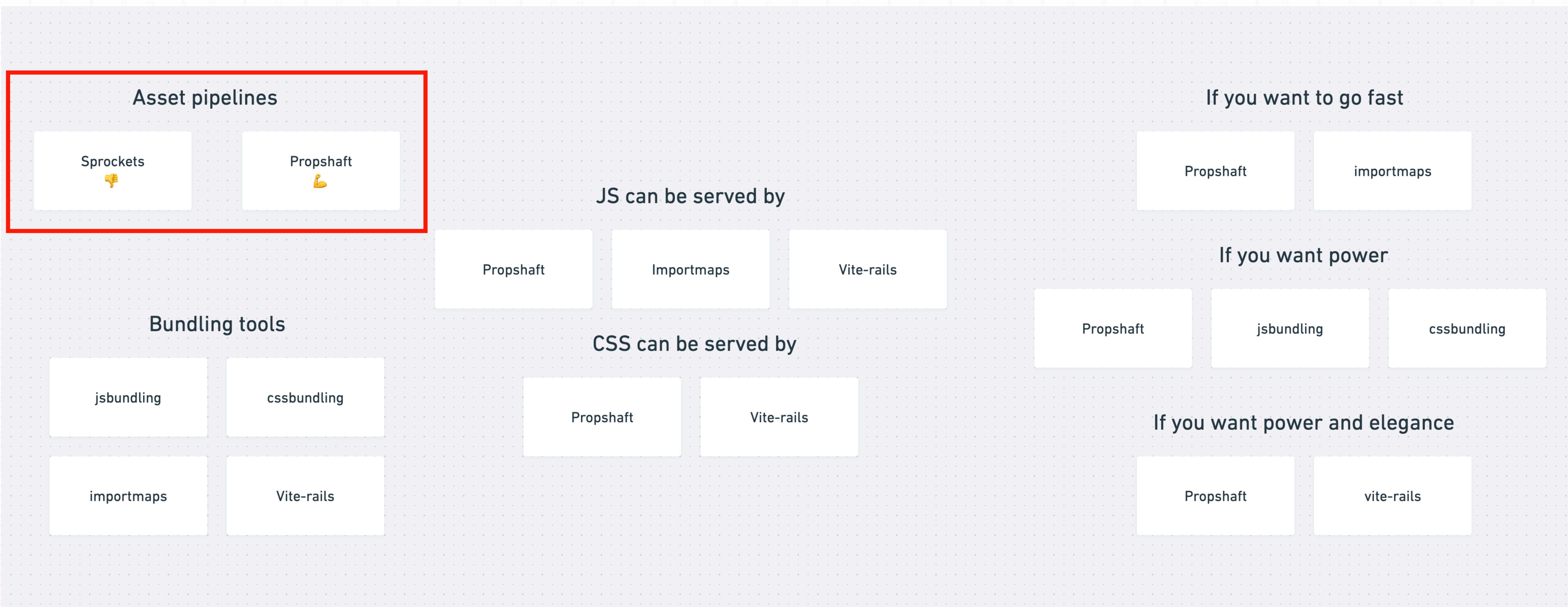
JS AND CSS BUNDLING GEMS

- Bundle using existing libraries
- For CSS: TailwindCSS, Bulma, others
- For JS: esbuild, bun, rollup, webpack
- They give you options

OTHER TOOLS

- **Shakapacker (previously Webpacker)**
- **vite-rails**

OVERVIEW



OVERVIEW

Asset pipelines

Sprockets
👎

Propshaft
💪

JS can be served by

Propshaft

Importmaps

Vite-rails

If you want to go fast

Propshaft

importmaps

Bundling tools

jsbundling

cssbundling

importmaps

Vite-rails

CSS can be served by

Propshaft

Vite-rails

If you want power

Propshaft

jsbundling

cssbundling

If you want power and elegance

Propshaft

vite-rails

OVERVIEW

Asset pipelines

Sprockets
👎

Propshaft
💪

JS can be served by

Propshaft

Importmaps

Vite-rails

Bundling tools

jsbundling

cssbundling

Propshaft

Vite-rails

importmaps

Vite-rails

If you want to go fast

Propshaft

importmaps

If you want power

Propshaft

jsbundling

cssbundling

If you want power and elegance

Propshaft

vite-rails

OVERVIEW

Asset pipelines

Sprockets
👎

Propshaft
💪

JS can be served by

Propshaft

Importmaps

Vite-rails

If you want to go fast

Propshaft

importmaps

Bundling tools

jsbundling

cssbundling

importmaps

Vite-rails

CSS can be served by

Propshaft

Vite-rails

If you want power

Propshaft

jsbundling

cssbundling

If you want power and elegance

Propshaft

vite-rails

OVERVIEW

Asset pipelines

Sprockets
👎

Propshaft
💪

JS can be served by

Propshaft

Importmaps

Vite-rails

If you want to go fast

Propshaft

importmaps

Bundling tools

jsbundling

cssbundling

CSS can be served by

Propshaft

Vite-rails

If you want power

Propshaft

jsbundling

cssbundling

importmaps

Vite-rails

If you want power and elegance

Propshaft

vite-rails

OVERVIEW

Asset pipelines

Sprockets
👎

Propshaft
💪

JS can be served by

Propshaft

Importmaps

Vite-rails

Bundling tools

jsbundling

cssbundling

CSS can be served by

Propshaft

Vite-rails

importmaps

Vite-rails

If you want to go fast

Propshaft

importmaps

If you want power

Propshaft

jsbundling

cssbundling

If you want power and elegance

Propshaft

vite-rails

OVERVIEW

Asset pipelines

Sprockets
👎

Propshaft
💪

JS can be served by

Propshaft

Importmaps

Vite-rails

If you want to go fast

Propshaft

importmaps

Bundling tools

jsbundling

cssbundling

importmaps

Vite-rails

CSS can be served by

Propshaft

Vite-rails

If you want power

Propshaft

jsbundling

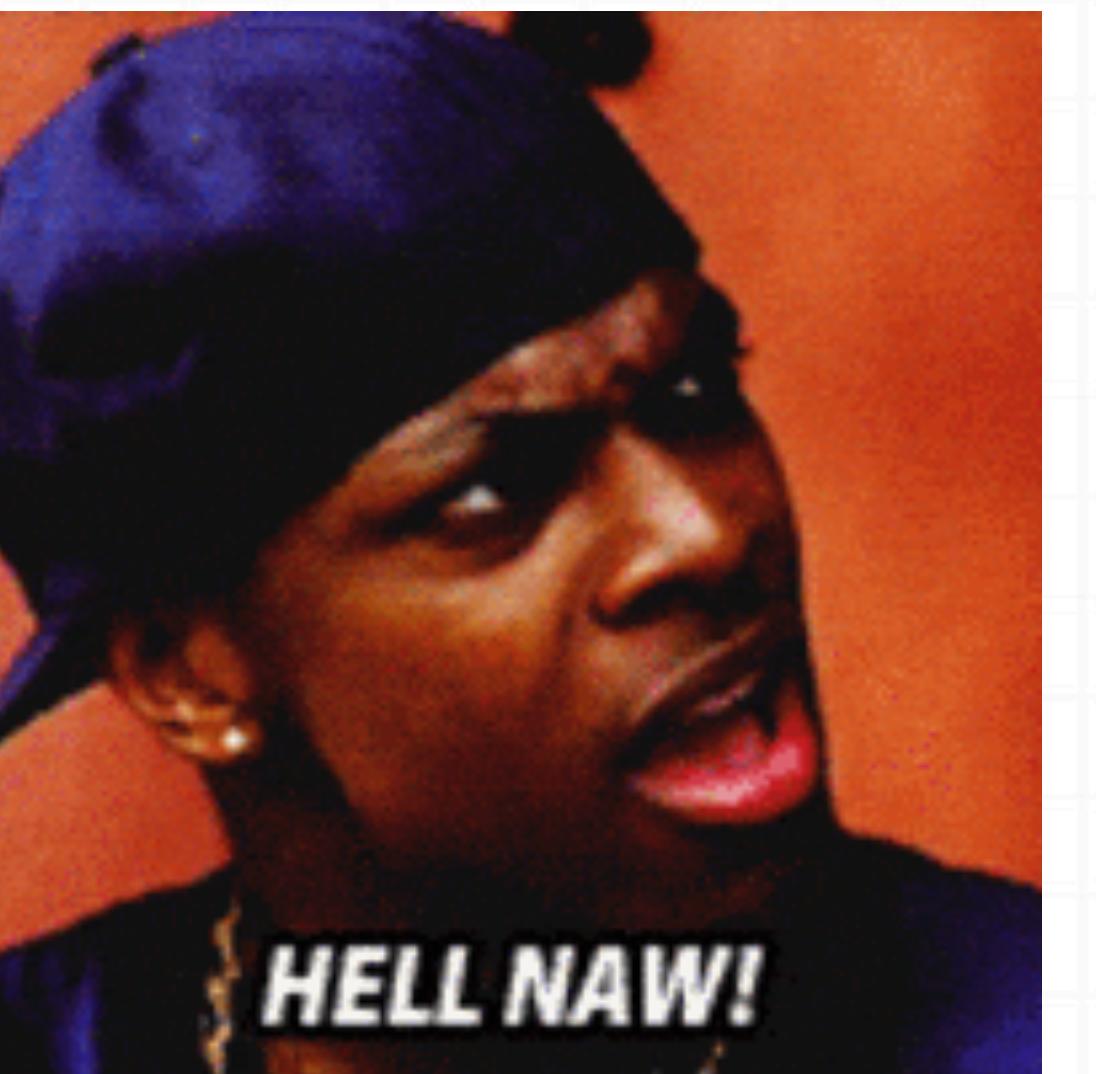
cssbundling

If you want power and elegance

Propshaft

vite-rails

THAT'S IT FOLKS!



HOW CAN WE LEARN TO BUILD RAILS ASSET PIPELINE EFFECTIVELY?



WHAT ARE WE BUILDING?



CONFSMITH EDITOR

REQUIREMENTS

- the user has to add a new gem to the parent app
- the user may use sprockets or propshaft
- the user may or may not have importmaps installed
- the user may or may not have node installed
- the user may or may not have a build step
- the user may want browser-safe JavaScript
- the user might need to import some other assets using yarn or npm
- we might want to use TypeScript, JSX, or other types of code that needs transpiling
- the user needs browser caching and fingerprinting
- the user might have common internal dependencies and we don't want to have conflicts with the parent app



IMPORTMAPS

```
# confsmith/config/importmap.rb
pin "confsmith"

# confsmith/lib/confsmith/engine.rb
module Confsmith
  class Engine < ::Rails::Engine
    isolate_namespace Confsmith

    # Add our own importmap config to the main app
    initializer "confsmith.importmap", before: "importmap" do |app|
      if Rails.application.respond_to?(:importmap)
        app.config.importmap.paths << Engine.root.join("config/importmap.rb")
      end
    end

    # Add our the javascript directory to the assets path
    initializer "confsmith.assets.precompile" do |app|
      if Rails.application.config.respond_to?(:assets)
        app.config.assets.paths << Engine.root.join("app/javascript/").to_s
      end
    end
  end
end
```

```
// app/javascript/confsmith.js
console.log('hi from confsmith')
```

REQUIREMENTS

- the user has to add a new gem to the parent app
- the user may use sprockets or propshaft
- the user may or may not have importmaps installed
- the user may or may not have node installed
- the user may or may not have a build step
- the user may want browser-safe JavaScript
- the user might need to import some other assets using yarn or npm
- we might want to use TypeScript, JSX, or other types of code that needs transpiling
- the user needs browser caching and fingerprinting
- the user might have common internal dependencies and we don't want to have conflicts with the parent app

WHAT ABOUT NPM PACKAGES?

- 1. We bundle it in the gem**
- 2. We tell the user to pin them**

BUNDLE THEM OURSELVES

SOLUTION 1.

```
# confsmith/config/importmap.rb
pin "confsmith"
pin "stimulus-confetti" # @1.0.1
pin "canvas-confetti" # @1.9.3

▶ ./bin/importmap
Pinned
npm:st
Pinned
npm:ca
initiali
  if Rai
    app.
    app.
    look
  end
end
}

<script type="importmap" data-turbo-track="reload">
  imports: {
    "confsmith": "/assets/confsmith-4ced8629.js", // ← added by our gem
    "stimulus-confetti": "/assets/stimulus-confetti-50236747.js", // ← added by our gem
    "canvas-confetti": "/assets/canvas-confetti-d9346ae1.js", // ← added by our gem
    "application": "/assets/application-98e8f90d.js",
    "@hotwired/turbo-rails": "/assets/turbo.min-3a2e143f.js",
    "@hotwired/stimulus": "/assets/stimulus.min-4b1e420e.js",
    "@hotwired/stimulus-loading": "/assets/stimulus-loading-1fc53fe7.js",
    "controllers/application": "/assets/controllers/application-3affb389.js",
    "controllers/hello_controller": "/assets/controllers/hello_controller-708796bd.js",
    "controllers": "/assets/controllers/index-ee64e1f1.js"
  }
}</script>
```

<https://ga.jspm.io/>

[://ga.jspm.io/](https://ga.jspm.io/)

if to

TELL THE USER TO PIN IT IN THEIR APP

SOLUTION 2.

```
```bash
./bin/importmap pin stimulus-confetti
````
```

REQUIREMENTS

- the user has to add a new gem to the parent app
- the user may use sprockets or propshaft
- the user may or may not have importmaps installed
- the user may or may not have node installed
- the user may or may not have a build step
- the user might need to import some other assets using yarn or npm
- we might want to use TypeScript, JSX, or other types of code that needs transpiling
- the user needs browser caching and fingerprinting
- the user might have common internal dependencies and we don't want to have conflicts with the parent app

HAVE A BUILD STEP IN THE GEM

```
bundle add jsbundling-rails

cd test/dummy

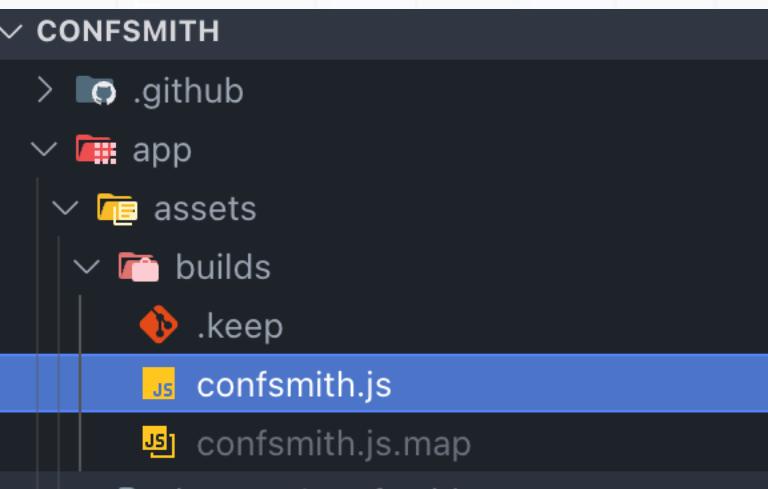
rails javascript:install:esbuild

# Manually move the generated files from the dummy app to the gem
# Manually move the generated files from the dummy app to the gem

▶ yarn build
yarn run v1.22.22
$ esbuild app/javascript/*.* --bundle --sourcemap --format=esm --outdir=app/assets/builds --public-path=/assets

  app/assets/builds/confsmith.js      32.9kb
  app/assets/builds/confsmith.js.map  128.0kb

✨ Done in 0.25s.
```



```
✓ CONFSMITH
  > .github
  ✓ app
    ✓ assets
      ✓ builds
        .keep
        confsmith.js
        confsmith.js.map
```

```
initializer "confsmith.assets.precompile" do |app|
  if Rails.application.config.respond_to?(:assets)
    app.config.assets.paths << Engine.root.join("app/javascript/").to_s
    app.config.assets.paths << Engine.root.join("vendor/javascript/").to_s
  end
end
```

REQUIREMENTS

- the user has to add a new gem to the parent app
- the user may use sprockets or propshaft
- the user may or may not have importmaps installed
- the user may or may not have node installed
- the user may or may not have a build step
- the user might need to import some other assets using yarn or npm
- we might want to use TypeScript, JSX, or other types of code that needs transpiling
- the user needs browser caching and fingerprinting
- the user might have common internal dependencies and we don't want to have conflicts with the parent app

JS BUNDLING

```
'rails new APP_NAME --javascript=esbuild'
```

```
// app/javascript/application.js
```

```
import "confsmith"
```

```
// use it here
```

CREATE AN NPM PACKAGE

SOLUTION 1.

The image shows a terminal window on the left and a GitHub package page on the right. The terminal output shows the command `npm publish` being run, with several npm warning messages about publishing to a local repository. The GitHub page for the package `confsmith` is displayed, showing the package details and a link to the source code.

```
▶ npm publish
npm warn public registry is deprecated, please switch to https://npmjs.com.
npm warn public run "npm pkg"
npm warn public registry is deprecated, please switch to https://npmjs.com.
npm warn public registry is deprecated, please switch to https://npmjs.com.
npm warn public registry is deprecated, please switch to https://npmjs.com.

{
  "name": "confsmith",
  "version": "1.0.0",
  "main": "index.js",
  "repository": "git@github.com:avo-hq/confsmith.git",
  "author": "Adrian <adrian@adrianthedev.com>",
  "license": "MIT",
  "description": "Dummy editor package.",
  "homepage": "https://github.com/avo-hq/confsmith",
  "module": "app/assets/builds/confsmith.js",
  "bugs": {
    "url": "https://github.com/avo-hq/confsmith/issues"
  },
  "publishConfig": {
    "access": "public"
  }
}

Readme Code (Beta) Public access
```

shing. Please
vo-hq/confsmi

Confsmith

REQUIREMENTS

- ✓ the user has to add a new gem to the parent app
- ✓ the user may use sprockets or propshaft
- ✓ the user may or may not have importmaps installed
- ✓ the user may or may not have node installed
- ✓ the user may or may not have a build step
- ✓ the user might need to import some other assets using yarn or npm
- ✓ we might want to use TypeScript, JSX, or other types of code that needs transpiling
- ✓ the user needs browser caching and fingerprinting
- ✓ the user might have common internal dependencies and we don't want to have conflicts with the parent app

A BUNCH

CHES

```
namespace :confsmith do
  desc "Symlink confsmith.js from the gem to vendor/javascript"
  task expose_js: :environment do
    require 'fileutils'

    # Find the confsmith gem specification
    gem_spec = Bundler.rubygems.find_name('confsmith').first

    if gem_spec.nil?
      puts "Error: confsmith gem not found. Make sure it's installed."
      exit 1
    end

    # Build the source path
    source_file = File.join(gem_spec.full_gem_path, 'app', 'assets', 'builds', 'confsmith.js')

    unless File.exist?(source_file)
      puts "Error: confsmith.js not found at #{source_file}"
      exit 1
    end

    # Create target directory if it doesn't exist
    target_dir = Rails.root.join('vendor', 'javascript')
    FileUtils.mkdir_p(target_dir)

    # Remove existing file or symlink if it exists
    target_file = target_dir.join('confsmith.js')
    FileUtils.rm_f(target_file) if File.exist?(target_file)

    # Create the symlink
    FileUtils.ln_s(source_file, target_file)

    puts "Successfully symlinked confsmith.js to #{target_file}"
  end
end
```

INJECT THE JS FILE YOURSELF

```
←!— app/views/layouts/application.html.erb —→  
<%= javascript_include_tag "confsmith", "data-turbo-track": "reload", type: "module" %>
```

https://github.com/avo-hq/avo/blob/main/app/views/avo/partials/_javascript.html.erb#L15

<https://github.com/avo-hq/avo/blob/main/app/javascript/late-registration.js>

https://github.com/avo-hq/avo/blob/main/app/components/avo/asset_manager/javascript_component.html.erb

```
<!-- app/view/layouts/application.html.erb -->
<script>
  let Config = {}
  Config.stimulus_controllers = []
</script>

<!-- inject your gem's js here -->
<%= javascript_include_tag "confsmith_controllers", "data-turbo-track": "reload", type: "module" %>

<script>
  Config.stimulus_controllers.push(['confsmith', 'confsmith_controller'])
</script>

<script>
  import application from "application"

  window.Config.stimulus_controllers.forEach(([name, controller]) => {
    application.register(name, controller)
  })
</script>

<%= javascript_include_tag "application", "data-turbo-track": "reload", type: "module" %>

<script>
  // register the rest of the controllers and bootstrap the app
</script>
```

```
<!-- app/view/layouts/application.html.erb -->  
<script>  
  let Config = {}  
  Config.stimulus_controllers = []  
</script>  
  
<!-- inject your gem's js here --&gt;<br/><%= javascript_include_tag "confsmith_controllers", "data-turbo-track": "reload", type: "module" %>  
  
<script>  
  Config.stimulus_controllers.push(['confsmith', 'confsmith_controller'])  
</script>  
  
<script>  
  import application from "application"  
  
  window.Config.stimulus_controllers.forEach(([name, controller]) => {  
    application.register(name, controller)  
  })  
</script>  
  
<%= javascript_include_tag "application", "data-turbo-track": "reload", type: "module" %>  
  
<script>  
  // register the rest of the controllers and bootstrap the app  
</script>
```

```
<!-- app/view/layouts/application.html.erb -->

<script>
  let Config = {}
  Config.stimulus_controllers = []
</script>

<!-- inject your gem's js here -->
<%= javascript_include_tag "confsmith_controllers", "data-turbo-track": "reload", type: "module" %>

<script>
  Config.stimulus_controllers.push(['confsmith', 'confsmith_controller'])
</script>

<script>
  import application from "application"

  window.Config.stimulus_controllers.forEach(([name, controller]) => {
    application.register(name, controller)
  })
</script>

<%= javascript_include_tag "application", "data-turbo-track": "reload", type: "module" %>

<script>
  // register the rest of the controllers and bootstrap the app
</script>
```

```
<!-- app/view/layouts/application.html.erb -->

<script>
  let Config = {}
  Config.stimulus_controllers = []
</script>

<!-- inject your gem's js here -->
<%= javascript_include_tag "confsmith_controllers", "data-turbo-track": "reload", type: "module" %>

<script>
  Config.stimulus_controllers.push(['confsmith', 'confsmith_controller'])
</script>

<script>
  import application from "application"

  window.Config.stimulus_controllers.forEach(([name, controller]) => {
    application.register(name, controller)
  })
</script>

<%= javascript_include_tag "application", "data-turbo-track": "reload", type: "module" %>

<script>
  // register the rest of the controllers and bootstrap the app
</script>
```

```
<!-- app/view/layouts/application.html.erb -->

<script>
  let Config = {}
  Config.stimulus_controllers = []
</script>

<!-- inject your gem's js here -->
<%= javascript_include_tag "confsmith_controllers", "data-turbo-track": "reload", type: "module" %>

<script>
  Config.stimulus_controllers.push(['confsmith', 'confsmith_controller'])
</script>

<script>
  import application from "application"

  window.Config.stimulus_controllers.forEach(([name, controller]) => {
    application.register(name, controller)
  })
</script>

<%= javascript_include_tag "application", "data-turbo-track": "reload", type: "module" %>

<script>
  // register the rest of the controllers and bootstrap the app
</script>
```

USE A MIDDLEWARE

```
"prod:build:js": "esbuild app/javascript/*.js --bundle --sourcemap --minify --outdir=public/avo-assets",
```



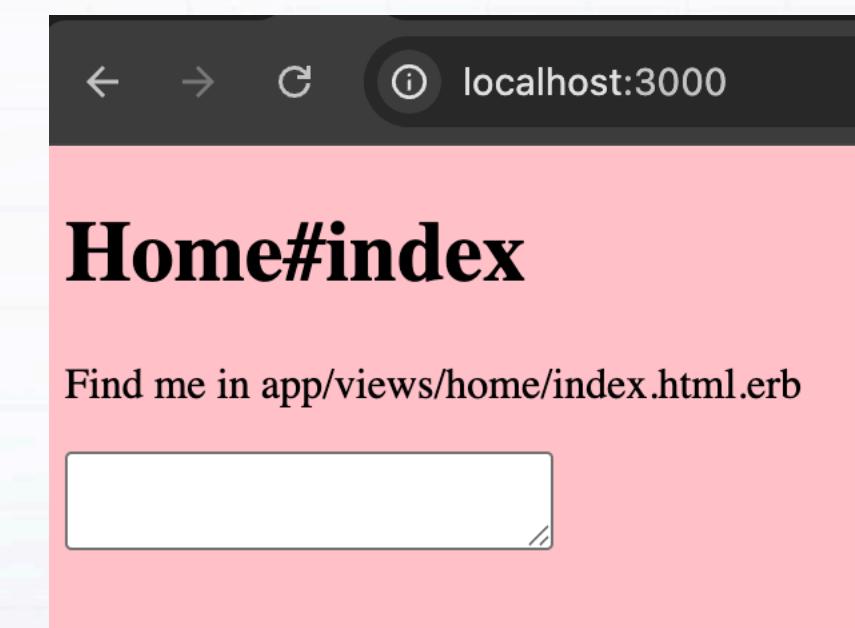
```
module Avo
  class Engine < ::Rails::Engine
    isolate_namespace Avo

    config.app_middleware.use(
      Rack::Static,
      urls: ["/avo-assets"],
      root: Avo::Engine.root.join("public")
    )
  end
end
```

WHAT ABOUT CSS?

```
←!— app/views/layouts/application.html.erb —→  
<%= stylesheet_link_tag :app, "data-turbo-track": "reload" %>  
<%= stylesheet_link_tag 'confsmith/application', "data-turbo-track": "reload" %>
```

```
/* confsmith/app/assets/stylesheets/confsmith/application.css */  
body {  
  background: pink;  
}
```

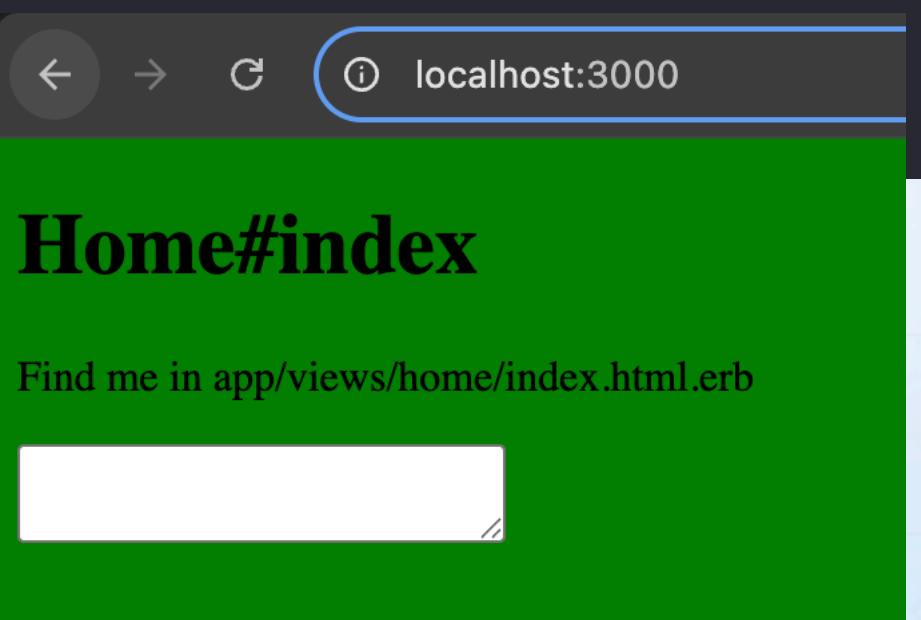


CSS IMPORTS SUPPORTED

```
←!— app/views/layouts/application.html.erb —→  
<%= stylesheet_link_tag :app, "data-turbo-track": "reload" %>  
<%= stylesheet_link_tag 'confsmith/application', "data-turbo-track": "reload" %>  
  
/* confsmith/app/assets/stylesheets/confsmith/application.css */  
@import url('./reset.css');
```



```
body {  
  background: pink;  
}  
  
/* confsmith/app/assets/stylesheets/confsmith/reset.css */  
body {  
  background: green !important;  
}
```



SEE YOU AT HACK SPACES TOMORROW

The image shows a dark-themed GitHub README page. At the top left is a 'README' button with a file icon. At the top right are edit and more options icons. The main title 'Railsconf '25 Avo Hack Space' is displayed in large white font. Below the title is a short paragraph: 'Thanks for wanting to get involved with Open Source through project we do at Avo.' Further down, there's information about the hack space: 'Location: Banquet Hall Hack Space Station: Yellow Station Schedule Time: July 9th, 2025 — 2:30PM ~ 5:00PM'. A note follows: 'As a reminder, Avo started out as an admin panel framework and slowly evolved into a complete toolkit to develop internal tools and operational software for your entire team.' Another paragraph states: 'We develop commercial software but our roots are with Open-source. We have a bunch of initiatives that we're supporting through [Open-source](#), [sponsorships](#), [events](#) and other ways.' A section titled 'Open Source' is present, followed by a list of open-source projects: 'Some of our more popular Open Source projects are [avo](#), [marksmith](#), [class_variants](#), [stimulus-confetti](#), [prop_initializer](#) and [active_storage-blurhash](#)'.

README

Railsconf '25 Avo Hack Space

Thanks for wanting to get involved with Open Source through project we do at Avo.

Location: Banquet Hall Hack Space Station: Yellow Station Schedule Time: July 9th, 2025 — 2:30PM ~ 5:00PM

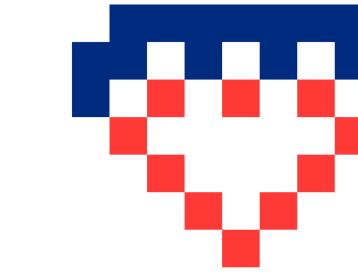
As a reminder, Avo started out as an admin panel framework and slowly evolved into a complete toolkit to develop internal tools and operational software for your entire team.

We develop commercial software but our roots are with Open-source. We have a bunch of initiatives that we're supporting through [Open-source](#), [sponsorships](#), [events](#) and other ways.

Open Source

Some of our more popular Open Source projects are [avo](#), [marksmith](#), [class_variants](#), [stimulus-confetti](#), [prop_initializer](#) and [active_storage-blurhash](#).

FRIENDLY.RB COLLECTIBLE AVAILABLE AT THE SILENT AUCTION



THANKS A BUNCH!

Confsmith: <https://github.com/avo-hq/confsmith>

Marksmith: <https://github.com/avo-hq/marksmith>

Avo: <https://github.com/avo-hq/avo>

<https://bsky.app/profile/adrianthedev.com>

<https://linkedin.com/in/adrianthedev>

<https://friendlyrb.com>

