

Playing with: Derailed Benchmark Gem

How much memory does gems in Rails use

about

def about

- Rails Software Engineer  
- @ghosteathuman

end

start of the journey

- Question:
 - How much memory does each gem in Rails use?

rails gems

- Action Cable: WebSockets Integration
- Action Controller: Controller in MVC
- Action Mailbox: Receive emails
- Action Mailer: Send emails
- Action View: View in MVC
- Action Text: Rich text content and editing
- Active Job: Jobs & Queuing
- Active Record: Model in MVC
- Active Storage: File & Storage Management
- Active Support: Ruby Extensions & Utilities

derailed benchmarks gem

- Benchmarking gem
- Static Benchmark
 - Get memory information from Gemfile without having to boot the app
- Dynamic Benchmark
 - Boot Rails app and run benchmarks again it

let's start

- Created new Rails 7 project with everything default
- Add derailed benchmarks gem and run a static benchmark

now let's compare

- Created new Rails 7 project with everything default **BUT**
 - skip Action Mailbox (using '—skip-action-mailbox' option in 'rails new')
- Add derailed benchmarks gem and run a static benchmark

what I expected to happen

- Without Action Mailbox, less memory is used

what actually happened

- There were no changes to the benchmark results
 - Action Mailbox was still showing up in the benchmark results

what?



so apparently

- derailed benchmark static benchmark looks at Gemfile / Gemfile.lock
- Skipping Action Mailbox doesn't remove it from Gemfile.lock
- Instead, it is commented out in 'applications.rb'

application.rb



```
require "rails"

# Pick the frameworks you want:
require "active_model/railtie"
require "active_job/railtie"
require "active_record/railtie"
require "active_storage/engine"
require "action_controller/railtie"
require "action_mailer/railtie"
# require "action_mailbox/engine"
require "action_text/engine"
require "action_view/railtie"
require "action_cable/engine"
require "sprockets/railtie"
require "rails/test_unit/railtie"
```

fine

- Let's try dynamic benchmarking now
- It's better than static benchmarking for sure, but:
 - Dynamic benchmarking hits a specific endpoint (can be configured using 'PATH_TO_HIT')
 - Which means the result obtained is path dependent.
 - What about background jobs?

wrapping up

- Benchmarking is pretty cool and pretty complex too. Look into it!
- Application Performance Monitoring (APM) might be better to obtain information from production

references

- Derailed Benchmarks GitHub Repository: https://github.com/zombocom/derailed_benchmarks
- Rails 7 Default Benchmark: <https://gist.github.com/ghosteathuman/3957c14cc521b9afe68717960ccdda63>
- Stack Overflow Question: <https://stackoverflow.com/questions/64364178/rails-6-how-to-disable-action-mailbox-and-action-text-from-loading-into-memory>

thanks!