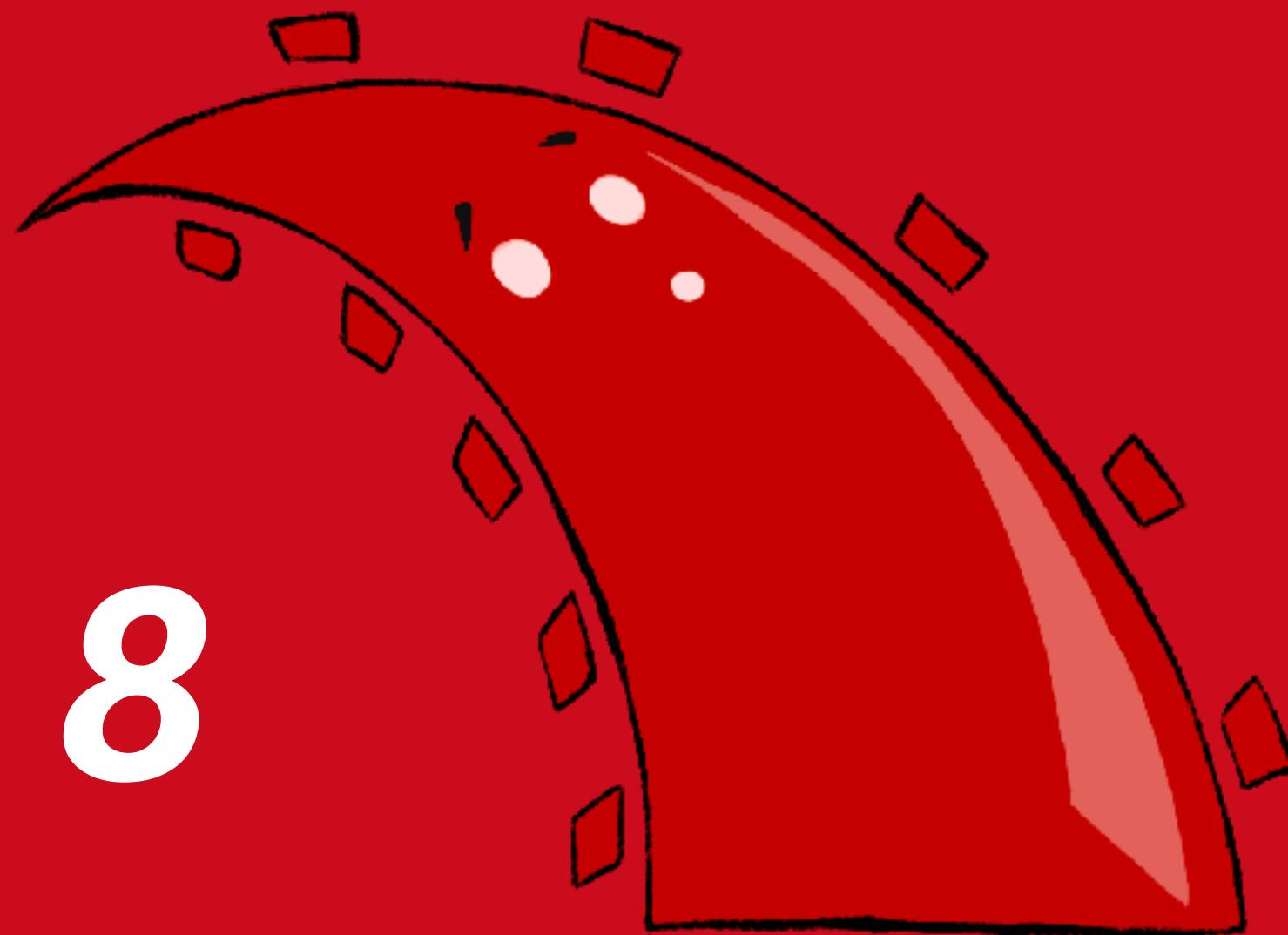


An Upgrade Handbook to Rails 8



Rails World 2024

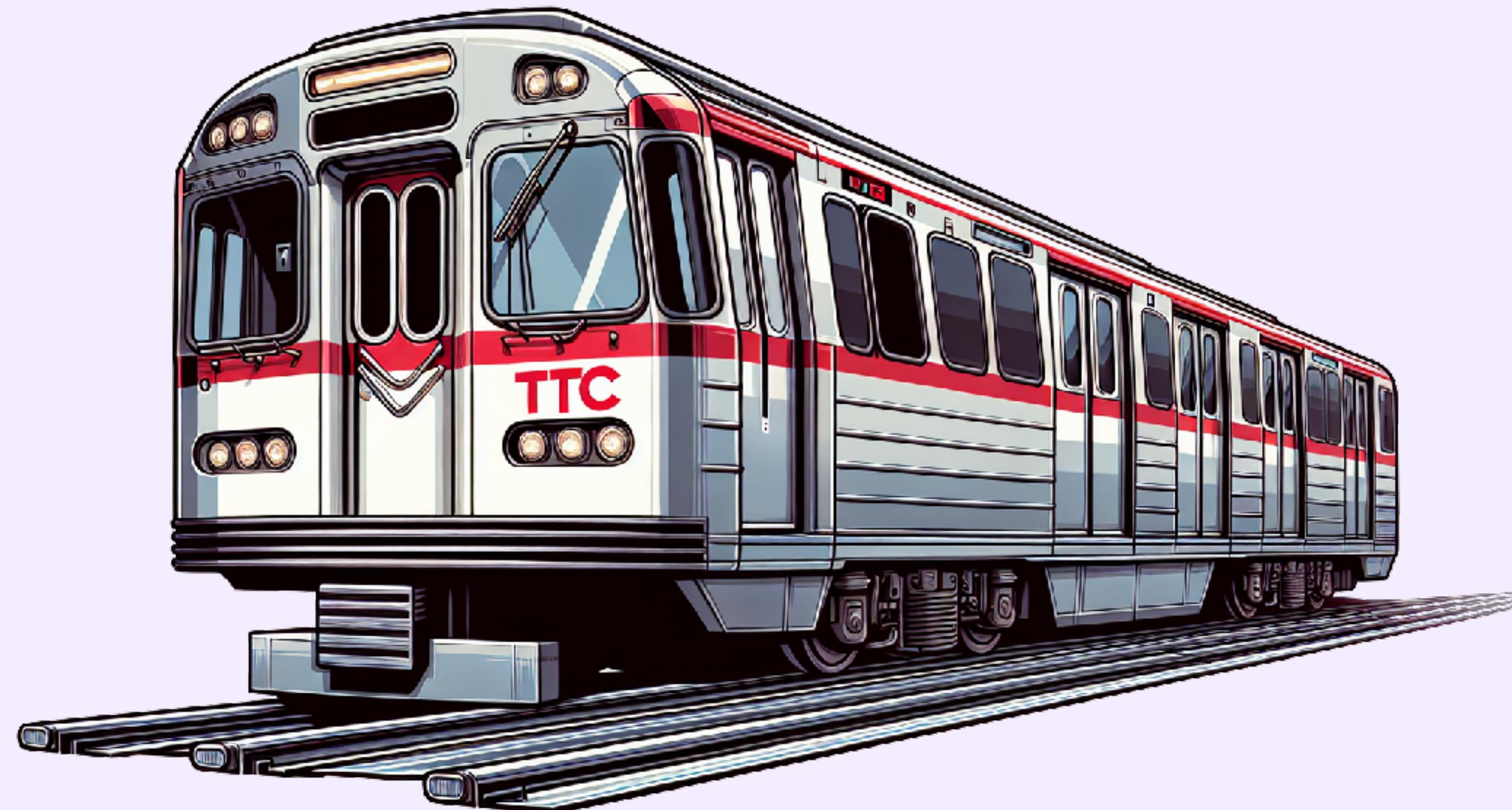
TORONTO

Population of 3 million

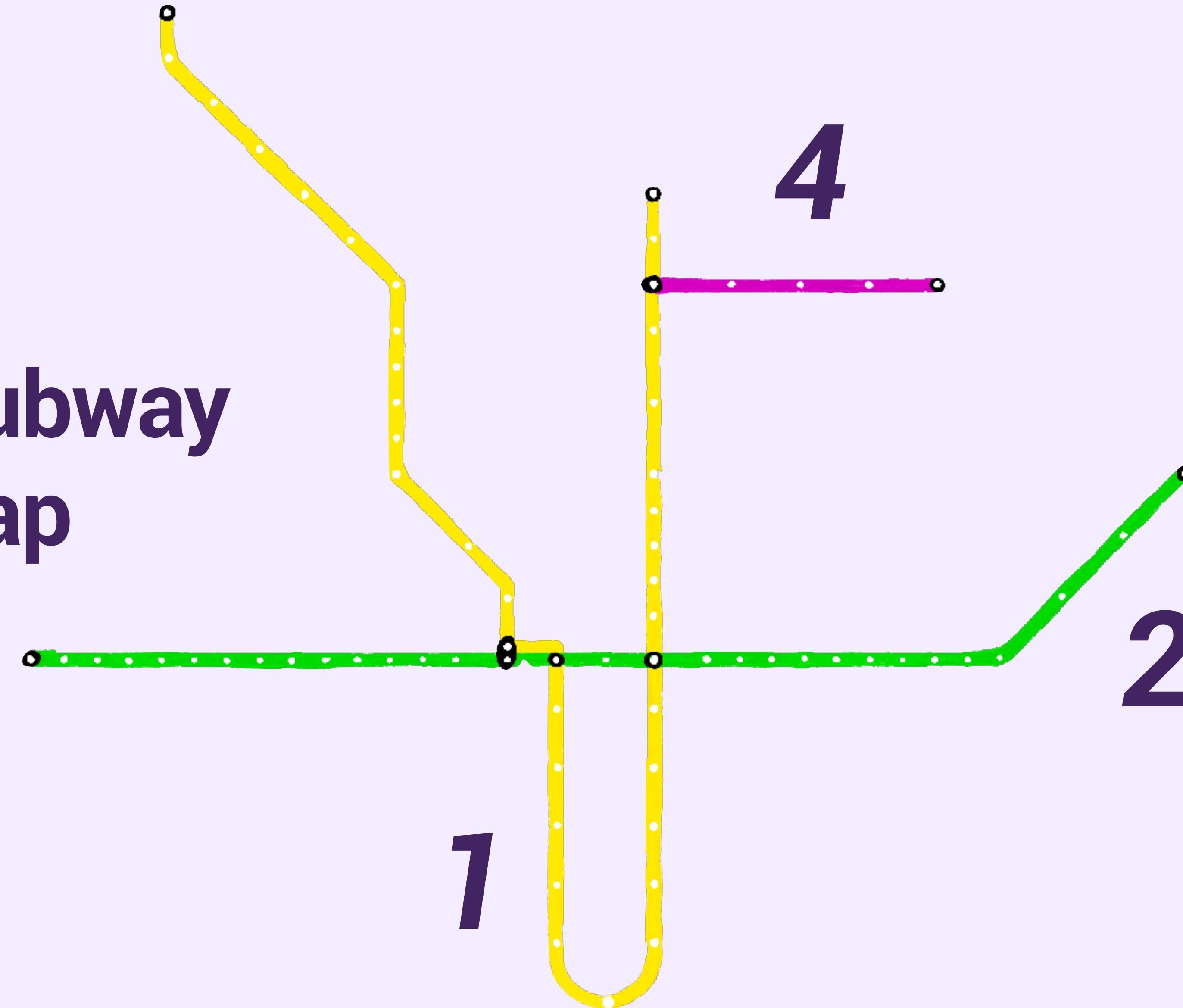


city of
TORONTO

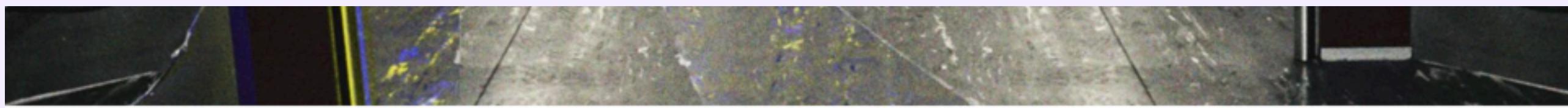
600k
Rides per day



TTC Subway Map



Let's not talk about what happened to Line 3...



WHO BROKE THE TTC?

Toronto used to have one of the best transit systems in North America. Now it's overcrowded, underfunded, unreliable and dangerous. The inside story of what went wrong and who's to blame

BY STEPHEN SPENCER DAVIS | JULY 6, 2023

turkishmelo0
19 Followers · 3430 Likes

See all videos

00:06 / 00:06

turkishmelo0 mali · 5-13

Follow

The worst transportation system ever #ttc #toronto #ttcjourney #ttctiktok #canada

Breaking News - Breaking News

The TTC Represents Everything Wrong with Toronto — and Capitalism

Matthew Puddister · [Follow](#)
11 min read · Oct 13, 2023

161



20k

**Subway delays
annually**

20k

**Subway delays
annually**

50

**Subway delays
per day**

20k

**Subway delays
annually**

50

**Subway delays
per day**

14%

**Of rides
considered
on time**

Why is the TTC so awful?

Hold that thought

Toronto Rail





I'm Jenny



**Replacement
Shuttle**



CENSORED



HYPED

The hype things we're going to cover



The hype things we're going to cover 😊

1. Rails upgrade process and the benefits

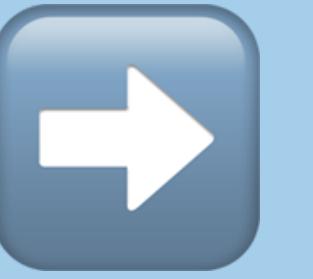
The hype things we're going to cover 😊

1. Rails upgrade process and the benefits
2. Upgrading to Rails 8

The hype things we're going to cover 😊

1. Rails upgrade process and the benefits
2. Upgrading to Rails 8
3. How Shopify upgrades at scale

Slides



How to upgrade Rails

guides.rubyonrails.org/

The screenshot shows a dark-themed version of the Ruby on Rails Guides website. At the top left is the "Rails Guides" logo with a red circular icon containing a white "R". To its right is the word "Guides" in white, followed by a dropdown menu showing "Version: Edge". On the far right of the header are links for "Home", "Guides Index", and "Contribute". The main title "Upgrading Ruby on Rails" is centered above a descriptive paragraph. Below the title is a section titled "1 General Advice" with a sub-section "1.1 Test Coverage" and a detailed paragraph about test coverage. To the right of the main content is a sidebar titled "Chapters" with a list of upgrade guides, each with a small icon and a brief description.

Upgrading Ruby on Rails

This guide provides steps to be followed when you upgrade your applications to a newer version of Ruby on Rails. These steps are also available in individual release guides.

1 General Advice

Before attempting to upgrade an existing application, you should be sure you have a good reason to upgrade. You need to balance several factors: the need for new features, the increasing difficulty of finding support for old code, and your available time and skills, to name a few.

1.1 Test Coverage

The best way to be sure that your application still works after upgrading is to have good test coverage before you start the process. If you don't have automated tests that exercise the bulk of your application, you'll need to spend time manually exercising all the parts that have changed. In the case of a Rails upgrade, that will mean every single piece of functionality in the application. Do yourself a favor and make sure your test coverage is good *before* you start an upgrade.

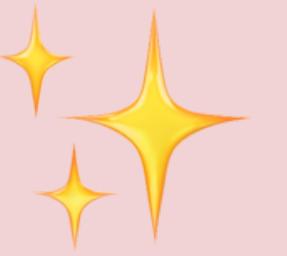
1.2 Ruby Versions

Chapters

- 1. General Advice**
 - Test Coverage
 - Ruby Versions
 - The Upgrade Process
 - The Update Task
 - Configure Framework Defaults
- 2. Upgrading from Rails 7.1 to Rails 7.2**
 - All tests now respect the `active_job.queue_adapter` config
- 3. Upgrading from Rails 7.0 to Rails 7.1**
 - Development and test environments `secret_key_base` file changed
 - Autoloaded paths are no longer in `$LOAD_PATH`
 - `configautoload.lib` and `configautoload.lib_once`

Prerequisites

Test coverage



Minimum Ruby version

Minimum Ruby version

3.1

1. Move to the latest patch version

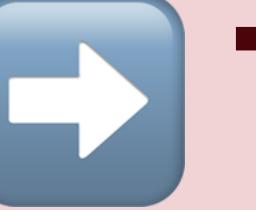
1. Move to the latest patch version

7.1.0  7.1.4

2. Fix tests and deprecated features

3. Move to the latest patch of the next minor

3. Move to the latest patch of the next minor

7.1.4  7.2.1

4. Run the update task

`bin/rails app:update`

How to upgrade Rails



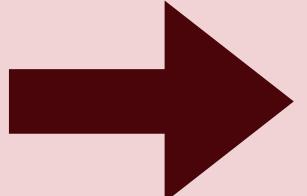
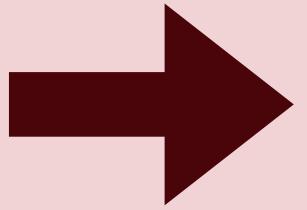
```
$ bin/rails app:update
      exist config
conflict config/application.rb
Overwrite /myapp/config/application.rb? (enter "h" for help) [Ynaqdh]
      force config/application.rb
      create config/initializers/new_framework_defaults_7_2.rb
```

5. Configure framework defaults

How to upgrade Rails

config/initializers/new_framework_defaults_7_2.rb

```
# Be sure to restart your server when you modify this file.  
#  
# This file eases your Rails 7.2 framework defaults upgrade.  
#  
# Uncomment each configuration one by one to switch to the new default.  
# Once your application is ready to run with all new defaults, you can remove  
# this file and set the `config.load_defaults` to `7.2`.  
#  
# Read the Guide for Upgrading Ruby on Rails for more info on each option.  
# https://guides.rubyonrails.org/upgrading\_ruby\_on\_rails.html  
  
###  
# Controls whether Active Job's `#perform_later` and similar methods automatically defer  
# the job queuing to after the current Active Record transaction is committed.  
#  
# Example:  
# Topic.transaction do  
#   topic = Topic.create(...)  
#   NewTopicNotificationJob.perform_later(topic)  
# end  
#  
# ...  
#+  
# Rails.application.config.active_job.enqueue_after_transaction_commit = :default  
  
###  
# Adds image/webp to the list of content types Active Storage considers as an image  
# Prevents automatic conversion to a fallback PNG, and assumes clients support WebP, as they support gif, jpeg, and png.  
# This is possible due to broad browser support for WebP, but older browsers and email clients may still not support  
# WebP. Requires imagemagick/libvips built with WebP support.  
#+  
# Rails.application.config.active_storage.web_image_content_types = %w[image/png image/jpeg image/gif image/webp]
```



How to upgrade Rails

config/application.rb



```
module MyApp
  class Application < Rails::Application
-    config.load_defaults(7.1)
+    config.load_defaults(7.2)
...

```

**Why upgrade Rails when a new version
comes out?**

Why is the TTC so awful?

Reduced Speed Zones

Effective September 10, 2024



TTC announces changes to 510 Spadina starting June 23 to accommodate major streetcar infrastructure upgrades

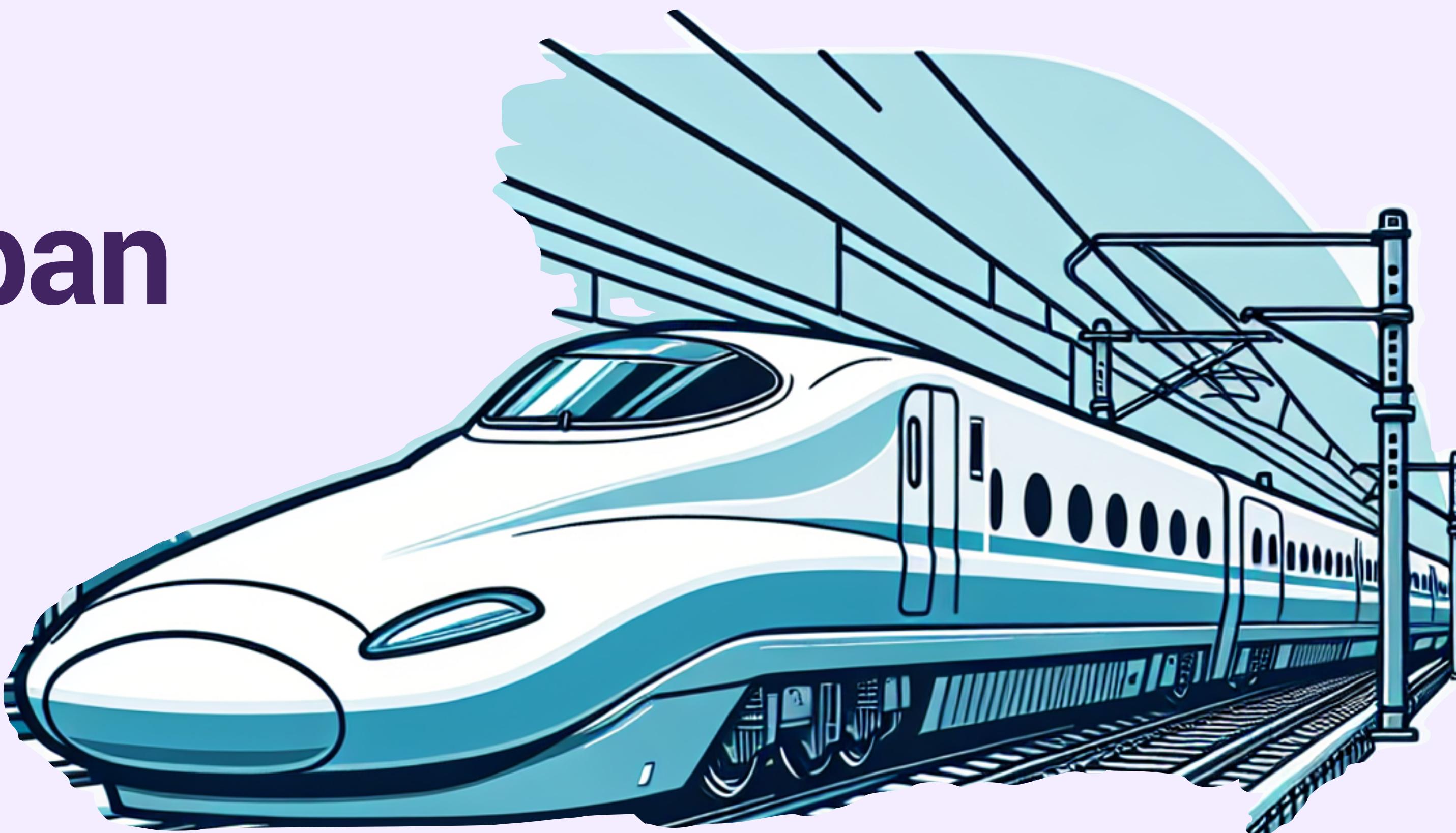
June 12, 2024

Lack of regular upgrades

Bandaid fixes

What does better look like?

Japan



Most efficient and reliable railways



WORLD CLASS Maintenance For JAPAN SHINKANSEN Bullet Train • Japan Railway Journal



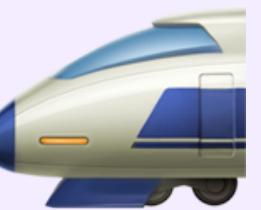
1st class City



=



1st class Infra



=



Happy citizens



Transit systems = Rails apps



Band-aids / Monkey patches

Extremely Outdated Rails App

■ rubyonrails-talk

Aug 2021

I'm in serious need of some help. I have an application running on Heroku on Ruby 2.3.8 and Rails 3.2. The stack on Heroku is no longer supported and needs major updating (my own fault for putting it off for so long). I can no longer push updates to my app until it is upgraded.

Upgrading would take a long long time. I've already started trying to get Rails 4.0 running after many hours am still struggling. My app has users and I'm afraid something is going to break and I wont be able to do anything about it. I've also looked at starting a fresh new app on the latest Rails version and trying to port my code but my Mac is old as well causing more issues. Any advice?



The logo consists of the word "Rails 8" in a bold, dark purple sans-serif font. Two yellow starburst icons, each containing three smaller yellow stars, flank the text. The starbursts are positioned on the left and right sides of the word.

Rails 8

Happy developers!

Why should you keep your Rails app up to date?

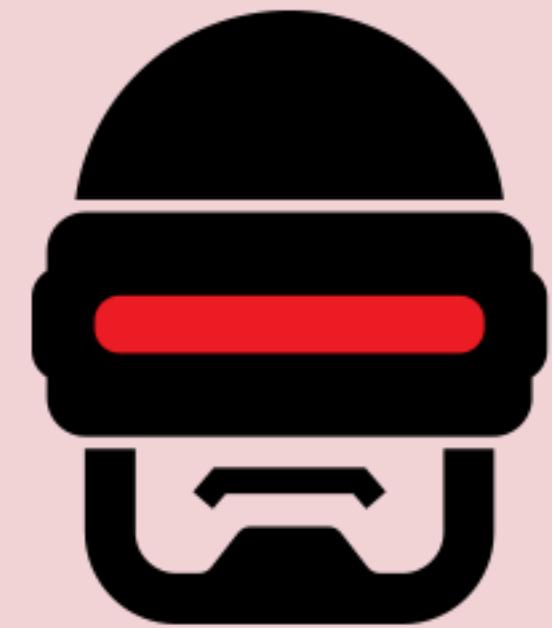
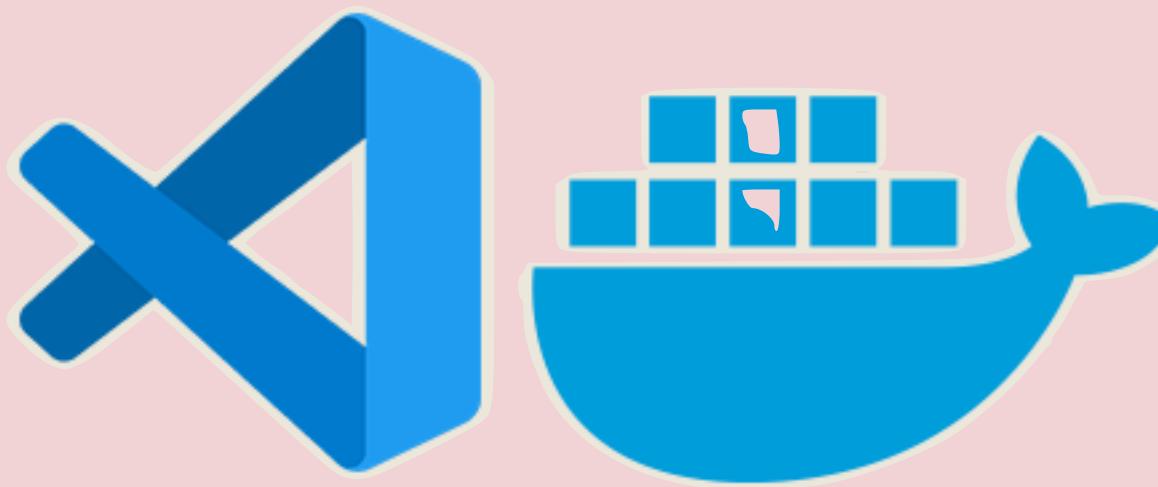
To NOT be the TTC of Rails apps

Why keep Rails apps up to date?

- 1. Keeps our apps safe and reliable**
- 2. Prevents costly large upgrades**
- 3. Promotes a first class dev experience**

Handbook to Rails 8

PWA



RuboCop

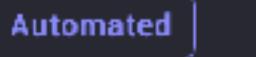
Let's upgrade an app!

Tracks TTC Service Alerts

[◀ Back](#)



[Follow](#) 

TTC Service Alerts
@ttcnotices@botsin.space
 Automated

Service advisories of the Toronto Transit Commission. twitter.com/ttcnotices

Bot by [@ndrwy](#)

JOINED
Mar 25, 2018

1.2K Posts 2 Following 243 Followers

[Posts](#) [Posts and replies](#) [Media](#)

 Replied to TTC Service Alerts

 **TTC Service Alerts** @ttcnotices 52s

47 Lansdowne Regular service has resumed near Caledonia Rd at Lawrence Ave West North Side.

 0    

 **TTC Service Alerts** @ttcnotices 12m

47 Lansdowne: Detour via Orfus Rd, Dufferin St and Glencairn Ave due to a collision.

 1    



😢 ⚡ TTC Service Alerts ⚡ 😢

47 Lansdowne Regular service has resumed near Caledonia Rd at Lawrence Ave West North Side.

Posted: 2 minutes ago

47 Lansdowne: Detour via Orfus Rd, Dufferin St and Glencairn Ave due to a collision.

Posted: 13 minutes ago

St Clair West: Elevator Back in service between St Clair Ave W north side entrance and bus and streetcar platform.

Posted: 22 minutes ago

66 Prince Edward: Detour northbound via Park Lawn Rd due to traffic congestion.

Posted: about 1 hour ago



🚨 😞 TTC Service Alerts 🚨 😞

Detour

Line 47 Lansdowne

Incident

47 Lansdowne: Detour via Orfus Rd, Dufferin St and Glencairn Ave due to a collision.

Posted 15 minutes ago

Resolution

47 Lansdowne Regular service has resumed near Caledonia Rd at Lawrence Ave West North Side.

Resolved 4 minutes ago

Event created: September 02, 2024 at 06:48 PM

Detour

Line 66 Prince

Incident

66 Prince Edward: Detour northbound via Park Lawn Rd due to traffic congestion.

Posted about 1 hour ago



Dashboard

Current Weather

Temperature: 20.2°C

Condition: Sunny

Should I wear a jacket? Nah, you're good!

Latest Service Alerts

Detour

about 1 hour ago

Line 66 Prince

66 Prince Edward: Detour northbound via Park Lawn Rd due to traffic congestion.

Detour

about 2 hours ago

Line 63 Ossington

63 Ossington: Detour via Ossington Ave, Queen St W and Shaw St due to Labour Day Parade.

Delay

about 3 hours ago

Line 501 Queen

501 Queen: Major delays near Queen St West at Dufferin St due to Labour Day Parade.

App is on 7.1.4



```
$ bundle update rails
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Using activesupport 7.2.1 (was 7.1.4)
Using activemodel 7.2.1 (was 7.1.4)
Using activejob 7.2.1 (was 7.1.4)
Using actionview 7.2.1 (was 7.1.4)
Using activerecord 7.2.1 (was 7.1.4)
Using actionpack 7.2.1 (was 7.1.4)
Using actioncable 7.2.1 (was 7.1.4)
Using activestorage 7.2.1 (was 7.1.4)
Using actionmailer 7.2.1 (was 7.1.4)
Using railties 7.2.1 (was 7.1.4)
Using actionmailbox 7.2.1 (was 7.1.4)
Using actiontext 7.2.1 (was 7.1.4)
Using rails 7.2.1 (was 7.1.4)
Bundle updated!
```

Deprecations and errors!

Extending Rails console

[rails/rails#51705](#)

“feeling lucky” command

jennyshen@Jennys-MacBook-Pro:~/toronto-on-rails

```
jennyshen@Jennys-MacBook-Pro ~/toronto-on-rails r main []
```

```
● ● ●  
  
module TorontoOnRails  
  class Application < Rails::Application  
    #...  
  
    console do  
      Rails::ConsoleMethods.include(Console::FeelingLucky)  
    end  
  end  
end
```

```
● ● ●  
  
module Console  
  module FeelingLucky  
    SONG = "<Insert rick roll lyrics>"  
  
    def feeling_lucky  
      SONG.split(',').each do |line|  
        puts line  
        `say #{line.strip}`  
      end  
    end  
  end  
end  
end
```



DEPRECATION WARNING: Extending Rails console through `Rails::ConsoleMethods` is deprecated and will be removed in Rails 8.0.

Please directly use IRB's extension API to add new commands or helpers to the `Ensabee` details, please visit: https://github.com/ruby/irb/blob/master/EXTEND_IRB.md
(called from block in <class:Application> at /Users/jennyshen/test/toronto-on-rails/config/application.rb:29)

Official IRB API

github.com/ruby/irb/blob/master/EXTEND_IRB.md

Helper Method + Classes

Helper Method

Returns a Ruby object

**Command
utility operations, displays info**

```
module Console
  class FeelingLucky < IRB::Command::Base
    category "Best commands"
    description "A command that you'll never forget"

    SONG = <<~SONG
      Never gonna give you up,
      # ...
    SONG

    def execute(*)
      SONG.split(',').each do |line|
        puts line
        `say #{line.strip}`
      end
    end
  end
end
```



```
module TorontoOnRails
  class Application < Rails::Application
    #...

    console do
      require "irb"

      IRB::Command.register :feeling_lucky, Console::FeelingLucky
    end
  end
end
```





toronto-on-rails(dev)> help

...

Context

show_doc	Look up documentation with RI.
ls	Show methods, constants, and variables.
show_source	Show the source code of a given method, class/module, or constant.
whereami	Show the source code around binding.irb again.

Best commands

feeling_lucky	A command that you'll never forget
---------------	------------------------------------



Rails console

reload!	Reloads the Rails application.
---------	--------------------------------

Helper methods

conf	Returns the current IRB context.
------	----------------------------------



DEPRECATION WARNING: `rails/console/helpers` has been deprecated and will be removed in Rails 8.0.
Please require `rails/console/methods` instead.



DEPRECATION WARNING: `rails/console/app` has been deprecated and will be removed in Rails 8.0.
Please require `rails/console/methods` instead.



Active job queue adapters

rails/rails#48585

config.active_job.queue_adapter = :<adapter>

ActionDispatch::IntegrationTest

ActionMailer::TestCase

ActiveJob::TestCase

Will still use TestAdapter 😱

queue_adapter is now consistent!



```
class BackfillPostsTest < ApplicationSystemTestCase
  include ActiveJob::TestHelper

  test "can trigger backfill job" do
    stub_mastodon_requests
    visit posts_path
    assert_enqueued_with(job: BackfillMastodonAlertsJob) do
      click_button "Refresh"
    end
  end

  #...
end
```

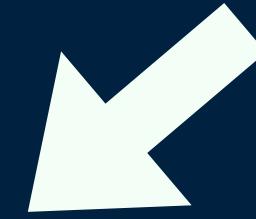




```
$ bin/rails test
```

Error:

```
BackfillMastodonAlertsJobTest#test_create_posts:  
ArgumentError: assert_enqueued_with requires the Active Job test adapter, you're  
using ActiveJob::QueueAdapters::SidekiqAdapter.  
    test/jobs/backfill_mastodon_alerts_job_test.rb:11:in `block in  
<class:BackfillMastodonAlertsJobTest>'
```



v ⌂ 3 config/environments/test.rb

	...	@@ -51,8 +51,9 @@
51	51	# Raise exceptions for disallowed deprecations.
52	52	config.active_support.disallowed_deprecation.r
53	53 +	config.active_job.queue_adapter = :test



Test is missing assertions

rails/rails#51625



```
$ bin/rails test
```

```
Test is missing assertions: `test_each_resolved_event_has_a_resolved_post`  
/Users/jennyshen/test/toronto-on-rails/test/models/event_test.rb:4
```

```
Test is missing assertions: `test_feeling_lucky_command_does't_explode`  
/Users/jennyshen/test/toronto-on-rails/test/console_test.rb:4
```

Identify broken tests

Read more at: railsatscale.com/2024-01-25-catching-assertionless-tests



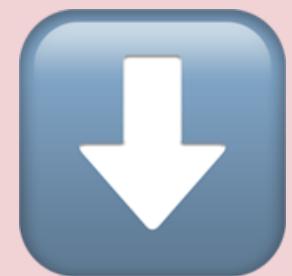
```
class EventTest < ActiveSupport::TestCase
  test "each resolved event has a resolved post" do
    events = Event.resolved # there's no resolved events

    events.each do |event|
      assert_not_nil event.resolution # doesn't reach the assert
    end
  end

  # ...
end
```



```
class ConsoleTest < ActiveSupport::TestCase
  test "feeling_lucky command doesn't explode" do
    system("echo 'abort unless respond_to?(:feeling_lucky)' | bin/rails console", exception: true)
  end
end
```



wrap assert_nothing_raised



```
class ConsoleTest < ActiveSupport::TestCase
  test "feeling_lucky command doesn't explode" do
    assert_nothing_raised do
      system("echo 'abort unless respond_to?(:feeling_lucky)' | bin/rails console", exception: true)
    end
  end
end
```

Enum syntax

rails/rails#50987



```
enum kind: [:detour, :delay, :security, :unknown]
```



DEPRECATION WARNING: Defining enums with keyword arguments is deprecated and will be removed in Rails 8.0. Positional arguments should be used instead:

```
enum :kind, [:detour, :delay, :security, :unknown]
(called from <class:Event> at /Users/jennyshen/test/toronto-on-rails/app/models/event.rb:9)
```

Removes the need of underscoring options

```
class Function > ApplicationRecord
  enum color: [:red, :blue],
    type: [:instance, :class],
    _scopes: false
```

Secrets

Rails 4.1 - Rails.application.secrets

config/secrets.yml



```
development:
  secret_key_base: 9aa7613af4777e365adc101f684bb50d
  weather_api_key: i_love_sunny_days

test:
  secret_key_base: 7ced99e07962cacce4caa231c51df6b5
  weather_api_key: i_love_rainy_days

production:
  secret_key_base: e8acc8ef9789a87bba41fb879dfb1a8a
  weather_api_key: i_love_hurricane_days
```

Rails 5.1 - Encrypted secrets

config/application.rb

```
● ● ●  
module TorontoOnRails  
  class Application < Rails::Application  
    config.read_encrypted_secrets = true  
  
    #...  
  end  
end
```



bin/rails secrets:setup



config/secrets.yml.key
config/secrets.yml.enc

Rails 5.2 - Rails.application.credentials

Environment based secrets

config/credentials/<env>.key

config/credentials/<env>.yml.enc

Rails 7.2



DEPRECATION WARNING: 'config.read_encrypted_secrets=' is deprecated and will be removed in Rails 8.0.

Rails.application.secrets is removed

Get started with credentials

bin/rails credentials:edit –environment <env>

Update task file changes

Configure puma-dev

```
3 +  
4 +  # puts "\n== Configuring puma-dev =="  
5 +  # system "ln -nfs #{APP_ROOT} ~/.puma-dev/#{APP_NAME}"  
6 +  # system "curl -Is https://#{APP_NAME}.test/up | head -n 1"
```

Puma thread count from 5 to 3

```
5 + # Puma starts a configurable number of processes (workers) and each process
6 + # serves each request in a thread from an internal thread pool.
7 +
8 + # The ideal number of threads per worker depends both on how much time the
9 + # application spends waiting for IO operations and on how much you wish to
10 + # to prioritize throughput over latency.
11 +
12 + # As a rule of thumb, increasing the number of threads will increase how much
13 + # traffic a given process can handle (throughput), but due to CRuby's
14 + # Global VM Lock (GVL) it has diminishing returns and will degrade the
15 + # response time (latency) of the application.
16 +
17 + # The default is set to 3 threads as it's deemed a decent compromise between
18 + # throughput and latency for the average Rails application.
19 + #
```

406-unsupported-browser.html.erb



The screenshot shows a code editor window with the following details:

- Title Bar:** Shows the path "66 public/406-unsupported-browser.html" and a close button.
- Code Area:** Displays the content of the file. The code is color-coded: blue for XML tags, red for CSS properties, and black for text content.
- Code Content:** The file contains an HTML document structure with a title, meta viewport, and a CSS rule for the error page.

```
@@ -0,0 +1,66 @@
...
...
...
1 + <!DOCTYPE html>
2 + <html>
3 + <head>
4 +   <title>Your browser is not supported (406)</title>
5 +   <meta name="viewport" content="width=device-width,initial-scale=1">
6 +   <style>
7 +     .rails-default-error-page {
8 +       background-color: #EFEFEF;
9 +       color: #2E2F30;
10 +      text-align: center;
11 +      font-family: arial, sans-serif;
12 +      margin: 0;
13 +    }
14 +
```

guides.rubyonrails.org/v7.2/7_2_release_notes.html#add-browser-version-guard-by-default



```
class ApplicationController < ActionController::Base
  # Allow only browsers natively supporting webp images, web push, badges, import
  maps, CSS nesting + :has
  allow_browser versions: :modern
end

class ApplicationController < ActionController::Base
  # All versions of Chrome and Opera will be allowed, but no versions of "internet
  explorer" (ie). Safari needs to be 16.4+ and Firefox 121+.
  allow_browser versions: { safari: 16.4, firefox: 121, ie: false }
end

class MessagesController < ApplicationController
  # In addition to the browsers blocked by ApplicationController, also block Opera
  below 104 and Chrome below 119 for the show action.
  allow_browser versions: { opera: 104, chrome: 119 }, only: :show
end
```

New framework defaults

Enqueue jobs after transaction commit

```
# backends that use the same database as Active Record as a queue, hence they
# don't need this feature.
#++
Rails.application.config.active_job.enqueue_after_transaction_commit = :default
```



```
Event.transaction do
  event = Event.create(incident_post: post)

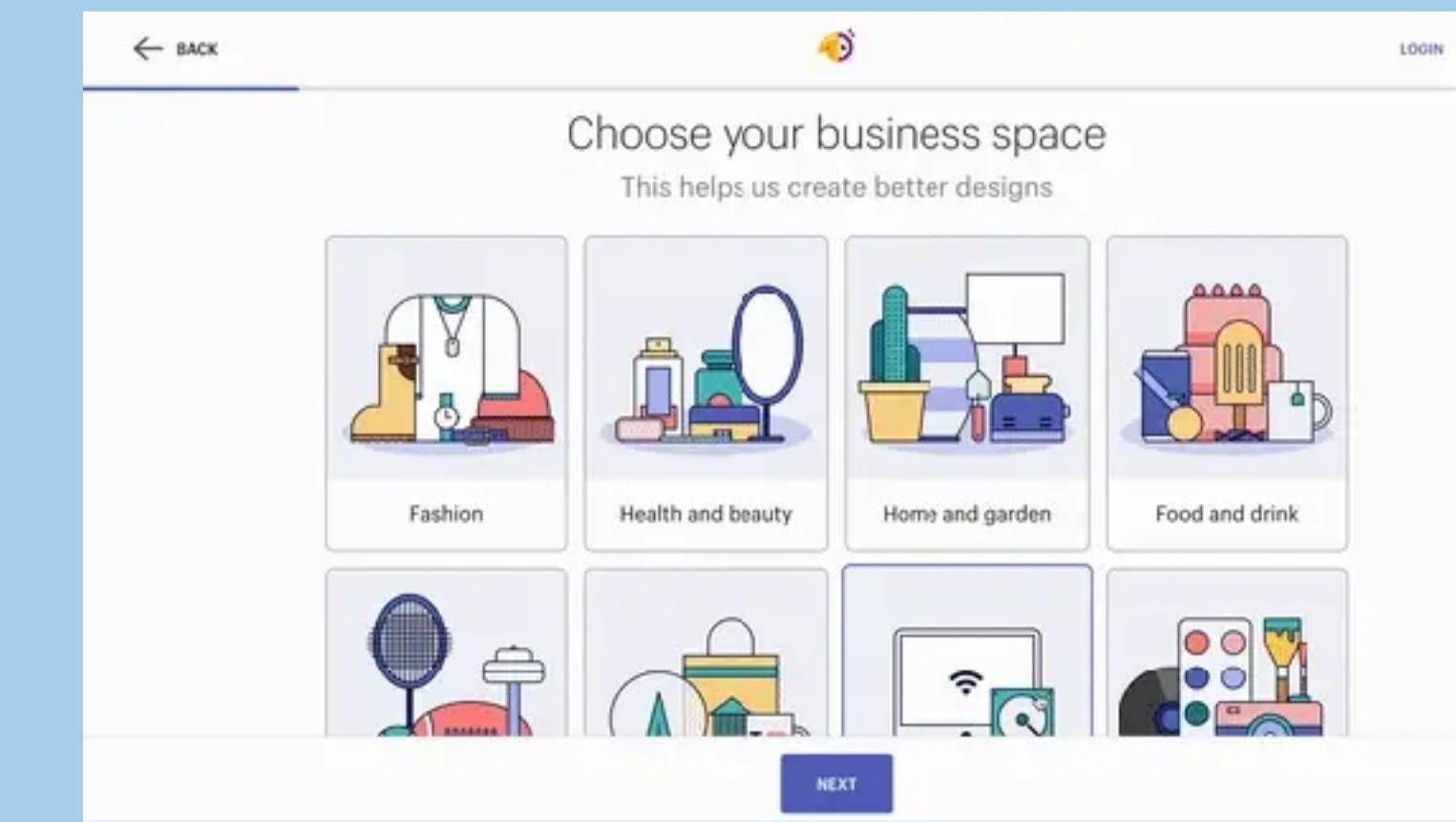
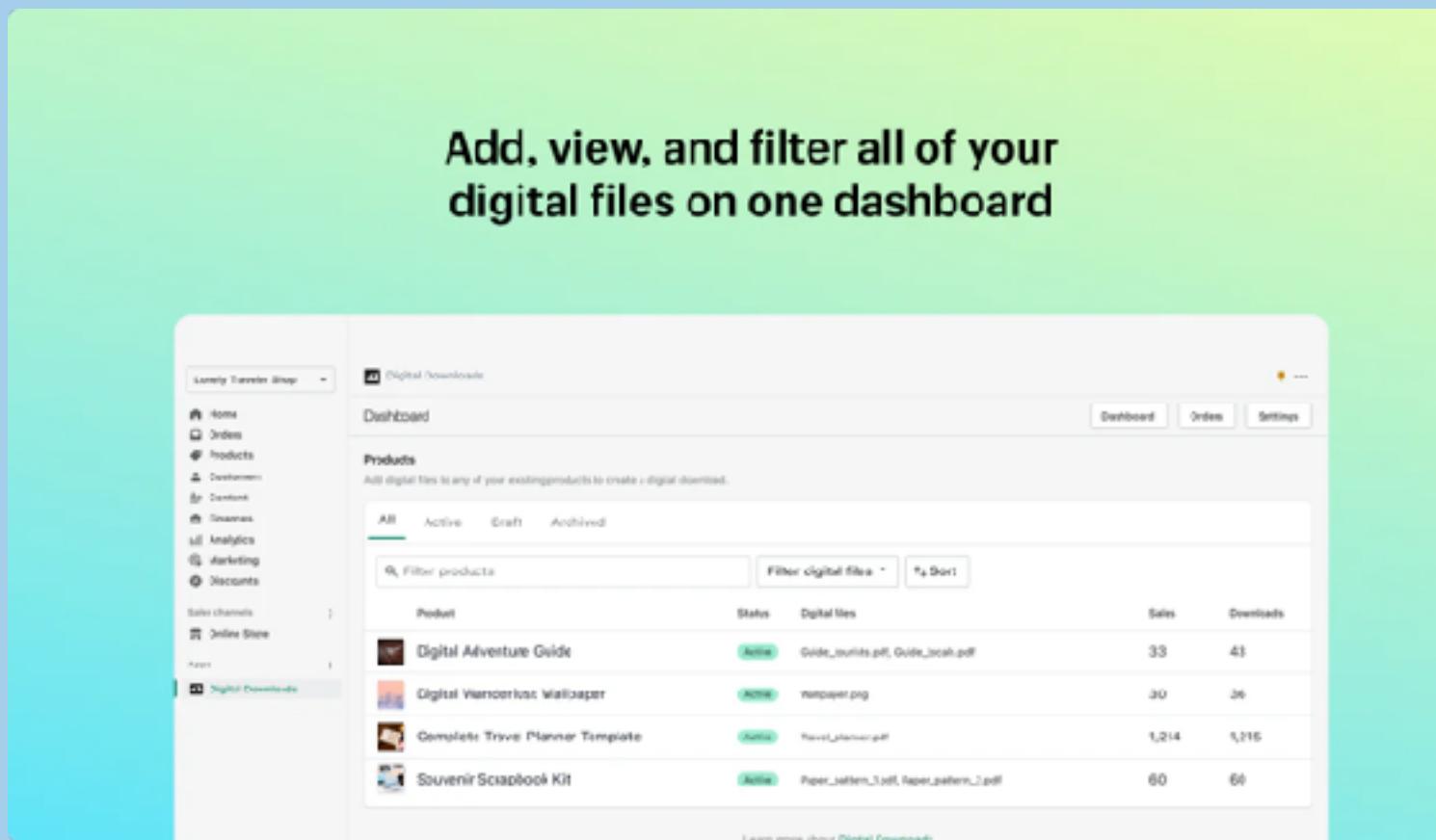
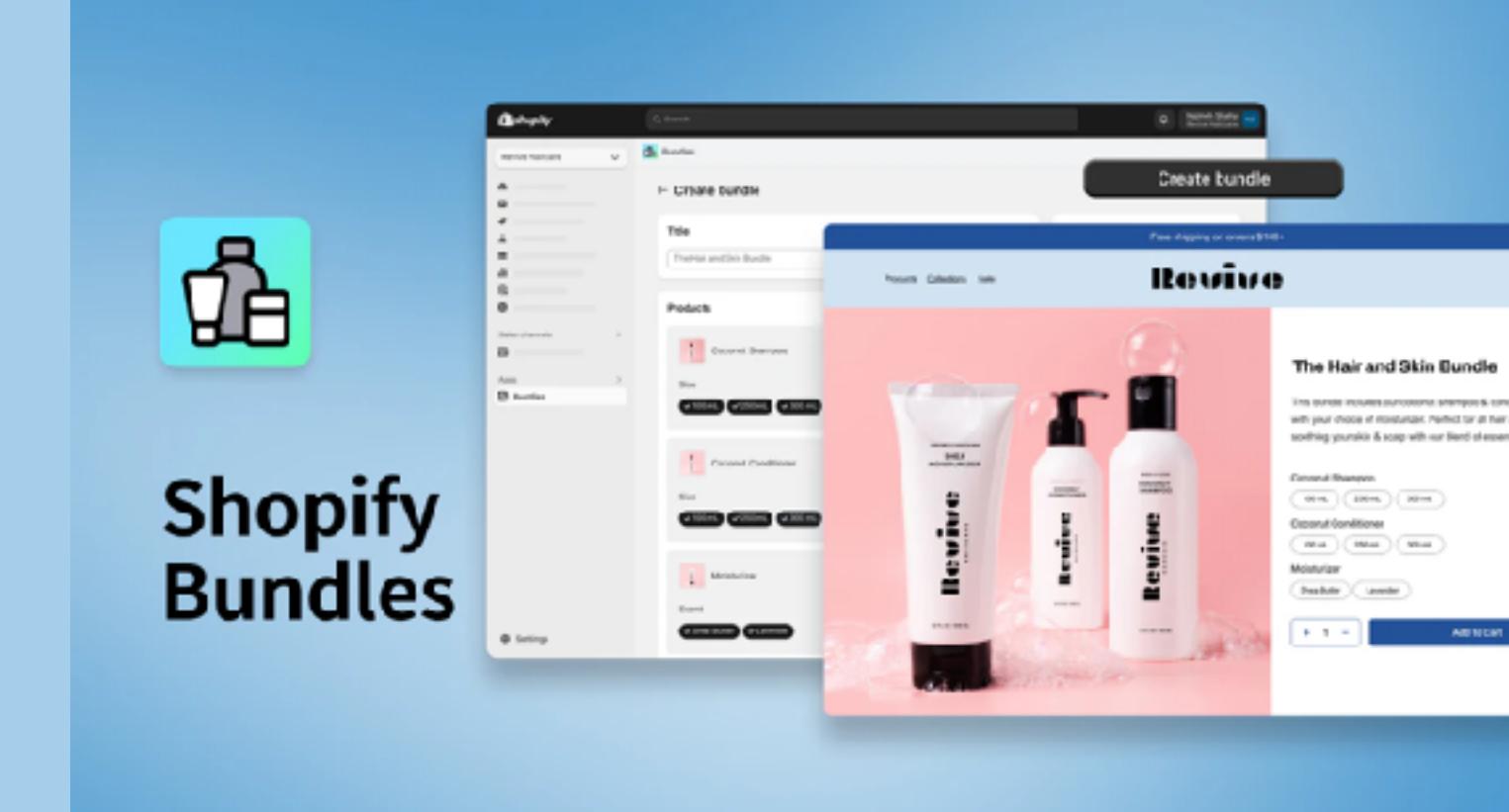
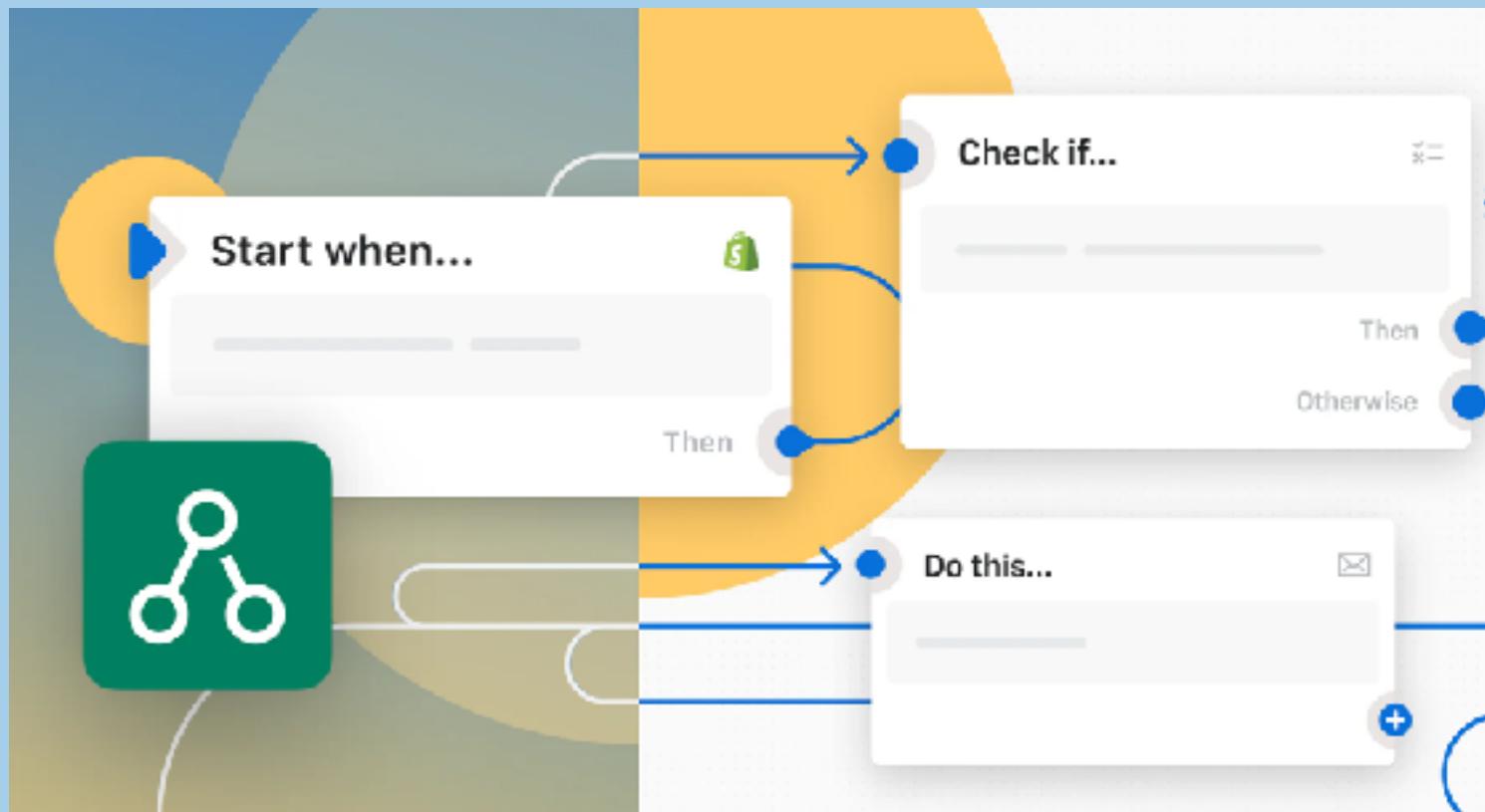
  NewEventNotificationJob.perform_later(event)
end
```

YJIT enabled by default

```
###  
# Enables YJIT as of Ruby 3.3, to bring sizeable performance improvements. If you are  
# deploying to a memory constrained environment you may want to set this to `false`.  
#++  
Rails.application.config.yjit = true
```

Upgrade complete!

Rails Upgrades at Scale



Shopify 2020

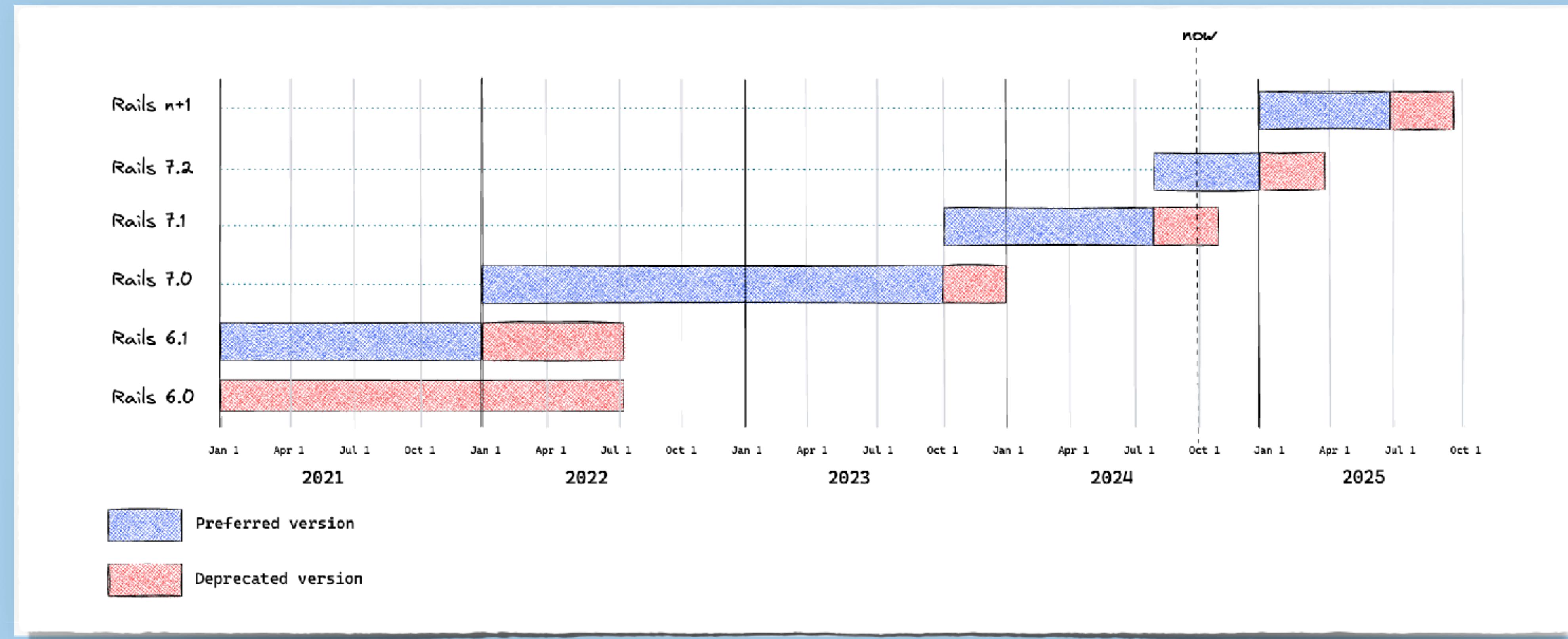
500+ Rails applications

~20 different Rails versions

Apps were out of date

Steps were unclear

- 1. How to ensure apps are up to date**
- 2. How to make upgrading easier**



Plan around an upgrade



Delete unused applications

2. How to make upgrading easier

rails_upgrade gem



Upgrading Ruby on Rails

This guide provides steps to be followed when you upgrade your applications to a newer version of Ruby on Rails. These steps are also available in individual release guides.

1 General Advice

Before attempting to upgrade an existing application, you should be sure you have a good reason to upgrade. You need to balance several factors: the need for new features, the increasing difficulty of finding support for old code, and your available time and skills, to name a few.

1.1 Test Coverage

The best way to be sure that your application still works after upgrading is to have good test coverage before you start the process. If you don't have automated tests that exercise the bulk of your application, you'll need to spend time manually exercising all the parts that have changed. In the case of a Rails upgrade, that will mean every single piece of functionality in the application. Do yourself a favor and make sure your test coverage is good *before* you start an upgrade.

1.2 Ruby Versions

Chapters

1. General Advice

- [Test Coverage](#)
- [Ruby Versions](#)
- [The Upgrade Process](#)
- [The Update Task](#)
- [Configure Framework Defaults](#)

2. Upgrading from Rails 7.1 to Rails 7.2

- [All tests now respect the `active_job.queue_adapter` config](#)

3. Upgrading from Rails 7.0 to Rails 7.1

- [Development and test environments `secret_key_base` file changed](#)
- [Autoloaded paths are no longer in `\$LOAD_PATH`](#)
- [`configautoload.lib` and `configautoload.lib_once`](#)

- 1. Determine the next step to upgrade your application**
- 2. Create and commit changes based on the step**
- 3. Output any context and follow ups before shipping**

rails_upgrade steps



```
STEPS = [  
  UpdateToLatestPatch,  
  AutocorrectDeprecations,  
  DisallowDeprecations,  
  UpdateToNextMinor,  
  SetupDeprecationCollection,  
  RunAppUpdateTask,  
  EnsureNewDefaults,  
]
```

Upgrade steps from the Rails guides

The process should go as follows:

1. Write tests and make sure they pass.
2. Move to the latest patch version after your current version.
3. Fix tests and deprecated features.
4. Move to the latest patch version of the next minor version.

1.4 The Update Task

1.5 Configure Framework Defaults



```
module RailsUpgrade
  class Step
    class << self
      delegate :met?, :meet, to: :new
    end

    def met?
      # check whether the step has already been met
    end

    def meet
      # make changes to the code to satisfy the step
      # build commit message and commit changes
    end
  end
end
```

```
1 module RailsUpgrade
2   class UpdateToLatestPatch < Step
3     def met?
4       return unless version = @version_detection.current_version
5       latest_patch = latest_patch_of_minor
6       return true if !latest_patch && version.prerelease?
7       version == latest_patch
8     end
9
10    def meet
11      latest_patch_version = latest_patch_of_minor.version
12
13      RailsUpgrade.patch_upgrade = true
14      @version_upgrader.upgrade("rails", latest_patch_version)
15
16      title("Bump Rails from #{@version_detection.current_version} to #{latest_patch_version}")
17      commit("Upgrade to Rails #{latest_patch_version}")
18      guidance(<<~MESSAGE)
19        ### Updating to the latest patch version
20
21        Bumps Rails from #{@version_detection.current_version} to #{latest_patch_version}.
22 MESSAGE
23 todo(<<~MESSAGE)
24   **Review the [changelog](https://github.com/rails/rails/releases/tag/v#{latest\_patch\_version}) \
25   to better understand the changes and how they might affect your application.**
26   Your application will be updated from #{@version_detection.current_version} to #{latest_patch_version} by this PR. \
27   Make sure that won't break anything by testing the changes in CI and staging (if possible)!
28 MESSAGE
29   :advance_with_changes
30   end
31 end
32 end
33
```

```
1 module RailsUpgrade
2   class UpdateToLatestPatch < Step
3     def met?
4       return unless version = @version_detection.current_version
5       latest_patch = latest_patch_of_minor
6       return true if !latest_patch && version.prerelease?
7       version == latest_patch
8     end
9
10    def meet
11      latest_patch_version = latest_patch_of_minor.version
12
13      RailsUpgrade.patch_upgrade = true
14      @version_upgrader.upgrade("rails", latest_patch_version)
15
16      title("Bump Rails from #{@version_detection.current_version} to #{latest_patch_version}")
17      commit("Upgrade to Rails #{latest_patch_version}")
18      guidance(<<~MESSAGE)
19        ### Updating to the latest patch version
```

```
6      return true if :latest_patch && version.prerelease?
7      version == latest_patch
8    end
9
10   def meet
11     latest_patch_version = latest_patch_of_minor&.version
12
13     RailsUpgrade.patch_upgrade = true
14     @version_upgrader.upgrade("rails", latest_patch_version)
15
16     title("Bump Rails from #{@version_detection.current_version} to #{latest_patch_version}")
17     commit("Upgrade to Rails #{latest_patch_version}")
18     guidance(<<~MESSAGE)
19       ### Updating to the latest patch version
20
21     Bumps Rails from #{@version_detection.current_version} to #{latest_patch_version}.
22 MESSAGE
23 todo(<<~MESSAGE)
24   **Review the [changelog](https://github.com/rails/rails/releases/tag/v#{latest\_patch\_version}) \
25   to better understand the changes and how they might affect your application.**
26   Your application will be updated from #{@version_detection.current_version} to #{latest_patch_version}
27   Make sure that won't break anything by testing the changes in CI and staging (if possible)!
```

Autocorrecting deprecations

Rails 6.1: "`URI.parser` is deprecated. Use `URI::DEFAULT_PARSER` instead."



```
return unless node.message.value == "parser"
return unless node.receiver&.respond_to?( :value )
return unless node.receiver.value.is_a?(SyntaxTree::Const)
return unless node.receiver.value.value == "URI"
```



```
def substitution
  "URI::DEFAULT_PARSER"
end
```



```
$ rails_upgrade
== Updating to latest patch ==
+ bundle update rails
+ git add .
+ git commit --message 'Upgrade to Rails 7.1.4' --quiet
== Updating RBI files for bumped gems ==
== Autocorrecting deprecations ==
== Disallowing deprecations ==
```

Bump Rails from 7.1.0 to 7.1.4

What is this PR doing?

Updating to the latest patch version

Bumps Rails from 7.1.0 to 7.1.4.

What do I need to do?

> [!IMPORTANT]

> Read through the following action items and complete them before merging this PR.

- [] **Review the [changelog](<https://github.com/rails/rails/releases/tag/v7.1.4>) to better understand the changes and how they might affect your application.**
Your application will be updated from 7.1.0 to 7.1.4 by this PR. Make sure that won't break anything by testing the changes in CI and staging (if possible)!
- [] **Merge this PR.** After completing any action items above, you can merge this pull request.

Solid Track application

- 1. Tracks all Shopify Rails apps and their versions**
- 2. Runs the rails_upgrade command and open PRs**
- 3. Repeat step 2 on a weekly basis**

Solid Track

Applications Internal ▾

Learn more about upgrading Rails at Shopify 🚧

Rails Applications

Search

Submit

Statistics

	7.2	7.1	7.0	6.1	6.0	5.2	nil
8.0.0.alpha	1	0	0	0	0	0	0
7.2.1	36	6	1	0	1	0	0
7.2.0.alpha	1	0	0	0	0	0	0
7.2.0	8	8	0	0	0	0	0
7.1.4	0	20	1	0	0	0	0

Shopify/solid_track

Rails Version: 7.2.1

Config Defaults Version: 7.2



App upgraded

Congratulations! 🎉 Your application is up-to-date!

Learn more about upgrading Rails at Shopify 🧑



Previous Rails Upgrade Runs

Success	Nothing to do!	Aug 27 2024	View upgrade job
Success	Use new framework defaults	Aug 26 2024	View Pull Request
Success	Use new framework defaults	Aug 23 2024	View upgrade job
Success	Bump Rails from 7.2.0 to 7.2.1	Aug 23 2024	View Pull Request
Success	Bump Rails from 7.1.3.4 to 7.2.0	Aug 13 2024	View Pull Request
Success	Bump Rails from 7.1.3.4 to 7.2.0	Aug 12 2024	View upgrade job
Success	Nothing to do!	Aug 05 2024	View upgrade job

Summary

Jobs

upgrade-rails

Run details

Usage

Workflow file

upgrade-rails

succeeded 5 days ago in 2m 15s

Search logs

↻ ⚙

Step	Description	Time
> ✓ Set up Ruby		14s
> ✓ Install rails_upgrade		1s
> ✓ Install rails_upgrade-shopify		1s
> ✓ Set up DATABASE_URL scheme		0s
> ✓ Setup the application before upgrading		13s
✓ Run the upgrade		40s

```
1 ► Run set -x
36 + bundle config --delete deployment
37 + err=0
38 + test '!' -f tmp/pull-request.txt
39 + rails_upgrade --require rails_upgrade-shopify
40 + DISABLE_SPRING=1 bin/rails runner 'IO.new(3).puts Rails.application.config.loaded_config_version' 3>&7
41 Configuration defaults (7.1) not up-to-date with Rails version (7.2).
42 Finishing current minor upgrade first.
43 == Updating to latest patch ==
44 == Updating RBI files for bumped gems ==
45 == Autocorrecting deprecations ==
46 + bundle exec rubocop --autocorrect-all
47 == Disallowing deprecations ==
```

[Solid Track] Rails Upgrade - Use new framework defaults #2281

Merged

Schwad merged 1 commit into `main` from `rails-upgrade/patch` 2 days ago

Conversation 1

Commits 1

Checks 1

Files changed 2



shopify-solidtrack (bot) commented 5 days ago · edited by Schwad

What is this PR doing?

Using the defaults for the current Rails version

Deletes the `config/initializers/new_framework_defaults_7_2.rb` file (containing the commented-out new defaults created during the Update task) and configures the application with the [defaults for the current version of Rails](#).

What do I need to do?

Important

Read through the following action items and complete them before merging this PR.

- Make sure that your application is ready to run with new defaults before merging this PR.** To test whether your application is ready, go through the instructions in the header of the `config/initializers/new_framework_defaults_7_2.rb` file.
- Merge this PR.** After completing any action items above, you can merge this pull request.

View more information about your service's Rails upgrades in your [service dashboard in Solid Track](#).

► More Information / Need help?



Reviewers

Schwad

Assignees

No one—assign

Labels

dependency

Projects

Ruby D

Status: Done

Milestone

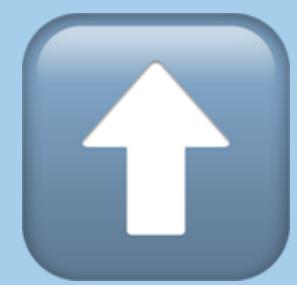
No milestone

Development

Successfully
these issues.
None yet



Upgrade calendar



rails_upgrade gem



Solid Track app



Shopify now

~400 Rails applications

on Rails 7.1 or later

Shopify now

**Super cool! Never thought that Rails upgrade
can be automated to the degree of being an
automatic PR** 

- A Shopify Dev using Solid Track

To recap

How to upgrade Rails and its importance

Tips to upgrade to the latest Rails

How Shopify upgrades at scale

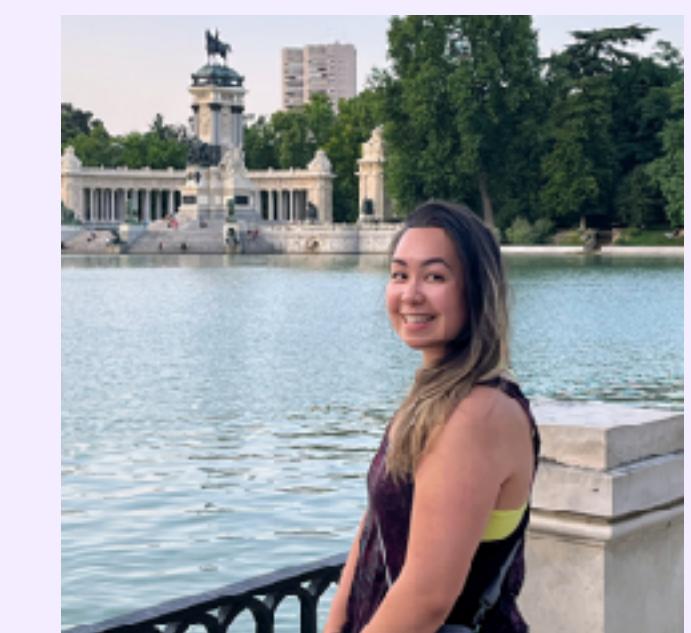
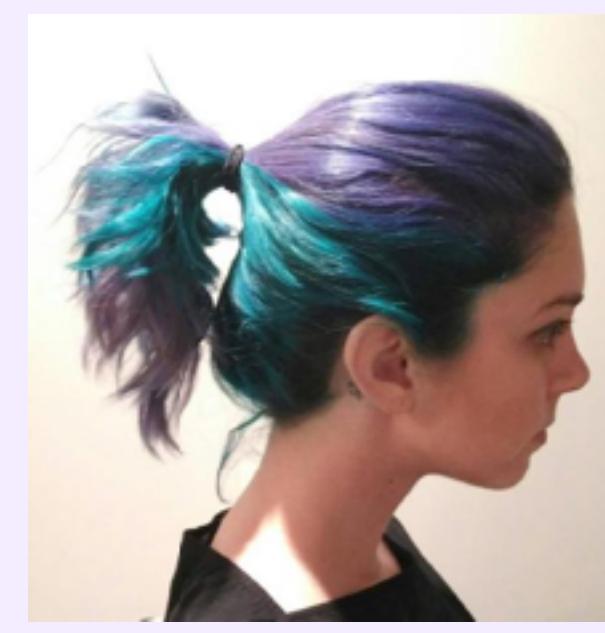
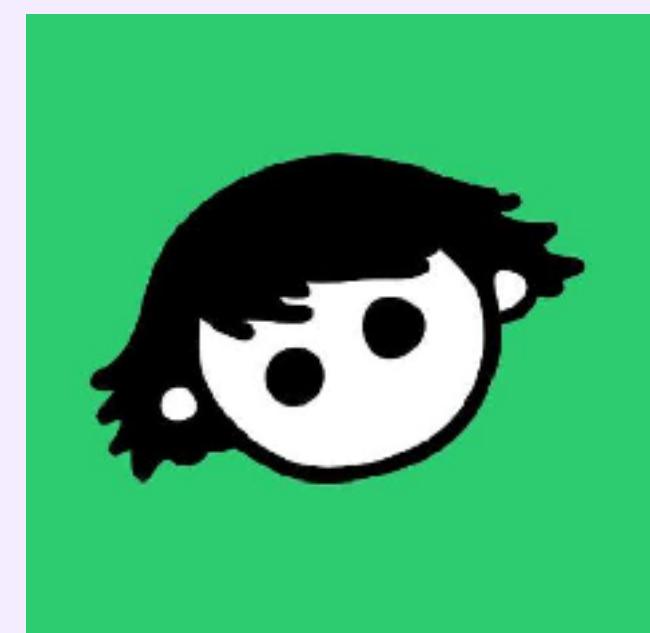
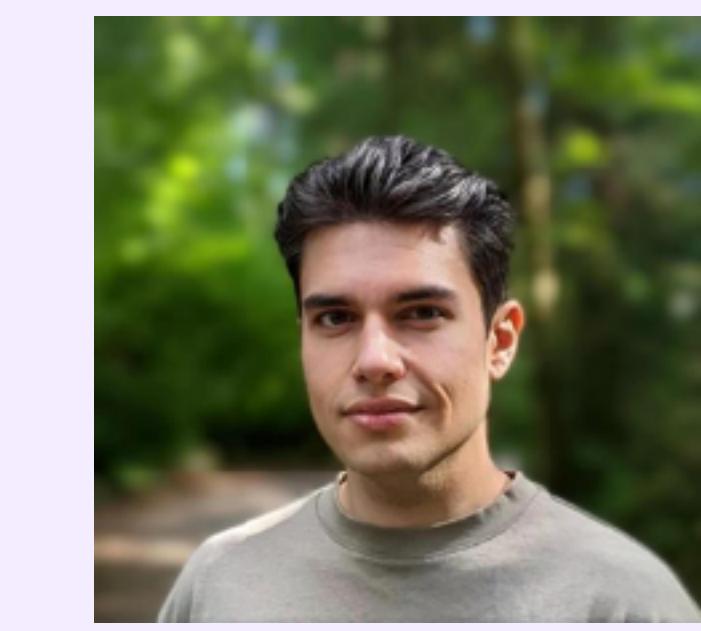
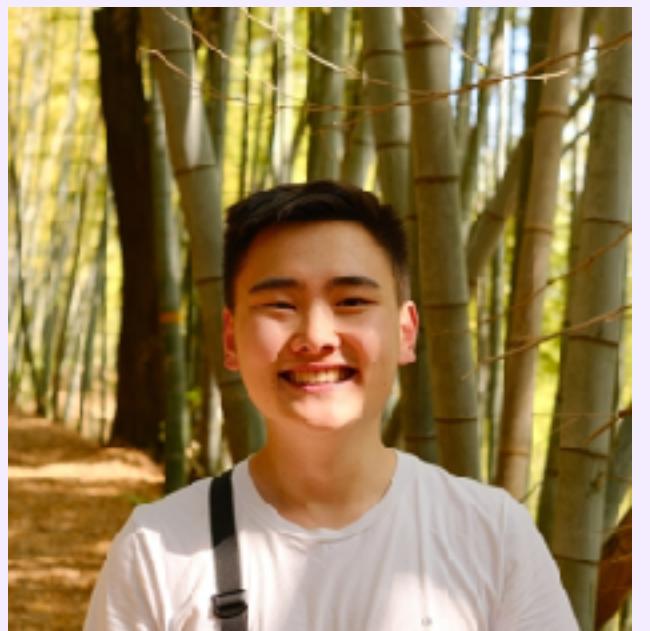
If there's one thing to take away

Keep your app up to date

*If you're running on Rails 3.2, your
version is 10 years old and so are your
problems*

*Eileen Uchitelle
RailsConf 2022 Keynote: The Success of Ruby on Rails*

Do not be the TTC of Rails apps



Thanks and happy upgrading!

Slides and more!



@jenshenny@ruby.social



@jenshenny

