

Rails 5.2

@bogdanlviv

Bogdan Denkovych

About me

It's my first talk! 🎉

About me

Vertalab

✓ Ruby Developer

About me

I ♥ vim



About me

I ♥ tmux



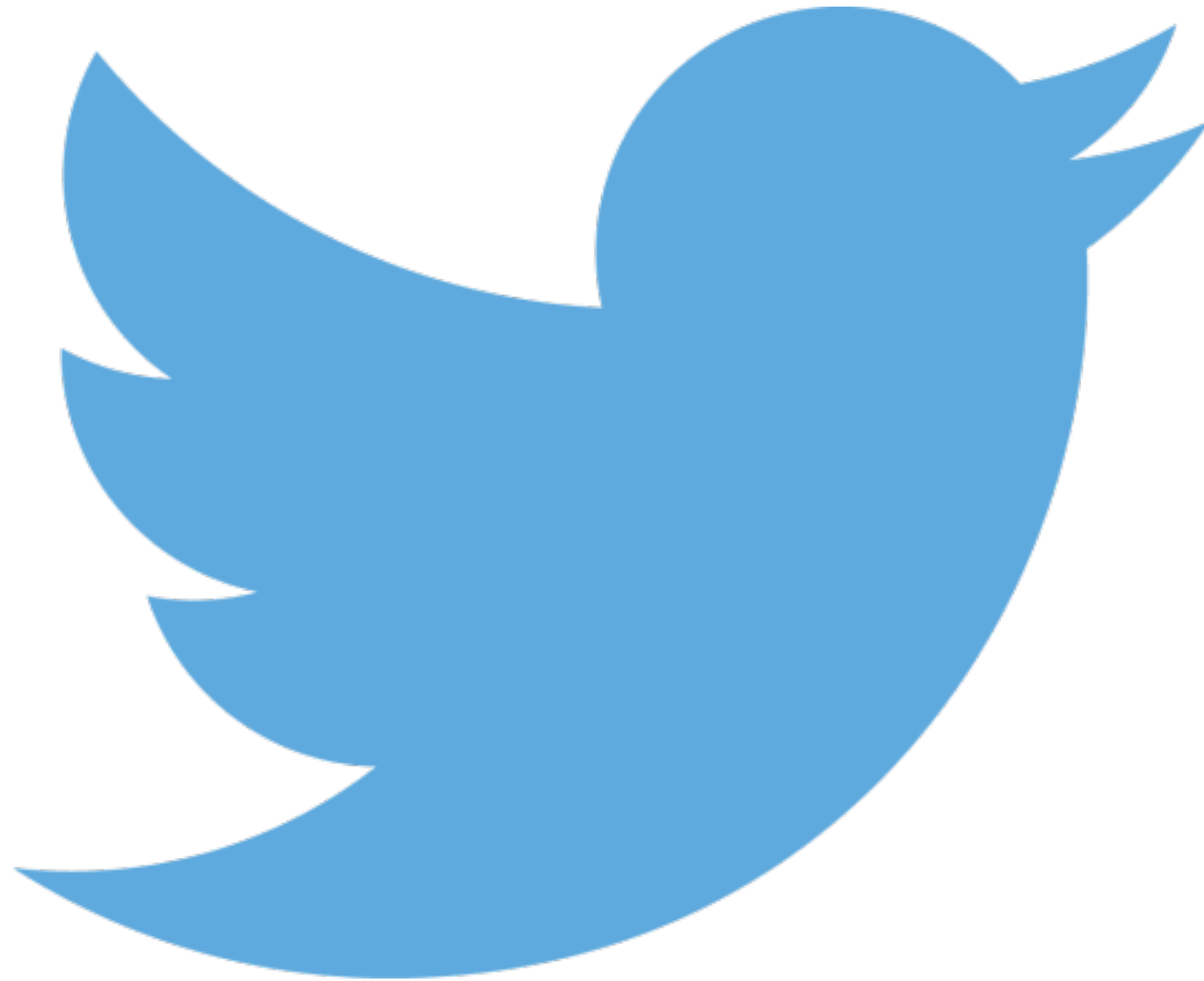
About me

bogdanvlviv.github.io



About me

@bogdanvLviv



Open Source



Open Source

Open Source Friday

[Contributors](#)

[Businesses](#)

[Maintainers](#)

[Sign up](#)



Your business runs on open source software

Once a week, every week. Take time to invest in
the open source software you use and love.

[Sign up with GitHub](#)

Your business depends on it

Companies need open source software to run effectively.

opensourcefriday.com


Open Source Friday

Open source is made by people just like you. This Friday, invest a few hours contributing to the software you use and love.

[Sign up with GitHub](#)



Open Source

 **RAILS CONTRIBUTORS**

#77 bogdanvlviv - All time

Showing 96 commits

Hash	Date	Message
703478d	14 Nov 2017	Fix activesupport/CHANGELOG.md [ci skip]
46a2f93	14 Nov 2017	Fix migration version in doc of #up_only
6f12334	08 Nov 2017	Change output log about skipping instalation of Active S...
d5ad637	08 Nov 2017	Remove redundant passing --skip-active-storage in test c...
3ff2c15	06 Nov 2017	Add cases to test combining validation conditions
90fe2a4	06 Nov 2017	Fix `bin/rails db:migrate` with specified `VERSION`
cb8553c	06 Nov 2017	Execution of `active_storage:install` should respect `--...
f4af77a	06 Nov 2017	Rails::Generators::Actions#execute_command allows option...
0835527	06 Nov 2017	`rails new` runs `rails active_storage:install`
4a835aa	06 Nov 2017	Add --skip-active-storage and do so automatically when -...
7308991	05 Nov 2017	Execute `ConfirmationValidator` validation when `_confir...
11f3f03	03 Nov 2017	Improve docs of ActionDispatch::Routing::Mapper
a37d03c	30 Oct 2017	Simplify implementation of `MySQLDatabaseTasks`
c4867b8	30 Oct 2017	Raise error if unsupported charset for mysql

All time

Today

This week

This month

This year

Releases

Edge

FAQ

About me

Ruby on Rails

- ✓ Contributor

March 26, 2016 - August 10, 2017

- ✓ Contributors Team Member

August 11, 2017 - present

Rails 4



Rails 5



Rails 5.0

Ruby 2.2.2+ required

Rails 5.0

Action Cable

Rails 5.0

Turbolinks 5

Rails 5.0

```
rails new my_api - -api
```

Rails 5.0

*Exclusive use of rails CLI
over Rake*

```
$ rake db:migrate
```

```
$ rails db:migrate
```

Rails 5.0

ar_internal_metadata

ar_internal_metadata

```
$ bin/rails dbconsole
> SELECT * FROM ar_internal_metadata;
key          |value          |created_at          |updated_at
environment|development|2017-09-11 23:14:10.815679|2017-09-11 23:14:10.815679
```

```
$ RAILS_ENV=test bin/rails dbconsole
> SELECT * FROM ar_internal_metadata;
key          |value|created_at          |updated_at
environment|test |2017-09-11 23:14:10.815679|2017-09-11 23:14:10.815679
```

```
$ RAILS_ENV=production bin/rails dbconsole
> SELECT * FROM ar_internal_metadata;
key          |value          |created_at          |updated_at
environment|production|2017-09-11 23:14:10.815679|2017-09-11 23:14:10.815679
```

ar_internal_metadata

*Rails 5 prevents
destructive action on
production database*

ar_internal_metadata

```
$ bin/rails db:drop
$ bin/rails db:drop:all
$ bin/rails db:purge
$ bin/rails db:purge:all
$ bin/rails db:schema:load
$ bin/rails db:structure:load
# ...
```

ar_internal_metadata

```
rails aborted!  
ActiveRecord::ProtectedEnvironmentError: You are attempting to run a destructive action against your 'production' database.  
If you are sure you want to continue, run the same command with the environment variable:  
DISABLE_DATABASE_ENVIRONMENT_CHECK=1
```

DISABLE_DATABASE_ENVIRONMENT_CHECK=1

```
config.active_record.protected_environments = ["production", "staging"]
```


Rails 5.1

Yarn Support

```
config.assets.paths << Rails.root.join('node_modules')
```

Rails 5.1

Optional Webpack support

```
gem "webpacker"
```

Rails 5.1

jQuery no longer a default dependency

jQuery was required by default in earlier versions of Rails to provide features like ``data-remote``, ``data-confirm`` and other parts of Rails' Unobtrusive JavaScript offerings. It is no longer required, as the UJS has been rewritten to use plain, vanilla JavaScript. This code now ships inside of Action View as ``rails-ujs``.

Rails 5.1

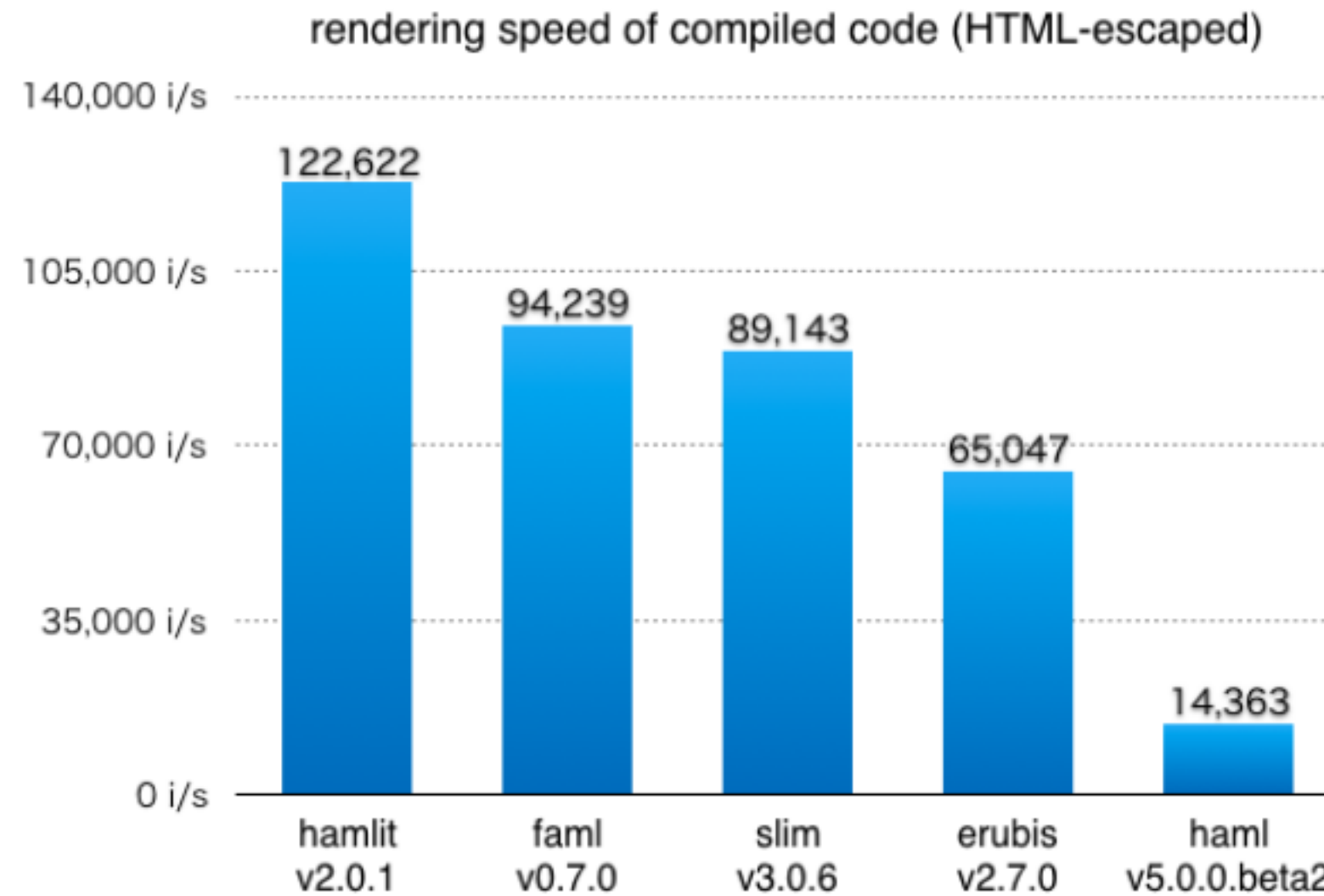
Unification of ``form_for`` and ``form_tag`` into ``form_with``

http://api.rubyonrails.org/v5.1/classes/ActionView/Helpers/FormHelper.html#method-i-form_with

Rails 5.1

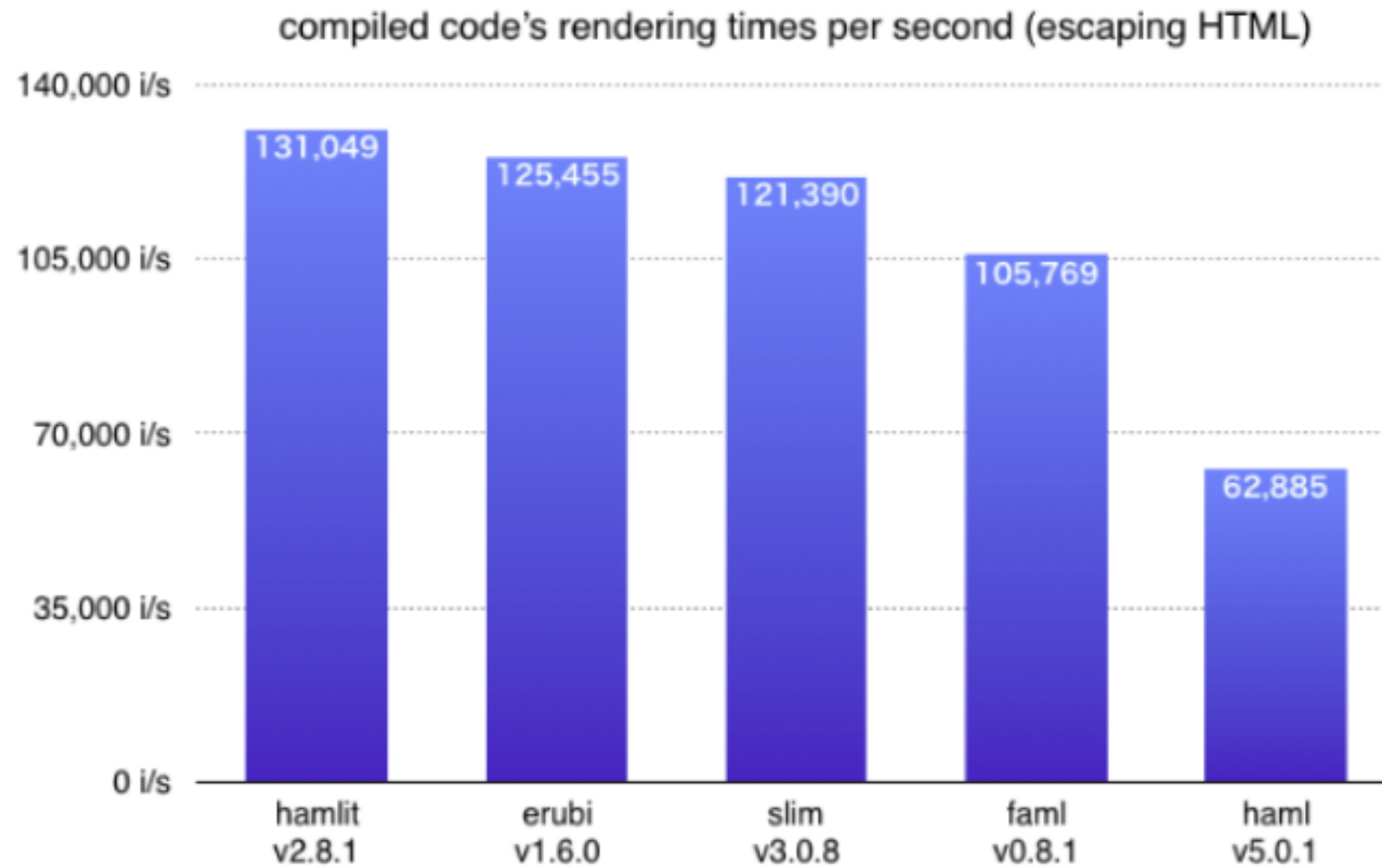
*Changed ActionView ERB
Handler from Erubis to
Erubi*

Erubis



```
hamlit v2.0.1: 122622.3 i/s
faml v0.7.0: 94239.1 i/s - 1.30x slower
slim v3.0.6: 89143.0 i/s - 1.38x slower
erubis v2.7.0: 65047.8 i/s - 1.89x slower
haml v5.0.0.beta.2: 14363.6 i/s - 8.54x slower
```

Erubi



```
hamlit v2.8.1: 131048.9 i/s
erubi v1.6.0: 125445.4 i/s - 1.04x slower
slim v3.0.8: 121390.4 i/s - 1.08x slower
faml v0.8.1: 100750.5 i/s - 1.30x slower
haml v5.0.1: 54882.6 i/s - 2.39x slower
```

Rails 5.1

*Capybara Integration with
Rails (AKA System Tests)*

System Tests

```
# test/application_system_test_case.rb
require "test_helper"

class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
  driven_by :selenium, using: :chrome, screen_size: [1400, 1400]
end
```

System Tests

```
# test/application_system_test_case.rb
require "test_helper"

class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
  driven_by :selenium, using: :firefox
end
```

System Tests

```
# test/application_system_test_case.rb
require "test_helper"
require "capybara/poltergeist"

class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
  driven_by :poltergeist, screen_size: [1400, 1400], options: { js_errors: true }
end
```

System Tests

By default, running `bin/rails test` won't run your system tests. Make sure to run `bin/rails test:system` to actually run them.

Rails 5.1

Encrypted secrets

``bin/rails secrets:setup`` generates the ``config/secrets.yml.key`` and ``config/secrets.yml.enc`` files.

```
$ export RAILS_MASTER_KEY=your_key  
$ export EDITOR=vi  
$ bin/rails secrets:edit
```

Rails 5.1

PostgreSQL & MySQL: Use big integer as primary key type for new tables

Jon McCartie, *Pavel Pravosud*

2147483647



Twitter API 
@TwitterAPI

Follow



We've scheduled the Twitpocalypse for
21:00 GMT (2pm Pacific/5pm Eastern). -
<http://bit.ly/2nXXe> ^MS

11:32 AM - 12 Jun 2009

18 Likes



18

Rails 5.1

Virtual/generated column support for MySQL 5.7.5+ and MariaDB 5.2.0+.

MySQL generated columns: <https://dev.mysql.com/doc/refman/5.7/en/create-table-generated-columns.html>

MariaDB virtual columns: <https://mariadb.com/kb/en/mariadb/virtual-computed-columns/>

Declare virtual columns with ``t.virtual name, type: ..., as: "expression"``.

Pass ``stored: true`` to persist the generated value (false by default).

Example:

```
create_table :generated_columns do |t|
  t.string :name
  t.virtual :upper_name, type: :string, as: "UPPER(name)"
  t.virtual :name_length, type: :integer, as: "LENGTH(name)", stored: true
  t.index :name_length # May be indexed, too!
end
```

Ryuta Kamizono

Rails 5.1

Added declarative exception handling via `ActiveJob::Base.retry_on` and `ActiveJob::Base.discard_on`.

Examples:

```
class RemoteServiceJob < ActiveJob::Base
  retry_on CustomAppException # defaults to 3s wait, 5 attempts
  retry_on AnotherCustomAppException, wait: ->(executions) { executions * 2 }
  retry_on ActiveRecord::StatementInvalid, wait: 5.seconds, attempts: 3
  retry_on Net::OpenTimeout, wait: :exponentially_longer, attempts: 10
  discard_on ActiveJob::DeserializationError

  def perform(*args)
    # Might raise CustomAppException or AnotherCustomAppException for something domain specific
    # Might raise ActiveRecord::StatementInvalid when a local db deadlock is detected
    # Might raise Net::OpenTimeout when the remote service is down
  end
end
```

DHH

Rails 5.1

RailsConf 2017: Rails 5.1: Awesome Features and Breaking Changes by Claudio B

<https://www.youtube.com/watch?v=9CWjFtCbrrM>

Rails 5.2



Rails 5.2

Add ActiveSupport::CurrentAttributes to provide a thread-isolated attributes singleton.
Primary use case is keeping all the per-request attributes easily available to the whole system.

DHH

CurrentAttributes

```
# app/models/current.rb  
class Current < ActiveSupport::CurrentAttributes  
  attribute :request_id, :user_agent, :ip_address  
end
```

CurrentAttributes

```
class ApplicationController < ActionController::Base
  before_action do
    Current.request_id = request.uuid
    Current.user_agent = request.user_agent
    Current.ip_address = request.ip
  end
end
```

```
Thread.current[] =
```

CurrentAttributes

```
class Event < ApplicationRecord
  before_create do
    self.request_id = Current.request_id
    self.user_agent = Current.user_agent
    self.ip_address = Current.ip_address
  end
end
```

```
Thread.current[]
```

Rails 5.2

Merging two relations representing nested joins no longer transforms the joins of the merged relation into LEFT OUTER JOIN. Example to clarify:

```
Author.joins(:posts).merge(Post.joins(:comments))
```

```
# Before the change:
```

```
#=> SELECT ... FROM authors INNER JOIN posts ON ... LEFT OUTER JOIN comments ON...
```

```
# After the change:
```

```
#=> SELECT ... FROM authors INNER JOIN posts ON ... INNER JOIN comments ON...
```

```
*Maxime Handfield Lapointe*
```


Rails 5.2

```
- gem 'therubyracer'  
+ gem 'mini_racer'
```

mini_racer



SamSaffron commented on May 30

Contributor

therubyracer is a relic, it depends on an ancient version of v8 that has multiple known security vulnerabilities

<https://github.com/cowboyd/therubyracer/blob/master/therubyracer.gemspec#L20>

In particular this is the equivalent of shipping a dependency to Chrome version 31.

Chrome 31 was released 2013-11-12

All places that talk about therubyracer or include it in templates should be replaced with mini_racer that depends on the most recent version of v8.

per: [#29276 \(comment\)](#)



29



schneems approved these changes on May 30

[View changes](#)

Much faster & quite stable.

Rails 5.2

ApplicationRecord is no longer generated when generating models. If you need to generate it, it can be created with ``rails g application_record``.

Lisa Ugray

Rails 5.2

```
# Gemfile  
gem 'bootsnap', '>= 1.1.0', require: false
```

Performance

- Discourse reports a boot time reduction of approximately 50%, from roughly 6 to 3 seconds on one machine;
- One of our smaller internal apps also sees a reduction of 50%, from 3.6 to 1.8 seconds;
- The core Shopify platform -- a rather large monolithic application -- boots about 75% faster, dropping from around 25s to 6.5s.

Read more <https://github.com/Shopify/bootsnap/blob/master/README.md>

Rails 5.2

Add `freeze_time` helper which freezes time to `Time.now` in tests.

Prathamesh Sonpatki

```
# test '#privorak28' do
#   Time.current # => Fri, 24 Nov 2017 19:20:59 EET +02:00
#   freeze_time do
#     sleep(1)
#     Tweet.create.created_at # => Fri, 24 Nov 2017 19:20:59 EET +02:00
#   end
#   Time.current # => Fri, 24 Nov 2017 19:21:00 EET +02:00
# end
def freeze_time(&block)
  travel_to Time.now, &block
end
```

Rails 5.2

Secrets -> Credentials

``config/secrets.yml.key` -> `config/master.key``

``config/secrets.yml.enc` -> `config/credentials.yml.enc``

`ENV["RAILS_MASTER_KEY"]`

``bin/rails secrets:edit` -> `bin/rails credentials:edit``

``Rails.application.secrets` -> `Rails.application.credentials``

Rails 5.2

```
- class_attribute :settings  
- self.settings = {}  
+ class_attribute :settings, default: {}
```

`:default` works for `:mattr_accessor`, `:mattr_reader`, `:mattr_writer`, `:cattr_accessor`, `:cattr_reader`, and `:cattr_writer` as well.

Rails 5.2

Add ``#up_only`` to database migrations for code that is only relevant when migrating up, e.g. populating a new column.

Rich Daley

```
class AddPublishedToPosts < ActiveRecord::Migration[5.2]
  def change
    add_column :posts, :published, :boolean, default: false

    up_only do
      execute "update posts set published = 'true'"
    end
  end
end
```


Rails 5.2



Aaron Patterson ✓

@tenderlove

Follow



YES! Rails 5.2 will ship with some H2 support [github.com/puma/puma/pull...](https://github.com/puma/puma/pull/1403)
[github.com/rails/rails/pu...](https://github.com/rails/rails/pull/25444)

Great job [@eileencodes](#)!!



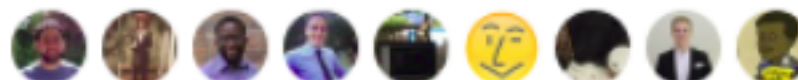
Implement H2 Early Hints for Rails by eileencodes · Pull R...

When puma/puma#1403 is merged Puma will support the Early Hints status code for sending assets before a request has finished. While the Early Hints spec is still in draft, this PR pr...

github.com

12:15 PM - 4 Oct 2017

56 Retweets 139 Likes



7



56



139

HTTP2 103 Early Hints

HTTP/1.1 103 Early Hints

Link: </style.css>; rel=preload; as=style

Link: </script.js>; rel=preload; as=script

HTTP/1.1 200 OK

Date: Fri, 26 May 2017 10:02:11 GMT

Content-Length: 1234

Content-Type: text/html; charset=utf-8


Link: </style.css>; rel=preload; as=style

Link: </script.js>; rel=preload; as=script

HTTP2 103 Early Hints

```
$ bin/rails s --early-hints
```

3.11.0 - Love Song

 nateberkopec released this 2 hours ago · 1 commit to master since this release



- 2 features:

- HTTP 103 Early Hints ([#1403](#))
- 421/451 status codes now have correct status messages attached ([#1435](#))

Rails 5.2

Add `:allow_other_host` option to `redirect_back` method.
When `allow_other_host` is set to `false`, the `redirect_back` will not allow a redirecting from a different host.
`allow_other_host` is `true` by default.

Tim Masliuchenko

Rails 5.2

Add headless chrome support to System Tests.

Yuji Yaginuma

:headless_chrome

```
# test/application_system_test_case.rb
require "test_helper"

class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
  driven_by :selenium, using: :headless_chrome
end
```

```
# Gemfile
gem "chromedriver-helper"
```

Rails 5.2

Built-in Redis cache store

Jeremy Daer

- ActiveSupport::Cache::FileStore
- ActiveSupport::Cache::MemCacheStore
- ActiveSupport::Cache::MemoryStore
- ActiveSupport::Cache::NullStore
- **ActiveSupport::Cache::RedisCacheStore**

Rails 5.2



Jeremy Daer

@bitsweat

Follow



New in Rails 5.2: SQL injection protection for ``order`` and ``pluck``.

Thanks to [@mastahyeti](#), whose patient, thorough pull request began 9 months ago:



Disallow raw SQL in dangerous AR methods by mastahye...

Some AR methods allow raw SQL where raw SQL is rarely necessary. This opens up applications to SQL injection vulnerabilities when a developer unwittingly passes params or...

[github.com](#)

9:25 AM - 14 Nov 2017

14 Retweets 40 Likes



2



14



40

Disallow raw SQL for #pluck and #order

```
Post.order("LENGTH(title)")
Post.pluck("DISTINCT(title)")
# ActiveSupport::Deprecation.warn(
#   "Dangerous query method (method whose arguments are used as raw " \
#   "SQL) called with non-attribute argument(s): " \
#   "#{unexpected.map(&:inspect).join(", ")}. Non-attribute " \
#   "arguments will be disallowed in Rails 6.0. This method should " \
#   "not be called with user-provided values, such as request " \
#   "parameters or model attributes. Known-safe values can be passed " \
#   "by wrapping them in Arel.sql()."
# )

Post.order(Arel.sql("LENGTH(title)"))
Post.pluck(Arel.sql("DISTINCT(title)"))
Post.order("title") # is OK
Post.pluck("title") # is OK
```

Rails 5.2

Protect from forgery by default

Rather than protecting from forgery in the generated ApplicationController, add it to ActionController::Base depending on ``config.action_controller.default_protect_from_forgery``. This configuration defaults to false to support older versions which have removed it from their ApplicationController, but is set to true for Rails 5.2.

Lisa Ugray

```
class ApplicationController < ActionController::Base
  # Protect from Cross-Site Request Forgery (CSRF)
  protect_from_forgery with: :exception
end
```

Rails 5.2

Active Storage

Active Storage

Active Storage facilitates uploading files to a cloud storage service like Amazon S3, Google Cloud Storage, or Microsoft Azure Storage and attaching those files to Active Record objects. It comes with a local disk-based service for development and testing and supports mirroring files to subordinate services for backups and migrations.

Active Storage

```
$ bin/rails active_storage:install
```

Active Storage

```
# This migration comes from active_storage (originally 20170806125915)
class CreateActiveStorageTables < ActiveRecord::Migration[5.2]
  def change
    create_table :active_storage_blobs do |t|
      t.string :key, null: false
      t.string :filename, null: false
      t.string :content_type
      t.text :metadata
      t.bigint :byte_size, null: false
      t.string :checksum, null: false
      t.datetime :created_at, null: false

      t.index [ :key ], unique: true
    end

    create_table :active_storage_attachments do |t|
      t.string :name, null: false
      t.references :record, null: false, polymorphic: true, index: false
      t.references :blob, null: false

      t.datetime :created_at, null: false

      t.index [ :record_type, :record_id, :name, :blob_id ], name: "index_active_storage_attachments_uniqueness", unique: true
    end
  end
end
```

Active Storage

- `ActiveStorage::Service::AzureStorageService`
- `ActiveStorage::Service::DiskService`
- `ActiveStorage::Service::GCSService`
- `ActiveStorage::Service::MirrorService`
- `ActiveStorage::Service::S3Service`
- AzureStorage, to manage attachments through Microsoft Azure Storage.
- Disk, to manage attachments saved directly on the hard drive.
- GCS, to manage attachments through Google Cloud Storage.
- S3, to manage attachments through Amazon S3.
- Mirror, to be able to use several services to manage attachments.
- `gem 'activestorage-openstack'`

Active Storage

```
# config/storage.yml
test_local:
  service: Disk
  root: <%= Rails.root.join("tmp/storage") %>

local:
  service: Disk
  root: <%= Rails.root.join("storage") %>
```


Active Storage

```
# config/storage.yml
amazon:
  service: S3
  access_key_id: <%= Rails.application.credentials.dig(:aws, :access_key_id) %>
  secret_access_key: <%= Rails.application.credentials.dig(:aws, :secret_access_key) %>
  region: us-east-1
  bucket: your_own_bucket
```

```
# Gemfile
gem "aws-sdk-s3", require: false
```

Active Storage

```
# config/storage.yml
google:
  service: GCS
  project: your_project
  keyfile: <%= Rails.root.join("path/to/gcs.keyfile") %>
  bucket: your_own_bucket
```

```
# Gemfile
gem "google-cloud-storage", "~> 1.3", require: false
```

Active Storage

```
# config/storage.yml
microsoft:
  service: AzureStorage
  path: your_azure_storage_path
  storage_account_name: your_account_name
  storage_access_key: <%= Rails.application.credentials.dig(:azure_storage, :storage_access_key) %>
  container: your_container_name
```

```
# Gemfile
gem "azure-storage", require: false
```

Active Storage

```
# config/environments/test.rb  
config.active_storage.service = :test_local
```

```
# config/environments/development.rb  
config.active_storage.service = :local
```

```
# config/environments/production.rb  
config.active_storage.service = :amazon
```

Active Storage extends AR

- #has_one_attached
- #has_many_attached

Active Storage

```
class User < ApplicationRecord
  has_one_attached :avatar
  has_many_attached :photos
  has_one_attached :video
  has_one_attached :pdf
end
```

Active Storage

```
<%= form.file_field :avatar %>
<%= form.file_field :photos, multiple: true%>
<%= form.file_field :video %>
<%= form.file_field :pdf %>
```

```
@user.avatar.attach(params.require(:avatar))
@user.photos.attach(params.require(:photos))
@user.video.attach(params.require(:video))
@user.pdf.attach(params.require(:pdf))
```

Active Storage

```
<%= image_tag user.avatar %>

# imagemagick
<%= image_tag user.avatar.variant(resize: "100x100") %>

# ffmpeg
<%= image_tag user.video.preview(resize: "300x300") %>

# mutool
<%= image_tag user.pdf.preview(resize: "300x300") %>
```


Active Storage

```
@user.avatar.purge
```

```
@user.photos.purge_later
```

```
@user.video.detach
```

Rails

```
https://github.com/rails/rails  
http://edgeguides.rubyonrails.org  
http://edgeapi.rubyonrails.org
```

Thanks!



Yay! You're on Rails!



Rails version: 5.2.0.alpha
Ruby version: 2.4.2 (x86_64-linux)