



Rails 6

Ankit, Terone and Ros

Growth Session #24 - May 16-17 2019

- Rails 6.0.0 rc1 is released
- Explore Rails 6 Features
- Update our Rails template

Removed

- uglifier
- coffee-rails
- capistrano-rails

Added

- webpacker



Replaced

- mini_magick -> image_processing
- chromedriver-helper -> webdrivers

- Bundle Javascript files
- Configure various plugins to process Javascript:
 - ES6
 - Compress assets

Webpacker - Changes

Webpacker is installed by default



config/

└─ webpack/

├─ environment.js

├─ development.js

├─ production.js

└─ test.js

└─ webpacker.yml

```
# Note: You must restart bin/webpack-dev-server for changes to take effect

default: &default
  source_path: app/javascript
  source_entry_path: packs
  public_root_path: public
  public_output_path: packs
  cache_path: tmp/cache/webpacker
  check_yarn_integrity: false
  webpack_compile_output: false

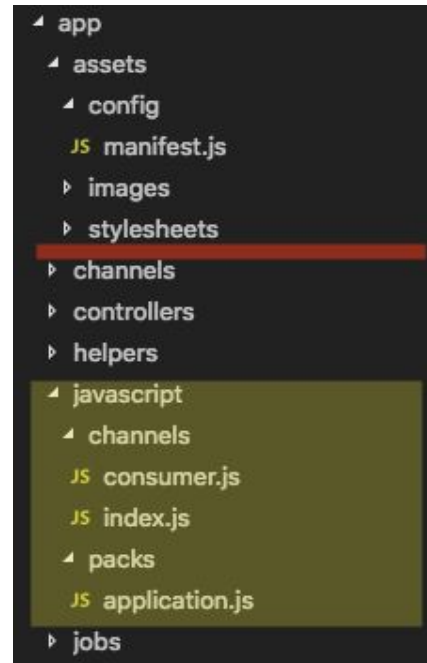
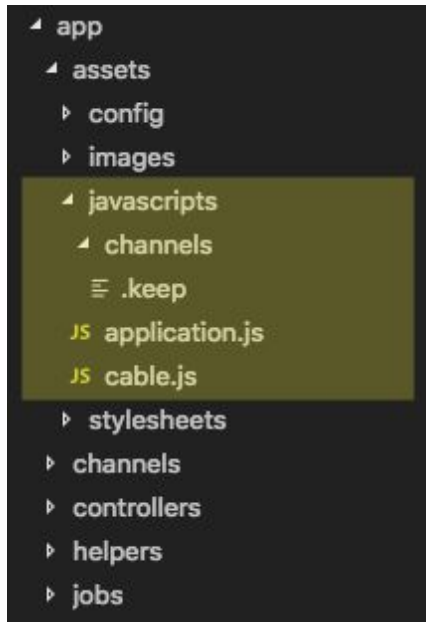
# Additional paths webpack should lookup modules
# ['app/assets', 'engine/foo/app/assets']
resolved_paths: []

# Reload manifest.json on all requests so we reload latest compiled packs
cache_manifest: false

# Extract and emit a css file
extract_css: false

static_assets_extensions:
```

JS folder structure change





Webpacker - Changes

- Sprocket is still used for css and images
- Some JS-related dependencies are added to `package.json`

```
{
  "name": "<%= app_name %>",
  "private": true,
  "dependencies": {
    "rails-ujs": ">=5.2.1" <%= unless options[:skip_turbolinks] %>,
    "turbolinks": "5.1.1" <%= end -%> <%= unless skip_active_storage? %>,
    "activestorage": ">=5.2.1" <%= end -%> <%= unless options[:skip_action_cable] %>,
    "actioncable": ">=5.2.1" <%= end -%>
  },
  "version": "0.1.0"
}
```

Update Rails Template

- Update gems 
- Need to remove some gems as it does not support Rails 6 yet
 - Pronto dirty words
 - Pronto rails schema
 - Ruby Critic
- Update javascript-related part in template
- Revisit again after stable version is released 

Upgrade Rails to 6.0.0.rc1 #111

Edit



rosle wants to merge 2 commits into `master` from `chore/upgrade-rails-6rc1` 

New Features

Rich text content and editing

- Handles embedded images
- Links, Quotes, etc.

Action Text - Why?

- Easy to use
- Simpler Models
- Functions out of the box

```
# /app/models/post.rb
class Post < ApplicationRecord
  has_rich_text :content
end

# /app/controllers/posts_controller.rb
def post_params
  params.require(:post).permit(:content)
end

# /app/views/posts/_form.html.erb
<div class="field">
  <%= form.rich_text_area :content %>
</div>
```

- Routes emails to controller-like mailboxes
- Ingresses for Mailgun, Mandrill, Postmark and SendGrid
- Inbound emails are turned into InboundEmail records using Active Record

Action Mailbox - Why?


Before:

```
class UserMailer < ApplicationMailer
  def receive(email)
    page = Page.find_by(address: email.to.first)
    page.emails.create(
      subject: email.subject,
      body: email.body
    )

    if email.has_attachments?
      email.attachments.each do |attachment|
        page.attachments.create({
          file: attachment,
          description: email.subject
        })
      end
    end
  end
end
```

Action Mailbox - Why?

After:



```
# app/mailboxes/application_mailbox.rb
class ApplicationMailbox < ActionMailbox::Base
  routing /^save@/i      => :forwards
  routing /@replies\./i => :replies
end
```

Action Mailbox - Why?

After:

```
class ApplicationMailbox < ActionMailbox::Base
  routing ->(inbound_email) { with_question?(inbound_email) } => :answer

  def self.with_question?(inbound_email)
    inbound_email.mail.subject.include?('QUESTION') || inbound_email.mail.body.include?('QUESTION')
  end
end

class AnswerFinderMailbox < ApplicationMailbox
  def process
    Reply.with_answer(mail.to.first).deliver_now
  end
end
```

Active Record (New Additions)

- Support for bulk insert queries.
- Multi Database support.
- New methods such as `pick`, `create_or_find_by`, `delete_by`, `destroy_by`, `touch_all`.

Bulk Insert

- Prior to Rails 6, we had to use activerecord-import gem for this purpose
- Single query to insert multiple records
- No validations and Callbacks are run

```
user_list = [{ email: 'ankit@mailinator.com', description: 'new_description 1'},  
             { email: 'ros@mailinator.com',   description: 'new_description 2'},  
             { email: 'thumb@mailinator.com', description: 'new_description 3'}]  
  
User.insert_all(user_list)  
  
INSERT INTO "users"("email","description") VALUES ('ankit@mailinator.com', 'desc1'),  
('ros@mailinator.com', 'desc2'), ('thumb@mailinator.com', 'desc3') ON CONFLICT DO NOTHING RETURNING  
"id"
```

Bulk Insert (upsert_all)

Can update the values in case of conflict



```
user_list = [{email: 'ankit@mailinator.com', description: 'new description'},  
             {email: 'raju@mailinator.com', description: 'desc2'},  
             {email: 'olivier@mailinator.com', description: 'desc2'}]
```

```
User.upsert_all(user_list, unique_by: :email)
```

```
INSERT INTO "users"("email","description") VALUES ('ankit@mailinator.com', 'new description'),  
('raju@mailinator.com', 'desc2'), ('olivier@mailinator.com', 'desc2') ON CONFLICT ("email") DO UPDATE  
SET "description"=excluded."description" RETURNING "id"
```

Multi Database Support

Let's say, we have a model with name `Product` as given below.

```
class Product < ApplicationRecord
  self.abstract_class = true
  connects_to database: { writing: :products_primary, reading: :products_read }
end
```

- This configuration will use **products_primary** database to perform write queries on **products** table
- This configuration will use **products_read** database to perform read queries on **products** table

Multiple Database Connections

connected_to by role

This block method can be used as given below.

```
ActiveRecord::Base.connected_to(role: :reading) do
  Product.first # finds product from replica connected to ApplicationRecord
end
```

It will create a database connection with :reading role.

connected_to by role

This block method can be used as given below.

```
ActiveRecord::Base.connected_to(database: :slow_replica) do
  Product.first
end
```

Action Cable Testing

- Rails 6 adds support for action cable testing
- Although it is not officially released on rspec yet, we can pull from their 4-0-dev branch to try it out.

Changes in gemfile



```
gem 'rails', '6.0.0.rc1'  
gem 'rspec-rails', git: 'https://github.com/rspec/rspec-rails', branch: '4-0-dev'
```

Change adapter to test in cable.yml



```
development:
  adapter: async

test:
  adapter: test

staging:
  adapter: redis
  url: <%= ENV["REDIS_URL"] %>/1
  channel_prefix: pttm-web_staging

production:
  adapter: redis
  url: <%= ENV["REDIS_URL"] %>/1
  channel_prefix: pttm-web_production
```

Example Test

```
def create
  if @bid.save
    auction.reload
    ActionCable.server.broadcast("auction_channel_#{auction.id}", payload)
  else
    flash[:alert] = @bid.errors.full_messages.to_sentence
  end

  redirect_to auction_path(auction)
end
```

```
it 'broadcasts to a channel' do
  auction = Fabricate(:active_auction, starting_price: 1, minimum_step: 0.5)
  user = Fabricate(:user, status: :active, role: :customer)

  sign_in(user)

  expect do
    post :create
  end.to have_broadcasted_to("auction_channel_#{action.id}")
end
```


Conclusion

- Final stable release is yet to come.
- Rails 6 is worth upgrading.
- Github, Basecamp, Shopify are already using Rails 6 rc1, so maybe we can start on this for new projects.

Long Live Rails :D



Thanks!

Contact Nimble

nimblehq.co

hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit
#2402-03, Klong Toei, Wattana, Bangkok
10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower
28 Queen's Road, Central, Hong Kong

