Ruby on Rails 4.1 Release Notes

January 13, 2015

Highlights in Rails 4.1:

- Spring application preloader
- `config/secrets.yml`
- Action Pack variants
- Action Mailer previews

These release notes cover only the major changes. To learn about various bug fixes and changes, please refer to the change logs or check out the list of commits in the main Rails repository on GitHub.

1 Upgrading to Rails 4.1

If you're upgrading an existing application, it's a great idea to have good test coverage before going in. You should also first upgrade to Rails 4.0 in case you haven't and make sure your application still runs as expected before attempting an update to Rails 4.1. A list of things to watch out for when upgrading is available in the Upgrading Ruby on Rails guide.

2 Major Features

2.1 Spring Application Preloader

Spring is a Rails application preloader. It speeds up development by keeping your application running in the background so you don't need to boot it every time you run a test, rake task or migration.

New Rails 4.1 applications will ship with "springified" binstubs. This means that `bin/rails` and `bin/rake` will automatically take advantage of preloaded spring environments.

Running rake tasks:

```
bin/rake test:models
```

Running a Rails command:

```
bin/rails console
```

Spring introspection:

```
$ bin/spring status
Spring is running:

 1182 spring server | my_app | started 29 mins ago
 3656 spring app    | my_app | started 23 secs ago | test mode
 3746 spring app    | my_app | started 10 secs ago | development mode
```

Have a look at the Spring README to see all available features.

See the Upgrading Ruby on Rails guide on how to migrate existing applications to use this feature.

2.2   `config/secrets.yml`

Rails 4.1 generates a new `secrets.yml` file in the `config` folder. By default, this file contains the application's `secret_key_base`, but it could also be used to store other secrets such as access keys for external APIs.

The secrets added to this file are accessible via `Rails.application.secrets`. For example, with the following `config/secrets.yml`:

```
development:
  secret_key_base: 3b7cd727ee24e8444053437c36cc66c3
  some_api_key: SOMEKEY
```

`Rails.application.secrets.some_api_key` returns `SOMEKEY` in the development environment.

See the Upgrading Ruby on Rails guide on how to migrate existing applications to use this feature.

2.3   Action Pack Variants

We often want to render different HTML/JSON/XML templates for phones, tablets, and desktop browsers. Variants make it easy.

The request variant is a specialization of the request format, like `:tablet`, `:phone`, or `:desktop`.

You can set the variant in a `before_action`:

```
request.variant = :tablet if request.user_agent =~ /iPad/
```

Respond to variants in the action just like you respond to formats:

```
respond_to do |format|
  format.html do |html|
    html.tablet # renders app/views/projects/show.html+tablet.erb
    html.phone { extra_setup; render ... }
  end
end
```

Provide separate templates for each format and variant:

```
app/views/projects/show.html.erb
app/views/projects/show.html+tablet.erb
app/views/projects/show.html+phone.erb
```

You can also simplify the variants definition using the inline syntax:

```
respond_to do |format|
  format.js         { render "trash" }
  format.html.phone { redirect_to progress_path }
  format.html.none  { render "trash" }
end
```

2.4   Action Mailer Previews

Action Mailer previews provide a way to see how emails look by visiting a special URL that renders them.

You implement a preview class whose methods return the mail object you'd like to check:

```
class NotifierPreview < ActionMailer::Preview
  def welcome
    Notifier.welcome(User.first)
  end
end
```

The preview is available in `http://localhost:3000/rails/mailers/notifier/welcome`, and a list of them in `http://localhost:3000/rails/mailers`.

By default, these preview classes live in `test/mailers/previews`. This can be configured using the `preview_path` option.

See its documentation for a detailed write up.

2.5   Active Record enums

Declare an enum attribute where the values map to integers in the database, but can be queried by name.

```
class Conversation < ActiveRecord::Base
  enum status: [ :active, :archived ]
end

conversation.archived!
conversation.active? # => false
conversation.status  # => "archived"

Conversation.archived # => Relation for all archived Conversations

Conversation.statuses # => { "active" => 0, "archived" => 1 }
```

See its documentation for a detailed write up.

2.6   Message Verifiers

Message verifiers can be used to generate and verify signed messages. This can be useful to safely transport sensitive data like remember-me tokens and friends.

The method `Rails.application.message_verifier` returns a new message verifier that signs messages with a key derived from secret_key_base and the given message verifier name:

```
signed_token = Rails.application.message_verifier(:remember_me).generate(token)
Rails.application.message_verifier(:remember_me).verify(signed_token) # => token

Rails.application.message_verifier(:remember_me).verify(tampered_token)
# raises ActiveSupport::MessageVerifier::InvalidSignature
```

2.7 Module#concerning

A natural, low-ceremony way to separate responsibilities within a class:

```
class Todo < ActiveRecord::Base
  concerning :EventTracking do
    included do
      has_many :events
    end

    def latest_event
      ...
    end

    private
      def some_internal_method
        ...
      end
  end
end
```

This example is equivalent to defining a `EventTracking` module inline, extending it with `ActiveSupport::Concern`, then mixing it in to the `Todo` class.

See its documentation for a detailed write up and the intended use cases.

2.8 CSRF protection from remote <script> tags

Cross-site request forgery (CSRF) protection now covers GET requests with JavaScript responses, too. That prevents a third-party site from referencing your JavaScript URL and attempting to run it to extract sensitive data.

This means any of your tests that hit `.js` URLs will now fail CSRF protection unless they use `xhr`. Upgrade your tests to be explicit about expecting XmlHttpRequests. Instead of `post :create, format:  :js`, switch to the explicit `xhr :post, :create, format:  :js`.

3 Railties

Please refer to the Changelog for detailed changes.

3.1   Removals

- Removed `update:application_controller` rake task.

- Removed deprecated `Rails.application.railties.engines`.

- Removed deprecated `threadsafe!` from Rails Config.

- Removed deprecated `ActiveRecord::Generators::ActiveModel#update_attributes` in favor of `ActiveRecord ::Generators::ActiveModel#update`.

- Removed deprecated `config.whiny_nils` option.

- Removed deprecated rake tasks for running tests: `rake test:uncommitted` and `rake test:recent`.

3.2   Notable changes

- The Spring application preloader is now installed by default for new applications. It uses the development group of the Gemfile, so will not be installed in production. (Pull Request)

- `BACKTRACE` environment variable to show unfiltered backtraces for test failures. (Commit)

- Exposed `MiddlewareStack#unshift` to environment configuration. (Pull Request)

- Added `Application#message_verifier` method to return a message verifier. (Pull Request)

- The `test_help.rb` file which is required by the default generated test helper will automatically keep your test database up-to-date with `db/schema.rb` (or `db/structure.sql`). It raises an error if reloading the schema does not resolve all pending migrations. Opt out with `config.active_record.maintain _test_schema = false`. (Pull Request)

- Introduce `Rails.gem_version` as a convenience method to return `Gem::Version.new(Rails.version)`, suggesting a more reliable way to perform version comparison. (Pull Request)

4   Action Pack

Please refer to the Changelog for detailed changes.

4.1   Removals

- Removed deprecated Rails application fallback for integration testing, set `ActionDispatch.test_app` instead.

- Removed deprecated `page_cache_extension` config.

- Removed deprecated `ActionController::RecordIdentifier`, use `ActionView::RecordIdentifier` instead.

- Removed deprecated constants from Action Controller:

| Removed | Successor | |:————————————|:————————————| | ActionController::AbstractRequest | ActionDispatch::Request | | ActionController::Request | ActionDispatch::Request | | ActionController::AbstractResponse | ActionDispatch::Response | | ActionController::Response | ActionDispatch::Response | | ActionController::Routing | ActionDispatch::Routing | | ActionController::Integration | ActionDispatch::Integration | | ActionController::IntegrationTe | ActionDispatch::IntegrationTest |

4.2   Notable changes

- `protect_from_forgery` also prevents cross-origin `<script>` tags. Update your tests to use `xhr :get`, `:foo, format:  :js` instead of `get :foo, format:  :js`. (Pull Request)

- `#url_for` takes a hash with options inside an array. (Pull Request)

- Added `session#fetch` method fetch behaves similarly to Hash#fetch, with the exception that the returned value is always saved into the session. (Pull Request)

- Separated Action View completely from Action Pack. (Pull Request)

- Log which keys were affected by deep munge. (Pull Request)

- New config option `config.action_dispatch.perform_deep_munge` to opt out of params "deep munging" that was used to address security vulnerability CVE-2013-0155. (Pull Request)

- New config option `config.action_dispatch.cookies_serializer` for specifying a serializer for the signed and encrypted cookie jars. (Pull Requests 1, 2 / More Details)

- Added `render :plain`, `render :html` and `render :body`. (Pull Request / More Details)

5   Action Mailer

Please refer to the Changelog for detailed changes.

5.1   Notable changes

- Added mailer previews feature based on 37 Signals mail_view gem. (Commit)

- Instrument the generation of Action Mailer messages. The time it takes to generate a message is written to the log. (Pull Request)

6   Active Record

Please refer to the Changelog for detailed changes.

6.1 Removals

- Removed deprecated nil-passing to the following `SchemaCache` methods: `primary_keys`, `tables`, `columns` and `columns_hash`.

- Removed deprecated block filter from `ActiveRecord::Migrator#migrate`.

- Removed deprecated String constructor from `ActiveRecord::Migrator`.

- Removed deprecated `scope` use without passing a callable object.

- Removed deprecated `transaction_joinable=` in favor of `begin_transaction` with a `:joinable` option.

- Removed deprecated `decrement_open_transactions`.

- Removed deprecated `increment_open_transactions`.

- Removed deprecated `PostgreSQLAdapter#outside_transaction?` method. You can use `#transaction_open?` instead.

- Removed deprecated `ActiveRecord::Fixtures.find_table_name` in favor of `ActiveRecord::Fixtures.default_fixture_model_name`.

- Removed deprecated `columns_for_remove` from `SchemaStatements`.

- Removed deprecated `SchemaStatements#distinct`.

- Moved deprecated `ActiveRecord::TestCase` into the Rails test suite. The class is no longer public and is only used for internal Rails tests.

- Removed support for deprecated option `:restrict` for `:dependent` in associations.

- Removed support for deprecated `:delete_sql`, `:insert_sql`, `:finder_sql` and `:counter_sql` options in associations.

- Removed deprecated method `type_cast_code` from Column.

- Removed deprecated `ActiveRecord::Base#connection` method. Make sure to access it via the class.

- Removed deprecation warning for `auto_explain_threshold_in_seconds`.

- Removed deprecated `:distinct` option from `Relation#count`.

- Removed deprecated methods `partial_updates`, `partial_updates?` and `partial_updates=`.

- Removed deprecated method `scoped`.

- Removed deprecated method `default_scopes?`.

- Remove implicit join references that were deprecated in 4.0.

- Removed `activerecord-deprecated_finders` as a dependency. Please see the gem README for more info.

- Removed usage of `implicit_readonly`. Please use `readonly` method explicitly to mark records as `readonly`. (Pull Request)

6.2   Deprecations

- Deprecated `quoted_locking_column` method, which isn't used anywhere.

- Deprecated `ConnectionAdapters::SchemaStatements#distinct`, as it is no longer used by internals. (Pull Request)

- Deprecated `rake db:test:*` tasks as the test database is now automatically maintained. See railties release notes. (Pull Request)

- Deprecate unused `ActiveRecord::Base.symbolized_base_class` and `ActiveRecord::Base.symbolized_sti_name` without replacement. Commit

6.3   Notable changes

- Default scopes are no longer overridden by chained conditions.

Before this change when you defined a `default_scope` in a model it was overridden by chained conditions in the same field. Now it is merged like any other scope. More Details.

- Added `ActiveRecord::Base.to_param` for convenient "pretty" URLs derived from a model's attribute or method. (Pull Request)

- Added `ActiveRecord::Base.no_touching`, which allows ignoring touch on models. (Pull Request)

- Unify boolean type casting for `MysqlAdapter` and `Mysql2Adapter`. `type_cast` will return `1` for `true` and `0` for `false`. (Pull Request)

- `.unscope` now removes conditions specified in `default_scope`. (Commit)

- Added `ActiveRecord::QueryMethods#rewhere` which will overwrite an existing, named where condition. (Commit)

- Extended `ActiveRecord::Base#cache_key` to take an optional list of timestamp attributes of which the highest will be used. (Commit)

- Added `ActiveRecord::Base#enum` for declaring enum attributes where the values map to integers in the database, but can be queried by name. (Commit)

- Type cast json values on write, so that the value is consistent with reading from the database. (Pull Request)

- Type cast hstore values on write, so that the value is consistent with reading from the database. (Commit)

- Make `next_migration_number` accessible for third party generators. (Pull Request)

- Calling `update_attributes` will now throw an `ArgumentError` whenever it gets a `nil` argument. More specifically, it will throw an error if the argument that it gets passed does not respond to to `stringify_keys`. (Pull Request)

- `CollectionAssociation#first`/`#last` (e.g. `has_many`) use a LIMITed query to fetch results rather than loading the entire collection. (Pull Request)

- `inspect` on Active Record model classes does not initiate a new connection. This means that calling `inspect`, when the database is missing, will no longer raise an exception. (Pull Request)

- Removed column restrictions for `count`, let the database raise if the SQL is invalid. (Pull Request)

- Rails now automatically detects inverse associations. If you do not set the `:inverse_of` option on the association, then Active Record will guess the inverse association based on heuristics. (Pull Request)

- Handle aliased attributes in ActiveRecord::Relation. When using symbol keys, ActiveRecord will now translate aliased attribute names to the actual column name used in the database. (Pull Request)

- The ERB in fixture files is no longer evaluated in the context of the main object. Helper methods used by multiple fixtures should be defined on modules included in `ActiveRecord::FixtureSet.context _class`. (Pull Request)

- Don't create or drop the test database if RAILS_ENV is specified explicitly. (Pull Request)

- `Relation` no longer has mutator methods like `#map!` and `#delete_if`. Convert to an `Array` by calling `#to _a` before using these methods. (Pull Request)

- `find_in_batches`, `find_each`, `Result#each` and `Enumerable#index_by` now return an `Enumerator` that can calculate its size. (Pull Request)

- `scope`, `enum` and Associations now raise on "dangerous" name conflicts. (Pull Request, Pull Request)

- `second` through `fifth` methods act like the `first` finder. (Pull Request)

- Make `touch` fire the `after_commit` and `after_rollback` callbacks. (Pull Request)

- Enable partial indexes for `sqlite >= 3.8.0`. (Pull Request)

- Make `change_column_null` revertible. (Commit)

- Added a flag to disable schema dump after migration. This is set to `false` by default in the production environment for new applications. (Pull Request)

7   Active Model

Please refer to the Changelog for detailed changes.

7.1   Deprecations

- Deprecate `Validator#setup`. This should be done manually now in the validator's constructor. (Commit)

7.2   Notable changes

- Added new API methods `reset_changes` and `changes_applied` to `ActiveModel::Dirty` that control changes state.

- Ability to specify multiple contexts when defining a validation. (Pull Request)

- `attribute_changed?` now accepts a hash to check if the attribute was changed `:from` and/or `:to` a given value. (Pull Request)

8   Active Support

Please refer to the Changelog for detailed changes.

8.1   Removals

- Removed `MultiJSON` dependency. As a result, `ActiveSupport::JSON.decode` no longer accepts an options hash for `MultiJSON`. (Pull Request / More Details)

- Removed support for the `encode_json` hook used for encoding custom objects into JSON. This feature has been extracted into the activesupport-json_encoder gem. (Related Pull Request / More Details)

- Removed deprecated `ActiveSupport::JSON::Variable` with no replacement.

- Removed deprecated `String#encoding_aware?` core extensions (`core_ext/string/encoding`).

- Removed deprecated `Module#local_constant_names` in favor of `Module#local_constants`.

- Removed deprecated `DateTime.local_offset` in favor of `DateTime.civil_from_format`.

- Removed deprecated `Logger` core extensions (`core_ext/logger.rb`).

- Removed deprecated `Time#time_with_datetime_fallback`, `Time#utc_time` and `Time#local_time` in favor of `Time#utc` and `Time#local`.

- Removed deprecated `Hash#diff` with no replacement.

- Removed deprecated `Date#to_time_in_current_zone` in favor of `Date#in_time_zone`.

- Removed deprecated `Proc#bind` with no replacement.

- Removed deprecated `Array#uniq_by` and `Array#uniq_by!`, use native `Array#uniq` and `Array#uniq!` instead.

- Removed deprecated `ActiveSupport::BasicObject`, use `ActiveSupport::ProxyObject` instead.

- Removed deprecated `BufferedLogger`, use `ActiveSupport::Logger` instead.

- Removed deprecated `assert_present` and `assert_blank` methods, use `assert object.blank?` and `assert object.present?` instead.

- Remove deprecated `#filter` method for filter objects, use the corresponding method instead (e.g. `#before` for a before filter).

- Removed 'cow' => 'kine' irregular inflection from default inflections. (Commit)

8.2   Deprecations

- Deprecated `Numeric#{ago,until,since,from_now}`, the user is expected to explicitly convert the value into an AS::Duration, i.e. `5.ago` => `5.seconds.ago` (Pull Request)

- Deprecated the require path `active_support/core_ext/object/to_json`. Require `active_support/core_ext/object/json` instead. (Pull Request)

- Deprecated `ActiveSupport::JSON::Encoding::CircularReferenceError`. This feature has been extracted into the activesupport-json_encoder gem. (Pull Request / More Details)

- Deprecated `ActiveSupport.encode_big_decimal_as_string` option. This feature has been extracted into the activesupport-json_encoder gem. (Pull Request / More Details)

- Deprecate custom `BigDecimal` serialization. (Pull Request)

8.3   Notable changes

- `ActiveSupport`'s JSON encoder has been rewritten to take advantage of the JSON gem rather than doing custom encoding in pure-Ruby. (Pull Request / More Details)

- Improved compatibility with the JSON gem. (Pull Request / More Details)

- Added `ActiveSupport::Testing::TimeHelpers#travel` and `#travel_to`. These methods change current time to the given time or duration by stubbing `Time.now` and `Date.today`.

- Added `ActiveSupport::Testing::TimeHelpers#travel_back`. This method returns the current time to the original state, by removing the stubs added by `travel` and `travel_to`. (Pull Request)

- Added `Numeric#in_milliseconds`, like `1.hour.in_milliseconds`, so we can feed them to JavaScript functions like `getTime()`. (Commit)

- Added `Date#middle_of_day`, `DateTime#middle_of_day` and `Time#middle_of_day` methods. Also added `midday`, `noon`, `at_midday`, `at_noon` and `at_middle_of_day` as aliases. (Pull Request)

- Added `Date#all_week/month/quarter/year` for generating date ranges. (Pull Request)

- Added `Time.zone.yesterday` and `Time.zone.tomorrow`. (Pull Request)

- Added `String#remove(pattern)` as a short-hand for the common pattern of `String#gsub(pattern,'')`. (Commit)

- Added `Hash#compact` and `Hash#compact!` for removing items with nil value from hash. (Pull Request)

- `blank?` and `present?` commit to return singletons. (Commit)

- Default the new `I18n.enforce_available_locales` config to `true`, meaning `I18n` will make sure that all locales passed to it must be declared in the `available_locales` list. (Pull Request)

- Introduce `Module#concerning`: a natural, low-ceremony way to separate responsibilities within a class. (Commit)

- Added `Object#presence_in` to simplify value whitelisting. (Commit)

9   Credits

See the full list of contributors to Rails for the many people who spent many hours making Rails, the stable and robust framework it is. Kudos to all of them.

10   Feedback

You're encouraged to help improve the quality of this guide.

Please contribute if you see any typos or factual errors. To get started, you can read our documentation contributions section.

You may also find incomplete content, or stuff that is not up to date. Please do add any missing documentation for master. Make sure to check Edge Guides first to verify if the issues are already fixed or not on the master branch. Check the Ruby on Rails Guides Guidelines for style and conventions.

If for whatever reason you spot something to fix but cannot patch it yourself, please open an issue.

And last but not least, any kind of discussion regarding Ruby on Rails documentation is very welcome in the rubyonrails-docs mailing list.

————————————————————————