Ruby on Rails 6.1 Release Notes

Highlights in Rails 6.1:

- Per-database Connection Switching
- Horizontal Sharding
- Strict Loading Associations
- Delegated Types
- Destroy Associations Async

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

Upgrading to Rails 6.1

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 6.0 in case you haven't and make sure your application still runs as expected before attempting an update to Rails 6.1. A list of things to watch out for when upgrading is available in the Upgrading Ruby on Rails guide.

Major Features

Per-database Connection Switching

Rails 6.1 provides you with the ability to switch connections per-database. In 6.0 if you switched to the `reading` role then all database connections also switched to the reading role. Now in 6.1 if you set `legacy_connection_handling` to `false` in your configuration, Rails will allow you to switch connections for a single database by calling `connected_to` on the corresponding abstract class.

Horizontal Sharding

Rails 6.0 provided the ability to functionally partition (multiple partitions, different schemas) your database but wasn't able to support horizontal sharding (same schema, multiple partitions). Rails wasn't able to support horizontal sharding because models in Active Record could only have one connection per-role per-class. This is now fixed and horizontal sharding with Rails is available.

1

Strict Loading Associations

Strict loading associations allows you to ensure that all your associations are loaded eagerly and stop N+1's before they happen.

Delegated Types

Delegated Types is an alternative to single-table inheritance. This helps represent class hierarchies allowing the superclass to be a concrete class that is represented by its own table. Each subclass has its own table for additional attributes.

Destroy Associations Async

Destroy associations async adds the ability for applications to `destroy` associations in a background job. This can help you avoid timeouts and other performance issues in your application when destroying data.

Railties

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated `rake notes` tasks.

- Remove deprecated `connection` option in the `rails dbconsole` command.

- Remove deprecated `SOURCE_ANNOTATION_DIRECTORIES` environment variable support from `rails notes`.

- Remove deprecated `server` argument from the rails server command.

- Remove deprecated support for using the `HOST` environment variable to specify the server IP.

- Remove deprecated `rake dev:cache` tasks.

- Remove deprecated `rake routes` tasks.

- Remove deprecated `rake initializers` tasks.

Deprecations

Notable changes

Action Cable

Please refer to the Changelog for detailed changes.

Removals

Deprecations

Notable changes

Action Pack

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated `ActionDispatch::Http::ParameterFilter`.
- Remove deprecated `force_ssl` at the controller level.

Deprecations

- Deprecate `config.action_dispatch.return_only_media_type_on_content_type`.

Notable changes

- Change `ActionDispatch::Response#content_type` to return the full Content-Type header.

Action View

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated `escape_whitelist` from `ActionView::Template::Handlers::ERB`.
- Remove deprecated `find_all_anywhere` from `ActionView::Resolver`.
- Remove deprecated `formats` from `ActionView::Template::HTML`.
- Remove deprecated `formats` from `ActionView::Template::RawFile`.
- Remove deprecated `formats` from `ActionView::Template::Text`.
- Remove deprecated `find_file` from `ActionView::PathSet`.
- Remove deprecated `rendered_format` from `ActionView::LookupContext`.
- Remove deprecated `find_file` from `ActionView::ViewPaths`.
- Remove deprecated support to pass an object that is not a `ActionView::LookupContext` as the first argument in `ActionView::Base#initialize`.
- Remove deprecated `format` argument `ActionView::Base#initialize`.
- Remove deprecated `ActionView::Template#refresh`.
- Remove deprecated `ActionView::Template#original_encoding`.

- Remove deprecated `ActionView::Template#variants`.

- Remove deprecated `ActionView::Template#formats`.

- Remove deprecated `ActionView::Template#virtual_path=`.

- Remove deprecated `ActionView::Template#updated_at`.

- Remove deprecated `updated_at` argument required on `ActionView::Template#initialize`.

- Remove deprecated `ActionView::Template.finalize_compiled_template_methods`.

- Remove deprecated `config.action_view.finalize_compiled_template_methods`

- Remove deprecated support to calling `ActionView::ViewPaths#with_fallback` with a block.

- Remove deprecated support to passing absolute paths to `render template:`.

- Remove deprecated support to passing relative paths to `render file:`.

- Remove support to template handlers that don't accept two arguments.

- Remove deprecated pattern argument in `ActionView::Template::PathResolver`.

- Remove deprecated support to call private methods from object in some view helpers.

Deprecations

Notable changes

- Require that `ActionView::Base` subclasses implement `#compiled_method_container`.

- Make `locals` argument required on `ActionView::Template#initialize`.

- The `javascript_include_tag` and `stylesheet_link_tag` asset helpers generate a `Link` header that gives hints to modern browsers about preloading assets. This can be disabled by setting `config.action_view.preload_links_header` to `false`.

Action Mailer

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated `ActionMailer::Base.receive` in favor of Action Mailbox.

Deprecations

Notable changes

Active Record

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated methods from `ActiveRecord::ConnectionAdapters::DatabaseLimits`.

  `column_name_length`   `table_name_length`   `columns_per_table`
  `indexes_per_table` `columns_per_multicolumn_index` `sql_query_length`
  `joins_per_query`

- Remove deprecated `ActiveRecord::ConnectionAdapters::AbstractAdapter#supports_multi_insert`

- Remove deprecated `ActiveRecord::ConnectionAdapters::AbstractAdapter#supports_foreign_keys`

- Remove deprecated `ActiveRecord::ConnectionAdapters::PostgreSQLAdapter#supports_ranges?`.

- Remove deprecated `ActiveRecord::Base#update_attributes` and
  `ActiveRecord::Base#update_attributes!`.

- Remove deprecated `migrations_path` argument in `ActiveRecord::ConnectionAdapter::SchemaStatem`

- Remove deprecated `config.active_record.sqlite3.represent_boolean_as_integer`.

- Remove deprecated methods from `ActiveRecord::DatabaseConfigurations`.

  `fetch each first values []=`

- Remove deprecated `ActiveRecord::Result#to_hash` method.

- Remove deprecated support for using unsafe raw SQL in `ActiveRecord::Relation`
  methods.

Deprecations

- Deprecate `ActiveRecord::Base.allow_unsafe_raw_sql`.

- Deprecate `database` kwarg on `connected_to`.

- Deprecate `connection_handlers` when `legacy_connection_handling` is
  set to false.

Notable changes

- MySQL: Uniqueness validator now respects default database collation, no
  longer enforce case-sensitive comparison by default.

- `relation.create` does no longer leak scope to class-level querying methods
  in initialization block and callbacks.

Before:

```ruby
User.where(name: "John").create do |john|
  User.find_by(name: "David") # => nil
end
```

After:

```ruby
User.where(name: "John").create do |john|
  User.find_by(name: "David") # => #<User name: "David", ...>
end
```

- Named scope chain does no longer leak scope to class-level querying methods.

```ruby
class User < ActiveRecord::Base
  scope :david, -> { User.where(name: "David") }
end
```

  Before:

```ruby
User.where(name: "John").david
# SELECT * FROM users WHERE name = 'John' AND name = 'David'
```

  After:

```ruby
User.where(name: "John").david
# SELECT * FROM users WHERE name = 'David'
```

- `where.not` now generates NAND predicates instead of NOR.

  Before:

```ruby
User.where.not(name: "Jon", role: "admin")
# SELECT * FROM users WHERE name != 'Jon' AND role != 'admin'
```

  After:

```ruby
User.where.not(name: "Jon", role: "admin")
# SELECT * FROM users WHERE NOT (name == 'Jon' AND role == 'admin')
```

- To use the new per-database connection handling applications must change `legacy_connection_handling` to false and remove deprecated accessors on `connection_handlers`. Public methods for `connects_to` and `connected_to` require no changes.

Active Storage

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated support to pass `:combine_options` operations to `ActiveStorage::Transformers::ImageProcessing`.

- Remove deprecated `ActiveStorage::Transformers::MiniMagickTransformer`.

- Remove deprecated `config.active_storage.queue`.

- Remove deprecated `ActiveStorage::Downloading`.

Deprecations

- Deprecate `Blob.create_after_upload` in favor of `Blob.create_and_upload`. (Pull Request)

Notable changes

- Add `Blob.create_and_upload` to create a new blob and upload the given `io` to the service. (Pull Request)
- `ActiveStorage::Blob#service_name` column was added. It is required that a migration is run after the upgrade. Run `bin/rails app:update` to generate that migration.

Active Model

Please refer to the Changelog for detailed changes.

Removals

Deprecations

Notable changes

- Active Model's errors are now objects with an interface that allows your application to more easily handle and interact with errors thrown by models. The feature includes a query interface, enables more precise testing, and access to error details.

Active Support

Please refer to the Changelog for detailed changes.

Removals

- Remove deprecated fallback to `I18n.default_locale` when `config.i18n.fallbacks` is empty.

- Remove deprecated `LoggerSilence` constant.

- Remove deprecated `ActiveSupport::LoggerThreadSafeLevel#after_initialize`.

- Remove deprecated `Module#parent_name`, `Module#parent` and `Module#parents`.

- Remove deprecated file `active_support/core_ext/module/reachable`.

- Remove deprecated file `active_support/core_ext/numeric/inquiry`.

- Remove deprecated file `active_support/core_ext/array/prepend_and_append`.

- Remove deprecated file `active_support/core_ext/hash/compact`.

- Remove deprecated file `active_support/core_ext/hash/transform_values`.

- Remove deprecated file `active_support/core_ext/range/include_range`.

- Remove deprecated `ActiveSupport::Multibyte::Chars#consumes?` and
  `ActiveSupport::Multibyte::Chars#normalize`.

- Remove deprecated `ActiveSupport::Multibyte::Unicode.pack_graphemes`,
  `ActiveSupport::Multibyte::Unicode.unpack_graphemes`, `ActiveSupport::Multibyte::Unicode.no`
  `ActiveSupport::Multibyte::Unicode.downcase`, `ActiveSupport::Multibyte::Unicode.upcase`
  and `ActiveSupport::Multibyte::Unicode.swapcase`.

- Remove deprecated `ActiveSupport::Notifications::Instrumenter#end=`.

Deprecations

- Deprecate `ActiveSupport::Multibyte::Unicode.default_normalization_form`.

Notable changes

Active Job

Please refer to the Changelog for detailed changes.

Removals

Deprecations

- Deprecate `config.active_job.return_false_on_aborted_enqueue`.

Notable changes

- Return `false` when enqueuing a job is aborted.

Action Text

Please refer to the Changelog for detailed changes.

Removals

Deprecations

Notable changes

- Add method to confirm rich text content existence by adding `?` after name
  of the rich text attribute. (Pull Request)

- Add `fill_in_rich_text_area` system test case helper to find a trix editor and fill it with given HTML content. (Pull Request)

- Add `ActionText::FixtureSet.attachment` to generate `<action-text-attachment>` elements in database fixtures. (Pull Request)

Action Mailbox

Please refer to the Changelog for detailed changes.

Removals

Deprecations

- Deprecate `Rails.application.credentials.action_mailbox.api_key` and `MAILGUN_INGRESS_API_KEY` in favor of `Rails.application.credentials.action_mailbox.signing` and `MAILGUN_INGRESS_SIGNING_KEY`.

Notable changes

Ruby on Rails Guides

Please refer to the Changelog for detailed changes.

Notable changes

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.