DO NOT READ THIS FILE ON GITHUB, GUIDES ARE PUBLISHED ON https://guides.rubyonrails.org.

Ruby on Rails 5.0 Release Notes

Highlights in Rails 5.0:

- Action Cable
- Rails API
- Active Record Attributes API
- Test Runner
- Exclusive use of `rails` CLI over Rake
- Sprockets 3
- Turbolinks 5
- Ruby 2.2.2+ required

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 5.0

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 4.2 in case you haven't and make sure your application still runs as expected before attempting an update to Rails 5.0. A list of things to watch out for when upgrading is available in the Upgrading Ruby on Rails guide.

Major Features

Action Cable

Action Cable is a new framework in Rails 5. It seamlessly integrates WebSockets with the rest of your Rails application.

Action Cable allows for real-time features to be written in Ruby in the same style and form as the rest of your Rails application, while still being performant and scalable. It's a full-stack offering that provides both a client-side JavaScript framework and a server-side Ruby framework. You have access to your full domain model written with Active Record or your ORM of choice.

See the Action Cable Overview guide for more information.

API Applications

Rails can now be used to create slimmed down API only applications. This is useful for creating and serving APIs similar to Twitter or GitHub API, that can be used to serve public-facing, as well as, for custom applications.

You can generate a new api Rails app using:

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

This will do three main things:

- Configure your application to start with a more limited set of middleware than normal. Specifically, it will not include any middleware primarily useful for browser applications (like cookies support) by default.

- Make `ApplicationController` inherit from `ActionController::API` instead of `ActionController::Base`. As with middleware, this will leave out any Action Controller modules that provide functionalities primarily used by browser applications.
- Configure the generators to skip generating views, helpers, and assets when you generate a new resource.

The application provides a base for APIs, that can then be configured to pull in functionality as suitable for the application's needs.

See the Using Rails for API-only Applications guide for more information.

Active Record attributes API

Defines an attribute with a type on a model. It will override the type of existing attributes if needed. This allows control over how values are converted to and from SQL when assigned to a model. It also changes the behavior of values passed to `ActiveRecord::Base.where`, which let's use our domain objects across much of Active Record, without having to rely on implementation details or monkey patching.

Some things that you can achieve with this:

- The type detected by Active Record can be overridden.
- A default can also be provided.
- Attributes do not need to be backed by a database column.

```
# db/schema.rb
create_table :store_listings, force: true do |t|
  t.decimal :price_in_cents
  t.string :my_string, default: "original default"
end
```

```
# app/models/store_listing.rb
class StoreListing < ActiveRecord::Base
end
```

```
store_listing = StoreListing.new(price_in_cents: '10.1')

# before
store_listing.price_in_cents # => BigDecimal.new(10.1)
StoreListing.new.my_string # => "original default"

class StoreListing < ActiveRecord::Base
  attribute :price_in_cents, :integer # custom type
  attribute :my_string, :string, default: "new default" # default value
  attribute :my_default_proc, :datetime, default: -> { Time.now } # default value
  attribute :field_without_db_column, :integer, array: true
end

# after
store_listing.price_in_cents # => 10
StoreListing.new.my_string # => "new default"
StoreListing.new.my_default_proc # => 2015-05-30 11:04:48 -0600
```

```
model = StoreListing.new(field_without_db_column: ["1", "2", "3"])
model.attributes # => {field_without_db_column: [1, 2, 3]}
```

Creating Custom Types:

You can define your own custom types, as long as they respond to the methods defined on the value type. The method `deserialize` or `cast` will be called on your type object, with raw input from the database or from your controllers. This is useful, for example, when doing custom conversion, like Money data.

Querying:

When `ActiveRecord::Base.where` is called, it will use the type defined by the model class to convert the value to SQL, calling `serialize` on your type object.

This gives the objects ability to specify, how to convert values when performing SQL queries.

Dirty Tracking:

The type of an attribute is allowed to change how dirty tracking is performed.

See its documentation for a detailed write up.

Test Runner

A new test runner has been introduced to enhance the capabilities of running tests from Rails. To use this test runner simply type `bin/rails test` .

Test Runner is inspired by `RSpec` , `minitest-reporters` , `maxitest` and others. It includes some of these notable advancements:

- Run a single test using line number of test.
- Run multiple tests pinpointing to line number of tests.
- Improved failure messages, which also add ease of re-running failed tests.
- Fail fast using `-f` option, to stop tests immediately on occurrence of failure, instead of waiting for the suite to complete.
- Defer test output until the end of a full test run using the `-d` option.
- Complete exception backtrace output using `-b` option.
- Integration with minitest to allow options like `-s` for test seed data, `-n` for running specific test by name, `-v` for better verbose output and so forth.
- Colored test output.

Railties

Please refer to the Changelog for detailed changes.

Removals

- Removed debugger support, use byebug instead. `debugger` is not supported by Ruby 2.2. (commit)

- Removed deprecated `test:all` and `test:all:db` tasks. (commit)

- Removed deprecated `Rails::Rack::LogTailer` . (commit)

- Removed deprecated `RAILS_CACHE` constant. (commit)

- Removed deprecated `serve_static_assets` configuration. (commit)
- Removed the documentation tasks `doc:app`, `doc:rails`, and `doc:guides`. (commit)
- Removed `Rack::ContentLength` middleware from the default stack. (Commit)

Deprecations

- Deprecated `config.static_cache_control` in favor of `config.public_file_server.headers`. (Pull Request)
- Deprecated `config.serve_static_files` in favor of `config.public_file_server.enabled`. (Pull Request)
- Deprecated the tasks in the `rails` task namespace in favor of the `app` namespace. (e.g. `rails:update` and `rails:template` tasks are renamed to `app:update` and `app:template`.) (Pull Request)

Notable changes

- Added Rails test runner `bin/rails test`. (Pull Request)
- Newly generated applications and plugins get a `README.md` in Markdown. (commit, Pull Request)
- Added `bin/rails restart` task to restart your Rails app by touching `tmp/restart.txt`. (Pull Request)
- Added `bin/rails initializers` task to print out all defined initializers in the order they are invoked by Rails. (Pull Request)
- Added `bin/rails dev:cache` to enable or disable caching in development mode. (Pull Request)
- Added `bin/update` script to update the development environment automatically. (Pull Request)
- Proxy Rake tasks through `bin/rails`. (Pull Request, Pull Request)
- New applications are generated with the evented file system monitor enabled on Linux and macOS. The feature can be opted out by passing `--skip-listen` to the generator. (commit, commit)
- Generate applications with an option to log to STDOUT in production using the environment variable `RAILS_LOG_TO_STDOUT`. (Pull Request)
- Enable HSTS with IncludeSubdomains header for new applications. (Pull Request)
- The application generator writes a new file `config/spring.rb`, which tells Spring to watch additional common files. (commit)
- Added `--skip-action-mailer` to skip Action Mailer while generating new app. (Pull Request)
- Removed `tmp/sessions` directory and the clear rake task associated with it. (Pull Request)
- Changed `_form.html.erb` generated by scaffold generator to use local variables. (Pull Request)
- Disabled autoloading of classes in production environment. (commit)

Action Pack

Please refer to the Changelog for detailed changes.

Removals

- Removed `ActionDispatch::Request::Utils.deep_munge` . (commit)
- Removed `ActionController::HideActions` . (Pull Request)
- Removed `respond_to` and `respond_with` placeholder methods, this functionality has been extracted to the responders gem. (commit)
- Removed deprecated assertion files. (commit)
- Removed deprecated usage of string keys in URL helpers. (commit)
- Removed deprecated `only_path` option on `*_path` helpers. (commit)
- Removed deprecated `NamedRouteCollection#helpers` . (commit)
- Removed deprecated support to define routes with `:to` option that doesn't contain `#` . (commit)
- Removed deprecated `ActionDispatch::Response#to_ary` . (commit)
- Removed deprecated `ActionDispatch::Request#deep_munge` . (commit)
- Removed deprecated `ActionDispatch::Http::Parameters#symbolized_path_parameters` . (commit)
- Removed deprecated option `use_route` in controller tests. (commit)
- Removed `assigns` and `assert_template` . Both methods have been extracted into the rails-controller-testing gem. (Pull Request)

Deprecations

- Deprecated all `*_filter` callbacks in favor of `*_action` callbacks. (Pull Request)
- Deprecated `*_via_redirect` integration test methods. Use `follow_redirect!` manually after the request call for the same behavior. (Pull Request)
- Deprecated `AbstractController#skip_action_callback` in favor of individual skip_callback methods. (Pull Request)
- Deprecated `:nothing` option for `render` method. (Pull Request)
- Deprecated passing first parameter as `Hash` and default status code for `head` method. (Pull Request)
- Deprecated using strings or symbols for middleware class names. Use class names instead. (commit)
- Deprecated accessing mime types via constants (e.g. `Mime::HTML`). Use the subscript operator with a symbol instead (e.g. `Mime[:html]`). (Pull Request)
- Deprecated `redirect_to :back` in favor of `redirect_back` , which accepts a required `fallback_location` argument, thus eliminating the possibility of a `RedirectBackError` . (Pull

Request)

- `ActionDispatch::IntegrationTest` and `ActionController::TestCase` deprecate positional arguments in favor of keyword arguments. (Pull Request)
- Deprecated `:controller` and `:action` path parameters. (Pull Request)
- Deprecated env method on controller instances. (commit)
- `ActionDispatch::ParamsParser` is deprecated and was removed from the middleware stack. To configure the parameter parsers use `ActionDispatch::Request.parameter_parsers=`. (commit, commit)

Notable changes

- Added `ActionController::Renderer` to render arbitrary templates outside controller actions. (Pull Request)
- Migrating to keyword arguments syntax in `ActionController::TestCase` and `ActionDispatch::Integration` HTTP request methods. (Pull Request)
- Added `http_cache_forever` to Action Controller, so we can cache a response that never gets expired. (Pull Request)
- Provide friendlier access to request variants. (Pull Request)
- For actions with no corresponding templates, render `head :no_content` instead of raising an error. (Pull Request)
- Added the ability to override default form builder for a controller. (Pull Request)
- Added support for API-only apps. `ActionController::API` is added as a replacement of `ActionController::Base` for this kind of applications. (Pull Request)
- Make `ActionController::Parameters` no longer inherits from `HashWithIndifferentAccess`. (Pull Request)
- Make it easier to opt in to `config.force_ssl` and `config.ssl_options` by making them less dangerous to try and easier to disable. (Pull Request)
- Added the ability of returning arbitrary headers to `ActionDispatch::Static`. (Pull Request)
- Changed the `protect_from_forgery` prepend default to `false`. (commit)
- `ActionController::TestCase` will be moved to its own gem in Rails 5.1. Use `ActionDispatch::IntegrationTest` instead. (commit)
- Rails generates weak ETags by default. (Pull Request)
- Controller actions without an explicit `render` call and with no corresponding templates will render `head :no_content` implicitly instead of raising an error. (Pull Request 1, 2)
- Added an option for per-form CSRF tokens. (Pull Request)
- Added request encoding and response parsing to integration tests. (Pull Request)

- Add `ActionController#helpers` to get access to the view context at the controller level. (Pull Request)
- Discarded flash messages get removed before storing into session. (Pull Request)
- Added support for passing collection of records to `fresh_when` and `stale?` . (Pull Request)
- `ActionController::Live` became an `ActiveSupport::Concern` . That means it can't be just included in other modules without extending them with `ActiveSupport::Concern` or `ActionController::Live` won't take effect in production. Some people may be using another module to include some special `Warden / Devise` authentication failure handling code as well since the middleware can't catch a `:warden` thrown by a spawned thread which is the case when using `ActionController::Live` . (More details in this issue)
- Introduce `Response#strong_etag=` and `#weak_etag=` and analogous options for `fresh_when` and `stale?` . (Pull Request)

Action View

Please refer to the Changelog for detailed changes.

Removals

- Removed deprecated `AbstractController::Base::parent_prefixes` . (commit)
- Removed `ActionView::Helpers::RecordTagHelper` , this functionality has been extracted to the record_tag_helper gem. (Pull Request)
- Removed `:rescue_format` option for `translate` helper since it's no longer supported by I18n. (Pull Request)

Notable Changes

- Changed the default template handler from `ERB` to `Raw` . (commit)
- Collection rendering can cache and fetches multiple partials at once. (Pull Request, commit)
- Added wildcard matching to explicit dependencies. (Pull Request)
- Make `disable_with` the default behavior for submit tags. Disables the button on submit to prevent double submits. (Pull Request)
- Partial template name no longer has to be a valid Ruby identifier. (commit)
- The `datetime_tag` helper now generates an input tag with the type of `datetime-local` . (Pull Request)
- Allow blocks while rendering with the `render partial:` helper. (Pull Request)

Action Mailer

Please refer to the Changelog for detailed changes.

Removals

- Removed deprecated `*_path` helpers in email views. (commit)
- Removed deprecated `deliver` and `deliver!` methods. (commit)

Notable changes

- Template lookup now respects default locale and I18n fallbacks. (commit)
- Added `_mailer` suffix to mailers created via generator, following the same naming convention used in controllers and jobs. (Pull Request)
- Added `assert_enqueued_emails` and `assert_no_enqueued_emails`. (Pull Request)
- Added `config.action_mailer.deliver_later_queue_name` configuration to set the mailer queue name. (Pull Request)
- Added support for fragment caching in Action Mailer views. Added new config option `config.action_mailer.perform_caching` to determine whether your templates should perform caching or not. (Pull Request)

Active Record

Please refer to the Changelog for detailed changes.

Removals

- Removed deprecated behavior allowing nested arrays to be passed as query values. (Pull Request)
- Removed deprecated `ActiveRecord::Tasks::DatabaseTasks#load_schema`. This method was replaced by `ActiveRecord::Tasks::DatabaseTasks#load_schema_for`. (commit)
- Removed deprecated `serialized_attributes`. (commit)
- Removed deprecated automatic counter caches on `has_many :through`. (commit)
- Removed deprecated `sanitize_sql_hash_for_conditions`. (commit)
- Removed deprecated `Reflection#source_macro`. (commit)
- Removed deprecated `symbolized_base_class` and `symbolized_sti_name`. (commit)
- Removed deprecated `ActiveRecord::Base.disable_implicit_join_references=`. (commit)
- Removed deprecated access to connection specification using a string accessor. (commit)
- Removed deprecated support to preload instance-dependent associations. (commit)
- Removed deprecated support for PostgreSQL ranges with exclusive lower bounds. (commit)
- Removed deprecation when modifying a relation with cached Arel. This raises an `ImmutableRelation` error instead. (commit)
- Removed `ActiveRecord::Serialization::XmlSerializer` from core. This feature has been extracted into the activemodel-serializers-xml gem. (Pull Request)

- Removed support for the legacy `mysql` database adapter from core. Most users should be able to use `mysql2`. It will be converted to a separate gem when we find someone to maintain it. (Pull Request 1, Pull Request 2)

- Removed support for the `protected_attributes` gem. (commit)

- Removed support for PostgreSQL versions below 9.1. (Pull Request)

- Removed support for `activerecord-deprecated_finders` gem. (commit)

- Removed `ActiveRecord::ConnectionAdapters::Column::TRUE_VALUES` constant. (commit)

Deprecations

- Deprecated passing a class as a value in a query. Users should pass strings instead. (Pull Request)

- Deprecated returning `false` as a way to halt Active Record callback chains. The recommended way is to `throw(:abort)`. (Pull Request)

- Deprecated `ActiveRecord::Base.errors_in_transactional_callbacks=`. (commit)

- Deprecated `Relation#uniq` use `Relation#distinct` instead. (commit)

- Deprecated the PostgreSQL `:point` type in favor of a new one which will return `Point` objects instead of an `Array` (Pull Request)

- Deprecated force association reload by passing a truthy argument to association method. (Pull Request)

- Deprecated the keys for association `restrict_dependent_destroy` errors in favor of new key names. (Pull Request)

- Synchronize behavior of `#tables`. (Pull Request)

- Deprecated `SchemaCache#tables`, `SchemaCache#table_exists?` and `SchemaCache#clear_table_cache!` in favor of their new data source counterparts. (Pull Request)

- Deprecated `connection.tables` on the SQLite3 and MySQL adapters. (Pull Request)

- Deprecated passing arguments to `#tables` - the `#tables` method of some adapters (mysql2, sqlite3) would return both tables and views while others (postgresql) just return tables. To make their behavior consistent, `#tables` will return only tables in the future. (Pull Request)

- Deprecated `table_exists?` - The `#table_exists?` method would check both tables and views. To make their behavior consistent with `#tables`, `#table_exists?` will check only tables in the future. (Pull Request)

- Deprecate sending the `offset` argument to `find_nth`. Please use the `offset` method on relation instead. (Pull Request)

- Deprecated `{insert|update|delete}_sql` in `DatabaseStatements`. Use the `{insert|update|delete}` public methods instead. (Pull Request)

- Deprecated `use_transactional_fixtures` in favor of `use_transactional_tests` for more clarity. (Pull Request)

- Deprecated passing a column to `ActiveRecord::Connection#quote`. (commit)

- Added an option `end` to `find_in_batches` that complements the `start` parameter to specify where to stop batch processing. (Pull Request)

Notable changes

- Added a `foreign_key` option to `references` while creating the table. (commit)
- New attributes API. (commit)
- Added `:_prefix`/`:_suffix` option to `enum` definition. (Pull Request, Pull Request)
- Added `#cache_key` to `ActiveRecord::Relation`. (Pull Request)
- Changed the default `null` value for `timestamps` to `false`. (commit)
- Added `ActiveRecord::SecureToken` in order to encapsulate generation of unique tokens for attributes in a model using `SecureRandom`. (Pull Request)
- Added `:if_exists` option for `drop_table`. (Pull Request)
- Added `ActiveRecord::Base#accessed_fields`, which can be used to quickly discover which fields were read from a model when you are looking to only select the data you need from the database. (commit)
- Added the `#or` method on `ActiveRecord::Relation`, allowing use of the OR operator to combine WHERE or HAVING clauses. (commit)
- Added `ActiveRecord::Base.suppress` to prevent the receiver from being saved during the given block. (Pull Request)
- `belongs_to` will now trigger a validation error by default if the association is not present. You can turn this off on a per-association basis with `optional: true`. Also deprecate `required` option in favor of `optional` for `belongs_to`. (Pull Request)
- Added `config.active_record.dump_schemas` to configure the behavior of `db:structure:dump`. (Pull Request)
- Added `config.active_record.warn_on_records_fetched_greater_than` option. (Pull Request)
- Added a native JSON data type support in MySQL. (Pull Request)
- Added support for dropping indexes concurrently in PostgreSQL. (Pull Request)
- Added `#views` and `#view_exists?` methods on connection adapters. (Pull Request)
- Added `ActiveRecord::Base.ignored_columns` to make some columns invisible from Active Record. (Pull Request)
- Added `connection.data_sources` and `connection.data_source_exists?`. These methods determine what relations can be used to back Active Record models (usually tables and views). (Pull Request)
- Allow fixtures files to set the model class in the YAML file itself. (Pull Request)
- Added ability to default to `uuid` as primary key when generating database migrations. (Pull Request)

- Added `ActiveRecord::Relation#left_joins` and `ActiveRecord::Relation#left_outer_joins`. (Pull Request)

- Added `after_{create,update,delete}_commit` callbacks. (Pull Request)

- Version the API presented to migration classes, so we can change parameter defaults without breaking existing migrations, or forcing them to be rewritten through a deprecation cycle. (Pull Request)

- `ApplicationRecord` is a new superclass for all app models, analogous to app controllers subclassing `ApplicationController` instead of `ActionController::Base`. This gives apps a single spot to configure app-wide model behavior. (Pull Request)

- Added ActiveRecord `#second_to_last` and `#third_to_last` methods. (Pull Request)

- Added ability to annotate database objects (tables, columns, indexes) with comments stored in database metadata for PostgreSQL & MySQL. (Pull Request)

- Added prepared statements support to `mysql2` adapter, for mysql2 0.4.4+, Previously this was only supported on the deprecated `mysql` legacy adapter. To enable, set `prepared_statements: true` in `config/database.yml`. (Pull Request)

- Added ability to call `ActionRecord::Relation#update` on relation objects which will run validations on callbacks on all objects in the relation. (Pull Request)

- Added `:touch` option to the `save` method so that records can be saved without updating timestamps. (Pull Request)

- Added expression indexes and operator classes support for PostgreSQL. (commit)

- Added `:index_errors` option to add indexes to errors of nested attributes. (Pull Request)

- Added support for bidirectional destroy dependencies. (Pull Request)

- Added support for `after_commit` callbacks in transactional tests. (Pull Request)

- Added `foreign_key_exists?` method to see if a foreign key exists on a table or not. (Pull Request)

- Added `:time` option to `touch` method to touch records with different time than the current time. (Pull Request)

- Change transaction callbacks to not swallow errors. Before this change any errors raised inside a transaction callback were getting rescued and printed in the logs, unless you used the (newly deprecated) `raise_in_transactional_callbacks = true` option.

  Now these errors are not rescued anymore and just bubble up, matching the behavior of other callbacks. (commit)

Active Model

Please refer to the Changelog for detailed changes.

Removals

- Removed deprecated `ActiveModel::Dirty#reset_#{attribute}` and `ActiveModel::Dirty#reset_changes`. (Pull Request)

- Removed XML serialization. This feature has been extracted into the activemodel-serializers-xml gem. (Pull Request)
- Removed `ActionController::ModelNaming` module. (Pull Request)

Deprecations

- Deprecated returning `false` as a way to halt Active Model and `ActiveModel::Validations` callback chains. The recommended way is to `throw(:abort)`. (Pull Request)
- Deprecated `ActiveModel::Errors#get`, `ActiveModel::Errors#set` and `ActiveModel::Errors#[]=` methods that have inconsistent behavior. (Pull Request)
- Deprecated the `:tokenizer` option for `validates_length_of`, in favor of plain Ruby. (Pull Request)
- Deprecated `ActiveModel::Errors#add_on_empty` and `ActiveModel::Errors#add_on_blank` with no replacement. (Pull Request)

Notable changes

- Added `ActiveModel::Errors#details` to determine what validator has failed. (Pull Request)
- Extracted `ActiveRecord::AttributeAssignment` to `ActiveModel::AttributeAssignment` allowing to use it for any object as an includable module. (Pull Request)
- Added `ActiveModel::Dirty#[attr_name]_previously_changed?` and `ActiveModel::Dirty#[attr_name]_previous_change` to improve access to recorded changes after the model has been saved. (Pull Request)
- Validate multiple contexts on `valid?` and `invalid?` at once. (Pull Request)
- Change `validates_acceptance_of` to accept `true` as default value apart from `1`. (Pull Request)

Active Job

Please refer to the Changelog for detailed changes.

Notable changes

- `ActiveJob::Base.deserialize` delegates to the job class. This allows jobs to attach arbitrary metadata when they get serialized and read it back when they get performed. (Pull Request)
- Add ability to configure the queue adapter on a per job basis without affecting each other. (Pull Request)
- A generated job now inherits from `app/jobs/application_job.rb` by default. (Pull Request)
- Allow `DelayedJob`, `Sidekiq`, `qu`, `que`, and `queue_classic` to report the job id back to `ActiveJob::Base` as `provider_job_id`. (Pull Request, Pull Request, commit)
- Implement a simple `AsyncJob` processor and associated `AsyncAdapter` that queue jobs to a `concurrent-ruby` thread pool. (Pull Request)
- Change the default adapter from inline to async. It's a better default as tests will then not mistakenly come to rely on behavior happening synchronously. (commit)

Active Support

Please refer to the Changelog for detailed changes.

Removals

- Removed deprecated `ActiveSupport::JSON::Encoding::CircularReferenceError` . (commit)
- Removed deprecated methods `ActiveSupport::JSON::Encoding.encode_big_decimal_as_string=` and `ActiveSupport::JSON::Encoding.encode_big_decimal_as_string` . (commit)
- Removed deprecated `ActiveSupport::SafeBuffer#prepend` . (commit)
- Removed deprecated methods from `Kernel` . `silence_stderr` , `silence_stream` , `capture` and `quietly` . (commit)
- Removed deprecated `active_support/core_ext/big_decimal/yaml_conversions` file. (commit)
- Removed deprecated methods `ActiveSupport::Cache::Store.instrument` and `ActiveSupport::Cache::Store.instrument=` . (commit)
- Removed deprecated `Class#superclass_delegating_accessor` . Use `Class#class_attribute` instead. (Pull Request)
- Removed deprecated `ThreadSafe::Cache` . Use `Concurrent::Map` instead. (Pull Request)
- Removed `Object#itself` as it is implemented in Ruby 2.2. (Pull Request)

Deprecations

- Deprecated `MissingSourceFile` in favor of `LoadError` . (commit)
- Deprecated `alias_method_chain` in favour of `Module#prepend` introduced in Ruby 2.0. (Pull Request)
- Deprecated `ActiveSupport::Concurrency::Latch` in favor of `Concurrent::CountDownLatch` from concurrent-ruby. (Pull Request)
- Deprecated `:prefix` option of `number_to_human_size` with no replacement. (Pull Request)
- Deprecated `Module#qualified_const_` in favour of the builtin `Module#const_` methods. (Pull Request)
- Deprecated passing string to define callback. (Pull Request)
- Deprecated `ActiveSupport::Cache::Store#namespaced_key` , `ActiveSupport::Cache::MemCachedStore#escape_key` , and `ActiveSupport::Cache::FileStore#key_file_path` . Use `normalize_key` instead. (Pull Request, commit)
- Deprecated `ActiveSupport::Cache::LocaleCache#set_cache_value` in favor of `write_cache_value` . (Pull Request)
- Deprecated passing arguments to `assert_nothing_raised` . (Pull Request)

- Deprecated `Module.local_constants` in favor of `Module.constants(false)` . (Pull Request)

Notable changes

- Added `#verified` and `#valid_message?` methods to `ActiveSupport::MessageVerifier` . (Pull Request)
- Changed the way in which callback chains can be halted. The preferred method to halt a callback chain from now on is to explicitly `throw(:abort)` . (Pull Request)
- New config option `config.active_support.halt_callback_chains_on_return_false` to specify whether ActiveRecord, ActiveModel, and ActiveModel::Validations callback chains can be halted by returning `false` in a 'before' callback. (Pull Request)
- Changed the default test order from `:sorted` to `:random` . (commit)
- Added `#on_weekend?` , `#on_weekday?` , `#next_weekday` , `#prev_weekday` methods to `Date` , `Time` , and `DateTime` . (Pull Request, Pull Request)
- Added `same_time` option to `#next_week` and `#prev_week` for `Date` , `Time` , and `DateTime` . (Pull Request)
- Added `#prev_day` and `#next_day` counterparts to `#yesterday` and `#tomorrow` for `Date` , `Time` , and `DateTime` . (Pull Request)
- Added `SecureRandom.base58` for generation of random base58 strings. (commit)
- Added `file_fixture` to `ActiveSupport::TestCase` . It provides a simple mechanism to access sample files in your test cases. (Pull Request)
- Added `#without` on `Enumerable` and `Array` to return a copy of an enumerable without the specified elements. (Pull Request)
- Added `ActiveSupport::ArrayInquirer` and `Array#inquiry` . (Pull Request)
- Added `ActiveSupport::TimeZone#strptime` to allow parsing times as if from a given timezone. (commit)
- Added `Integer#positive?` and `Integer#negative?` query methods in the vein of `Integer#zero?` . (commit)
- Added a bang version to `ActiveSupport::OrderedOptions` get methods which will raise an `KeyError` if the value is `.blank?` . (Pull Request)
- Added `Time.days_in_year` to return the number of days in the given year, or the current year if no argument is provided. (commit)
- Added an evented file watcher to asynchronously detect changes in the application source code, routes, locales, etc. (Pull Request)
- Added thread_m/cattr_accessor/reader/writer suite of methods for declaring class and module variables that live per-thread. (Pull Request)
- Added `Array#second_to_last` and `Array#third_to_last` methods. (Pull Request)

- Publish `ActiveSupport::Executor` and `ActiveSupport::Reloader` APIs to allow components and libraries to manage, and participate in, the execution of application code, and the application reloading process. (Pull Request)

- `ActiveSupport::Duration` now supports ISO8601 formatting and parsing. (Pull Request)

- `ActiveSupport::JSON.decode` now supports parsing ISO8601 local times when `parse_json_times` is enabled. (Pull Request)

- `ActiveSupport::JSON.decode` now return `Date` objects for date strings. (Pull Request)

- Added ability to `TaggedLogging` to allow loggers to be instantiated multiple times so that they don't share tags with each other. (Pull Request)

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.