

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

Ruby on Rails 4.0 Release Notes

Highlights in Rails 4.0:

- Ruby 2.0 preferred; 1.9.3+ required
- Strong Parameters
- Turbolinks
- Russian Doll Caching

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 4.0

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 3.2 in case you haven't and make sure your application still runs as expected before attempting an update to Rails 4.0. A list of things to watch out for when upgrading is available in the Upgrading Ruby on Rails guide.

Creating a Rails 4.0 application

```
# You should have the 'rails' RubyGem installed
$ rails new myapp
$ cd myapp
```

Vendoring Gems

Rails now uses a **Gemfile** in the application root to determine the gems you require for your application to start. This **Gemfile** is processed by the Bundler gem, which then installs all your dependencies. It can even install all the dependencies locally to your application so that it doesn't depend on the system gems.

More information: Bundler homepage

Living on the Edge

Bundler and **Gemfile** makes freezing your Rails application easy as pie with the new dedicated **bundle** command. If you want to bundle straight from the Git repository, you can pass the **--edge** flag:

```
$ rails new myapp --edge
```

If you have a local checkout of the Rails repository and want to generate an application using that, you can pass the `--dev` flag:

```
$ ruby /path/to/rails/railties/bin/rails new myapp --dev
```

Major Features

Rails 4.0

Upgrade

- **Ruby 1.9.3** (commit) - Ruby 2.0 preferred; 1.9.3+ required
- **New deprecation policy** - Deprecated features are warnings in Rails 4.0 and will be removed in Rails 4.1.
- **ActionPack page and action caching** (commit) - Page and action caching are extracted to a separate gem. Page and action caching requires too much manual intervention (manually expiring caches when the underlying model objects are updated). Instead, use Russian doll caching.
- **ActiveRecord observers** (commit) - Observers are extracted to a separate gem. Observers are only needed for page and action caching, and can lead to spaghetti code.
- **ActiveRecord session store** (commit) - The ActiveRecord session store is extracted to a separate gem. Storing sessions in SQL is costly. Instead, use cookie sessions, memcache sessions, or a custom session store.
- **ActiveModel mass assignment protection** (commit) - Rails 3 mass assignment protection is deprecated. Instead, use strong parameters.
- **ActiveResource** (commit) - ActiveResource is extracted to a separate gem. ActiveResource was not widely used.
- **vendor/plugins removed** (commit) - Use a **Gemfile** to manage installed gems.

ActionPack

- **Strong parameters** (commit) - Only allow permitted parameters to update model objects (`params.permit(:title, :text)`).
- **Routing concerns** (commit) - In the routing DSL, factor out common sub-routes (`comments` from `/posts/1/comments` and `/videos/1/comments`).
- **ActionController::Live** (commit) - Stream JSON with `response.stream`.
- **Declarative ETags** (commit) - Add controller-level etag additions that will be part of the action etag computation.
- **Russian doll caching** (commit) - Cache nested fragments of views. Each fragment expires based on a set of dependencies (a cache key). The cache key is usually a template version number and a model object.
- **Turbolinks** (commit) - Serve only one initial HTML page. When the user navigates to another page, use `pushState` to update the URL and use

AJAX to update the title and body.

- **Decouple ActionView from ActionController** (commit) - ActionView was decoupled from ActionPack and will be moved to a separated gem in Rails 4.1.
- **Do not depend on ActiveRecord** (commit) - ActionPack no longer depends on ActiveRecord.

General

- **ActiveModel::Model** (commit) - **ActiveModel::Model**, a mixin to make normal Ruby objects to work with ActionPack out of box (ex. for **form_for**)
- **New scope API** (commit) - Scopes must always use callables.
- **Schema cache dump** (commit) - To improve Rails boot time, instead of loading the schema directly from the database, load the schema from a dump file.
- **Support for specifying transaction isolation level** (commit) - Choose whether repeatable reads or improved performance (less locking) is more important.
- **Dalli** (commit) - Use Dalli memcache client for the memcache store.
- **Notifications start & finish** (commit) - ActiveSupport instrumentation reports start and finish notifications to subscribers.
- **Thread safe by default** (commit) - Rails can run in threaded app servers without additional configuration.

NOTE: Check that the gems you are using are threadsafe.

- **PATCH verb** (commit) - In Rails, PATCH replaces PUT. PATCH is used for partial updates of resources.

Security

- **match do not catch all** (commit) - In the routing DSL, match requires the HTTP verb or verbs to be specified.
- **html entities escaped by default** (commit) - Strings rendered in erb are escaped unless wrapped with **raw** or **html_safe** is called.
- **New security headers** (commit) - Rails sends the following headers with every HTTP request: **X-Frame-Options** (prevents clickjacking by forbidding the browser from embedding the page in a frame), **X-XSS-Protection** (asks the browser to halt script injection) and **X-Content-Type-Options** (prevents the browser from opening a jpeg as an exe).

Extraction of features to gems

In Rails 4.0, several features have been extracted into gems. You can simply add the extracted gems to your **Gemfile** to bring the functionality back.

- Hash-based & Dynamic finder methods (GitHub)

- Mass assignment protection in Active Record models (GitHub, Pull Request)
- ActiveRecord::SessionStore (GitHub, Pull Request)
- Active Record Observers (GitHub, Commit)
- Active Resource (GitHub, Pull Request, Blog)
- Action Caching (GitHub, Pull Request)
- Page Caching (GitHub, Pull Request)
- Sprockets (GitHub)
- Performance tests (GitHub, Pull Request)

Documentation

- Guides are rewritten in GitHub Flavored Markdown.
- Guides have a responsive design.

Railties

Please refer to the Changelog for detailed changes.

Notable changes

- New test locations `test/models`, `test/helpers`, `test/controllers`, and `test/mailers`. Corresponding rake tasks added as well. (Pull Request)
- Your app's executables now live in the `bin/` directory. Run `rake rails:update:bin` to get `bin/bundle`, `bin/rails`, and `bin/rake`.
- Threadsafe on by default
- Ability to use a custom builder by passing `--builder` (or `-b`) to `rails new` has been removed. Consider using application templates instead. (Pull Request)

Deprecations

- `config.threadsafe!` is deprecated in favor of `config.eager_load` which provides a more fine grained control on what is eager loaded.
- `Rails::Plugin` has gone. Instead of adding plugins to `vendor/plugins` use gems or bundler with path or git dependencies.

Action Mailer

Please refer to the Changelog for detailed changes.

Notable changes

Deprecations

Active Model

Please refer to the Changelog for detailed changes.

Notable changes

- Add `ActiveModel::ForbiddenAttributesProtection`, a simple module to protect attributes from mass assignment when non-permitted attributes are passed.
- Added `ActiveModel::Model`, a mixin to make Ruby objects work with Action Pack out of box.

Deprecations

Active Support

Please refer to the Changelog for detailed changes.

Notable changes

- Replace deprecated `memcache-client` gem with `dalli` in `ActiveSupport::Cache::MemCacheStore`.
- Optimize `ActiveSupport::Cache::Entry` to reduce memory and processing overhead.
- Inflections can now be defined per locale. `singularize` and `pluralize` accept locale as an extra argument.
- `Object#try` will now return nil instead of raise a `NoMethodError` if the receiving object does not implement the method, but you can still get the old behavior by using the new `Object#try!`.
- `String#to_date` now raises `ArgumentError: invalid date` instead of `NoMethodError: undefined method 'div' for nil:NilClass` when given an invalid date. It is now the same as `Date.parse`, and it accepts more invalid dates than 3.x, such as:

```
# ActiveSupport 3.x
"asdf".to_date # => NoMethodError: undefined method `div' for nil:NilClass
"333".to_date  # => NoMethodError: undefined method `div' for nil:NilClass

# ActiveSupport 4
"asdf".to_date # => ArgumentError: invalid date
"333".to_date  # => Fri, 29 Nov 2013
```

Deprecations

- Deprecate `ActiveSupport::TestCase#pending` method, use `skip` from `minitest` instead.
- `ActiveSupport::Benchmarkable#silence` has been deprecated due to its lack of thread safety. It will be removed without replacement in Rails 4.1.
- `ActiveSupport::JSON::Variable` is deprecated. Define your own `#as_json` and `#encode_json` methods for custom JSON string literals.
- Deprecates the compatibility method `Module#local_constant_names`, use `Module#local_constants` instead (which returns symbols).
- `BufferedLogger` is deprecated. Use `ActiveSupport::Logger`, or the logger from Ruby standard library.
- Deprecate `assert_present` and `assert_blank` in favor of `assert object.blank?` and `assert object.present?`

Action Pack

Please refer to the Changelog for detailed changes.

Notable changes

- Change the stylesheet of exception pages for development mode. Additionally display also the line of code and fragment that raised the exception in all exceptions pages.

Deprecations

Active Record

Please refer to the Changelog for detailed changes.

Notable changes

- Improve ways to write `change` migrations, making the old `up & down` methods no longer necessary.
 - The methods `drop_table` and `remove_column` are now reversible, as long as the necessary information is given. The method `remove_column` used to accept multiple column names; instead use `remove_columns` (which is not revertible). The method `change_table` is also reversible, as long as its block doesn't call `remove`, `change` or `change_default`
 - New method `reversible` makes it possible to specify code to be run when migrating up or down. See the Guide on Migration

- New method `revert` will revert a whole migration or the given block. If migrating down, the given migration / block is run normally. See the Guide on Migration
- Adds PostgreSQL array type support. Any datatype can be used to create an array column, with full migration and schema dumper support.
- Add `Relation#load` to explicitly load the record and return `self`.
- `Model.all` now returns an `ActiveRecord::Relation`, rather than an array of records. Use `Relation#to_a` if you really want an array. In some specific cases, this may cause breakage when upgrading.
- Added `ActiveRecord::Migration.check_pending!` that raises an error if migrations are pending.
- Added custom coders support for `ActiveRecord::Store`. Now you can set your custom coder like this:


```
store :settings, accessors: [ :color, :homepage ], coder: JSON
```
- `mysql` and `mysql2` connections will set `SQL_MODE=STRICT_ALL_TABLES` by default to avoid silent data loss. This can be disabled by specifying `strict: false` in your `database.yml`.
- Remove `IdentityMap`.
- Remove automatic execution of `EXPLAIN` queries. The option `active_record.auto_explain_threshold_in_seconds` is no longer used and should be removed.
- Adds `ActiveRecord::NullRelation` and `ActiveRecord::Relation#none` implementing the null object pattern for the `Relation` class.
- Added `create_join_table` migration helper to create HABTM join tables.
- Allows PostgreSQL hstore records to be created.

Deprecations

- Deprecated the old-style hash based finder API. This means that methods which previously accepted “finder options” no longer do.
- All dynamic methods except for `find_by...` and `find_by...!` are deprecated. Here’s how you can rewrite the code:
 - * ``find_all_by...`` can be rewritten using ``where(...)``.
 - * ``find_last_by...`` can be rewritten using ``where(...).last``.
 - * ``scoped_by...`` can be rewritten using ``where(...)``.
 - * ``find_or_initialize_by...`` can be rewritten using ``find_or_initialize_by(...)``.
 - * ``find_or_create_by...`` can be rewritten using ``find_or_create_by(...)``.
 - * ``find_or_create_by...!`` can be rewritten using ``find_or_create_by!(...)``.

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.