

---

## Combustion

Combustion is a library to help you test your Rails Engines in a simple and effective manner, instead of creating a full Rails application in your spec or test folder.

It allows you to write your specs within the context of your engine, using only the parts of a Rails app you need.

### Usage

Get the gem into either your gemspec or your Gemfile, depending on how you manage your engine's dependencies:

```
1 # gemspec
2 gem.add_development_dependency 'combustion', '~> 1.3'
3
4 # Gemfile
5 gem 'combustion', '~> 1.3'
```

In your `spec_helper.rb`, get Combustion to set itself up - which has to happen before you introduce `rspec/rails` and - if being used - `capybara/rails`. Here's an example within context:

```
1 require 'bundler'
2
3 Bundler.require :default, :development
4
5 # If you're using all parts of Rails:
6 Combustion.initialize! :all
7 # Or, load just what you need:
8 # Combustion.initialize! :active_record, :action_controller
9
10 require 'rspec/rails'
11 # If you're using Capybara:
12 # require 'capybara/rails'
13
14 RSpec.configure do |config|
15   config.use_transactional_fixtures = true
16 end
```

Please note that using `:all` as an argument for `Combustion.initialize!` will load all key parts of Rails that are considered essential for that version. For example: `ActiveJob` is only loaded for Rails 4.2 onwards, and `Sprockets` is only loaded for Rails 3.1-6.1 (as it is no longer part of the default set in 7.0).

You'll also want to run the generator that creates a minimal set of files expected by Rails - run this in the directory of your engine:

---

```
1 combust
2
3 # or, if bundling with the git repo:
4 bundle exec combust
```

Minitest support is considered to be experimental, but it's certainly working for some. Comments on others' experiences are welcome!

What Combustion is doing is setting up a Rails application at `spec/internal` - but you only need to add the files within that directory that you're going to use. Read on for some detail about what that involves.

If you want to use Cucumber, I recommend starting with these notes in issue #16 from Niklas Cathor.

### Configuring a different test app directory

If you want your app to be located somewhere other than `spec/internal`, then make sure you configure it before you call `Combustion.initialize!`:

```
1 Combustion.path = 'spec/dummy'
2 Combustion.initialize! :all
```

### Configuring which Rails modules should be loaded.

By default, Combustion doesn't come with any of the Rails stack. You can customise this though - just pass in what you'd like loaded to the `Combustion.initialize!` call:

```
1 Combustion.initialize! :active_record, :action_controller,
2                       :action_view, :sprockets
```

And then in your engine's Gemfile:

```
1 group :test do
2   gem 'activerecord'
3   gem 'actionpack' # action_controller, action_view
4   gem 'sprockets'
5 end
```

Make sure to specify the appropriate version that you want to use.

ActiveSupport and Railties are always loaded, as they're an integral part of Rails.

---

## Using Models and ActiveRecord

If you're using ActiveRecord, then there are two critical files within your internal Rails app at `spec/internal` that you'll need to modify:

- `config/database.yml`
- `db/schema.rb`

Both follow the same structure as in any normal Rails application - and the schema file lets you avoid migrations, as it gets run whenever the test suite starts. Here's a quick sample (note that tables are overwritten if they already exist - this is necessary):

```
1 ActiveRecord::Schema.define do
2   create_table(:pages, :force => true) do |t|
3     t.string :name
4     t.text   :content
5     t.timestamps
6   end
7 end
```

**Disabling Database Preparation** If you are preparing your own database manually or through different processes, you can disable different parts of the setup process by the following flags: `:database_reset`, `:load_schema`, and `:database_migrate`. All default to true.

```
1 Combustion.initialize! :active_record,
2   :database_reset => false,
3   :load_schema    => false
```

## Configuring Combustion to initialise the test db from a .sql file instead of schema.rb

Name the file `structure.sql` and configure Combustion to use it before initialising:

```
1 Combustion.schema_format = :sql
2 Combustion.initialize! :all
```

Any models that aren't provided by your engine should be located at `spec/internal/app/models`.

## Using ActionController and ActionView

You'll only need to add controllers and views to your internal Rails app for whatever you're testing that your engine doesn't provide - this may be nothing at all, so perhaps you don't even need `spec/internal/app/views` or `spec/internal/app/controllers` directories.

---

However, if you're doing any testing of your engine's controllers or views, then you're going to need routes set up for them - so modify `spec/internal/config/routes.rb` accordingly:

```
1 Rails.application.routes.draw do
2   resources :pages
3 end
```

Just like in a standard Rails app, if you have a mounted engine, then its routes are accessible through whatever it has been loaded as.

### Customizing Rails application settings

If you would like to specify any Rails configuration parameter, you can do it without creating any environment file, simply passing a block to `Combustion.initialize!` like this:

```
1 Combustion.initialize! :all do
2   config.active_record.whitelist_attributes = false
3 end
```

Values given through the `initialize!` block will be set during Rails initialization process, exactly before the corresponding environment file inside `spec/internals/config/environments` is loaded (when that file exists), overriding Combustion's defaults.

Parameters defined in, for instance, `spec/internals/config/environments/test.rb`, would override Combustion's defaults and also config settings passed to `initialize!`.

### Using other Rails-focused libraries

Be aware that other gems may require parts of Rails when they're loaded, and this could cause some issues with Combustion's own setup. You may need to manage the loading yourself by setting `:require` to `false` in your Gemfile for the gem in question, and then requiring it manually in your `spec_helper`. View issue #33 for an example with `FactoryBot`.

### Environment and Logging

Your tests will execute within the test environment for the internal Rails app - and so logs are available at `spec/internal/log/test.log`. You should probably create that log directory so Rails doesn't complain.

---

## Rack it up

Once you've got this set up, you can fire up your test environment quite easily with Rack - a `config.ru` file is provided by the generator. Just run `rackup` and visit `http://localhost:9292`.

## Get your test on!

Now you're good to go - you can write specs within your engine's spec directory just like you were testing a full Rails application - models in `spec/models`, controllers in `spec/controllers`. If you bring Capybara into the mix, then the standard helpers from that will be loaded as well.

```
1 require 'spec_helper'
2
3 describe Page do
4   describe '#valid' do
5     it 'requires a name' do
6       # This is just an example. Go write your own tests!
7     end
8   end
9 end
```

## Compatibility

The current test matrix covers MRI 2.4 to 3.1, and Rails 3.1 to 7.0. It will possibly work on older versions and other Ruby implementations as well.

You can also use Combustion with multiple versions of Rails to test compatibility across them. Appraisal is a gem that can help with this, and a good starting reference is the Thinking Sphinx test suite, which runs against multiple versions of Rails.

## Limitations and Known Issues

Combustion is currently written with the expectation it'll be used with RSpec, but others have got it working with Minitest. I'd love to make this more flexible - if you want to give it a shot before I get around to it, patches are very much welcome.

I've not tried using this with Cucumber, but it should work in theory without too much hassle. Let me know if I'm wrong!

---

## **Contributing**

Please note that this project now has a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

Contributions are very much welcome - but keep in mind the following:

- Keep patches in a separate branch
- Don't mess with the version or history file. I'll take care of that when the patch is merged in.

The tests are extremely minimal, and patches to extend the suite are especially welcome.

## **Credits**

Copyright (c) 2011-2021, Combustion is developed and maintained by Pat Allan, and is released under the open MIT Licence. Many thanks to HyperTiny for encouraging its development, and all who have contributed patches.