


Getting To Know My Enemies


Time and Time Zones in Rails


B'more on Rails

December 11, 2025

Rosanna Speller

 github.com/r-spell

 gitlab.com/r-spell

 linkedin.com/in/rjspeller

- Learned to code in Python & Java
- 8 years working in Rails (with some JS)
 - 6 years as the "Rails Expert" of my team
 - Work at a smallish tech non-profit (launchcode.org)
- This is my first tech talk!

My enemies

- Timezones
- Daylight savings time
- The letter S

Versions

- Rails 8.1
- Ruby 3.4

- An online store

- Products and subscribers

TODO: Link it

date, datetime, time in the database migration

```
class AddColumnsToProducts < ActiveRecord::Migration[8.1]
  def change
    add_column :products, :foo, :date
    add_column :products, :bar, :datetime
    add_column :products, :stuff, :time
  end
end
```

in the database schema

date, datetime, time

in the form helpers

```
<div>
  <%= form.label :foo, style: "display: block" %>
  <%= form.date_field :foo %>
</div>

<div>
  <%= form.label :bar, style: "display: block" %>
  <%= form.datetime_field :bar %>
</div>

<div>
  <%= form.label :stuff, style: "display: block" %>
  <%= form.time_field :stuff %>
</div>
```

date, datetime, time in the form helpers (part 2)

Foo


Bar


Stuff

date, datetime, time in the form helpers (part 3)

Foo

Bar

Stuff

date, datetime, time


in the Rails Console

```
timewarp(dev):010> Product.last
  Product Load (6.3ms)  SELECT "products".* ...
=>
#<Product:0x000000001226dac90
  id: 8,
  created_at: "2025-12-07 04:54:08.975221000 +0000",
  name: "Example Product",
  updated_at: "2025-12-07 04:54:09.007336000 +0000",
  inventory_count: 4,
  foo: "2025-12-17",
  bar: "2025-12-24 12:34:00.0000000000 +0000",
  stuff: "2000-01-01 12:34:00.0000000000 +0000">
timewarp(dev):011>
```


The attributes related to the `datetime` AND `time` form inputs and column types both look like a date and a time.

date, datetime, time in the form helpers (part 3)

Foo

Bar

Stuff

date, datetime, time what about the parameters?

Using the Rails **debug** gem:

```
[45, 53] in ~/timewarp/app/controllers/products_controller.rb
 45|
 46|     def product_params
 47|       x = params.expect(product:
 48|         [ :name, :description, :featured_image, :inventory_count, :foo,
 49|           :bar, :stuff ]
 50|       )
=> 50|       debugger
 51|       x
 52|     end
 53| end

=>#0      ProductsController#product_params at
=> ~/timewarp/app/controllers/products_controller.rb:50
#1      ProductsController#create at
=> ~/timewarp/app/controllers/products_controller.rb:17
# and 81 frames (use `bt' command for all frames)
(rdbg) x
#<ActionController::Parameters {"name" => "Another Example", "description" =>
=> "<div>blah</div>", "inventory_count" => "2", "foo" => "2025-12-17", "bar"
=> => "2025-12-24T12:34", "stuff" => "12:34"} permitted: true>
```

TODO: put footnote about gem

Let's highlight those attributes a little better!

```
#<ActionController::Parameters {"name" => "Another Example",  
  "description" => "<div>blah</div>", "inventory_count" =>  
  "2", "foo" => "2025-12-17", "bar" => "2025-12-24T12:34",  
  "stuff" => "12:34"} permitted: true>
```

TODO: see if can see how looks in DB but I'm not super familiar with sqlite

Classes

Date, DateTime, Time

We looked at the form inputs and column types:

- `date`
- `datetime`
- `time`

There's also Classes:

- `Date`
- `DateTime`
- `Time`

Surely these `Date`, `DateTime` and `Time` classes will have values like those of the `date`, `datetime` and `time` column types, and form helpers right?

Date, DateTime, Time

In Ruby and Rails

`Date`, `DateTime` and `Time` are the each types of Ruby objects.¹

Rails has `ActiveSupport` core extensions for these 3 classes.²

¹Ruby Docs for `Date`, `DateTime` and `Time`

²Rails Docs for `Date`, `DateTime` and `Time`

Date, DateTime, Time

in the console

```
timewarp(dev):013> date = Date.today  
=> Sun, 07 Dec 2025  
timewarp(dev):014> date_time = DateTime.now  
=> Sun, 07 Dec 2025 17:03:13 -0500  
timewarp(dev):015> time = Time.now  
=> 2025-12-07 17:03:17.282426 -0500
```

The instances of `DateTime` AND `Time` both
look like a date and a time.

They look like pretty much the same date and
time even!

Date, DateTime, Time in the console (cont.)

```
timewarp(dev):013> date = Date.today
=> Sun, 07 Dec 2025
timewarp(dev):014> date_time = DateTime.now
=> Sun, 07 Dec 2025 17:03:13 -0500
timewarp(dev):015> time = Time.now
=> 2025-12-07 17:03:17.282426 -0500
timewarp(dev):016> date.acts_like?(:date)
=> true
timewarp(dev):017> date.acts_like?(:time)
=> false
timewarp(dev):018> date_time.acts_like?(:date)
=> true
timewarp(dev):019> date_time.acts_like?(:time)
=> true
timewarp(dev):020> time.acts_like?(:date)
=> false
timewarp(dev):021> time.acts_like?(:time)
=> true
```

`DateTime` is acts like a date and a time, so that must be the best one if I want to give a date and a time, right?

Why is there a `rubocop` that complains when people use `DateTime`?³

`DateTime` must be the best one, right?

³See Rubocop Docs about `DateTime cop`

NO!

DateTime is actually **deprecated!**

Date, DateTime, Time

Prefer Time to DateTime

At least as far back as Ruby 3.0 Ruby says:
DateTime class is considered deprecated. Use
Time class. ⁴

⁴[Ruby Docs](#)

Date, DateTime, Time

Prefer Time to DateTime

At least as far back as Ruby 3.0 Ruby says:
DateTime class is considered deprecated. Use
Time class. ⁵

⁵[Ruby Docs](#)

Date, DateTime, Time

DateTime vs. Time

TODO explain more about DateTime vs Time

date, datetime, time

back in the console

Remember **foo** is a **date**, **bar** is a **datetime** and **stuff** is a **time** in terms of *column type* in the database

```
timewarp(dev):011> product = Product.last
timewarp(dev):012> product.foo.class
=> Date
timewarp(dev):013> product.bar.class
=> ActiveSupport::TimeWithZone
timewarp(dev):014> product.stuff.class
=> ActiveSupport::TimeWithZone
```

Time, TimeZone, and TimeWithZone in the console

```
timewarp(dev):028> now = Time.now  
=> 2025-12-07 18:53:04.5808 -0500  
timewarp(dev):029> now.class  
=> Time  
timewarp(dev):030> current = Time.current  
=> 2025-12-07 23:53:25.035459000 UTC +00:00  
timewarp(dev):031> current.class  
=> ActiveSupport::TimeWithZone
```

Time, TimeZone, and TimeWithZone in the console part 2

```
timewarp(dev):036> now = Time.now  
=> 2025-12-07 18:56:09.391246 -0500  
timewarp(dev):037> current = Time.current  
=> 2025-12-07 23:56:15.552126000 UTC +00:00  
timewarp(dev):038> now.zone  
=> "EST"  
timewarp(dev):039> current.zone  
=> "UTC"
```

Thank you!! Questions?

<https://github.com/piazzai/arguelles>

<https://ctan.org/pkg/beamertheme-arguelles>

TODO citations about using pandoc