

# Migrations

**To Create a Blank Migration:** rails g migration <name>

**To Add Columns:** rails g migration Add<Anything>To<TableName> [columnName:type]

**To Remove Columns:** rails g migration Remove<Anything>From<TableName> [columnName:type]

## Column Options

```
default:      <value>
limit:        30
null:         false
first:        true
after:        :email
unique:       true
```

## Migration Methods\*

```
create_table  change_table
drop_table    add_column
change_column rename_column
remove_column add_index
remove_index
```

## Active Record Supported Types

```
:primary_key :string
:text         :integer
:float        :decimal
:datetime     :timestamp
:time         :date
:binary       :boolean
```

\* See documentation for syntax

## Remove Column

```
rails g migration RemoveAgeFromZombies age:integer
```

```
class RemoveAgeFromZombies < ActiveRecord::Migration
  def up
    remove_column :zombies, :age
  end
  def down
    add_column :zombies, :age, :integer
  end
end
```

## Add Column

```
rails g migration AddEmailToZombies email:string
```

```
class AddEmailToZombies < ActiveRecord::Migration
  def up
    add_column :zombies, :email, :string
  end
  def down
    remove_column :zombies, :email
  end
end
```

## Create Table

```
rails g migration CreateZombiesTable name:string, bio:text, age:integer
```

```
class CreateZombiesTable < ActiveRecord::Migration
  def up
    create_table :zombies do |t|
      t.string :name
      t.text :bio
      t.integer :age
      t.timestamps
    end
  end
  def down
    drop_table :zombies
  end
end
```

## Don't Forget to Rake!

```
$ rake db:migrate
Run all missing migrations

$ rake db:rollback
Rollback the previous migration

$ rake db:setup
Create db, load schema & seed

$ rake db:schema:dump
Dump the current db state
```

db/schema.rb

## Resources:

<http://guides.rubyonrails.org/migrations.html>

# Rails Command Line

Command	Shortcut	Description
rails new <app name>		# creates a new Rails application
rails server	rails s	# starts up the Rails server
rails generate	rails g	# generates template code
rails console	rails c	# starts up the Rails console
rails dbconsole	rails db	# starts up the Rails db console

## Help:

All commands can be run with -h  
for more information

.....

## Generate Examples

```
rails g scaffold zombie name:string bio:text age:integer

rails g migration RemoveEmailFromUser email:string

rails g mailer ZombieMailer decomp_change lost_brain
```

## Named Scope

```
class Zombie < ActiveRecord::Base
  # Ruby 1.9 lambda syntax
  scope :rotting, -> { where(rotting: true) }

  # Also acceptable
  scope :fresh, lambda { where("age < 20") }
  scope :recent, proc { order("created_at desc").limit(3) }
end
```

## Examples

```
Zombie.rotting
Zombie.fresh
Zombie.recent
Zombie.rotting.recent
```

## Callbacks

before_validation	after_validation
before_save	after_save
before_create	after_create
before_update	after_update
before_destroy	after_destroy

```
after_create :send_welcome_email
before_save :encrypt_password
before_destroy :set_deleted_flag
after_update {|zombie| logger.info "Zombie #{zombie.id} updated" }
```

## Self Scope

Reading attributes does not require “self” but setting attributes does require “self”

```
class Zombie < ActiveRecord::Base
  before_save :make_rotting
  def make_rotting
    if age > 20
      self.rotting = true
    end
  end
end
```

## Relationship Options

```
dependent: :destroy
foreign_key: :undead_id
primary_key: :zid
validate: true
```

## Relational Includes Examples\*

```
@zombies = Zombie.includes(:brain).all
@recent = Zombie.recent.includes(:brain)
```

\* To avoid extra queries

# REST & Routes

## Rake Routes ..... ➤ Generates

```
$ rake routes
```

- Prints your RESTful routes

```
zombies      GET /zombies
              POST /zombies
new_zombie    GET /zombies/new
edit_zombie   GET /zombies/:id/edit
zombie        GET /zombies/:id
              PATCH /zombies/:id
              PUT /zombies/:id
              DELETE /zombies/:id
```

```
{:action=>"index", :controller=>"zombies"}
{:action=>"create", :controller=>"zombies"}
{:action=>"new", :controller=>"zombies"}
{:action=>"edit", :controller=>"zombies"}
{:action=>"show", :controller=>"zombies"}
{:action=>"update", :controller=>"zombies"}
{:action=>"update", :controller=>"zombies"}
{:action=>"destroy", :controller=>"zombies"}
```

## Example Link To Usage

```
<%= link_to 'All Zombies', zombies_path %>
<%= link_to 'New Zombie', new_zombie_path %>
<%= link_to 'Edit Zombie', edit_zombie_path(@zombie) %>
<%= link_to 'Show Zombie', zombie_path(@zombie) %>
<%= link_to 'Show Zombie', @zombie %>
<%= link_to 'Delete Zombie', @zombie, method: :delete %>
```

## Relative Paths ... ➤ Path Generated

```
zombies_path    /zombies
new_zombie_path  /zombies/new
```

## Absolute Paths ... ➤ URL Generated

```
zombies_url      http://localhost:3000/zombies
new_zombie_url    http://localhost:3000/zombies/new
```

# Forms

## Example

```
<%= form_for(@zombie) do |f| %>
  <%= f.text_field :name %>
  <%= f.text_area :bio %>

  <%= f.select :decomp, ['fresh', 'rotting', 'stale'] %>
  <%= f.select :decomp, [['fresh', 1], ['rotting', 2], ['stale', 3]] %>

  <%= f.radio_button :decomp, 'fresh', checked: true %>
  <%= f.radio_button :decomp, 'rotting' %>
  <%= f.radio_button :decomp, 'stale' %>

  <%= f.check_box :rotting %>

  <%= f.submit %>
<% end %>
```

## Alternate Text Input Helpers

```
<%= f.password_field :password %>
<%= f.number_field :price %>
<%= f.range_field :quantity %>
<%= f.email_field :email %>
<%= f.url_field :website %>
<%= f.telephone_field :mobile %>
```

# Nested Routes

1

## app/configure/routes.rb

```
TwitterForZombies::Application.routes.draw do
  resources :zombies do
    resources :tweets
  end
end
```

2

## app/controller/tweets\_controller.rb

```
class TweetsController < ApplicationController
  before_action :get_zombie

  def get_zombie
    @zombie = Zombie.find(params[:zombie_id])
  end

  def show
    @tweet = @zombie.tweets.find(params[:id])
  end

  def create
    @tweet = @zombie.tweets.new(params[:tweet])
    if @tweet.save
      redirect_to [@zombie, @tweet]
    else
      render action: "new"
    end
  end

  def index
    @tweets = @zombie.tweets
  end
end
```

/zombies/4/tweets/2  
params = { :zombie\_id => 4, :id => 2 }

/zombies/4/tweets  
params = { :zombie\_id => 4 }

3

## app/views/tweets/\_form.html.erb

```
<%= form_for([@zombie, @tweet]) do |f| %>
```

4

## app/views/tweets/index.html.erb

```
<% @tweets.each do |tweet| %>
  <tr>
    <td><%= tweet.body %></td>
    <td><%= link_to 'Show', [@zombie, tweet] %></td>
    <td><%= link_to 'Edit', edit_zombie_tweet_path(@zombie, tweet) %></td>
    <td><%= link_to 'Destroy', [@zombie, tweet], method: :delete %></td>
  </tr>
<% end %>

<%= link_to 'New Tweet', new_zombie_tweet_path(@zombie) %>
```

## Look Up URL Helpers

```
$ rake routes
```

# View Partials & View Helpers

## app/views/tweets/new.html.erb

```
<h1>New tweet</h1>
<%= render 'form' %> .....
<%= link_to 'Back', zombie_tweets_path(@zombie) %>
```

## app/views/tweets/edit.html.erb

```
<h1>Editing tweet</h1>
<%= render 'form' %>
```

## app/views/tweets/\_form.html.erb

```
<%= form_for(@tweet) do |f| %>
  ...
<%= end %>
```

- Partials start with an underscore

## View Helper

```
<div id="tweet_<%= tweet.id %>" class="tweet">
  <%= tweet.body %>
</div>
```

## Same As

```
<%= div_for tweet do %>
  <%= tweet.body %>
<% end %>
```

## Calls

```
dom_id(@tweet) -> #tweet_2
```

## View Helper

```
<%= @zombies.each do |zombie| %>
  <%= render zombie %>
<% end %>
```

## Same As

```
<%= render @zombies %>
```

## Looks For

```
views/zombies/_zombie.html.erb
```

## Additional Helpers

## Result

```
<%= truncate("I need brains!", :length => 10) %>
```

I need bra...

```
<%= truncate("I need brains!", :length => 10, :separator => ' ') %>
```

I need...

```
I see <%= pluralize(Zombie.count, "zombie") %>
```

I see 2 zombies / I see 1 zombie

```
His name was <%= @zombie.name.titleize %>
```

His name was Ash Williams

```
Ash's zombie roles are <%= @role_names.to_sentence %>
```

Ash's zombie roles are Captain, and Solidier.

```
He was buried alive <%= time_ago_in_words @zombie.created_at %> ago
```

He was buried alive 2 days ago

```
Price is <%= number_to_currency 13.5 %>
```

Price is \$13.50

```
Ash is <%= number_to_human 13234355423 %> years old %>
```

Ash is 13.2 billion years old

# Creating a Mailer

**Generator:** rails g mailer ZombieMailer decomp\_change lost\_brain

## Mailer Class Example - app/mailers/zombie\_mailer.rb

```
class ZombieMailer < ActionMailer::Base
  default from: "from@example.com"

  def decomp_change(zombie)
    @zombie = zombie
    @last_tweet = @zombie.tweets.last

    attachments['z.pdf'] = File.read("#{Rails.root}/public/zombie.pdf")
    mail to: @zombie.email, subject: 'Your decomp stage has changed'
  end
end
```

## Mailer Text Views - app/views/zombie\_mailer/decomp\_change.text.erb

```
Greetings <%= @zombie.name %>
```

## Mailer HTML Views - app/views/zombie\_mailer/decomp\_change.html.erb

```
<h1>Greetings <%= @zombie.name %></h1>
```

## Sending Mail - app/models/zombie.rb

```
ZombieMailer.decomp_change(@zombie).deliver
```

## Additional Options

```
from: my@email.com
cc: my@email.com
bcc: my@email.com
reply_to: my@email.com
```

### Mass Mailing Notes:

Mass mailing is best done outside of Rails. You can use gems for services like MadMimi.com if you plan on sending a lot of mail.

### Resources:

[http://guides.rubyonrails.org/action\\_mailer\\_basics.html](http://guides.rubyonrails.org/action_mailer_basics.html)

# Assets & Asset Paths

## Asset Tag Helpers

```
<%= javascript_include_tag "custom" %>
```

```
<%= stylesheet_link_tag "style" %>
```

```
<%= image_tag "rails.png" %>
```

## Asset Paths in Stylesheets - app/assets/stylesheets/zombie.css.erb

```
form.new_zombie input.submit {
  background-image: url(<%= asset_path('button.png') %>);
}
```

## Using SASS, CoffeeScript Assets

To compile with other CSS/JS helpers, just add the necessary extension.

```
app/assets/stylesheets/zombie.css.scss.erb
app/assets/javascripts/zombies.js.coffee
```

### Resources:

<http://sass-lang.org>  
<http://jashkenas.github.com/coffee-script/>

# CoffeeScript & jQuery

## JavaScript & jQuery

app/assets/javascripts/zombie.js

```
$(document).ready(function() {  
  $('#show-bio').click(function(event) {  
    event.preventDefault();  
    $(this).hide();  
    $('#field#bio').show();  
  })  
})
```

## CoffeeScript & jQuery

app/assets/javascripts/zombie.js.coffee

```
$(document).ready ->  
  $('#show-bio').click (event) ->  
    event.preventDefault()  
    $(this).hide()  
    $('#field#bio').show()
```

## CoffeeScript AJAX Example

```
$(document).ready ->  
  $('#div#custom_phase2 form').submit (event) ->  
    event.preventDefault()  
    url = $(this).attr('action')  
    custom_decomp = $('#div#custom_phase2 #zombie_decomp').val()  
  
  $.ajax  
    type: 'patch'  
    url: url  
    data: { zombie: { decomp: custom_decomp } }  
    dataType: 'json'  
    success: (json) ->  
      $('#decomp').text(json.decomp).effect('highlight')  
      $('#div#custom_phase2').fadeOut() if json.decomp == "Dead (again)"
```

# Sass & CSS

## CSS

app/assets/stylesheets/zombie.css.erb

```
form.new_zombie {  
  border: 1px dashed gray;  
}  
  
form.new_zombie .field#bio {  
  display: none;  
}  
  
form.new_zombie input.submit {  
  background-image: url(<%= asset_path('button.png') %>);  
}
```

## Sass

app/assets/stylesheets/zombie.css.scss.erb

```
form.new_zombie {  
  border: 1px dashed gray;  
}  
  
.field#bio {  
  display: none;  
}  
  
input.submit {  
  background-image: url(<%= asset_path('button.png') %>);  
}
```

## To Remove Sass/CoffeeScript Default Asset Generation

### Gemfile

Remove

```
gem 'sass-rails'  
gem 'coffee-script'
```

then rerun 'bundle install'



# Sprockets & Application.js/.css

## app/assets/javascripts/application.js

Contains a manifest of the JavaScript files we use

```
=====
//= require jquery ..... Looks for jquery.js in all asset paths
//= require jquery_ujs

//= require shared ..... Loads: lib/assets/javascripts/shared.js.coffee
//= require_tree .
```

## app/assets/stylesheet/application.css

Contains a manifest of the stylesheets we use

```
=====
/*
*= require reset
*= require_self ..... Styles in this file are included after the reset stylesheet
*= require_tree .
*/

form.new_zombie {
  border: 1px dashed gray;
}
```

# Rendering / HTTP Status Codes

## Responds to Example

app/controllers/zombies\_controller.rb

```
class ZombiesController < ApplicationController
  def show
    @zombie = Zombie.find(params[:id])
    respond_to do |format|
      format.html do
        if @zombie.decomp == 'Dead (again)'
          render :dead_again ..... Renders app/views/zombies/dead_again.html.erb
        end
        ..... Renders app/views/zombies/show.html.erb
      end
      format.json { render json: @zombie } ..... Renders JSON
    end
  end
end
```

## HTTP Status Codes

200 :ok	401 :unauthorized
201 :created	102 :processing
422 :unprocessable_entity	404 :not_found

## JSON Rendering Examples

```
render json: @zombie.errors, status: :unprocessable_entity
render json: @zombie, status: :created, location: @zombie
```

# Custom Routes

## Types

### :member

acts on a single resource

### :collection

acts on a collection of resources

## Route

get :decomp, on: :member

patch :decay, on: :member

get :fresh, on: :collection

post :search, on: :collection

## URL

/zombies/:id/decomp

/zombies/:id/decay

/zombies/fresh

/zombies/search

## Examples

```
<%= link_to 'Fresh zombies', fresh_zombies_path %>
<%= form_tag(search_zombies_path) do |f| %>
<%= link_to 'Get decomp', decomp_zombie_path(@zombie) %>
<%= form_for @zombie, url: decay_zombie_path(@zombie) %>
```

# Custom JSON Responses

## Examples

```
@zombie.to_json(only: :name)
```

```
{ "name" : "Eric" }
```

```
@zombie.to_json(except: [:created_at, :updated_at, :id, :email, :bio])
```

```
{ "age":25, "decomp":"Fresh", "name":"Eric", "rotting":false }
```

```
@zombie.to_json(only: [:name, :age])
```

```
{ "name" : "Eric", "age": 25 }
```

```
@zombie.to_json(include: :brain, except: [:created_at, :updated_at, :id])
```

```
{
  "age":25,
  "bio":"I am zombified",
  "decomp":"Fresh",
  "email":"zom@bied.com",
  "name":"Eric",
  "rotting":false,
  "brain": { "flavor":"Butter", "status":"Smashed", "zombie_id":3 }
}
```

## 1

### Make a Remote Link or Form Call

```
<%= link_to 'delete', zombie, method: :delete, remote: true %>
```

```
<a href="/zombies/5" data-method="delete" data-remote="true" rel="nofollow">delete</a>
```

```
<%= form_for (@zombie, remote: true) do |f| %>
```

```
<form action="/zombies" data-remote="true" method="post">
```

## 2

### Ensure the Controller Can Accept JavaScript Calls

```
respond_to do |format|  
  format.js  
end
```

## 3

### Write the JavaScript to Return to the Client

app/views/zombies/<action name>.js.erb

```
$('#<%= dom_id(@zombie) %>').fadeOut();
```

### Other jQuery Actions

```
$('#<selector>').append(<content or jQuery object>);  
$('#<selector>').effect('highlight'); .....► Requires jQuery UI  
$('#<selector>').text(<string>);
```