Intro to Rails

#NICAR16

Al Shaw
@A_L

http://j.mp/nicar16-rails

> rails new nicar16
Do you need a web framework?

1. You have too much data to bake
2. You need user input
3. You need dynamic processing that can't be done in the browser
Do you need Rails?

1. Your app is bigger than a couple of pages
2. You have a big or complicated database
3. You need users and sessions
4. Or, you just want a place to organize your work
Sinatra

get '/' do
  "hello world"
end
Rails

Controller
ORM
Views
Assets
Authentication
Migrations
Email
JS

...
OK fine let's install it

> gem install rails

Fetching: activesupport-4.2.6.gem (100%)
Successfully installed activesupport-4.2.6
Fetching: actionview-4.2.6.gem (100%)
Successfully installed actionview-4.2.6
Fetching: actionpack-4.2.6.gem (100%)
Successfully installed actionpack-4.2.6
Fetching: activejob-4.2.6.gem (100%)
Successfully installed activejob-4.2.6
Fetching: actionmailer-4.2.6.gem (100%)
Successfully installed actionmailer-4.2.6
Fetching: activemodel-4.2.6.gem (100%)
Successfully installed activemodel-4.2.6
Fetching: activerecord-4.2.6.gem (100%)
Successfully installed activerecord-4.2.6
Fetching: railties-4.2.6.gem (100%)
Successfully installed railties-4.2.6
Fetching: rails-4.2.6.gem (100%)
Successfully installed rails-4.2.6
9 gems installed

whuu? I said I wanted rails
rails new nicar16

create README.rdoc
create Rakefile
create config.ru
create .gitignore
create Gemfile
create app
create app/assets/javascripts/application.js
create app/assets/stylesheets/application.css
create app/controllers/application_controller.rb
create app/helpers/application_helper.rb
create app/views/layouts/application.html.erb
create app/assets/images/.keep
create app/mailers/.keep
create app/controllers/concerns/.keep
create app/models/.keep
create bin
create bin/bundle
create bin/rails
create bin/rake
create bin/setup
create config
create config/routes.rb
create config/application.rb
create config/environment.rb
create config/secrets.yml
create config/environments
create config/environments/development.rb
create config/environments/production.rb
create config/environments/test.rb
create config/initializers
create config/initializers/assets.rb
create config/initializers/backtrace_silencers.rb
create config/initializers/cookies_serializer.rb
create config/initializers/filter_parameter_logging.rb
create config/initializers/inclusions.rb
create config/initializers/mime_types.rb
create config/initializers/session_store.rb
create config/initializers/wrap_parameters.rb
create config/locales
create config/locales/en.yml
create config/database.yml
create db
create db/seeds.rb
create lib
create lib/tasks
create lib/tasks/.keep
create lib/assets
create lib/assets/.keep
create log
create log/.keep
create public
create public/404.html
create public/422.html
create public/500.html
create public/favicon.ico
create public/robots.txt
create test/fixtures
create test/fixtures/.keep
create test/controllers
create test/controllers/.keep
create test/mailers
create test/mailers/.keep
create test/models
create test/models/.keep
create test/helpers
create test/helpers/.keep
create test/integration
create test/integration/.keep
create test/test_helper.rb
create tmp/cache
create tmp/cache/assets
create vendor/assets/javascripts
create vendor/assets/javascripts/.keep
create vendor/assets/stylesheets
create vendor/assets/stylesheets/.keep

run bundle install
What is all this stuff?

- Gemfile
- Gemfile.lock
- README.rdoc
- Rakefile
- app
- bin
- config
- config.ru
- db
- lib
- log
- public
- test
- tmp
- vendor

90% of your time

- assets
- controllers
- helpers
- mailers
- models
- views

Sublime Text
Stay on the path!

“convention over configuration”

http://guides.rubyonrails.org/
MVC

Model

School

Controller

SchoolsController

View

views/schools/show.html.erb

Router

resources :schools

http://yourapp.com/schools/1

the internet
1. Migrations

> cd nicar16
> rails generate migration AddSchoolsTable

invoke  active_record
create    db/migrate/20160308170855_add_schools_table.rb
1. Migrations

db/migrate/20160308170855_add_schools_table.rb

```ruby
class AddSchoolsTable < ActiveRecord::Migration
  def change
    end
  end
end
```
1. Migrations

db/migrate/20160308170855_add_schools_table.rb

class AddSchoolsTable < ActiveRecord::Migration
def change
  create_table :schools do |t|
    t.string  :name
    t.string  :kind
    t.integer :students
    t.integer :teachers
  end
end
end
1. Migrations

> rake db:migrate

== 20160308170855 AddSchoolsTable: migrating ================================
-- create_table(:schools)
  -> 0.0009s
== 20160308170855 AddSchoolsTable: migrated (0.0009s) =======================
OK, where is this database?

cfg/database.yml

default: &default
  adapter: sqlite3
  pool: 5
  timeout: 5000

development:
  <<: *default
  database: db/development.sqlite3

production:
  <<: *default
  database: db/production.sqlite3

You can use Postgres or MySQL too!

Or, mix 'n match between development and production
2. Add a model

app/models/school.rb

class School < ActiveRecord::Base
end
3. Add some data

> rails console
Loading development environment (Rails 4.2.6)
irb(main):001:0>
3. Add some data

```
irb(main):002:0> s = School.new
=> #<School id: nil, name: nil, kind: nil, students: nil, teachers: nil>
```

![Flickr: John Kannenberg](image)
3. Add some data

```ruby
irb(main):003:0> s.name = "Acme High School"
=> "Acme High School"
irb(main):004:0> s.kind = "high"
=> "high"
irb(main):005:0> s.students = 500
=> 500
irb(main):007:0> s.teachers = 80
=> 80
```
3. Add some data

```ruby
irb(main):008:0> s.save

(0.2ms)  begin transaction
SQL (1.5ms)  INSERT INTO "schools" ("name", "kind", "students", "teachers") VALUES (?, ?, ?, ?)  
[["name", "Acme High School"], ["kind", "high"], ["students", 500], ["teachers", 80]]
(1.6ms)  commit transaction
=> true
```
Another way

irb(main):009:0> School.create({:name => "Magic High School", :kind => "high", :students => 100, :teachers => 10})

(1.1ms) begin transaction
SQL (0.3ms)  INSERT INTO "schools" ("name", "kind", "students", "teachers") VALUES (?, ?, ?, ?) [["name", "Magic High School"], ["kind", "high"], ["students", 100], ["teachers", 10]]
(1.4ms) commit transaction
=> #<School id: 2, name: "Magic High School", kind: "high", students: 100, teachers: 10>
ActiveRecord magic

```ruby
irb(main):010:0> schools = School.all

School Load (0.1ms) SELECT "schools".* FROM "schools"

irb(main):011:0> schools = School.where(:kind => "high")

School Load (0.1ms) SELECT "schools".* FROM "schools" WHERE "schools"."kind" = ? [["kind", "high"]]

irb(main):013:0> schools = School.where("students > ?", 100)

School Load (0.1ms) SELECT "schools".* FROM "schools" WHERE (students > 100)
=> #<ActiveRecord::Relation [#<School id: 1, name: "Acme High School", kind: "high", students: 500, teachers: 80>]
```
ActiveRecord magic

> School.find(1)
> School.first
> School.last
> School.find_by(:name => "Acme High School")
> School.where.not(:kind => "high")
> School.select("name").where(:kind => "high")

http://guides.rubyonrails.org/active_record_querying.html
Let's get it on a web page

app/controllers/schools_controller.rb
app/views/schools/index.html.erb
app/views/schools/show.html.erb
4. Controllers

app/controllers/schools_controller.rb

class SchoolsController < ApplicationController
end
4. Controllers

app/controllers/schools_controller.rb

class SchoolsController < ApplicationController
  def index
    end
  end

  def show
    end
  end
end
4. Controllers

app/controllers/schools_controller.rb

class SchoolsController < ApplicationController
  def index
    @schools = School.all
  end

  def show
    @school = School.find(params[:id])
  end
end
5. Views

app/views/schools/index.html.erb

<ul>
  <% @schools.each do |s| %>
    <li><%= link_to s do %><%= s.name %><% end %></li>
  <% end %>
</ul>
5. Views

app/views/schools/show.html.erb

<h1><%= @school.name %></h1>
<p>Students: <%= @school.students %></p>
<p>Teachers: <%= @school.teachers %></p>
5. Routes: wire it together

config/routes.rb

Rails.application.routes.draw do
  resources :schools
end
Did it work?

> rails server

=> Booting WEBrick
=> Rails 4.2.6 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server

[2016-03-08 13:47:25] INFO  WEBrick 1.3.1
[2016-03-08 13:47:25] INFO  ruby 2.2.2 (2015-04-13) [x86_64-darwin14]
[2016-03-08 13:47:25] INFO  WEBrick::HTTPServer#start: pid=3172 port=3000
Did it work?

> rails server

=> Booting WEBrick
=> Rails 4.2.6 application starting in development on localhost:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server

[2016-03-08 13:47:25] INFO  WEBrick 1.3.1
[2016-03-08 13:47:25] INFO  ruby 2.2.2 (2015-04-13) [x86_64-darwin14]
[2016-03-08 13:47:25] INFO  WEBrick::HTTPServer#start: pid=3172 port=3000
Did it work?

http://localhost:3000/schools
Did it work?

http://localhost:3000/schools/1
Route the homepage to schools#index

cfg/routes.rb

Rails.application.routes.draw do
  root :to => "schools#index"
  resources :schools
end
Now you know Rails!

Flickr: Wally Gobetz
What if I want to load an entire dataset at once?

rake
Rake for data loading

http://j.mp/nicar-rails-csv  →  db/initial/schools.csv
Rake for data loading

lib/tasks/load.rake

desc "load the data"
task :csv => :environment do
  # load code here
end
Rake for data loading

> rake -T csv

rake csv  # load the data

Remember our migration?

> rake -T migrate

rake db:migrate  # Migrate the database (options: VERSION=x, VERBOSE=false, SCOPE=blog)
rake db:migrate:status  # Display status of migrations
desc "load the data"
task :csv => :environment do
  require 'csv'
  csv = "#{Rails.root.to_s}/db/initial/schools.csv"
  CSV.foreach(csv, :headers => true) do |row|
    s = School.new
    row.each do |key,val|
      s[key] = val
    end
    p s
    s.save
  end
end
Rake for data loading

lib/tasks/load.rake

> rake csv

#<School id: nil, name: "Kozey", kind: "elementary", students: 500, teachers: 120>
#<School id: nil, name: "Stoltenberg", kind: "high", students: 2500, teachers: 500>
#<School id: nil, name: "Sipes", kind: "middle", students: 1000, teachers: 200>
#<School id: nil, name: "Gottlieb", kind: "elementary", students: 500, teachers: 120>
#<School id: nil, name: "Boehm", kind: "elementary", students: 500, teachers: 120>
Rake for data loading

lib/tasks/load.rake

> rake csv

#<School id: nil, name: "Kozey", kind: "elementary", students: 500, teachers: 120>
#<School id: nil, name: "Stoltenberg", kind: "high", students: 2500, teachers: 500>
#<School id: nil, name: "Sipes", kind: "middle", students: 1000, teachers: 200>
#<School id: nil, name: "Gottlieb", kind: "elementary", students: 500, teachers: 120>
#<School id: nil, name: "Boehm", kind: "elementary", students: 500, teachers: 120>
Getting a little fancier

What if we want pages for school kind?
app/controllers/schools_controller.rb

class SchoolsController < ApplicationController
  def index
    @schools = School.all
  end

  def show
    @school = School.find(params[:id])
  end

  def kind
    @schools = School.where(:kind => params[:kind])
  end
end
app/views/schools/kind.html.erb

<ul>
  <% @schools.each do |s| %>
    <li><%= link_to s do %><%= s.name %><% end %></li>
  <% end %>
</ul>

Look familiar?
Route it up

cfg/routes.rb

Rails.application.routes.draw do
  match "/schools/kind/:kind", => "schools#kind", :via => "get"
  root :to => "schools#index"
  resources :schools
end
Rails.application.routes.draw do
  match "#/schools/kind/:kind", => "schools#kind", :via => "get"
  root :to => "schools#index"
  resources :schools
end
DRY it up with partials

views/schools/_school_list.html.erb

<ul>
  <% @schools.each do |s| %>
  <li><%= link_to s do %><%= s.name %><% end %></li>
  <% end %>
</ul>
DRY it up with partials

views/schools/index.html.erb
views/schools/kind.html.erb

<%= render :partial => "school_list" %>
<p>See all <%= link_to kind_path("elementary") do %>&gt;Elementary<% end %>,
<%= link_to kind_path("middle") do %>&gt;Middle<% end %> and
<%= link_to kind_path("high") do %>&gt;High<% end %> schools.</p>
<%= render :partial => "school_list" %>
Agh!

**NoMethodError in Schools#index**

Showing /Users/ashaw/nicar16/app/views/schools/index.html.erb where line #1 raised:

undefined method `kind_path' for #<#<Class:0x007fb942002c30>:0x007fb93c7e9418>

Extracted source (around line #1):

```ruby
<p>See all <%= link_to kind_path("elementary") do %>Elementary<% end %>,</p>
<%= link_to kind_path("middle") do %>Middle<% end %> and
<%= link_to kind_path("high") do %>High<% end %> schools.</p>
<%= render :partial => "school_list" %>
```

Rails.root: /Users/ashaw/nicar16

[Application Trace](#)  |  [Framework Trace](#)  |  [Full Trace](#)

app/views/schools/index.html.erb:1:in `app_views_schools_index_html.erb'
Sometimes you have to add the magic yourself

config/routes.rb

Rails.application.routes.draw do
  match "#/schools/kind/:kind", => "schools#kind",
    via => "get", :as => "kind"
  root :to => "schools#index"
  resources :schools
end
Sometimes you have to add the magic yourself

config/routes.rb

Rails.application.routes.draw do
  match "#/schools/kind/:kind", => "schools#kind", :via => "get", :as => "kind"
  root :to => "schools#index"
  resources :schools
end

See all Elementary, Middle and High schools.

- Acme High School
- Magic High School
- Kozey
- Stoltenberg
- Sipes
- Gottlieb
- Boehm
Joins!
Schools have districts, right?
Joins!

Schools have districts, right?

> rails generate migration AddDistrictsTable

  invoke  active_record
  create    db/migrate/20160309183752_add_districts_table.rb
class AddDistrictsTable < ActiveRecord::Migration
  def change
    create_table :districts do |t|
      t.string :name
    end

    add_column :schools, :district_id, :integer
  end
end
app/models/district.rb

class District < ActiveRecord::Base
end
app/models/district.rb

class District < ActiveRecord::Base
  has_many :schools
end

app/models/school.rb

class District < ActiveRecord::Base
  belongs_to :district
end
class Order < ActiveRecord::Base
  belongs_to :customer
end

class Customer < ActiveRecord::Base
  has_many :orders
end
Create a district

irb(main):002:0> d = District.create({:name => "Denver Schools"})
(0.2ms)  begin transaction
  SQL (0.4ms)  INSERT INTO "districts" ("name") VALUES (?)  [["name", "Denver Schools"]]
(0.7ms)  commit transaction
=> #<District id: 1, name: "Denver Schools">

Flickr: John Kannenberg
Add a school to our new district

irb(main):002:0> s = School.find(1)
irb(main):003:0> s.district = d
irb(main):004:0> s.save
irb(main):005:0> s
=> #<School id: 1, name: "Acme High School", kind: "high", students: 500, teachers: 80, district_id: 1>

Under the hood:

SQL (0.4ms) UPDATE "schools" SET "district_id" = ? WHERE "schools"."id" = ? [["district_id", 1], ["id", 1]]
Get all the schools in a district

```ruby
irb(main):001:0> District.first.schools
=> [#<ActiveRecord::Associations::CollectionProxy [...>
```

Under the hood:

- **District Load (0.7ms)**
  ```sql
  SELECT "districts". (*) FROM "districts"
  ORDER BY "districts"."id" ASC LIMIT 1
  ```

- **School Load (0.6ms)**
  ```sql
  SELECT "schools". (*) FROM "schools" WHERE "schools"."district_id" = ?
  ```

Watch out, joins can get expensive!
You can figure out how to add controllers and views for districts, right?

app/controllers/districts_controller.rb
app/views/districts/index.html.erb
app/views/districts/show.html.erb
config/routes.rb
class DistrictsController < ApplicationController
  def show
    @district = District.find(params[:id])
    @schools = @district.schools
  end
end
We basically just made this
We basically just made this
Next steps...

guides.rubyonrails.org
Next steps...

*Pluralsight screencasts (formerly PeepCode)*

[Website Screenshot](https://www.pluralsight.com/courses/ruby-on-rails3-fundamentals)
Thank you!

@A_L
al.shaw@propublica.org