

Using Padrino

introduction to a simple Ruby framework

About me



amazing team
Rails app, full-stack
connecting brands and influencers



co-founder
travelling platform for vegans & vegetarians
lifestyle blog



co-organizer
electronics, 3d printing, Internet of Things
many cities, more to come



Rails workshops
started with them 2 years ago
now trying to pay back by coaching

Story of trying Ruby web frameworks

project:
to build personal website

Episode 1:

Rails

\$ rails new my_awesome_page

app folder, bin files, config, lib,
logs, public, tests, vendor folder....

assets, controllers, concerns,
helpers, mailers, models, views...

```
anita@hot-pink-dot ~/Dev/various [master *]
± % rails new rails_devfest2014
create
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/models/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
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/inflections.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/boot.rb
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
Fetching gem metadata from https://rubygems.org/.....
```

Google is your best friend: “ruby frameworks”

The Ruby Toolbox

Home ► Web Apps, Services & Interaction ► Web App Frameworks

Web App Frameworks

📊 84.454



Ruby on Rails

📊 49.694



Rack

📊 22.616



Sinatra

📊 3.451



Padrino

📊 1.582



Cramp

Episode 2: Sinatra

1. `gem install sinatra`
2. create `.rb` file
3. `ruby my_awesome_website.rb`
4. go to `http://localhost:4567/`

```
my_awesome_website.rb x
1  require 'sinatra'
2
3  get '/' do
4    "Hello World, it's Anita here!"
5  end
6
```

```
± % ruby my_awesome_website.rb
== Sinatra/1.4.5 has taken the stage on 4567 for development with backup from Thin
Thin web server (v1.6.3 codename Protein Powder)
Maximum connections set to 1024
Listening on localhost:4567, CTRL+C to stop
127.0.0.1 - - [21/Nov/2014 17:02:08] "GET / HTTP/1.1" 200 29 0.0065
127.0.0.1 - - [21/Nov/2014 17:02:08] "GET /favicon.ico HTTP/1.1" 404 448 0.0012
█
```



```
get '/' do
  haml :index
end

get '/about' do
  haml :about
end

get '/contact' do
  haml :contact
end

__END__

@@index
%html
  %body
    %h1 Welcome to Our Site

@@about
%html
  %body
    %h1 About Our Site

@@contact
%html
  %body
    %h1 Contact Us
```

super easy to store views
and templates

structure of files needs to
be build manually

tons of options through
gems

light, fast and 100%
custom app

Episode 3:
Let's get (a little) crazy

- static pages
- blog
- gallery of funny lamas
- contact form



http://www.funnyjunk.com/funny_pictures/3079665/My+little+lama

“Writing a more complex Sinatra application (or rather, an application using, amongst other things, Sinatra) requires the developer to spend way more time thinking about architecture and tooling. This is both an advantage a disadvantage.”

Konstantin Haase

source: <http://www.sitepoint.com/rails-or-sinatra-the-best-of-both-worlds/>



[CODE](#) [BLOG](#) [GUIDES](#) [API](#) [CONTRIBUTE](#) [CHANGES](#) [TEAM](#)

The Elegant Ruby Web Framework



```
gem install padrino
padrino g project myapp -d datamapper -b
cd myapp
padrino g admin
padrino rake dm:migrate seed
padrino start
```

SINATRA CORE
DROP-IN ADMIN
LIGHTWEIGHT
AGNOSTIC
COMPREHENSIVE

"Padrino is a ruby framework built upon the Sinatra web library. (...) Padrino was created to make it **fun and easy to code** more advanced web applications while still adhering to the spirit that makes Sinatra great!"

Episode 4:
Happy ending

1. `$ gem install padrino`
2. `$ padrino g project my_awesome_app -d activerecord`
3. `$ bundle install`
4. `$ padrino start`
5. go to `http://localhost:3000/`

```
├─ app
|   ├── controllers
|   ├── helpers
|   └─ views
|       └─ layouts
├─ config
├─ public
|   ├── images
|   ├── javascripts
|   └─ stylesheets
└─ tmp
```

Sinatra doesn't know this ditty.



Try this:

```
# in app.rb
class FirstPadrinoApp::App
  get '/' do
    "Hello World"
  end
end
```

1. \$ padrino g admin
2. \$ bundle install
3. \$ padrino rake ar:create
4. \$ padrino rake ar:migrate
5. \$ padrino rake seed

PADRINOAccounts

ListNewEdit

Name:

Foo

Ex: a simple text

Surname:

Bar

Ex: a simple text

Email:

anita.jaszewska@gmail.com

Ex: a simple text

Password:

Ex: a simple text

Password confirmation:

Ex: a simple text

Role:

admin

Ex: a simple text

SaveSave and continueCancel

1. \$ padrino g model post title:string body:text -a app
2. \$ padrino rake ar:migrate
3. \$ padrino g controller posts get:index get:show
4. following MVC pattern - create views
5. \$ padrino g admin_page post

PADRINO

PostsAccounts

List

New

Edit

Title:

I'm a first post, yay!

Ex: a simple text

Body:

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale. Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

Nori grape silver beet broccoli kombu beet greens fava bean potato quandong celery. Bunya nuts black-eyed pea prairie turnip leek lentil turnip greens parsnip. Sea lettuce lettuce water chestnut eggplant winter purslane fennel azuki bean earthnut pea sierra leone bologi leek soko chicory celtuce parsley jicama salsify.

Ex: a simple text

SaveSave and continueCancel

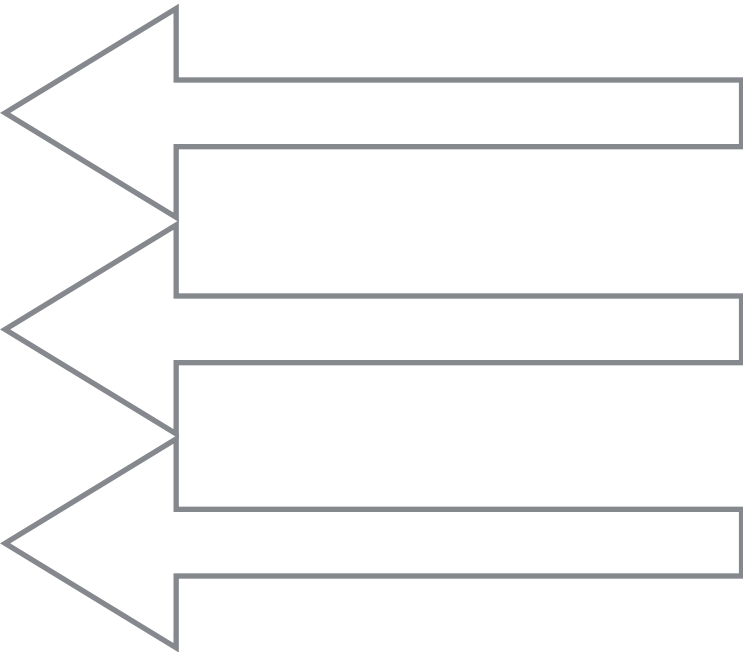

```

  first_padrino_app
  |
  |-- admin
  |   |-- controllers
  |   |   |-- accounts.rb
  |   |   |-- base.rb
  |   |   |-- posts.rb
  |   |   |-- sessions.rb
  |   |-- views
  |   |   |-- accounts
  |   |   |-- base
  |   |   |-- errors
  |   |   |-- layouts
  |   |   |-- posts
  |   |   |-- sessions
  |   |-- app.rb
  |
  |-- app
  |   |-- controllers
  |   |   |-- posts.rb
  |   |-- helpers
  |   |-- models
  |   |   |-- post.rb
  |   |-- views
  |   |   |-- layouts
  |   |   |-- posts
  |   |   |   |-- _post.haml
  |   |   |   |-- index.haml
  |   |   |   |-- show.haml
  |   |-- app.rb
  |
  |-- bin
  |
  |-- config
  |
  |-- db
  |   |-- migrate
  |   |   |-- 001_create_accounts.rb
  |   |   |-- 002_create_posts.rb
  |   |   |-- first_padrino_app_development.db
  |   |-- schema.rb
  |   |-- seeds.rb
  |
  |-- lib

```



admin files



controller

model

views



db migrations

PRETTY NEAT



memegenerator.net

**BUT THERE IS
MORE**

```
# app/controllers/session.rb
```

```
post :create do
```

```
  email do
```

```
    from "tony@reyes.com"
```

```
    to "john@smith.com"
```

```
    subject "Welcome!"
```

```
    body "Body"
```

```
    via :sendmail
```

```
  end
```

```
end
```

Class List

[Classes](#) | [Methods](#) | [Files](#)

Search:

- ▶ [Admin](#)
- [Application](#) < Base
- ▶ [Cache](#)
- ▶ [Generators](#)
- ▼ [Helpers](#)
 - [AssetTagHelpers](#)
 - [FormHelpers](#)**
 - [FormatHelpers](#)
 - [NumberHelpers](#)
 - ▶ [OutputHelpers](#)
 - [RenderHelpers](#)
 - [TagHelpers](#)
 - [TranslationHelpers](#)
- ▶ [Logger](#) < Object
- ▶ [Mailer](#)
- [Mounter](#) < Object
- ▶ [Reloader](#)
- ▶ [Rendering](#)
- [Router](#) < Object
- ▶ [Routing](#)
- [Server](#) < Server
- [Tasks](#)

[Libraries](#) » [padrino/padrino-framework \(master\)](#) » [Index \(F\)](#) » [Padrino](#) » [Helpers](#) » [FormHelpers](#)

Module: Padrino::Helpers::FormHelpers

Defined in: `padrino-helpers/lib/padrino-helpers/form_helpers.rb`

Overview

Helpers related to producing form related tags and inputs into templates.

Instance Method Summary

[collapse](#)

`#button_tag(caption, options = {}) ⇒ String`

Constructs a button input from the given options.

`#checkbox_tag(name, options = {}) ⇒ String`

Constructs a check_box from the given options.

`#error_message_on(object, field, options = {}) ⇒ String`

Returns a string containing the error message attached to the method on the object if one exists.

`#error_messages_for(*objects, options = {}) ⇒ String`

Constructs list html for the errors for a given symbol.

`#field_set_tag(*args, &block) ⇒ String`

Constructs a field_set to group fields with given options.

`#fields_for(object, settings = {}, &block) ⇒ String`

Constructs form fields for an object using given or default form_builder Used within an existing form to allow alternate objects within one form.

`#file_field_tag(name, options = {}) ⇒ String`

Constructs a file field input from the given options

ALL Padrino projects are automatically mountable into other Padrino applications

from basic level

```
Padrino.mount("blog").to("/blog")  
Padrino.mount("website").to("/website")  
Padrino.mount("app").to("/")
```

to advanced support

```
# Adds support for matching an app to a host string or pattern  
Padrino.mount("Blog").to("/").host("blog.example.org")  
Padrino.mount("Admin").host("admin.example.org")  
Padrino.mount("WebSite").host(/.*\.?example.org/)  
Padrino.mount("Foo").to("/foo").host("bar.example.org")
```

Epilogue:
Padrino is cool

So, what is cool about Padrino?

- structure between Rails and Sinatra
- clear and neat routing
- generators and templates to create apps fast
- built-in, good looking admin
- helpers that make life easier
- simple mailer
- mountable projects

anita.jaszewska@gmail.com
@anitajaszewska
letsleaveitempty

