

Padrino

The elegant web framework

<http://www.padrinorb.com/>



Who am I?

Who am I?



Joshua Hull
@joshbuddy

Who am I?

<http://www.github.com/joshbuddy/padrino-presentation>

Joshua Hull
@joshbuddy

Why you should care

Why you should care

Built on top of Sinatra

Why you should care

Built on top of Sinatra

```
require 'rubygems'  
require 'sinatra'  
get '/hi' do  
  "Hello World!"  
end
```

Why you should care

Built on top of Sinatra

Search for "Sinatra inspired web frameworks"

Language	Name
JavaScript	Sammy
Scala	Scalatra
Perl	Mojolicious
Python	itty
.Net	Martin
Haskell	Bird
Java	play
PHP	Fitzgerald

Why you should care

Sinatra is awesome!

Minimalism

DIY

Take what you need

Easy to learn

Why you should care

Sinatra needs help!

Keeps re-inventing the same thing

There are common patterns

Why you should care

Sinatra needs help!

"Using partials in your views is a great way to keep them clean. Since Sinatra takes the hands off approach to framework design, you'll have to implement a partial handler yourself."

– Sinatra Book

What is Padrino?

Common set of Sinatra extensions

What is Padrino?

Generation	Create Padrino applications, models, controllers
Routing	Full url named routes, named params, respond_to, before/after filter support.
Output Helpers	Content helpers such as: content_for, capture_html, concat_content
Tag Helpers	View helpers such as: tag, content_tag, input_tag
Asset Helpers	View helpers such as: link_to, image_tag
Text Helpers	Useful formatting like: relative_time_ago, js_escape_html
Form Helpers	Builder support such as: form_tag, form_for, field_set_tag
Mailer	Fast and simple delivery support for sending emails (akin to ActionMailer)
Admin	Built-in admin interface (akin to Django) with authentication
Logging	Provide a unified logger that can interact with your ORM or any library
Reloading	Automatically reloads server code during development
Localization	Full support for I18n localization

What is Padrino?

Built from Sinatra, leverages existing extensions

Built on top of Rack, leverages existing middleware

Use what you want, leave what you don't

What you get

MVC structure, similar to Rails

Generators to build the common first steps

Easy to migrate from Sinatra to Padrino

Generation

Generation

Component	Default	Aliases	Options
orm	none	-d	mongomapper, mongoid, activerecord, datamapper, couchrest, ohm, mongomatic
test	none	-t	bacon, shoulda, cucumber, testspec, riot, rspec
script	none	-s	prototype, rightjs, jquery, mootools, extcore, dojo
rendering	haml	-e	erb, haml, erubis
stylesheet	none	-c	sass, less, scss
mock	none	-m	rr, mocha

Generation

sequel + rspec

```
$ padrino g project catalog -d sequel -t rspec  
$ cd catalog  
$ padrino g model product
```

activerecord + riot

```
$ padrino g project catalog -d activerecord -t riot  
$ cd catalog  
$ padrino g model product
```

Generation

activerecord + riot

padrino g model product

```
app/models/product.rb
```

```
class Product < ActiveRecord::Base  
  
end
```

```
test/models/product_test.rb
```

```
require File.expand_path(File.dirname(__FILE__) + '/../test_config.rb')  
  
context "Product Model" do  
  context 'can be created' do  
    setup do  
      Product.new  
    end  
  
    asserts("that record is not nil") { !topic.nil? }  
  end  
end
```

Generation

sequel + rspec

padrino g model product

```
app/models/product.rb
```

```
class Product < Sequel::Model  
  
end
```

```
spec/models/product_spec.rb
```

```
require File.expand_path(File.dirname(__FILE__) + '/../spec_helper.rb')  
  
describe "Product Model" do  
  it 'can be created' do  
    @product = Product.new  
    @product.should_not be_nil  
  end  
end
```

Generation

padrino g project tiny-proj -i

```
app/app.rb
app/controllers.rb
app/helpers.rb
app/mailers.rb
app/views
app/views/mailers
config/apps.rb
config/boot.rb
config.ru
Gemfile
public
public/favicon.ico
```

Generation

padrino g project normal-proj

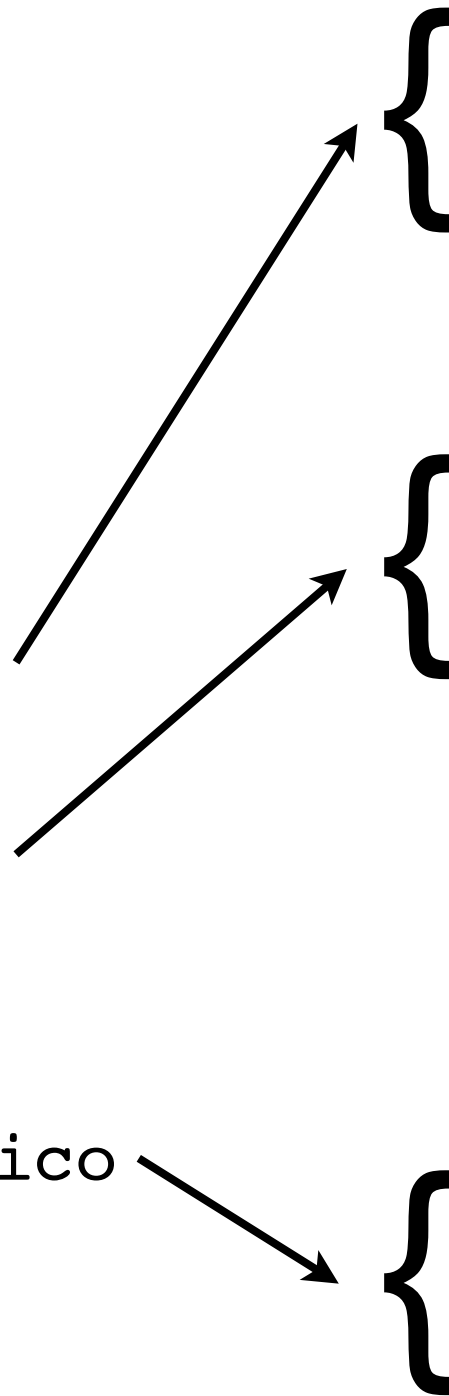
```
app/app.rb  
app/controllers  
app/helpers  
config/apps.rb  
config/boot.rb  
config.ru  
Gemfile  
public/favicon.ico
```

Padrino application

app/app.rb
app/controllers
app/helpers
config/apps.rb
config/boot.rb
config.ru
Gemfile
public/favicon.ico

Rails application

{
app/controllers/application_controller.rb
app/helpers/application_helper.rb
app/models
app/views
app/views/layouts
config/boot.rb
config/database.yml
config/environment.rb
config/environments/development.rb
config/environments/production.rb
config/environments/test.rb
config/initializers/backtrace_silencers.rb
config/initializers/inflections.rb
config/initializers/mime_types.rb
config/initializers/new_rails_defaults.rb
config/initializers/session_store.rb
config/locales/en.yml
config/routes.rb
db/seeds.rb
public/404.html
public/422.html
public/500.html
public/favicon.ico
public/images
public/images/rails.png
public/index.html
public/javascripts
public/javascripts/application.js
public/javascripts/controls.js
public/javascripts/dragdrop.js
public/javascripts/effects.js
public/javascripts/prototype.js
public/robots.txt
public/stylesheets
Rakefile



Generation

Controller

```
$ padrino g controller Product get:index get:new post:create
```

Generation

Controller

```
$ padrino g controller Product get:index get:new post:create
```

```
Catalog.controllers :product do
```

```
  get :index do  
  end
```

```
  get :new do  
  end
```

```
  post :create do  
  end
```

```
end
```


Generation

Model

```
$ padrino g model User name:string age:integer email:string
```

Generation

Model

```
$ padrino g model User name:string age:integer email:string
```

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.string :name
      t.integer :age
      t.string :email
    end
  end

  def self.down
    drop_table :users
  end
end
```

Generation

Migration

```
$ padrino g migration AddFieldsToUsers last_login:datetime crypted_password:string
```

Generation

Migration

```
$ padrino g migration AddFieldsToUsers last_login:datetime crypted_password:string
```

```
class AddFieldsToUsers < ActiveRecord::Migration
  def self.up
    change_table :users do |t|
      t.datetime :last_login
      t.string :crypted_password
    end
  end

  def self.down
    change_table :users do |t|
      t.remove :last_login
      t.remove :crypted_password
    end
  end
end
```

Generation

Mailer

```
$ padrino g mailer UserNotifier confirm_account welcome inactive_account
```

Generation

Mailer

```
$ padrino g mailer UserNotifier confirm_account welcome inactive_account
```

```
Catalog.mailer :user_notifier do

  email :confirm_account do
    # Your mailer code goes here
  end

  email :welcome do
    # Your mailer code goes here
  end

  email :inactive_account do
    # Your mailer code goes here
  end

end
```

Generation

Sub Application

Generation

Sub Application

```
$ padrino g project catalog  
$ padrino g app one  
$ padrino g app two
```


Generation

Sub Application

\$ padrino g project catalog → /
\$ padrino g app one → /one
\$ padrino g app two → /two

Generation

Sub Application

\$ padrino g project catalog → /
\$ padrino g app one → /one
\$ padrino g app two → /two

\$ padrino g controller base --app one
\$ padrino g controller base --app two
\$ padrino g controller base
\$ padrino g mailer mail --app one
\$ padrino g mailer mail --app two
\$ padrino g mailer mail

Routing

Routing

```
Catalog.controllers do
  get '/' do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get '/' do
    # ...
  end
end
```



```
Catalog.controllers do
  get :index do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get '/' do
    # ...
  end
end
```



`url(:index) # "/"`

```
Catalog.controllers do
  get :index do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get '/account/:id' do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get '/account/:id' do
    # ...
  end
end
```



```
Catalog.controllers do
  get :account, :with => :id do
    # ...
  end
end
```


Routing

```
Catalog.controllers do
  get '/account/:id' do
    # ...
  end
end
```



```
url(:account, :id => 3)
# "/account/3"
```

```
Catalog.controllers do
  get :account, :with => :id do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get '/account/:id' do
    # ...
  end
end
```



```
url(:account, 3)
# "/account/3"
```

```
Catalog.controllers do
  get :account, :with => :id do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get :account, :map => "/account/:id" do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get :account, :map => "/account/:id" do
    # ...
  end
end
```

```
Catalog.controllers do
  get :account, :map => "/account/id/:id" do
    # ...
  end
end
```

```
Catalog.controllers do
  get :account, :map => "/account-:id" do
    # ...
  end
end
```



Routing

```
Catalog.controllers do
  get :index do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get :index do
    # ...
  end
end
```

Routing

```
Catalog.controllers do
  get :index do
    # ...
  end
end
```

Namespacing
Provides
Filters
Nesting
Layouts

Routing

Namespacing

```
Catalog.controllers do  
  get :index do  
    # ...  
  end  
end
```


Routing

Namespacing

```
Catalog.controllers :admin do  
  get :index do  
    # ...  
  end  
end
```

Routing

Namespacing

```
Catalog.controllers :admin do  
  get :index do  
    # ...  
  end  
end
```

```
url(:admin, :index) # "/admin/index"
```

Routing

Namespacing

```
Catalog.controllers do
  controller :admin do
    get :index
    # ...
  end
end
end
```

```
url(:admin, :index) # "/admin/index"
```

Routing

Namespacing

```
Catalog.controllers do
  controller :admin do
    get :index
    # ...
    url(:admin, :index)
    # "/admin/index"
  end
end

controller :user do
  get :index
  # ...
  url(:user, :index)
  # "/user/index"
end
```

Routing

Provides

Routing

Provides

```
Catalog.controllers :admin do  
  get :index do  
    # ...  
  end  
end
```

Routing

Provides

```
Catalog.controllers :admin do
  get :index, :provides => :js do
    # ... return some javascript
  end
end
```

Routing

Provides

```
Catalog.controllers :admin do
  get :index, :provides => [:js, :json, :xml] do
    case content_type
      when :js then # ...
      when :json then # ...
      when :xml then # ...
    end
  end
end
```


Routing

Provides

```
Catalog.controllers :admin do
  get :index, :provides => [:js, :json, :xml, :any] do
    case content_type
      when :js, :json then # ...
      when :xml then # ...
      else # ...
    end
  end
end
```

Routing

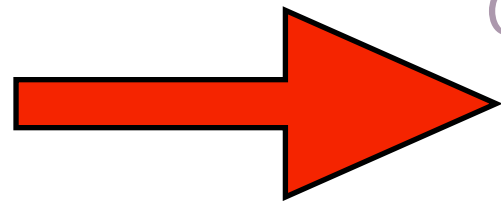
Provides

```
Catalog.controllers :admin do
  get :index, :provides => [:js, :any] do
    # ...
  end
end

  get :show, :provides => [:js, :any] do
    # ...
  end
end
```

Routing

Provides



```
Catalog.controllers :admin do  
  provides :js, :any
```

```
  get :index do  
    # ... return some javascript  
  end
```

```
  get :show do  
    # ... return some javascript  
  end  
end
```

Routing

Filters

Routing

Filters

```
Catalog.controllers do
  get :index do
    # ...
  end
end
```

Routing

Filters

```
Catalog.controllers do
  before { @section = 'main' }
  get :index do
    # ...
  end
end
```

Routing

Filters

```
Catalog.controllers do
```

```
  before { @section = 'main' }
```


```
  get :index do
```

```
    # ...
```

```
  end
```

```
end
```

Use this to go to a
different set of
actions



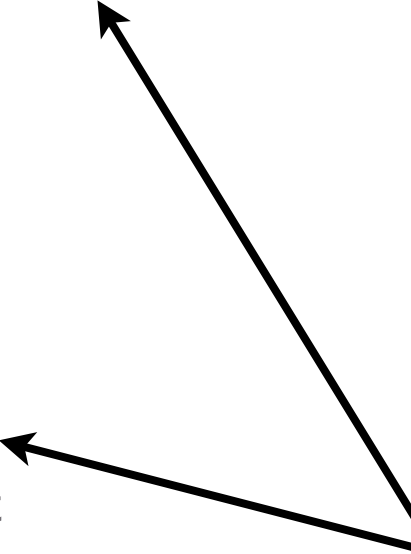
Routing

Filters

```
Catalog.controllers do
  controller do
    before { @user.admin? }
    get :index do
      # admin get
    end
  end

  controller do
    get :index do
      # normal get
    end
  end
end
```

Use this to go to a
different set of
actions



Routing

Filters

```
Catalog.controllers do
  get :index do
    # ...
  end
  after { cleanup }
end
```

Routing

Nesting

Routing

Nesting

```
Catalog.controllers :details do  
  get :index do  
    # /details  
  end  
end
```

Routing

Nesting

```
Catalog.controllers :details do
  get :index, :parent => :product do
    # /product/:product_id/details
  end
end
```

Routing

Nesting

```
Catalog.controllers :details do
  get :index, :parent => [:product, :section] do
    # /product/:product_id/section/:section_id/details
  end
end
```

Routing

Nesting

```
Catalog.controllers :details, :parent => :user do  
  get :index, :parent => :product do  
    # /user/:user_id/product/:product_id/details  
  end  
end
```

Routing

Layouts

Routing

Layouts

```
Catalog.controllers do
  get :index do
    # ...
  end
end
```


Routing

Layouts

```
Catalog.controllers do
  disable :layout
  get :index
    # /details
end
end
```

Routing

Layouts

```
Catalog.controllers do
  layout :product
  get :index
  # /details
end
end
```

Routing

Layouts

```
Catalog.controllers do
  controller :product do
    layout :product
    get :index
    # ...
  end
end

controller :blog do
  layout :blog
  get :index
  # ...
end
end
```

Output Helpers

Output Helpers

content_for & yield_content

app/views/site/index.haml

```
-content_for :assets do  
  =stylesheet_link_tag 'index', 'custom'
```

Output Helpers

content_for & yield_content

app/views/site/index.haml

```
-content_for :assets do  
  =stylesheet_link_tag 'index', 'custom'
```

app/views/layout.haml

```
%head  
  %title Example  
  =stylesheet_link_tag 'style'  
  =yield_content :assets
```

Output Helpers

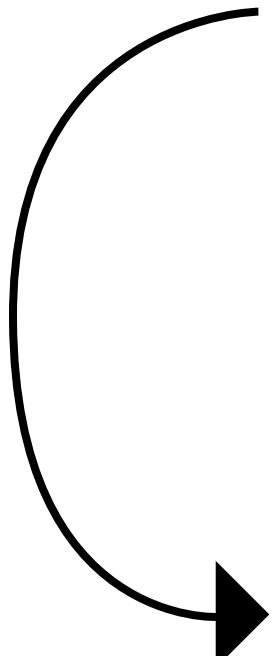
content_for & yield_content

app/views/site/index.haml

```
-content_for :assets do  
  =stylesheet_link_tag 'index', 'custom'
```

app/views/layout.haml

```
%head  
  %title Example  
  =stylesheet_link_tag 'style'  
  =stylesheet_link_tag 'index', 'custom'
```



Output Helpers

concat_content & capture_html

```
def wrap_with_tag(tag_name, &block)
  inner_form_html = capture_html(&block)
  concat_content "<#{tag_name}>" + inner_form_html + "</#{tag_name}>"
end
```


Output Helpers

concat_content & capture_html

```
def wrap_with_tag(tag_name, &block)
  inner_form_html = capture_html(&block)
  concat_content "<#{tag_name}>" + inner_form_html + "</#{tag_name}>"
end
```

app/views/site/index.haml

```
-wrap_with_tag 'p' do
  What an awesome paragraph!
```

Output Helpers

concat_content & capture_html

```
def wrap_with_tag(tag_name, &block)
  inner_form_html = capture_html(&block)
  concat_content "<#{tag_name}>" + inner_form_html + "</#{tag_name}>"
end
```

app/views/site/index.haml

-wrap_with_tag 'p'

What an awesome paragraph!

<p>What an awesome paragraph!</p>

Tag Helpers

Tag Helpers

`tag(name, options = {})`

Tag Helpers

`tag(name, options = {})`

```
tag :img, :src => 'thumbnail.png'
```

```
''
```

Tag Helpers

tag(name, options = {})

```
tag :img, :src => 'thumbnail.png'
```

```
''
```

```
tag :p, :class => 'important', :content => 'Awesome sauce'
```

```
'<p class="important">Awesome sauce</p>'
```

Tag Helpers

`content_tag(name, content, options={})`

Tag Helpers

`content_tag(name, content, options={})`

`content_tag(:p, "Wow, another paragraph")`

`'<p>Wow, another paragraph</p>'`

Tag Helpers

`content_tag(name, content, options={})`

`content_tag(:p, "Wow, another paragraph")`

`'<p>Wow, another paragraph</p>'`

`content_tag(:p, "Wow, another paragraph", :class => 'important')`

`'<p class="important">Wow, another paragraph</p>'`

Tag Helpers

`content_tag(name, content, options={})`

`content_tag(:p, "Wow, another paragraph")`

`'<p>Wow, another paragraph</p>'`

`content_tag(:p, "Wow, another paragraph", :class => 'important')`

`'<p class="important">Wow, another paragraph</p>'`

`content_tag(:p) { "And some more paragraph content" }`

`'<p>And some more paragraph content</p>'`

Tag Helpers

`input_tag(type, options={})`

Tag Helpers

`input_tag(type, options={})`

```
input_tag :text, :class => "name"
```

```
'<input type="text" class="name"/>'
```

Tag Helpers

`input_tag(type, options={})`

```
input_tag :text, :class => "name"
```

```
'<input type="text" class="name"/>'
```

```
input_tag :password, :class => "sekrit"
```

```
'<input type="password" class="sekrit"/>'
```

Asset Helpers

flash_tag(kind, options={})

```
flash_tag(:notice, :class => 'flash', :id => 'flash-notice')
```

```
'<div class="flash" id="flash-notice">Your notice</div>'
```

link_to(*args, &block)

```
link_to 'click me', '/dashboard', :class => 'linky'
```

```
'<a href="/dashboard" class="linky">click me</a>'
```

```
link_to 'click me', '/dashboard', :class => 'linky', :if => @foo.present?
```

```
link_to 'click me', '/dashboard', :class => 'linky', :unless => @foo.blank?
```

```
link_to 'click me', '/dashboard', :class => 'linky', :unless => :current
```

```
link_to('/dashboard', :class => 'linky') { "click me" }
```

mail_to(email, caption=nil, mail_options={})

```
mail_to "mailto:test@demo.com", "My Email", :class => "demo"
```

```
"<a class=\"demo\" href=\"mailto:test@demo.com\">My Email</a>"
```

Asset Helpers

image_tag(url, options={})

`image_tag('icons/avatar.png')`

`''`

stylesheet_link_tag(*sources)

`stylesheet_link_tag 'style', 'application', 'layout'`

`<link type="text/css" rel="stylesheet" media="screen" href="/stylesheets/style.css?1282639722" />`
`<link type="text/css" rel="stylesheet" media="screen" href="/stylesheets/application.css?1282639722" />`
`<link type="text/css" rel="stylesheet" media="screen" href="/stylesheets/layout.css?1282639722" />`

javascript_include_tag(*sources)

`javascript_include_tag 'application', 'base'`

`<script type="text/javascript" src="/javascripts/application.js?1282639968"></script>`
`<script type="text/javascript" src="/javascripts/base.js?1282639968"></script>`

favicon_tag(source, options={})

`favicon_tag 'images/favicon.ico', :type => 'image/ico'`

`'<link type="image/ico" rel="icon" href="/images/favicon.ico?1282640162" />'`

feed_tag(mime, source, options={})

`feed_tag :rss, "/blog/post.rss"`

`'<link type="application/rss+xml" rel="alternate" title="rss" href="/blog/post.rss" />'`

Form Helpers

```
- form_tag '/destroy', :class => 'destroy-form', :method => 'delete' do
  = flash_tag(:notice)
  - field_set_tag do
    %p
    = label_tag :username, :class => 'first'
    = text_field_tag :username, :value => params[:username]
    %p
    = label_tag :password, :class => 'first'
    = password_field_tag :password, :value => params[:password]
    %p
    = label_tag :strategy
    = select_tag :strategy, :options => ['delete', 'destroy'], :selected => 'delete'
    %p
    = check_box_tag :confirm_delete
  - field_set_tag(:class => 'buttons') do
    = submit_tag "Remove"
```


Form Helpers

```
- form_tag '/destroy', :class => 'destroy-form', :method => 'delete' do
  = flash_tag(:notice)
  - field_set_tag do
    %p
    = label_tag :username, :class => 'first'
    = text_field_tag :username, :value => params[:username]
    %p
    = label_tag :password, :class => 'first'
    = password_field_tag :password, :value => params[:password]
    %p
    = label_tag :strategy
    = select_tag :strategy, :options => ['delete', 'destroy'], :selected => 'delete'
    %p
    = check_box_tag :confirm_delete
  - field_set_tag(:class => 'buttons') do
    = submit_tag "Remove"
```

Form Helpers

```
- form_tag '/destroy', :class => 'destroy-form', :method => 'delete' do
  = flash_tag(:notice)
- field_set_tag do
  %p
    = label_tag :username, :class => 'first'
    = text_field_tag :username, :value => params[:username]
  %p
    = label_tag :password, :class => 'first'
    = password_field_tag :password, :value => params[:password]
  %p
    = label_tag :strategy
    = select_tag :strategy, :options => ['delete', 'destroy'], :selected => 'delete'
  %p
    = check_box_tag :confirm_delete
- field_set_tag(:class => 'buttons') do
  = submit_tag "Remove"
```

Form Helpers

```
- form_tag '/destroy', :class => 'destroy-form', :method => 'delete' do
  = flash_tag(:notice)
  - field_set_tag do
    %p
    = label_tag :username, :class => 'first'
    = text_field_tag :username, :value => params[:username]
    %p
    = label_tag :password, :class => 'first'
    = password_field_tag :password, :value => params[:password]
    %p
    = label_tag :strategy
    = select_tag :strategy, :options => ['delete', 'destroy'], :selected => 'delete'
    %p
    = check_box_tag :confirm_delete
  - field_set_tag(:class => 'buttons') do
    = submit_tag "Remove"
```

Form Builders

```
- form_for @user, '/register', :id => 'register' do |f|
  = f.error_messages
  %p
  = f.label :username, :caption => "Nickname"
  = f.text_field :username
  %p
  = f.label :email
  = f.text_field :email
  %p
  = f.label :password
  = f.password_field :password
  %p
  = f.label :is_admin, :caption => "Admin User?"
  = f.check_box :is_admin
  %p
  = f.label :color, :caption => "Favorite Color?"
  = f.select :color, :options => ['red', 'black']
  %p
  - fields_for @user.location do |location|
    = location.text_field :street
    = location.text_field :city
  %p
  = f.submit "Create", :class => 'button'
```

Form Builders

```
- form_for @user, '/register', :id => 'register' do |f|
  = f.error_messages
  %p
  = f.label :username, :caption => "Nickname"
  = f.text_field :username
  %p
  = f.label :email
  = f.text_field :email
  %p
  = f.label :password
  = f.password_field :password
  %p
  = f.label :is_admin, :caption => "Admin User?"
  = f.check_box :is_admin
  %p
  = f.label :color, :caption => "Favorite Color?"
  = f.select :color, :options => ['red', 'black']
  %p
  - fields_for @user.location do |location|
    = location.text_field :street
    = location.text_field :city
  %p
  = f.submit "Create", :class => 'button'
```

Form Builders

```
- form_for @user, '/register', :id => 'register' do |f|
  = f.error_messages
  %p
  = f.label :username, :caption => "Nickname"
  = f.text_field :username
  %p
  = f.label :email
  = f.text_field :email
  %p
  = f.label :password
  = f.password_field :password
  %p
  = f.label :is_admin, :caption => "Admin User?"
  = f.check_box :is_admin
  %p
  = f.label :color, :caption => "Favorite Color?"
  = f.select :color, :options => ['red', 'black']
  %p
  - fields_for @user.location do |location|
    = location.text_field :street
    = location.text_field :city
  %p
  = f.submit "Create", :class => 'button'
```

Form Builders

```
- form_for @user, '/register', :id => 'register' do |f|  
  = f.error_messages  
  %p  
  = f.label :username, :caption => "Nickname"  
  = f.text_field :username  
  %p  
  = f.label :email  
  = f.text_field :email  
  %p  
  = f.label :password  
  = f.password_field :password  
  %p  
  = f.label :is_admin, :caption => "Admin User?"  
  = f.check_box :is_admin  
  %p  
  = f.label :color, :caption => "Favorite Color?"  
  = f.select :color, :options => ['red', 'black']  
  %p  
  - fields_for @user.location do |location|  
    = location.text_field :street  
    = location.text_field :city  
  %p  
  = f.submit "Create", :class => 'button'
```

Form Builders

```
- form_for @user, '/register', :id => 'register' do |f|
  = f.error_messages
%p
  = f.label :username, :caption => "Nickname"
  = f.text_field :username
%p
  = f.label :email
  = f.text_field :email
%p
  = f.label :password
  = f.password_field :password
%p
  = f.label :is_admin, :caption => "Admin User?"
  = f.check_box :is_admin
%p
  = f.label :color, :caption => "Favorite Color?"
  = f.select :color, :options => ['red', 'black']
%p
  - fields_for @user.location do |location|
    = location.text_field :street
    = location.text_field :city
%p
  = f.submit "Create", :class => 'button'
```


Form Builders

```
- form_for @user, '/register', :id => 'register' do |f|
  = f.error_messages
  %p
  = f.label :username, :caption => "Nickname"
  = f.text_field :username
  %p
  = f.label :email
  = f.text_field :email
  %p
  = f.label :password
  = f.password_field :password
  %p
  = f.label :is_admin, :caption => "Admin User?"
  = f.check_box :is_admin
  %p
  = f.label :color, :caption => "Favorite Color?"
  = f.select :color, :options => ['red', 'black']
  %p
  - fields_for @user.location do |location|
    = location.text_field :street
    = location.text_field :city
  %p
  = f.submit "Create", :class => 'button'
```

What if your
object doesn't
implement
#errors?

Form Builders

`AbstractFormBuilder`

Form Builders

Abstract FormBuilder

error_messages

error_message_on

label

hidden_field

text_field

text_area

password_field

select

check_box

radio_button

file_field

submit

image_submit

Form Builders

AbstractFormBuilder

```
=form_for @user, '/register', :builder => 'MyCustomFormBuilder'
```

```
set :default_builder, 'MyCustomFormBuilder'
```

Format Helpers

simple_format

```
simple_format("hello\nworld")
```

```
"<p>hello<br/>world</p>"
```

pluralize

```
pluralize(2, 'person')
```

```
'2 people'
```

word_wrap

```
word_wrap("Once upon a time", :line_width => 8)
```

```
"Once upon\na time"
```

truncate

```
truncate("Once upon a time in a galaxy", :length => 9)
```

```
"Once upon..."
```

highlight

```
highlight("Highlight this word", "this")
```

```
'Highlight <strong class="highlight">this</strong> word'
```

escape_html

```
# alias for Rack::Utils.escape_html
```

strip_tags

```
strip_tags("This <em>won't</em> work")
```

```
"This won't work"
```

distance_of_time_in_words

```
distance_of_time_in_words(2.days.ago)
```

```
"2 days"
```

time_ago_in_words

```
time_ago_in_words(2.days.ago)
```

```
"2 days ago"
```

js_escape_html

```
js_escape_html("\n\r<p>hello</p>\r\n")
```

```
"\\n\\r<p>hello<\\r\\n>"
```

Render Helpers

Render Helpers

The old busted (Sinatra)

erb :index

Render Helpers

The old busted (Sinatra)

erb `:index`

views/index.erb

```
<h1>Hey</h1>
```

```
<p>Hello world <%=@name%></p>
```


Render Helpers

The old busted (Sinatra)

erb :index

views/index.erb

```
<h1>Hey</h1>  
<p>Hello world <%=@name%></p>
```

views/admin/index.erb

erb : "admin/index"

erb "admin/index".to_sym

Render Helpers

The old busted (Sinatra)

erb :index

views/index.erb

```
<h1>Hey</h1>
<p>Hello world <%=@name%></p>
```

views/admin/index.erb

erb : "admin/index"

erb "admin/index".to_sym

The new hotness (Padrino)

render :erb, "index"

render :haml, "index"

render "index"

Render Helpers

The old busted (Sinatra)

```
erb :index
```

views/index.erb

```
<h1>Hey</h1>  
<p>Hello world <%=@name%></p>
```

views/admin/index.erb

```
erb : "admin/index"
```

```
erb "admin/index".to_sym
```

The new hotness (Padrino)

```
render :erb, "index"
```

```
render :haml, "index"
```

```
render "index"
```

Finds the first available engine.

Render Helpers

partial

```
partial 'photo/item'  
uses "views/photo/_item.haml"
```

```
partial 'photo/item', :object => @item  
uses "views/photo/_item.haml"
```

```
%p  
  My item's name is #{item.name}.
```

```
partial 'photo/item', :collection => @items  
uses "views/photo/_item.haml"
```

```
%p  
  My item's name is #{item.name} at #{item_counter}
```

Custom Helpers

```
MyApplication.helpers do
  def render_pony
    "pony time!"
  end
end
```

Custom Helpers

```
module MyAwesomeHelper  
  def awesome  
    "Your are awesome"  
  end  
end
```

```
MyApplication.helpers MyAwesomeHelper
```

Custom Helpers

```
module MyAwesomeHelper  
  def awesome  
    "Your are awesome"  
  end  
end
```

```
MyApplication.helpers MyAwesomeHelper
```



Unobtrusive Javascript Helpers

Unobtrusive Javascript Helpers

Forms

Link confirmation

Links

Custom request method
on link

Unobtrusive Javascript Helpers

Forms

Link confirmation

Link

Custom request method
on link



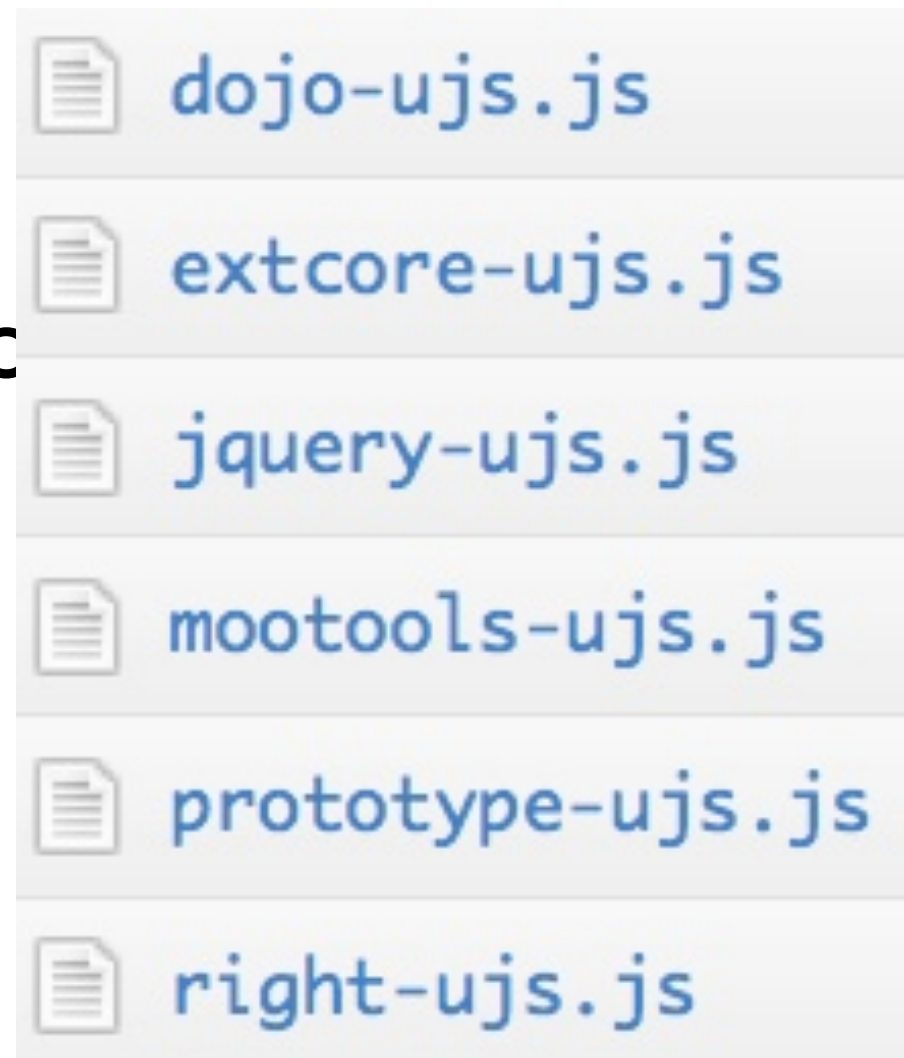
Unobtrusive Javascript Helpers

Forms

Link confirmation

Link

Custom



Unobtrusive Javascript Helpers

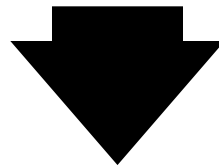
Forms

```
- form_for @user, url(:items, :create, :format => :js), :remote => true do |f|  
  = f.text_field :username  
  = f.submit
```

Unobtrusive Javascript Helpers

Forms

```
- form_for @user, url(:items, :create, :format => :js), :remote => true do |f|  
  = f.text_field :username  
  = f.submit
```



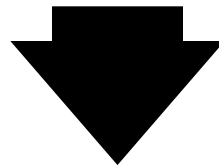
Generates

```
<form data-remote="true" action="/items/create.js" method="post">  
  <input type="text" id="post_title" name="post[title]">  
  <input type="submit" value="Create">  
</form>
```

Unobtrusive Javascript Helpers

Forms

```
- form_for @user, url(:items, :create, :format => :js), :remote => true do |f|  
  = f.text_field :username  
  = f.submit
```



Generates

```
<form data-remote="true" action="/items/create.js" method="post">  
  <input type="text" id="post_title" name="post[title]">  
  <input type="submit" value="Create">  
</form>
```

Unobtrusive Javascript Helpers

Links

Unobtrusive Javascript Helpers

Links

Remote Links

```
= link_to "add item", url(:items, :new, :format => :js), :remote => true
```

```
<a href="/items/new.js" data-remote="true">add item</a>
```


Unobtrusive Javascript Helpers

Links

Remote Links

```
= link_to "add item", url(:items, :new, :format => :js), :remote => true
```

```
<a href="/items/new.js" data-remote="true">add item</a>
```

Link Confirmations

```
= link_to "delete item", url(:items, :destroy, :format => :js), :confirm => "Are You Sure?"
```

```
<a data-confirm="Are You Sure?" href="/posts/destroy/7">[destroy]</a>
```

Unobtrusive Javascript Helpers

Links

Remote Links

```
= link_to "add item", url(:items, :new, :format => :js), :remote => true
```

```
<a href="/items/new.js" data-remote="true">add item</a>
```

Link Confirmations

```
= link_to "delete item", url(:items, :destroy, :format => :js), :confirm => "Are You Sure?"
```

```
<a data-confirm="Are You Sure?" href="/posts/destroy/7">[destroy]</a>
```

Custom HTTP Methods on Links

```
= link_to "logout", url(:session, :destroy, :format => :js), :method => :delete
```

```
<a data-method="delete" href="/posts/destroy/7" rel="nofollow">[destroy]</a>
```

Padrino Admin

Padrino Admin

Padrino Admin

Products

Accounts

ProfileLogout

ListNew

All Product

Id	Name	Price	
1	this is great	12.0	Edit Delete

Copyright © 2010 Your Site - Powered by [Padrino v.0.9.14](#)

Simple Block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Links

Link 1

Link 2

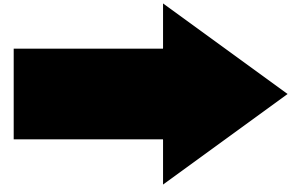
Padrino Admin

Usage

```
$ padrino g model post title:string body:text  
$ padrino rake dm:migrate # or ar:migrate  
$ padrino g admin_page post  
$ padrino start
```

Padrino Admin

Usage



```
$ padrino g model post title:string body:text  
$ padrino rake dm:migrate # or ar:migrate  
$ padrino g admin_page post  
$ padrino start
```

Padrino Admin

ORM Agnostic

supports

datamapper, sequel, activerecord, mongomapper, mongoid, couchrest

Padrino Admin

ORM Agnostic

supports

datamapper, sequel, activerecord, mongomapper, mongoid, couchrest

ActiveRecord

```
put :update, :with => :id do
  @account = Account.find(params[:id])
  if @account.update_attributes(params[:account])
    flash[:notice] = 'Account was successfully updated.'
    redirect url(:accounts, :edit, :id => @account.id)
  else
    render 'accounts/edit'
  end
end
```


Padrino Admin

ORM Agnostic

supports

datamapper, sequel, activerecord, mongomapper, mongoid, couchrest

Sequel

```
put :update, :with => :id do
  @account = Account[params[:id]]
  if @account.modified! && @account.update(params[:account])
    flash[:notice] = 'Account was successfully updated.'
    redirect url(:accounts, :edit, :id => @account.id)
  else
    render 'accounts/edit'
  end
end
```

Padrino Admin

Admin Authentication

```
class MyEcommerce < Padrino::Application
  enable :authentication
  enable :store_location
  set    :login_page, "/login"

  access_control.roles_for :any do |role|
    role.protect "/customer/orders"
    role.protect "/cart/checkout"
  end
end
```

Padrino Admin

Admin Authentication

```
class MyEcommerce < Padrino::Application
  enable :authentication
  enable :store_location
  set    :login_page, "/login"

  access_control.roles_for :any do |role|
    role.protect "/customer/orders"
    role.protect "/cart/checkout"
  end
end
```

Padrino Admin

Admin Authentication

```
class MyEcommerce < Padrino::Application
  enable :authentication
  enable :store_location
  set    :login_page, "/login"

  access_control.roles_for :any do |role|
    role.protect "/customer/orders"
    role.protect "/cart/checkout"
  end
end
```

Padrino Admin

Admin Authentication

Not just for admin!

Padrino Admin

Admin Authentication

```
access_control.roles_for :any do |role|  
  role.protect "/"  
  role.allow "/sessions"  
end
```

```
access_control.roles_for :admin do |role|  
  role.project :settings, "/settings"  
end
```

```
access_control.roles_for :editor do |role|  
  role.project :posts, "/posts"  
  role.project :categories, "/categories"  
end
```

Padrino Admin

Admin Authentication

```
access_control.roles_for :any do |role|  
  role.protect "/"  
  role.allow "/sessions"  
end
```

```
access_control.roles_for :admin do |role|  
  role.protect :settings, "/settings"  
end
```

```
access_control.roles_for :editor do |role|  
  role.protect :posts, "/posts"  
  role.protect :categories, "/categories"  
end
```

Padrino Admin

Admin Authentication

```
access_control.roles_for :any do |role|  
  role.protect "/"  
  role.allow "/sessions"  
end
```

```
access_control.roles_for :admin do |role|  
  role.project :settings, "/settings"  
end
```

```
access_control.roles_for :editor do |role|  
  role.project :posts, "/posts"  
  role.project :categories, "/categories"  
end
```


Padrino Admin

Admin Authentication

```
access_control.roles_for :any do |role|  
  role.protect "/"  
  role.allow "/sessions"  
end
```

```
access_control.roles_for :admin do |role|  
  role.project :settings, "/settings"  
end
```

```
access_control.roles_for :editor do |role|  
  role.project :posts, "/posts"  
  role.project :categories, "/categories"  
end
```

Padrino Mailer

Padrino Mailer

```
post :register do
  # ...
  email(:from => "donotreply@mysite.com", :to => user.email,
        :subject => "Welcome!", :body=>"Thanks for signing up")
end
```

Padrino Mailer

```
MyAppName.mailer :sample do
  email :registration_email do |name, email|
    from 'admin@site.com'
    to email
    subject 'Welcome to the site!'
    locals :name => name, :email => email
    render 'registration_email'
    content_type :html          # optional, defaults to :plain
    via :sendmail              # optional, to smtp if defined otherwise sendmail
  end
end
```

Padrino Mailer

```
MyAppName.mailer :sample do
  email :registration_email do |name, email|
    from 'admin@site.com'
    to email
    subject 'Welcome to the site!'
    locals :name => name, :email => email
    render 'registration_email'
    content_type :html          # optional, defaults to :plain
    via :sendmail              # optional, to smtp if defined otherwise sendmail
  end
end
```

```
$ padrino g mailer Sample registration_email
```

Padrino Mailer

```
MyAppName.mailer :sample do
  email :registration_email do |name, email|
    from 'admin@site.com'
    to email
    subject 'Welcome to the site!'
    locals :name => name, :email => email
    render 'registration_email'
    content_type :html          # optional, defaults to :plain
    via :sendmail              # optional, to smtp if defined otherwise sendmail
  end
end
```

views/mailers/sample/registration_email.erb

This is the body of the email and can access the `<%= name %>` variable.
That's all there is to defining the body of the email which can be in plain text or html.

Padrino Mailer

```
MyAppName.mailer :sample do
  email :registration_email do |name, email|
    from 'admin@site.com'
    to email
    subject 'Welcome to the site!'
    locals :name => name, :email => email
    render 'registration_email'
    content_type :html          # optional, defaults to :plain
    via :sendmail              # optional, to smtp if defined otherwise sendmail
  end
end
```

views/mailers/sample/registration_email.erb

This is the body of the email and can access the `<%= name %>` variable.
That's all there is to defining the body of the email which can be in plain text or html.

Padrino Mailer

Multi-part Goodness

Padrino Mailer

Multi-part Goodness

```
mailer :sample do
  email :email_with_parts do
    from 'admin@site.com'
    # ...
    text_part { render('path/to/basic.text') }
    html_part render('path/to/basic.html')
  end
end
```

Padrino Mailer

Multi-part Goodness

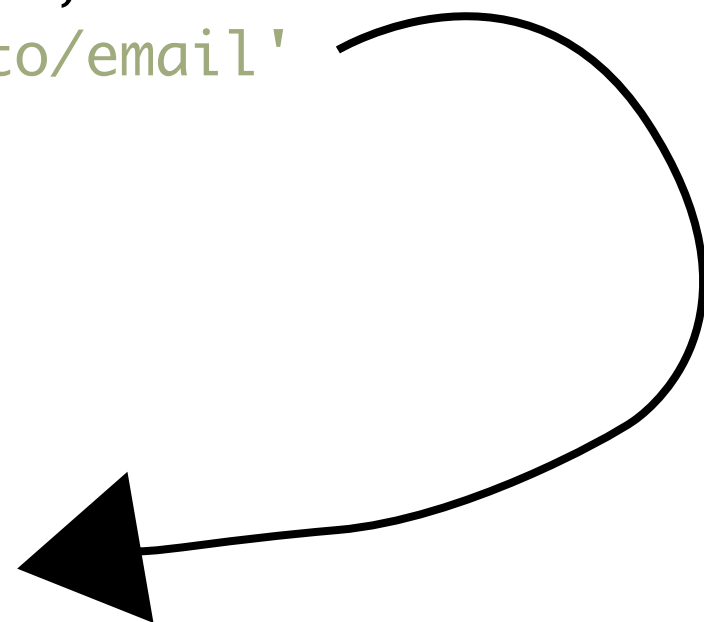
```
mailer :sample do
  email :email_with_parts do
    from 'admin@site.com'
    # ...
    provides :plain, :html
    render 'path/to/email'
  end
end
```

Padrino Mailer

Multi-part Goodness

```
mailer :sample do
  email :email_with_parts do
    from 'admin@site.com'
    # ...
    provides :plain, :html
    render 'path/to/email'
  end
end
```

path/to/basic.html.erb
path/to/basic.plain.erb



Padrino Mailer

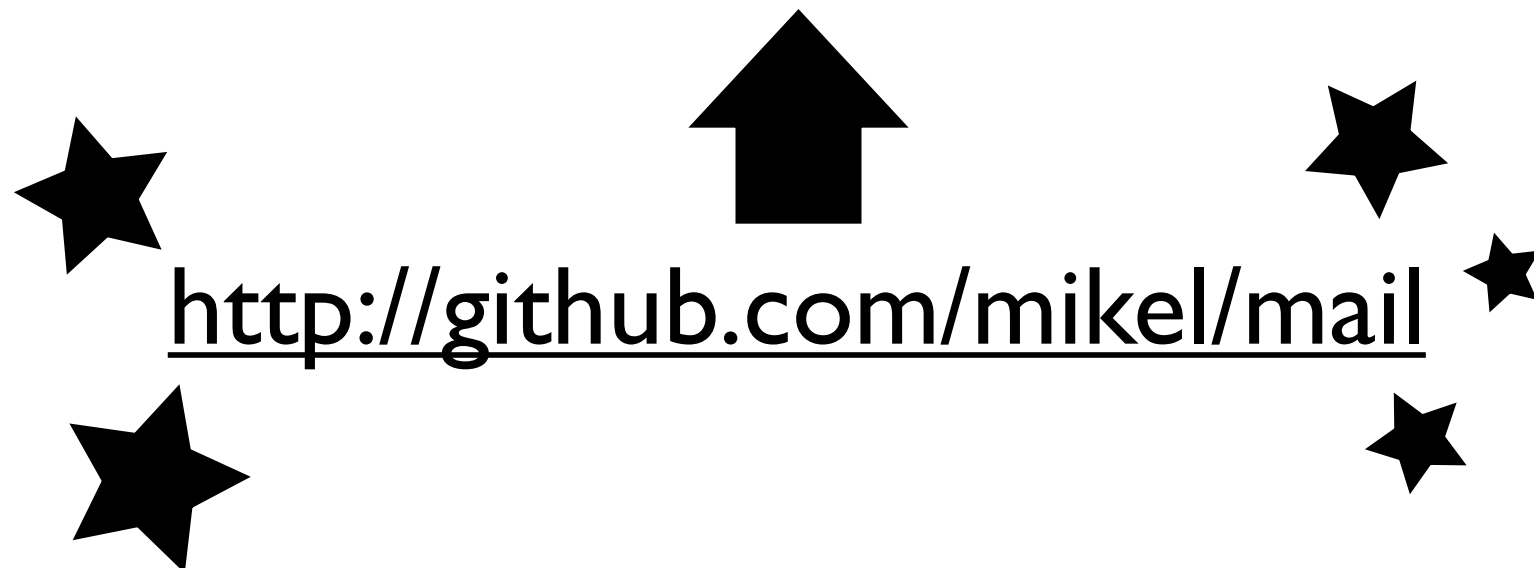
File Attachments

```
mailer :sample do
  email :email_with_files do
    from 'admin@site.com'
    # ...
    body "Here are your files!"
    add_file :filename => 'somefile.png', :content => File.read('/somefile.png')
    add_file '/full/path/to/someotherfile.png'
  end
end
```

Padrino Mailer

File Attachments

```
mailer :sample do
  email :email_with_files do
    from 'admin@site.com'
    # ...
    body "Here are your files!"
    add_file :filename => 'somefile.png', :content => File.read('/somefile.png')
    add_file '/full/path/to/someotherfile.png'
  end
end
```



Padrino Mailer

Defaults

app/app.rb

```
set :mailer_defaults, :from => 'admin@site.com'
```

in your mailer

```
MyAppName.mailers :sample do
  defaults :content_type => 'html'
  email :registration do |name, age|
    # Uses default 'content_type' and 'from' values but can also overwrite them
    to      'user@domain.com'
    subject 'Welcome to the site!'
    locals  :name => name
    render  'registration'
  end
end
```

Padrino Mailer

Defaults

app/app.rb

```
set :mailer_defaults, :from => 'admin@site.com'
```

in your mailer

```
MyAppName.mailers :sample do
  defaults :content_type => 'html'
  email :registration do |name, age|
    # Uses default 'content_type' and 'from' values but can also overwrite them
    to      'user@domain.com'
    subject 'Welcome to the site!'
    locals  :name => name
    render  'registration'
  end
end
```

Padrino Mailer

Defaults

app/app.rb

```
set :mailer_defaults, :from => 'admin@site.com'
```

in your mailer

```
MyAppName.mailers :sample do
  defaults :content_type => 'html'
  email :registration do |name, age|
    # Uses default 'content_type' and 'from' values but can also overwrite them
    to      'user@domain.com'
    subject 'Welcome to the site!'
    locals  :name => name
    render  'registration'
  end
end
```


Localization

Localization

Currently Supported

English	Russian
Spanish	Brazilian
German	Ukrainian
Italian	Czech
French	Turkish
Danish	Dutch
Polish	

Localization

Currently Supported

English

Russian

Spanish

Brazilian

German

Ukrainian

Italian

Czech

French

Turkish

Danish

Polish

Polish



Localization

Currently Supported

padrino-core.yml

padrino-admin.yml

padrino-admin-orm.yml

padrino-helper.yml

Email padrinorb@gmail.com or fork!

Localization

In Your Application

Localization

In Your Application

config/boot.rb

```
I18n.locale = :de
```

app/locale/de.yml

```
de:  
  foo: bar
```

```
I18n.t("foo")  
# => "bar"
```

Localization

In Your Application

config/boot.rb

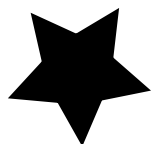
```
I18n.locale = :de
```

app/locale/de.yml

```
de:  
  foo: bar
```

```
I18n.t("foo")  
# => "bar"
```

<http://github.com/svenfuchs/i18n>



Localization

Models

```
create_table :products do |t|  
  t.string :price  
  t.string :name  
end
```

```
$ padrino rake ar:translate
```

app/locale/models/product/it.yml

```
it:  
  models:  
    product:  
      name: Product  
      attributes:  
        id: Id  
        name: Name  
        price: Price
```


Localization

Forms

```
-form_for :account, url(:accounts_create, :format => :js), :remote => true do |f|  
  %table  
    %tr  
      %td=f.label :name  
      %td=f.text_field :name  
    %tr  
      %td=f.label :surname  
      %td=f.text_field :surname  
    %tr  
      %td=f.label :role  
      %td=f.select :role, :options => access_control.roles
```

Localization

Forms

```
-form_for :account, url(:accounts_create, :format => :js), :remote => true do |f|  
  %table  
    %tr  
      %td=f.label :name  
      %td=f.text_field :name  
    %tr  
      %td=f.label :surname  
      %td=f.text_field :surname  
    %tr  
      %td=f.label :role  
      %td=f.select :role, :options => access_control.roles
```

Localization

Forms

```
-form_for :account, url(:accounts_create, :format => :js), :remote => true do |f|  
  %table  
    %tr  
      %td=f.label :name  
      %td=f.text_field :name  
    %tr  
      %td=f.label :surname  
      %td=f.text_field :surname  
    %tr  
      %td=f.label :role  
      %td=f.select :role, :options => access_control.roles  
  
  en, account, name => "Name"
```

Templates & Plugins

Templates

```
$ padrino g project my_project --template path/to/my_template.rb
```

Templates

```
$ padrino g project my_project --template path/to/my_template.rb
```

Thor::Actions + Padrino Template
DSL

Templates

```
# We generate a basic project
say
dev = yes?("Are you using padrino-dev?").present?
tiny = yes?("Do you need a tiny structure?").present?
adapter = ask("SQL adapter for ActiveRecord (sqlite, mysql, postgres):")

project :test => :none, :renderer => :haml, :script => :jquery, :orm => :activerecord, :dev => dev, :tiny => tiny, :adapter => adapter

say "=> Installing exception notifier", :magenta
execute_runner :plugin, :exception_notifier

say
exception_subject = ask("Tell me the subject of the exception email", fetch_app_name)
exception_from = ask("Tell me the sender of the email", "exceptions@lipsiasoft.com")
exception_to = ask("Tell me the recipient email", "help@lipsiasoft.com")

exception_tpl = <<-RUBY
set :exceptions_subject, "#{fetch_app_name}"
set :exceptions_from, "#{exception_from}"
set :exceptions_to, "#{exception_to}"
set :exceptions_page, "#{base/' unless tiny?}errors"
RUBY
```

...

Plugins

```
$ padrino g plugin --list
```


Plugins

```
$ padrino g plugin --list
```

```
$ padrino g plugin --list | wc -l  
# 26
```

Plugins

960	deflect
access	disqus
ar_permalink_index	exception_notifier
ar_permalink	flash_session
ar_textile	googleanalytics
ar_translate	hoptoad
auto_locale	maintenance
blueprint	omniauth
bug	openid
carrierwave	payment
codehighlighter	recaptcha
coderay	rewrite
coffee	tripoli

Plugins

googleanalytics_plugin.rb

```
##  
# Template to get GoogleAnalytics on Padrino  
# prereqs:  
# sudo gem install rack-google-analytics  
# http://github.com/leehambley/rack-google-analytics  
#  
ANALYTICS = <<-ANALYTICS  
app.use Rack::GoogleAnalytics, :tracker => 'UA-xxxxxx-x'  
ANALYTICS  
require_dependencies 'rack-google-analytics'  
initializer :analytics, ANALYTICS
```

Benchmarks!

Benchmarks!

```
# Rendering a string inline
Merb 1.1.0      => 1749.97 rps
Padrino 0.9.10  => 1629.15 rps
Sinatra 1.0.0   => 1537.78 rps
Rails 3.beta3   => 381.76 rps
Ramaze 2010.04.04 => 270.08 rps

# Rendering a basic erb template
Merb 1.1.0      => 1490.8 rps
Padrino 0.9.10  => 1416.84 rps
Sinatra 1.0.0   => 1157.89 rps
Rails 3.0.beta3 => 330.58 rps
Ramaze 2010.04.04 => 254.23 rps

# Rendering a simulated simple app
Padrino 0.9.10  => 675.79 rps
Sinatra 1.0.0   => 652.0 rps
Merb 1.1.0      => 642.29 rps
Rails 3.0.beta3 => 201.86 rps
Ramaze 2010.04.04 => 130.62 rps
```

<http://www.padrinorb.com/blog/padrino-0-9-10-released-built-for-speed>

Benchmarks!

	Padrino	Ramaze	units
Time taken for tests	38.184	51.072	<i>seconds</i>
Complete requests	10000	10000	
Failed requests	0	0	
Write errors	0	0	
Total transferred	16430000	17440000	<i>bytes</i>
HTML transferred	13320000	14050000	<i>bytes</i>
Requests per second	261.89	195.80	<i>[#/sec] (mean)</i>
Time per request	381.837	510.717	<i>[ms] (mean)</i>
Time per request	3.818	5.107	<i>[ms] (mean, across all concurrent requests)</i>
Transfer rate	420.20	333.48	<i>[Kbytes/sec] received</i>
Resident Memory per Worker	41.9	26.3	<i>MB</i>

<http://ramblings.gibberishcode.net/archives/ramaze-vs-padrino-benchmarks/82>

The Team



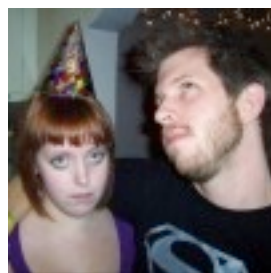
Davide D'Agostino
(DaDDyE)



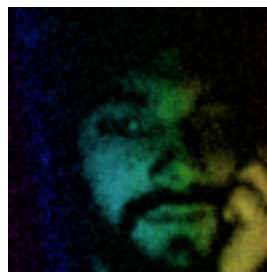
Nathan Esquenazi
(nesquena)



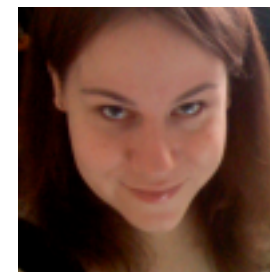
Arthur Chiu
(achiu)



Joshua Hull
(joshbuddy)



Florian Gilcher
(skade)



Lori Holden
(LHolden)

The Team



John E.Vincent
(Iusis)

Getting In Touch

<http://www.padrinorb.com/>

#padrino @freenode

<http://github.com/padrino>

<https://twitter.com/padrinorb>

Thank you!