Rails Intro

dotan@paracode.com
@jondot

The Productivity Buzz

vs

The Scaffold Demo

```
$ sudo gem install rails
C:\gem install rails

$ rails new MyApp
$ cd MyApp
$ bundle install

$ rails g scaffold Task title:string body:string
created:datetime priority:integer
$ rake db:migrate
$ rails s  (open browser http://localhost:3000/tasks)
```

```
=> Booting WEBrick
=> Rails 3.0.6 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2011-06-13 23:08:31] INFO  WEBrick 1.3.1
[2011-06-13 23:08:31] INFO  ruby 1.8.7 (2010-08-16) [i386-mingw32]
[2011-06-13 23:08:31] INFO  WEBrick::HTTPServer#start: pid=5156 port=3000
```

The Obligatory MVC Slide

MVC

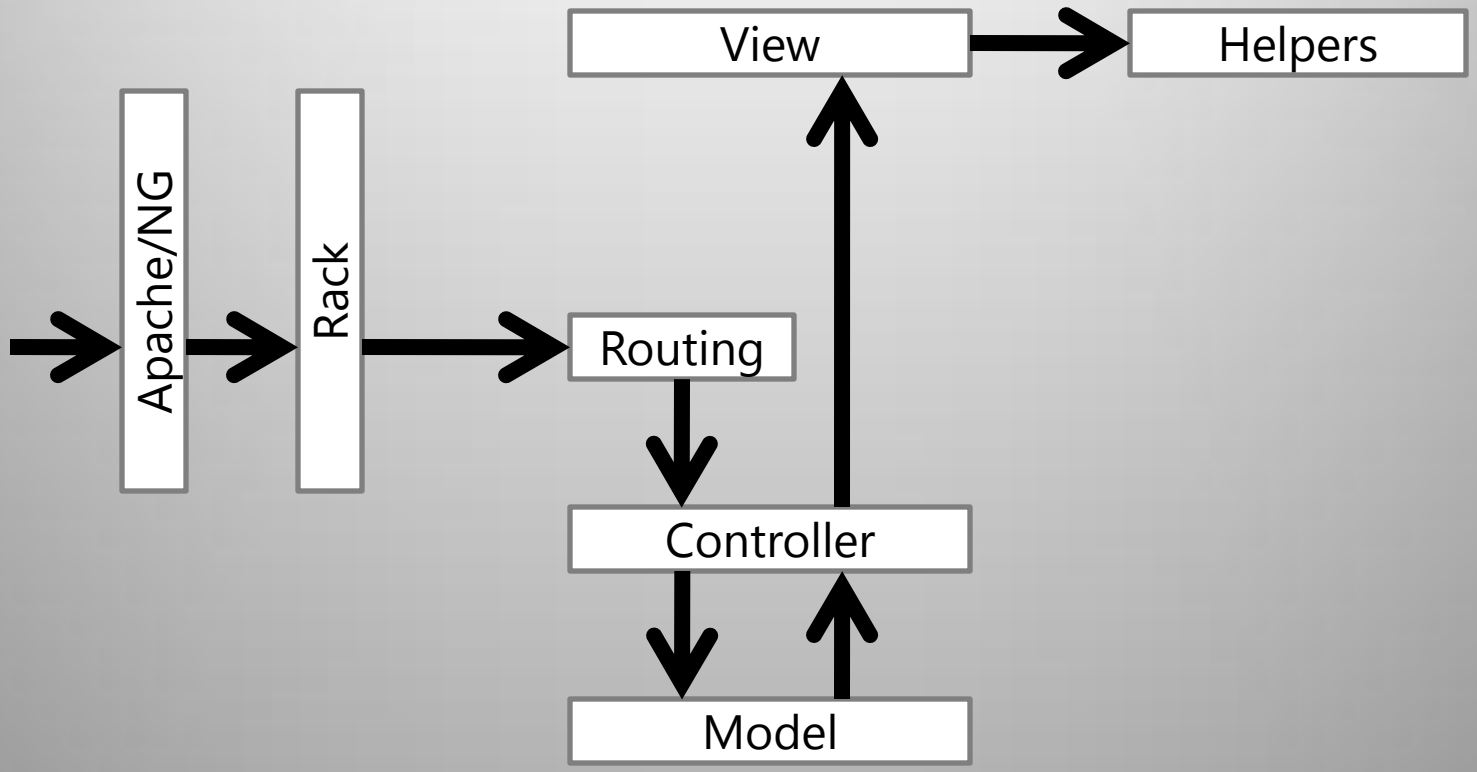
:)

Application Layout

- Some of the more interesting elements:
  - Gemfile
  - Rakefile
  - /config
    - /environments
    - /initializers
    - environment.rb
  - /app
    - **/models**
    - **/controllers**
    - **/views**
  - /lib
  - /test
  - vendor
    - /plugin
  - /public
    - /javascripts
    - /css

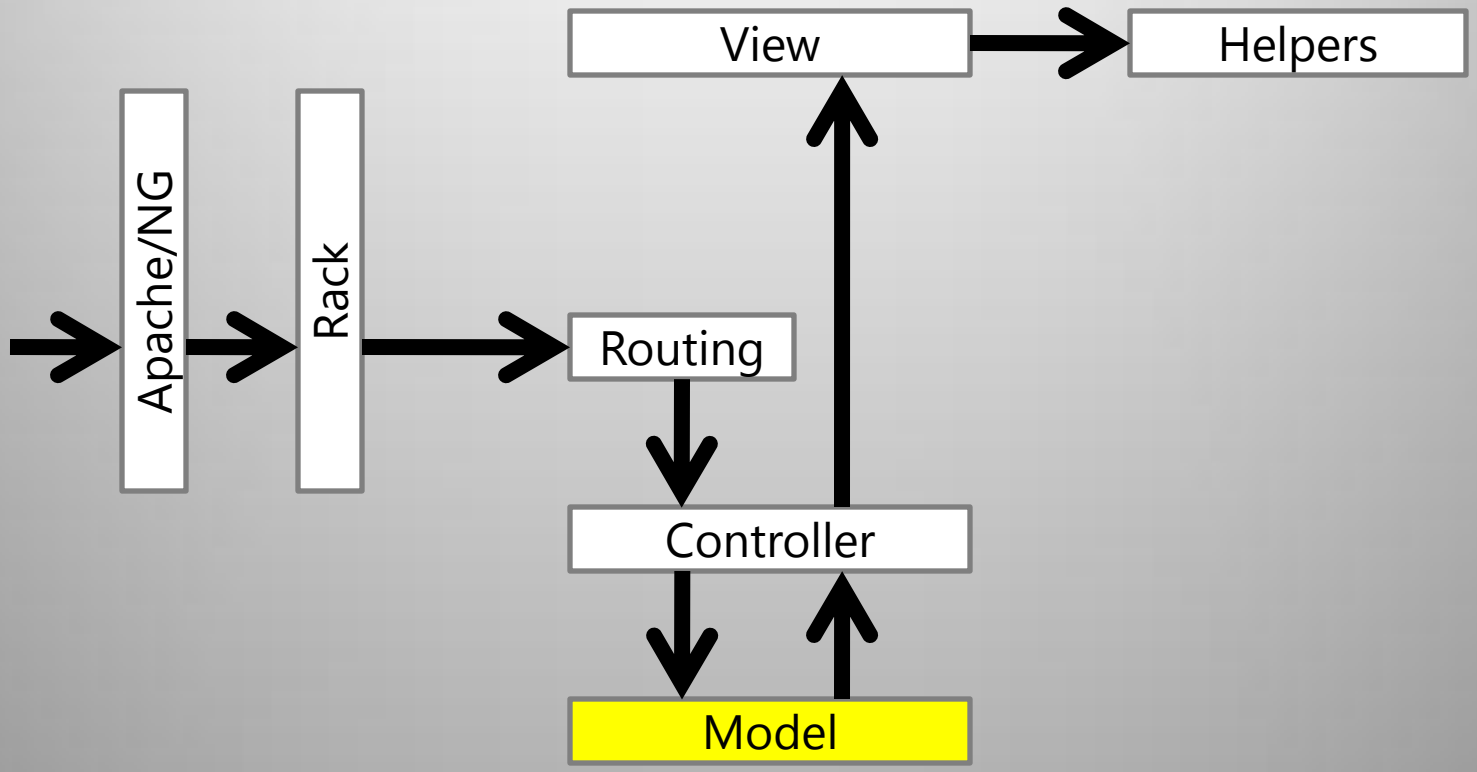
Terminology

- Route
- Controller#Action
- Model
- View
- Helper


View
Helpers
Apache/NG
Rack
Routing
Controller
Model

Convention Over Configuration

"*Software design paradigm which seeks to decrease the number of decisions that developers need to make, gaining simplicity, but not necessarily losing flexibility*


View
Helpers
Apache/NG
Rack
Routing
Controller
Model

Models

```ruby
# app/models/task.rb
class Task < ActiveRecord::Base
end
```

Fields pulled from DB (that convention thing) and injected in runtime

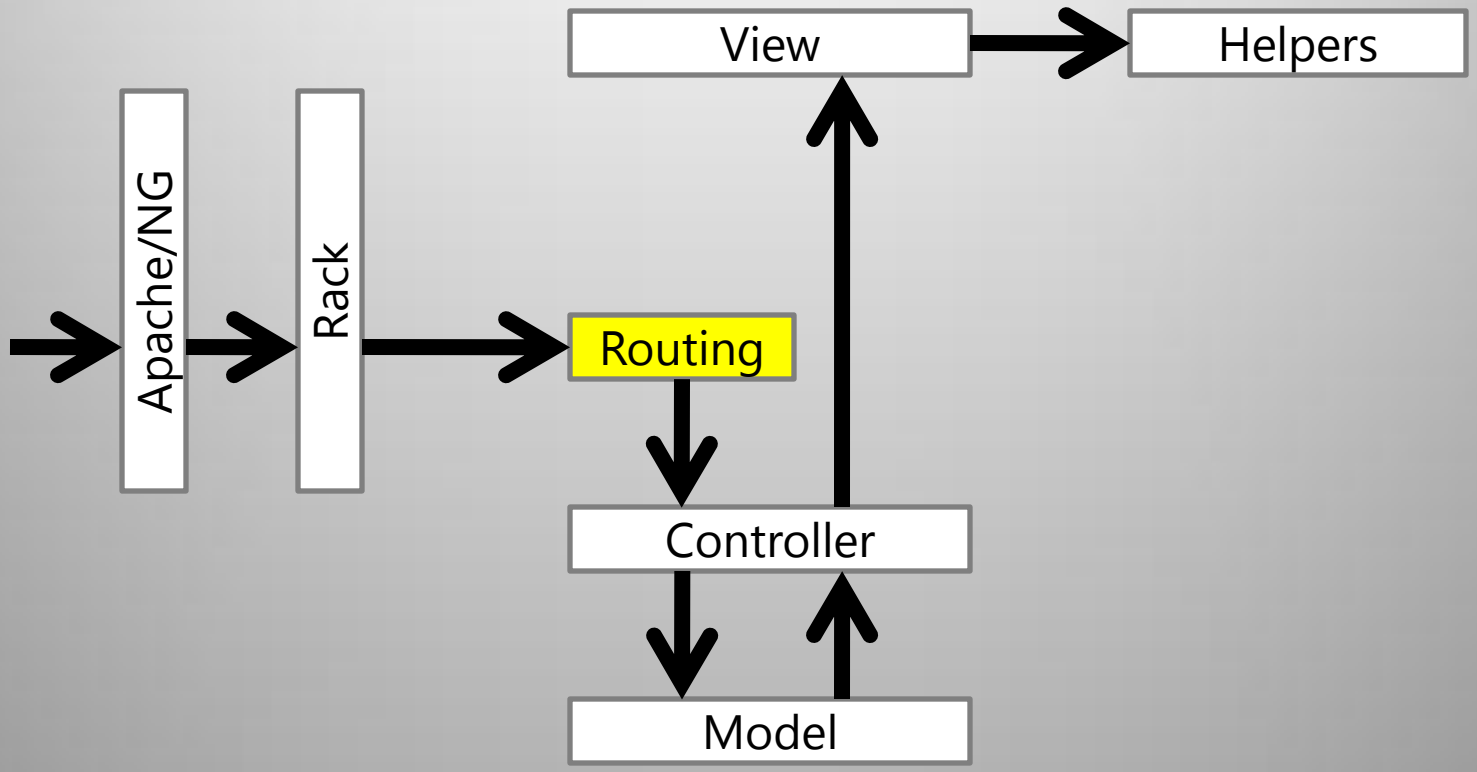
```ruby
# db/migrate/...create_tasks.rb
class CreateTasks < ActiveRecord::Migration
  def self.up
    create_table :tasks do |t|
      t.string :title
      t.string :body
      t.datetime :created
      t.integer :priority

      t.timestamps
    end
  end

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

```ruby
# ActiveRecord
@tasks = Task.all
@task = Task.find(params[:id])
@task = Task.new
@task.save
```

Migrate up and down. Creates our table and columns using a neat DSL.

Models – Meta Blitz

```
class Outpost < ActiveRecord::Base
  has_many :locations, :order=>'created_at DESC'
  belongs_to :user
  validates_presence_of :nid, :base_location
  validates_uniqueness_of :nid
  validates_format_of :base_location, :with=>/^[a-zA-Z0-9_-]{1,128}$/
  after_initialize :defaults



  #
  # callbacks
  #
  before_create :build_default_location
```


View
Helpers
Apache/NG
Rack
Routing
Controller
Model

Routing

- config/routes.rb
- $ rake routes

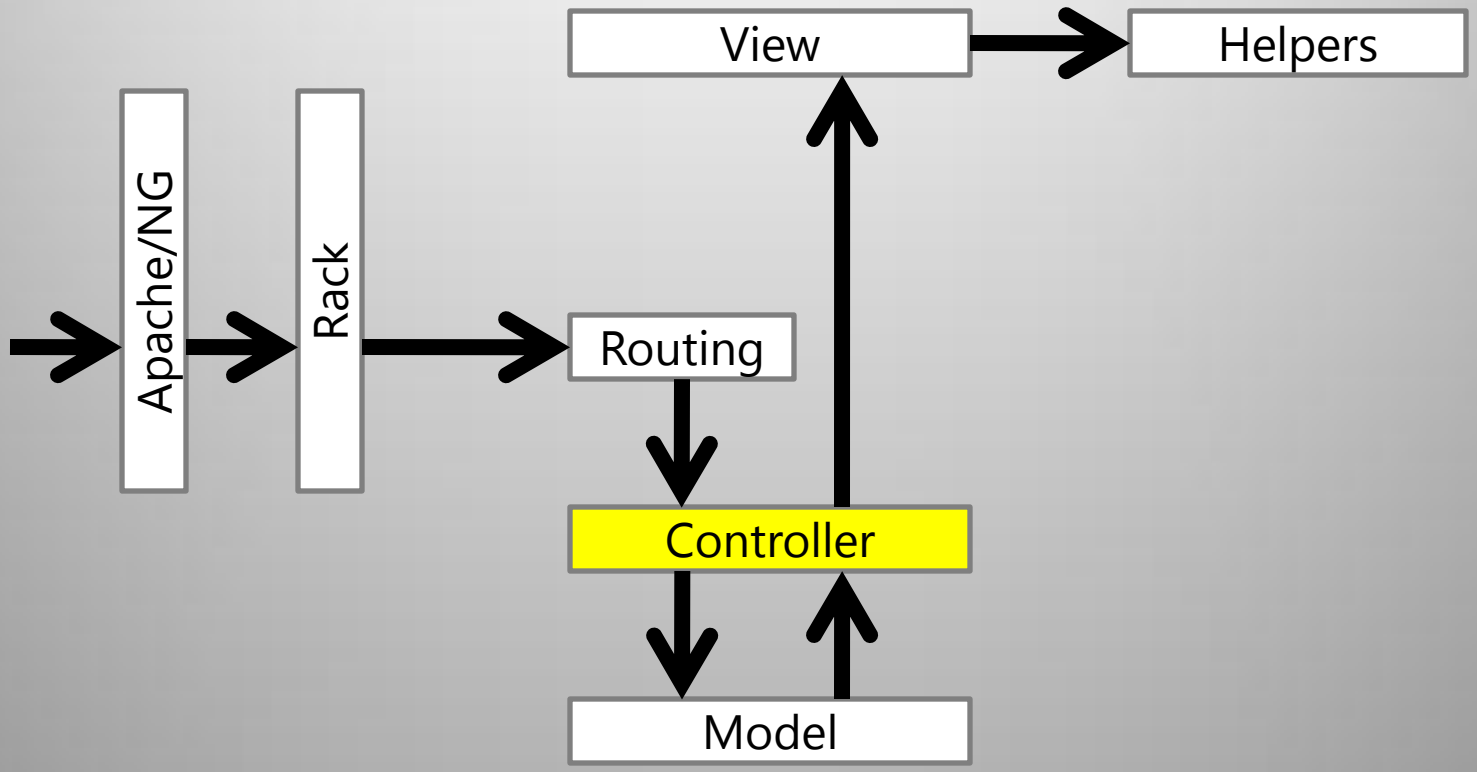
- RESTful routes
  - **`resources :entities`**
- Matching patterns
  - **`match ‘uri/pattern’ => ‘controller#action’`**
- Nested resources
- RESTful members
- Rack apps mounting
- More..

```
OutpostApp::Application.routes.draw do
  resources :foos

  devise_for :users


  resources :outposts do
    resources :locations, :only=>[:index] do
      resources :uploads, :only => [:show] #downl
    end
  end

  mount UploaderApp, :at  => '/submit'
  mount DownloaderApp, :at  => '/download'

  root :to => "outposts#index"
```

Mounting rack apps

App root

Declaring RESTful , nested resources


View
Helpers
Apache/NG
Rack
Routing
Controller
Model

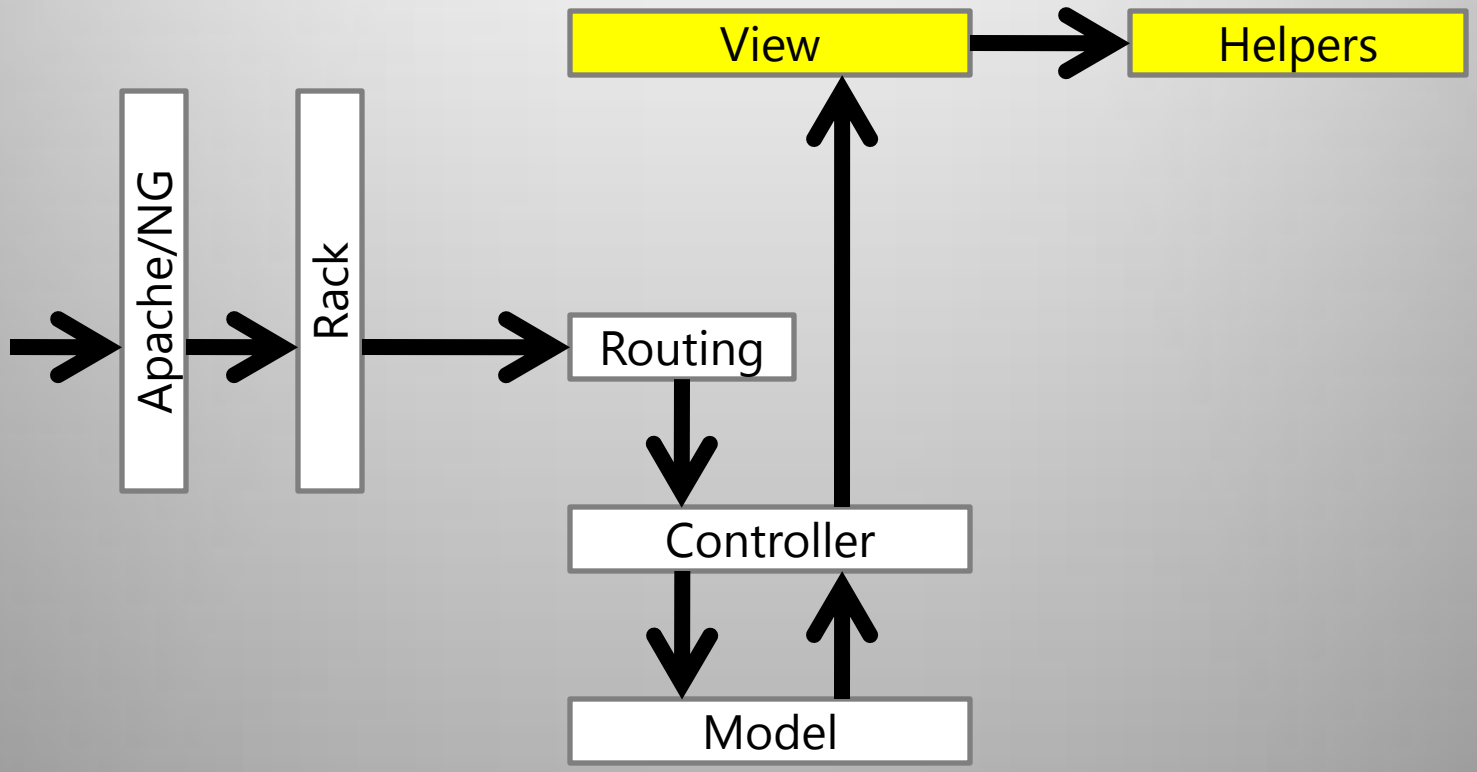
Controllers

Our model

Respond by Content Type

```
class TasksController < ApplicationController
  # GET /tasks
  # GET /tasks.xml
  def index
    @tasks = Task.all

    respond_to do |format|
      format.html # index.html.erb
      format.xml  { render :xml => @tasks }
    end
  end

  # GET /tasks/1
  # GET /tasks/1.xml
  def show
    @task = Task.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml  { render :xml => @task }
    end
  end

  # GET /tasks/new
  # GET /tasks/new.xml
  def new
    @task = Task.new
```

- Scaffolded CRUD
- RESTful (incl. Http verbs)
- ivars copied into views
- Cotent type detection/response
- Flow control (render, redirect, etc)
- Auth, CSRF, & more baked in.


View
Helpers
Apache/NG
Rack
Routing
Controller
Model

Views

Server side code in %%'s

```
<%= form_for(@task) do |f| %>
  <% if @task.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@task.errors.count, "error") %> pr
...from being saved:</h2>

      <ul>
      <% @task.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <%= f.label :title %><br />
    <%= f.text_field :title %>
  </div>
  <div class="field">
    <%= f.label :body %><br />
    <%= f.text_field :body %>
  </div>
```

- Layouts
- Partials
- Pluggable view engines: Erb, Haml, etc (also see Tilt)
- Unobtrusive Ajax

Thanks

dotan@paracode.com
@jondot

Questions?