

# Rails from the Ground Up!

with Jared Richardson  
RoleModel Software

# Rails

## Part One

### An Introduction

# Roadmap

- 0) Live demo
- 1) An overview of Rails & components
- 2) A light application (card deck)
- 3) Write a blog

# Fixing the Demo

```
rake db:drop:all
```

```
rake db:create
```

```
rails destroy scaffold page
```

```
rails generate scaffold author name:string
```

```
rails generate scaffold page content:text title:string  
tag:string resource:string author:references ip:string
```

# Routes

```
match 'pages' => 'pages#index'  
match 'page/new' => 'pages#new'  
match 'page/index' => 'pages#index'  
match 'page/edit' => 'pages#edit'  
match ':controller(/:action(/:id))(format)'
```

# Adding Authentication

<https://github.com/plataformatec/devise#readme>

add devise to Gemfile

rails generate devise:install

Then use... before\_filter :authenticate\_user!

# More Devise Links

<https://github.com/railsapps/rails3-devise-rspec-cucumber#readme>

<http://railsapps.github.com/tutorial-rails-devise-rspec-cucumber.html>

# Ruby on Rails

Ruby != Rails

Web framework

Many projects in one

Released in ~2005

Ruby had been in use for a decade!



DHH

David Heinemeier Hansson

Invented

Still leading



# Goal

Scratch his project's "itch"

Needed the framework

Abstracted it out of real work

Open sourced it

# Philosophy

Convention over configuration

DRY

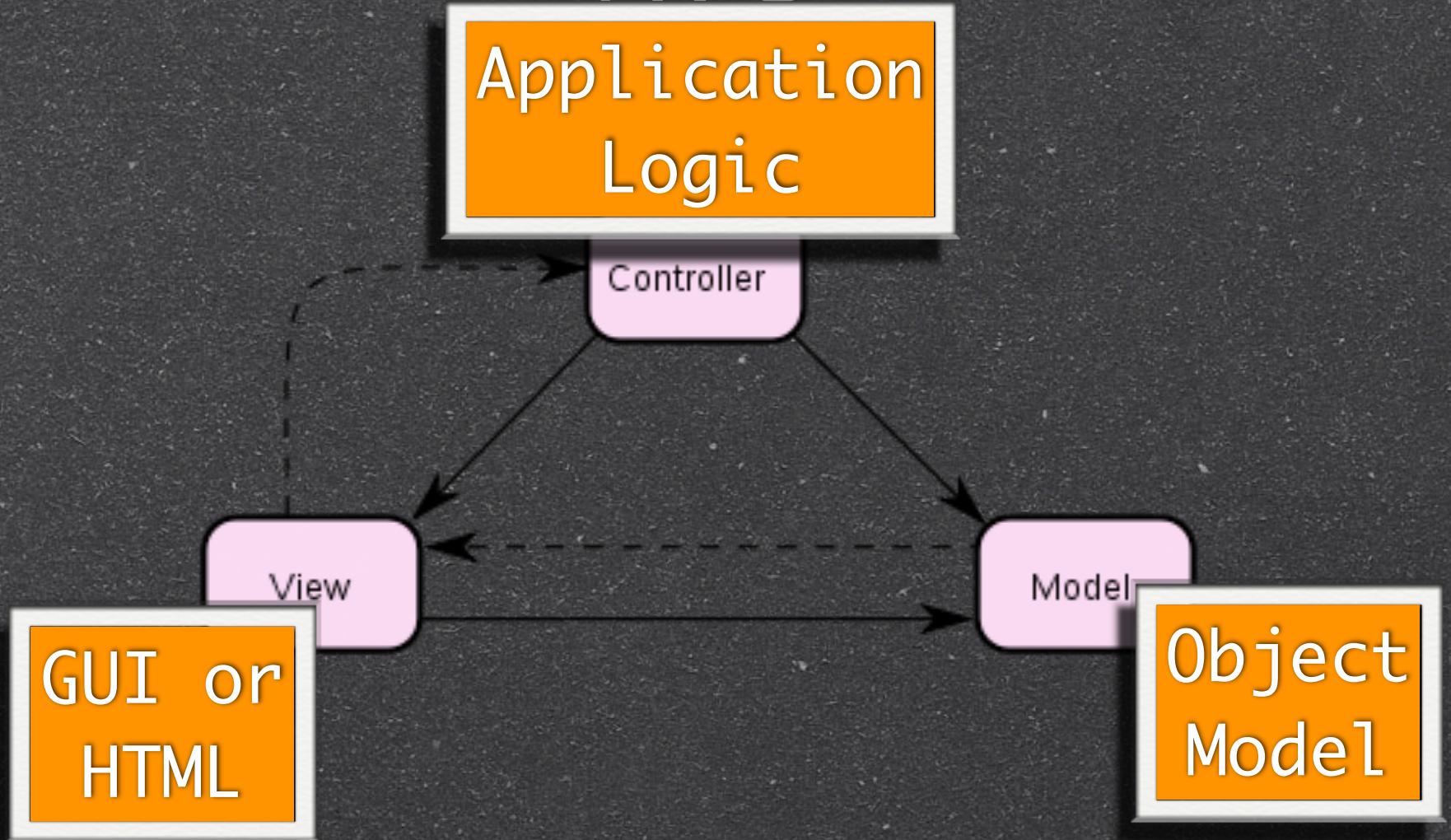
# Installation

rvm use 1.9.3

gem install rails

ALREADY DONE, RIGHT?

# MVC



# Diving In

# rails

type "rails"

rails new <app\_name>

# War!

rails new war

cd war

ls

What happened?

What's there?

# What Happened?

Directories created

Gems installed (via bundler)

app

config

db

doc

lib

log

public

Rakefile

script

test

vmp

vendor

app

assets

controllers

helpers

mailers

models

views

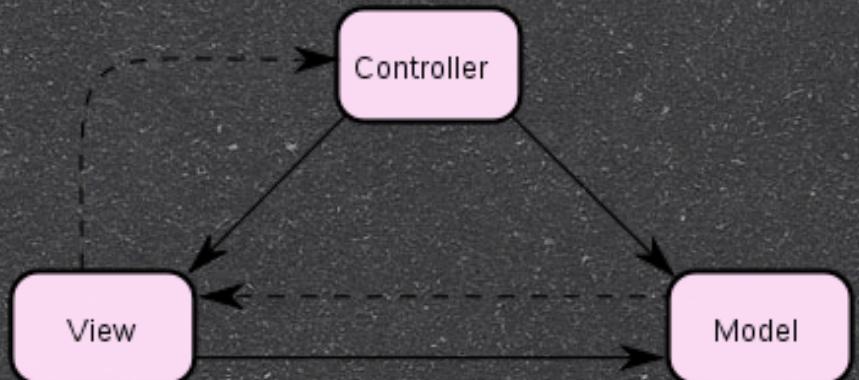
# app/assets

ignore for now

a spot for images, javascript, etc

Rails "pre-compiles" content

# app/controllers



C from MVC

Code that "controls" program flow

Usually one per model

# app/helpers

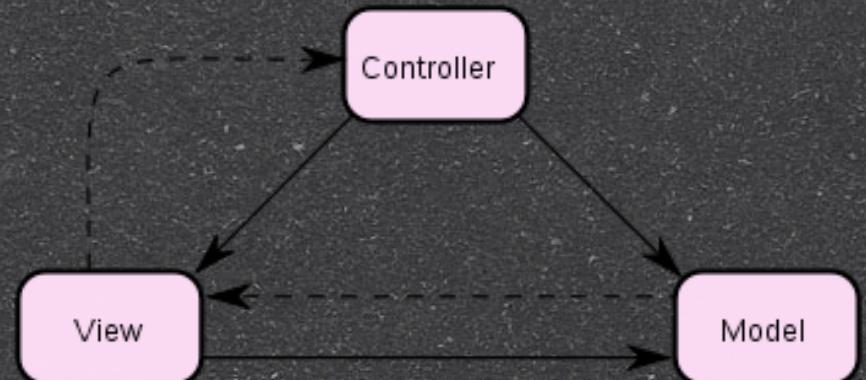
"helper" code

Code that's shared among many objects

# app/mailers

Ignore for now

# app/models

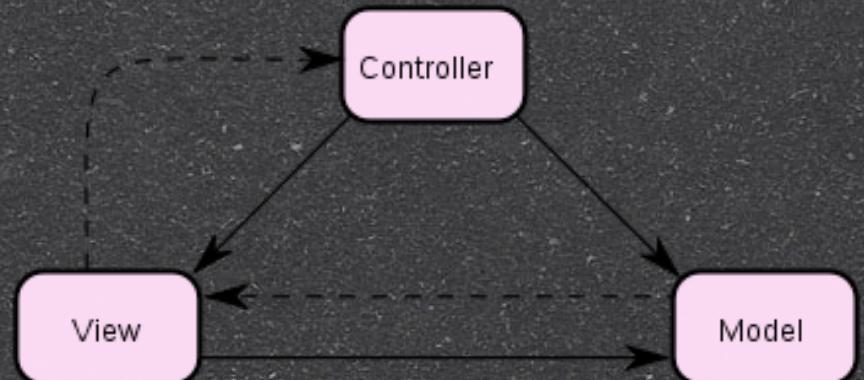


M of MVC

Data models

One for each object in the system

# app/views



V of MVC

Holds HTML, markup, & templates

Views render data for the user

# config

database.yml

environments

routes.rb

# database.yml

yaml => Yet Another Another Language

YAML Ain't a Markup Language

<keyword>:

<keyword>: value

# database.yml

```
development:
```

```
  adapter: sqlite3
```

```
  database: db/development.sqlite3
```

```
test:
```

```
  adapter: sqlite3
```

```
  database: db/test.sqlite3
```

# environments

Directory holds profiles

Profiles referenced in database.yml

-development

-test

-production

# routes.rb

Look familiar?

Open the file

Review the samples

Everything is commented out

# Start Your Server

`rails server`

or

`rails s`

# Look For . . .

```
jareds-MacBook:war jared$ rails server
=> Booting WEBrick
=> Rails 3.2.0.rc1 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2012-02-13 15:15:11] INFO  WEBrick 1.3.1
[2012-02-13 15:15:11] INFO  ruby 1.9.3 (2011-10-30) [x86_64-darwin10.8.0]
[2012-02-13 15:15:11] INFO  WEBrick::HTTPServer#start: pid=57029
port=3000
```

# What Did It Do?

<http://localhost:3000>



# Welcome aboard

You're riding Ruby on Rails!

[About your application's environment](#)

---

## Getting started

Here's how to get rolling:

1. Use `rails generate` to create your models and controllers

To see all available options, run it without parameters.

2. Set up a default route and remove `public/index.html`

Routes are set up in `config/routes.rb`.

3. Create your database

Run `rake db:create` to create your database. If you're not using SQLite (the default), edit `config/database.yml` with your username and password.

# New Terminal Window

cd into war

rails -h

```
jareds-MacBook:war jared$ rails -h
Rails is not currently installed on this system. To get the latest version, simply type:
$ sudo gem install rails
You can then rerun your "rails" command.
```

# rvm to the rescue

Create a new file

.rvmrc

rvm use 1.9.3

cd ..

cd war

```
jareds-MacBook:war jared$ vi .rvmrc
jareds-MacBook:war jared$ cd ..
jareds-MacBook:code jared$ cd war/
=====
= NOTICE
=====
= RVM has encountered a new or modified .rvmrc file in the current directory =
= This is a shell script and therefore may contain any shell commands. =
=
= Examine the contents of this file carefully to be sure the contents are =
= safe before trusting it! ( Choose view below to view the contents ) =
=====
Do you wish to trust this .rvmrc file? (/Users/jared/Dropbox/Rails from the Ground Up/code/war/.rvmrc)
y[es], n[o], v[iew], c[ancel]> y
Using /Users/jared/.rvm/gems/ruby-1.9.3-p0
jareds-MacBook:war jared$
```

# 2nd Time is Easy!

```
cd ..
```

```
cd war
```

# Available Rake Tasks

rake -T

rake stats

rake about

rake db:setup

What looks most useful?

`rails -h`

`rails generate`

Usage: `rails generate GENERATOR [args] [options]`

...

# `rails generate`

Focus on:

model

controller

migration

# Create a Template

```
rails generate model card_deck
```

# What Happened?

```
invoke  active_record
       create db/migrate/
20120213203148_create_card_decks.rb
       create app/models/card_deck.rb
invoke  test_unit
       create test/unit/card_deck_test.rb
       create test/fixtures/card_decks.yml
```

# What Does This Mean?

```
class CardDeck < ActiveRecord::Base  
end
```

# Make it Your Own!

We'll return later...

You'll paste in your War code

Migrate tests

# Where Do Tests Go?

war/test

war/test/fixtures

war/test/functional

war/test/integration

war/test/performance

war/test/unit

# Open Your Test

```
war/test/unit/card_deck_test.rb

require 'test_helper'

class CardDeckTest < ActiveSupport::TestCase

# def test_something do
#   assert true
# end

end
```

# Switching Gears

# Rails Components

Action Pack

Action Mailer

Active Model

Active Resource

Active Support

Railties

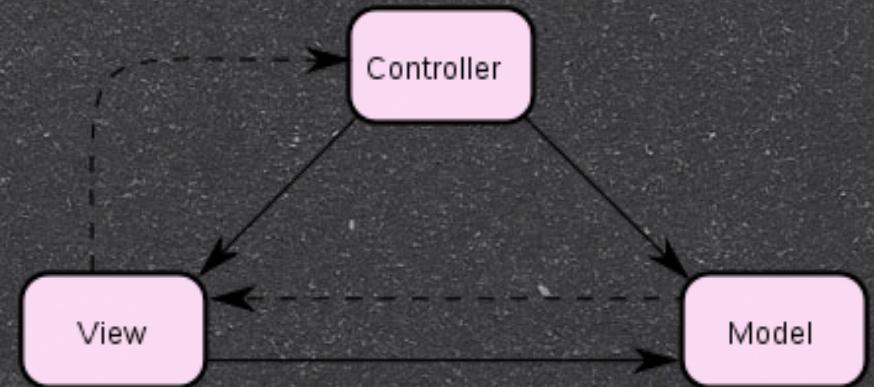
# Action Pack

The VC of MVC

Action Controller

Action View

Action Dispatch



# Action Pack Action Controller

Recognizes URLs

Dispatches to a controller's action

***Routes incoming requests***

# Action Controller Routes

<http://localhost/customers/21>

match "/customers/:id" => "customers#show"

Controller (class) => customer

Action (method) => show

Params (variables) => { :id => 21 }

OR.. customer.show(21)

# Action Pack

## Action View

Works with Action Controller

Renders the various markup pages

# Action Pack Action Dispatch

Advanced topic

Routes to more than just Rails

Any Rack application

# Back to Rails Components

Action Pack

Action Mailer

Active Model

Active Resource

Active Support

Railties

# Action Mailer

Three guesses? :)

Handles email services

More later...

# Back to Rails Components

Action Pack

Action Mailer

Active Model

Active Resource

Active Support

Railties

# Active Model

Interface between Action Pack and ORMs

Default ORM with Rails?

Active Record

# Active Model Active Record

Base for Rails models

Database independence

CRUD/ACID

Search

Model relations

# Your Model

card\_deck

# Remember . . .

invoke active\_record

create db/migrate/  
20120213203148\_create\_card\_decks.rb

create app/models/card\_deck.rb

invoke test\_unit

create test/unit/card\_deck\_test.rb

create test/fixtures/card\_decks.yml

# DB Migrations

What happens when you run:

`rake db:migrate`

# `rails dbconsole`

Reads `config/database.yml`

Drops you into the "right" console

Limited support (`mysql`, `postgres`,  
`sqlite*`)

"`rails db`" also works.

.help

.databases

.tables

.dump card\_decks

.dump schema\_migrations

select \* from card\_decks;

select \* from schema\_migrations;

# rails console

New window

rails console

cd = CardDeck.new

```
1.9.3p0 :001 > cd = CardDeck.new
SCHEMA (33.6ms) PRAGMA table_info("card_decks")
SCHEMA (0.2ms) SELECT name FROM sqlite_master WHERE type = 'table'
AND NOT name = 'sqlite_sequence' AND name = "card_decks"
SCHEMA (0.1ms) PRAGMA table_info("card_decks")
=> #<CardDeck id: nil, created_at: nil, updated_at: nil>
1.9.3p0 :002 > 
```

```
1.9.3p0 :001 > cd = CardDeck.new
SCHEMA (33.6ms) PRAGMA table_info("card_decks")
SCHEMA (0.2ms) SELECT name FROM sqlite_master WHERE type = 'table'
AND NOT name = 'sqlite_sequence' AND name = "card_decks"
SCHEMA (0.1ms) PRAGMA table_info("card_decks")
=> #<CardDeck id: nil, created_at: nil, updated_at: nil>
1.9.3p0 :002 > 
```

```
1.9.3p0 :001 > cd = CardDeck.new
SCHEMA (33.6ms) PRAGMA table_info("card_decks")
SCHEMA (0.2ms) SELECT name FROM sqlite_master WHERE type = 'table'
AND NOT name = 'sqlite_sequence' AND name = "card_decks"
SCHEMA (0.1ms) PRAGMA table_info("card_decks")
=> #<CardDeck id: nil, created_at: nil, updated_at: nil>
1.9.3p0 :002 > 
```

# Inspect and Save

cd.inspect

cd.save!

cd.inspect

# Again

```
cd = CardDeck.new
```

```
cd.inspect
```

```
cd.save!
```

```
cd.inspect
```

# ActiveRecord Tricks

cd\_1 = CardDeck.first

cd\_2 = CardDeck.last

CardDeck.count

CardDeck.methods.sort

CardDeck.methods.sort.each { |m|

  puts m

}

CardDeck.validators

More later...

# Return to rails dbconsole

.dump card\_decks

.explain ON

select \* from card\_decks;

# Back to Rails Components

Action Pack

Action Mailer

Active Model

Active Resource

Active Support

Railties

# Active Resource

Active Model for REST

Maps objects to RESTful web services

# REST?

Representational state transfer

Protocol for network services

CRUD

Post/Get/Put/Delete

# Examples

A Collection

<http://example.com/resources/>

An Element

<http://example.com/resources/item17>

# Back to Rails Components

Action Pack

Action Mailer

Active Model

Active Resource

Active Support

Railties

# Active Support

Utility classes

Ruby Extensions

# Railties

Core Rails code

Glues it all together

Builds new rails applications

# Back to Rails Components

Action Pack

Action Mailer

Active Model

Active Resource

Active Support

Railties

# Real World Rails Components

rails command line utilities

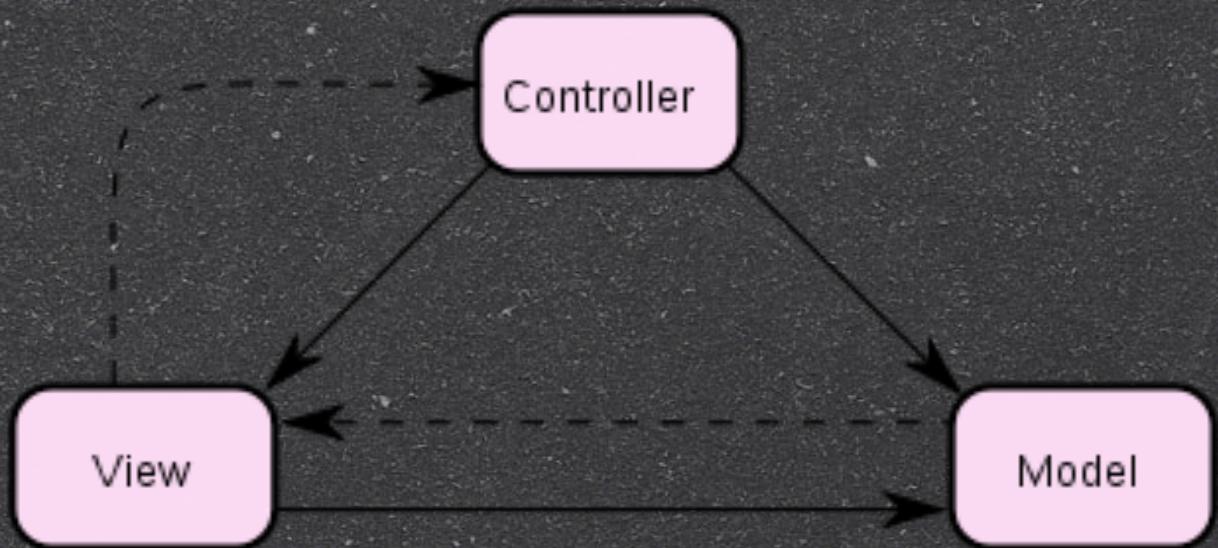
Active Record

Models

Controllers

Views

Routes



# Next...

Our first rails application!