

# Ruby on Rails 3

Robert Crida  
Stuart Corbishley

Clue Technologies

# Topic Overview

- What is Rails
- New in Rails 3
- New Project
- Generators
- MVC
- Active Record
- haml
- UJS
- RVM
- Bundler
- Migrations
- Factory Girl
- RSpec

# What is Rails

- Ruby is a dynamic, reflective, general purpose object-oriented programming language
- Syntax inspired by Perl
- Smalltalk-like features
- Originated in Japan in mid-1990's
- Rails is library written in Ruby for developing web applications

# New in Rails 3

- Core components decoupled and replaceable
- New query engine based on ARel
- Dependency management with Bundler
- Brand new Mail component improves rx/tx
- AJAX strategy changed to UJS

# Getting Started

- Build a new project framework with

`rails new  
project_name`

```
$ rails new spin
create
create  README
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/views/layouts/application.html.erb
create  app/mailers
create  app/models
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/environments
create  config/environments/development.rb
create  config/environments/production.rb
create  config/environments/test.rb
create  config/locales
create  config/locales/en.yml
create  config/boot.rb
create  config/database.yml
create  db
create  db/seeds.rb
create  doc
create  doc/README_FOR_APP
create  lib
create  lib/tasks
create  lib/tasks/.gitkeep
create  log
create  log/server.log
```

# Generators

- Scripts used to template infrastructure
- Get up and running quickly
- `rails g scaffold`

```
$ rails g scaffold model age:int name:string
invoke active_record
create db/migrate/20101115190852_create_models.rb
create app/models/model.rb
invoke test_unit
create test/unit/model_test.rb
create test/fixtures/models.yml
route resources :models
invoke scaffold_controller
create app/controllers/models_controller.rb
invoke erb
create app/views/models
create app/views/models/index.html.erb
create app/views/models/edit.html.erb
create app/views/models/show.html.erb
create app/views/models/new.html.erb
create app/views/models/_form.html.erb
invoke test_unit
create test/functional/models_controller_test.rb
invoke helper
create app/helpers/models_helper.rb
invoke test_unit
create test/unit/helpers/models_helper_test.rb
invoke stylesheets
create public/stylesheets/scaffold.css
```

# MVC

- Model
  - Business rules (relationships, scopes, validations)
- View
  - Presentation
- Controller
  - Restful actions (index, show, new, edit, create, update, destroy)

# Active Record

- Implements Active Record pattern
- Convention over configuration
- Infers attributes from table schema (method missing)
  - `_changed?`, `_was?`
  - `new`
  - `update_attributes`
  - `errors`

# Model Relationships

- belongs\_to
  - class Book { belongs\_to :author }
- has\_one
  - class Customer { has\_one :most\_recent\_order,  
:class\_name => 'Order',  
:order => 'created\_at DESC' }
- has\_many
  - class Library {  
has\_many :books  
has\_many :authors, :through => :books }

# Migrations

- Migrations describe incremental changes to the database schema
- Database agnostic
- Can be database specific
- Helps automate deployment

```
class CreateServiceDesks < ActiveRecord::Migration
  def self.up
    create_table :service_desks do |t|
      t.string :login,           :null => false
      t.string :email,          :null => false
      t.string :crypted_password, :null => false
      t.string :password_salt,  :null => false
      t.string :persistence_token, :null => false
      t.string :single_access_token, :null => false
      t.string :perishable_token, :null => false
      t.integer :login_count,    :null => false, :default => 0
      t.integer :failed_login_count, :null => false, :default => 0

      t.boolean :administrator, :null => false, :default => false

      t.timestamps
    end
  end

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

# Scopes

- Allow you to filter selections of a model or a has\_many relationship
  - `Book.recent_books`
  - `library.books.recent_books`
- Chainable
  - `Book.recent_books.childrens_books`
  - Effectively ANDs the filters in a single SQL statement
- Example:
  - `scope :recent_books, order('created_at DESC')`

# Validations

- For checking validity of model state prior to persisting
- Built-in standards
  - `validates_presence_of :name`
  - `validates_format_of :name, :with => regex`
  - `validates_numericality_of :age`
  - Etc
- Customizable

# ActionController

- Controls flow of application
  - Handles redirection
  - Receives requests
  - Renders responses using view code
- Restful, 7 standard actions
  - index, show, edit, new, create, update, destroy
- Can create other actions

# Example Index Action

```
def index
  @search = Model.search(params[:search])

  respond_to do |format|
    format.html { @models = @search.all.paginate :page => params[:page] }

    format.xml  { render :xml => @search.all }

    format.js   { render :partial => 'models/list', :locals => {:models =>
@search.all.paginate(:page => params[:page]) } }

    format.csv  { @models = @search.all }

  end
End
```

# ActionView

- Templated html with nested ruby code (erb)
- Include nested templates using 'partials'
- Form helpers – construct forms from models

```
<%= form_for @user do |f| %>
  <p>
    <b><%= f.label :name %></b>
    <%= f.text_field :name %>
  </p>
  <p>
    <%= f.submit "Create" %>
  </p>
<%- end %>
```

```
<form action="/users" method="post">
  <p>
    <b><label>Name</label></b>
    <input name="user[name]" type="text"/>
  </p>
  <p>
    <input type="submit" value="Create"/>
  </p>
</form>
```

# haml

- Much nicer than erb
- Infers tag closure from indentation
- sass for css

```
<%= form_for @user do |f| %>
  <p>
    <b><%= f.label :name %></b>
    <%= f.text_field :name %>
  </p>
  <p>
    <%= f.submit "Create" %>
  </p>
<%- end %>
```

```
= form_for @user do |f|
  %p
    %b
      = f.label :name
    = f.text_field :name
  %p
    = f.submit "Create"
```

# UJS

- Unobtrusive JavaScript
- Deprecating inline JavaScript
- Decoupled from Prototype
- Drivers for Prototype and JQuery, others...
- Now requires knowledge of JavaScript to implement AJAX – but far more powerful

# RVM

- Allows multiple versions of Ruby to be installed
- Gemsets are collections of libraries that can be associated with Ruby versions
- Usually for development but can work with production too

# Bundler

- Gemfile specifies which gems used by project
- Gemfile.lock fixes version of gem
- Isolates gems available to app
- `bundler install`
- `bundler update gem`

```
source 'http://rubygems.org'  
gem 'rails'
```

```
gem 'pg'  
gem 'authlogic'  
gem 'cancan'  
gem 'haml'  
gem 'meta_search'  
gem 'will_paginate', '~> 3.0.pre2'  
gem 'formtastic'  
gem 'stomp'  
gem 'jquery-rails'
```

```
group :test, :development do  
  gem 'rspec'  
  gem 'rspec-rails'  
end
```

```
group :test do  
  gem 'webrat'  
  gem 'factory_girl_rails'  
  gem 'shoulda'  
  gem 'autotest'  
  gem 'autotest-rails'  
end
```

# Factory Girl

- Power model creation factory to support testing
- Understands associations between objects
- Create objects
- Build hash of parameters to describe objects

```
Factory.sequence :email do |n|  
  "email%d@address.com" % n  
end
```

```
Factory.define :user do |f|  
  f.login { Factory.next :login }  
  f.email { Factory.next :email }  
  f.password "password"  
  f.password_confirmation "password"  
end
```

# RSpec

```
describe "User" do
  it "should have a login" do
    Factory.build(:user, :login => nil).should
    have_at_least(1).errors_on(:login)
  end
end
```

**FIN**

Questions?