

A close-up photograph of a young child with dark hair, smiling broadly. The child is wearing a bright pink, textured knit hooded jacket. They are seated in a stroller with a blue and white plaid fabric backrest. The background is slightly out of focus, showing a cluttered indoor space with various items like a wooden chair and a television. A semi-transparent dark grey banner with white text is overlaid across the middle of the image.

Happiness with Ruby on Rails

www.new-bamboo.co.uk

What is it?

- Nothing new... Well sort of.
- MVC
- Convention over configuration
- Ruby front to back

The three A's

- Activerecord
- Activesupport
- Actionpack

Activererecord

- Rich models
- No XML configuration files
- DSL to describe model relationships
- DSL to describe validation

```
class Bamboo < ActiveRecord::Base

  has_one :plant_pot
  has_many :leaves
  belongs_to :earth

  before_create :assign_plant_pot

  def grow_by(inches)
    self.height += inches
  end

  protected

  def assign_plant_pot
    self.plant_pot = PlantPot.new
  end

end
```

```
class UsersController < ApplicationController
  before_filter :setup_user, :except => [:index]

  def index
    @users = User.find(:all, :conditions => ["active = true", ])
    # @users = User.find_by_active(true)
  end

  def new; end

  def edit; end

  def create
    @user.attributes = params[:user]
    @user.save!
    flash[:notice] = "Your details were saved"
  rescue ActiveRecord::RecordInvalid
    flash[:error] = "There was a problem saving your details"
  end
  alias :update :create

  protected

  def setup_user
    @user = params[:id].blank? ? User.new : User.find(params[:id])
  rescue ActiveRecord::RecordNotFound
    render :file => "#{RAILS_ROOT}/public/404.html", :status => '404
Not Found'
  end
end
```

REST

- URI's become resources <http://www.new-bamboo.co.uk/users/234>
- POST - Create
- PUT - Update
- DELETE - Destroy
- GET - Show/Edit/Index

Routes

```
ActionController::Routing::Routes.draw do |map|  
  map.resources :user do |u|  
    # Singleton resource  
    u.resource :account  
  end  
end
```

/users
/users/2
/users/2;edit
/users/2/account

Routes

- `users_url` (GET) - Index
- `user_url(id)` (GET) - Show
- `user_url(id)` (DELETE) - Destroy
- `user_url(id)` (PUT) - Update

So what?

```
respond_to do |format|
  format.html {render }
  format.js {render :layout => false and return}
  format.xml {@user.to_xml}
  format.mob { render :partial => 'my_mobile_view'}
end
```

RJS

```
if @borrower.errors.size > 0
  error_message = ""
  @borrower.errors.each {|k,v| error_message << "\n* #{humanize(k)} #{v}"}
  page.alert('There were problems adding the borrower:' << error_message)
else
  page['borrowers_panel'].replace :partial =>
  'enquiries/creating/borrowers'
  page.visual_effect :highlight, @borrower.dom_id
end
page.call "ajax_loader.hide()"
```

Beautiful testing

Test::Unit

```
def test_should_have_a_name
  @enquirer = Enquirer.new
  assert !@enquirer.save
  assert_equal "is required", @enquirer.errors.on(:name)
end
```

RSpec

```
context "An auction that we want to remove" do

  setup do
    @auction = Auction.new
    @auction.attributes = AuctionSpecHelper.valid_auction_attributes
    @auction.stub!(:save!).and_return(:true)
  end

  specify "should be able to deactivate auction" do
    @auction.deactivate!
    @auction.active?.should_equal false
  end

end
```

So what's the difference?

- Vocabulary
- Hopefully a transition in thinking

It's Ruby that makes me
happy:)