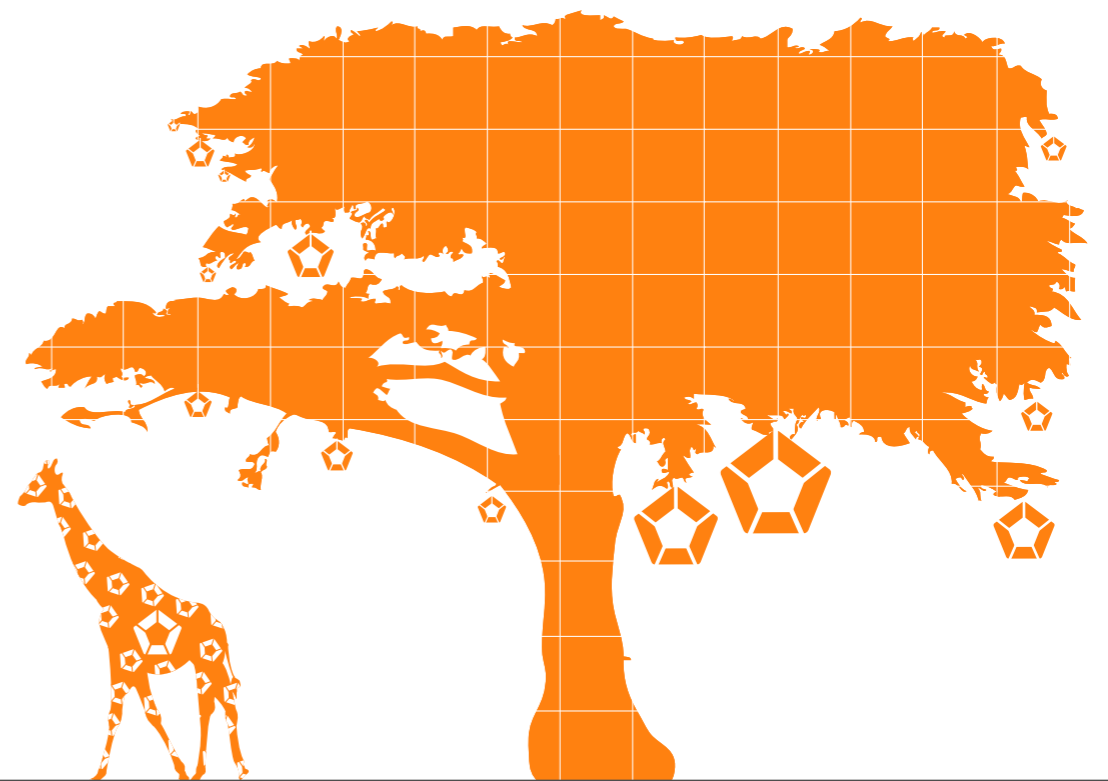




**James Golick**

**James Golick**  
[jamesgolick.com](http://jamesgolick.com)

giraffesoft™ 

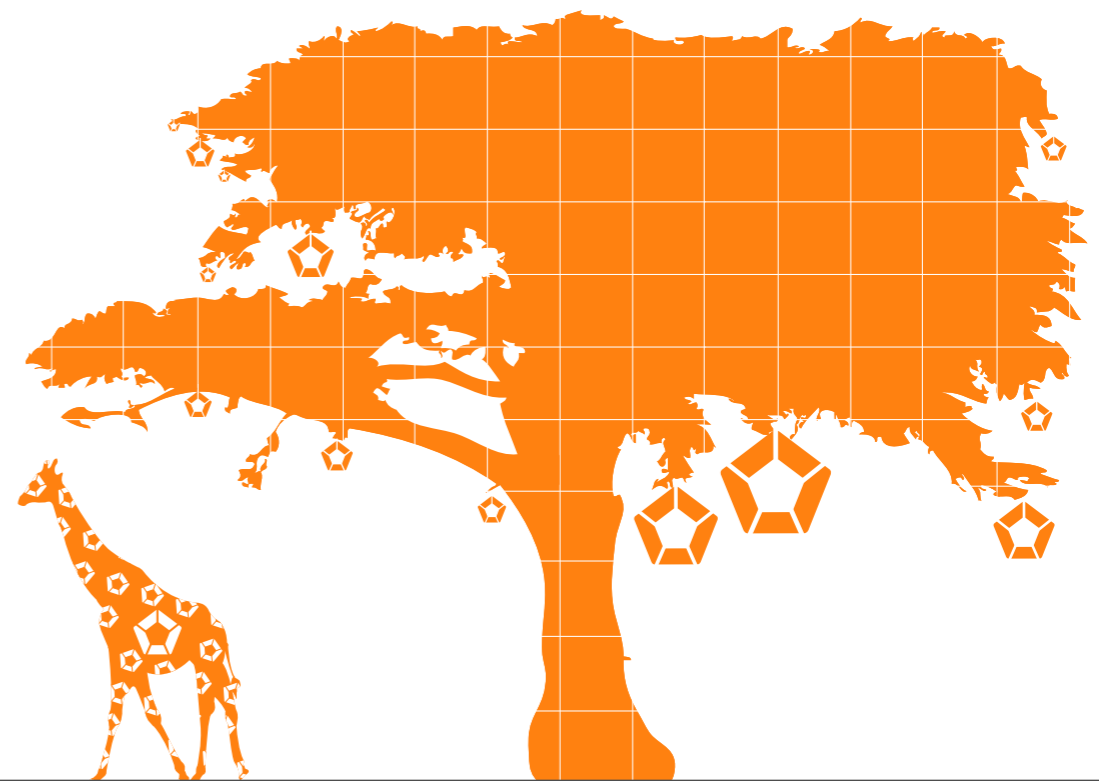


giraffesoft™ 

Past Clients:

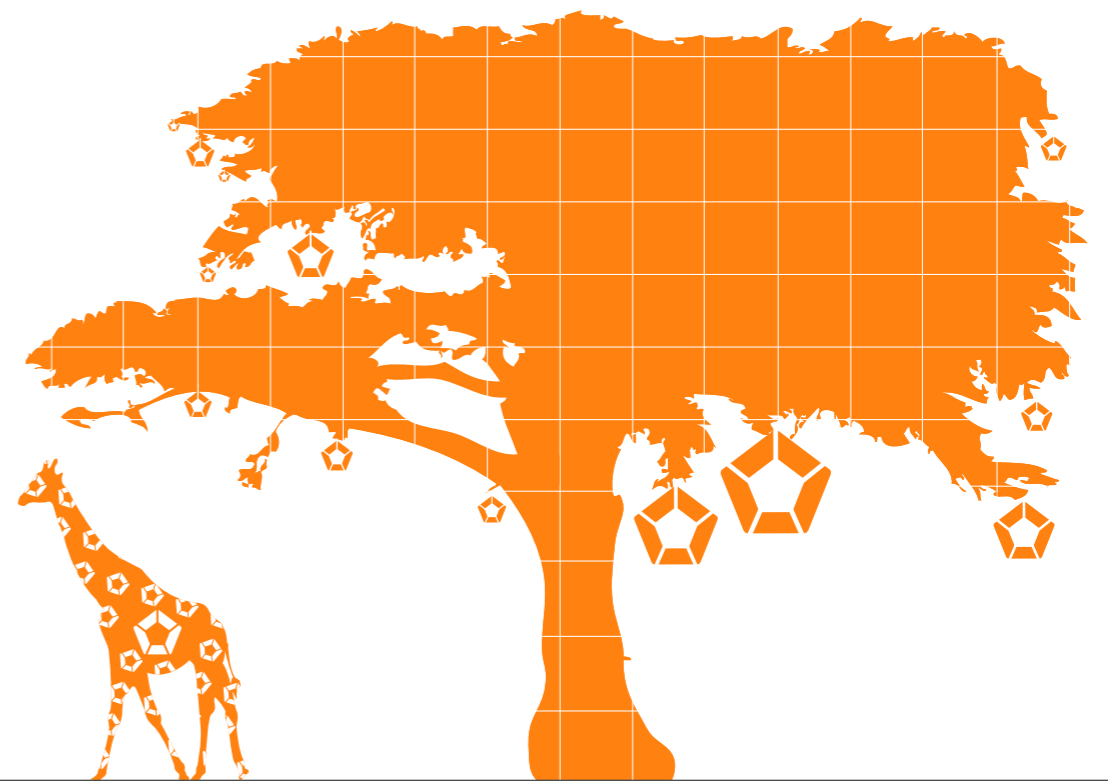


Arktyp



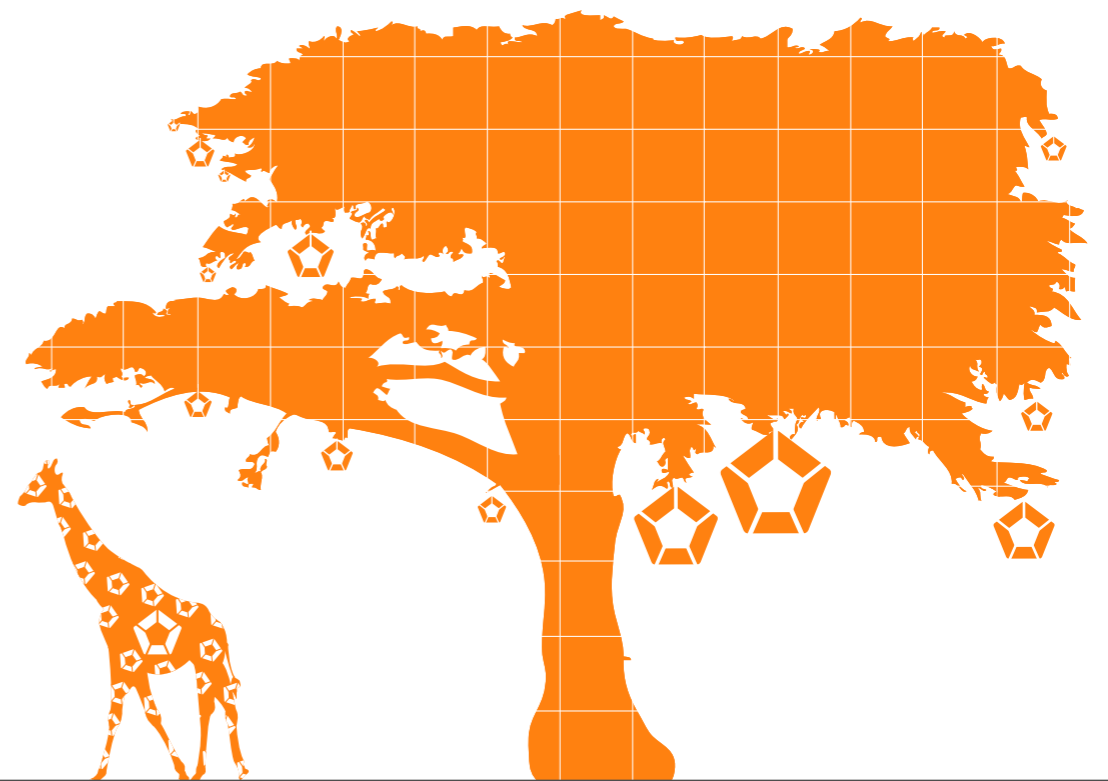


## Plugins I've Known and Loved





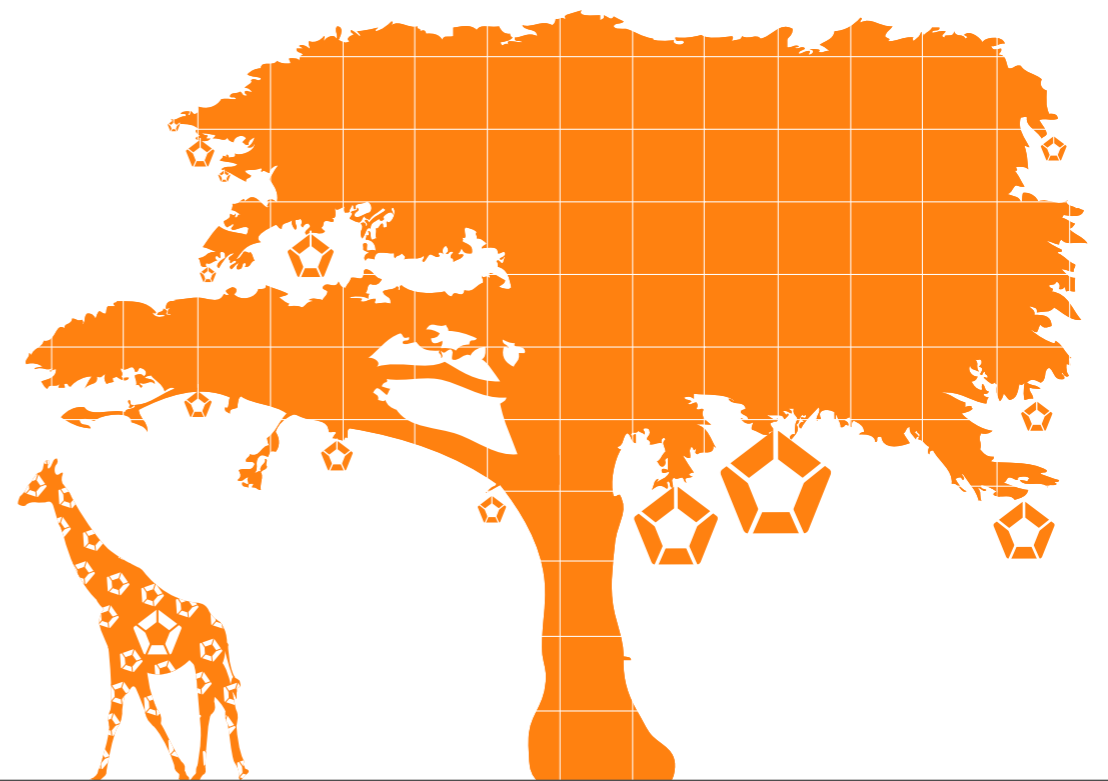
## Plugins I've Known and Loved #1





## Plugins I've Known and Loved #1

make\_resourceful





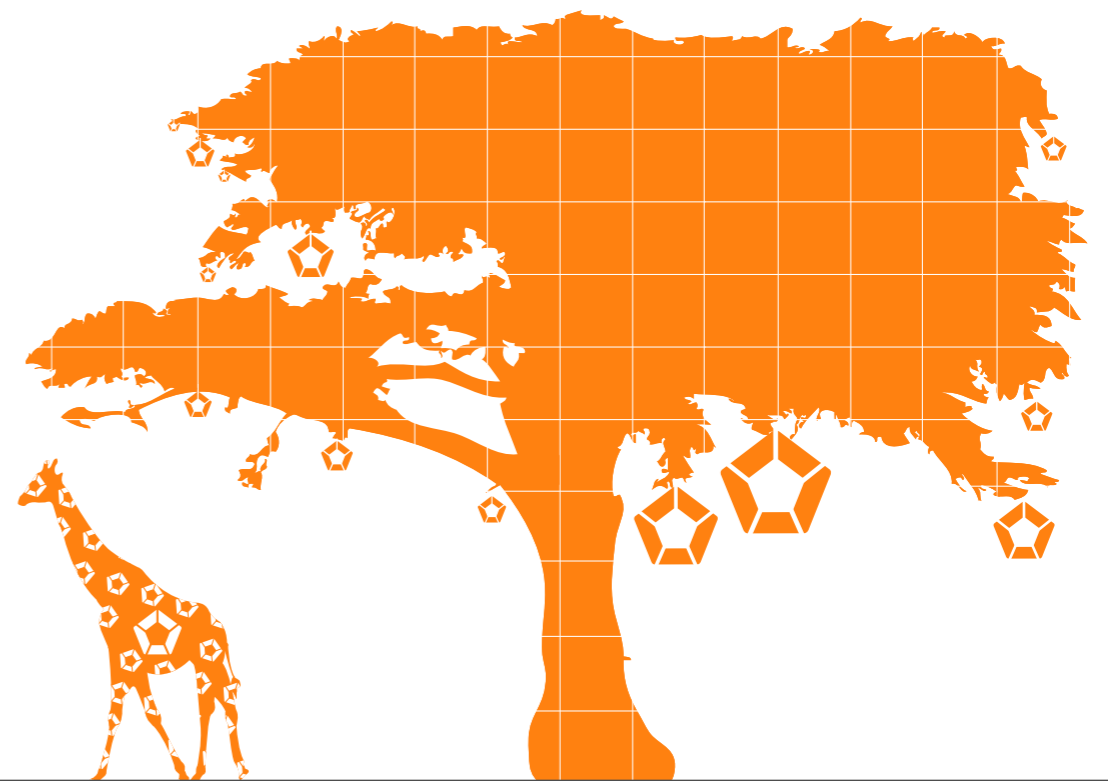


## Plugins I've Known and Loved #1

make\_resourceful

+

shoulda



# Make Resourceful

Seriously DRY controllers



```
class PostsController < ApplicationController
```

```
  def index
```

```
    @posts = Post.find(:all)
```

```
    respond_to do |format|
```

```
      format.html # index.rhtml
```

```
      format.xml { render :xml => @posts.to_xml }
```

```
    end
```

```
  end
```

```
  def show
```

```
    @post = Post.find(params[:id])
```

```
    respond_to do |format|
```

```
      format.html # show.rhtml
```

```
      format.xml { render :xml => @post.to_xml }
```

```
    end
```

```
  end
```

```
  def new
```

```
    @post = Post.new
```

```
  end
```

```
  def edit
```

```
    @post = Post.find(params[:id])
```

```
  end
```

```
  def create
```

```
    @post = Post.new(params[:post])
```

```
    respond_to do |format|
```

```
      if @post.save
```

```
        flash[:notice] = 'Post was successfully created.'
```

```
        format.html { redirect_to post_url(@post) }
```

```
        format.xml { head :created, :location => post_url(@post) }
```

```
      else
```

```
        format.html { render :action => "new" }
```

```
        format.xml { render :xml => @post.errors.to_xml }
```

```
      end
```

```
    end
```

```
  end
```

```
end
```

```
  def update
```

```
    @post = Post.find(params[:id])
```

```
    respond_to do |format|
```

```
      if @post.update_attributes(params[:post])
```

```
        flash[:notice] = 'Post was successfully updated.'
```

```
        format.html { redirect_to post_url(@post) }
```

```
        format.xml { head :ok }
```

```
      else
```

```
        format.html { render :action => "edit" }
```

```
        format.xml { render :xml => @post.errors.to_xml }
```

```
      end
```

```
    end
```

```
  end
```

```
  def destroy
```

```
    @post = Post.find(params[:id])
```

```
    @post.destroy
```

```
    respond_to do |format|
```

```
      format.html { redirect_to posts_url }
```

```
      format.xml { head :ok }
```

```
    end
```

```
  end
```

```
end
```

# That's It?

```
class PostsController < ApplicationController  
  
  make_resourceful do  
    actions :all  
  end  
  
end
```

# Helpers

# Helpers

- collection

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all
  end

  private
  def current_objects
    @current_objects ||= current_model.find(:all, :page => {:size => 10, :current => params[:page]})
  end

end
```

# Helpers

- collection
- creation

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all
  end

  private
  def current_objects
    @current_objects ||= current_model.find(:all, :page => {:size => 10, :current => params[:page]})
  end

  def build_object
    @current_object = Post.new params[:post]
  end

end
```

# Helpers

- collection
- creation
- param

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all
  end

  private

  def current_objects
    @current_objects ||= current_model.find(:all, :page => {:size => 10, :current => params[:page]})
  end

  def build_object
    @current_object = Post.new params[:post]
  end

  def current_param
    params[:id]
  end

end
```



# Helpers

- collection
- creation
- param
- object fetching

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all
  end

  private

  def current_objects
    @current_objects ||= current_model.find(:all, :page => {:size => 10, :current => params[:page]})
  end

  def build_object
    @current_object = Post.new params[:post]
  end

  def current_param
    params[:id]
  end

  def current_object
    @current_object ||= current_model.find(current_param)
  end

end
```

# Callbacks

# Callbacks

- before

```
class PostsController < ApplicationController  
  
  make_resourceful do  
    actions :all  
  
    before :create do  
      current_object.author = current_user  
    end  
  end  
  
end
```

# Callbacks

- before
- after

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all

    before :create do
      current_object.author = current_user
    end

    after :create do
      flash[:notice] = "Successfully created..."
    end
  end
end
```

# Callbacks

- before
- after
- response

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all

    before :create do
      current_object.author = current_user
    end

    after :create do
      flash[:notice] = "Successfully created..."
    end

    response_for :create do |wants|
      wants.rjs
    end
  end
end
```

# Publish

```
class PostsController < ApplicationController  
  
  make_resourceful do  
    actions :all  
    publish :xml, :json, :yaml, :attributes => [:author, :title, :body]  
  end  
  
end
```

# Publish

```
class PostsController < ApplicationController

  make_resourceful do
    actions :all
    publish :xml, :json, :yaml, :attributes => [:author, :title, :body, {:comments => [:author, :body]}]
  end
end
```

**What about testing?**



# Shoulda

Seriously DRY tests



```
class PostsControllerTest < Test::Unit::TestCase
```

```
  def setup
```

```
    @controller = PostsController.new
```

```
    @request = ActionController::TestRequest.new
```

```
    @response = ActionController::TestResponse.new
```

```
  end
```

```
  def test_should_get_index
```

```
    get :index
```

```
    assert_response :success
```

```
    assert assigns(:posts)
```

```
  end
```

```
  def test_should_get_new
```

```
    get :new
```

```
    assert_response :success
```

```
  end
```

```
  def test_should_create_post
```

```
    old_count = Post.count
```

```
    post :create, :post => { }
```

```
    assert_equal old_count+1, Post.count
```

```
    assert_redirected_to post_path(assigns(:post))
```

```
  end
```

```
  def test_should_show_post
```

```
    get :show, :id => 1
```

```
    assert_response :success
```

```
  end
```

```
  def test_should_get_edit
```

```
    get :edit, :id => 1
```

```
    assert_response :success
```

```
  end
```

```
  def test_should_update_post
```

```
    put :update, :id => 1, :post => { }
```

```
    assert_redirected_to post_path(assigns(:post))
```

```
  end
```

```
  def test_should_destroy_post
```

```
    old_count = Post.count
```

```
    delete :destroy, :id => 1
```

```
    assert_equal old_count-1, Post.count
```

```
    assert_redirected_to posts_path
```

```
  end
```

```
end
```

# That's It?

```
class PostsControllerTest < Test::Unit::TestCase

  def setup
    @controller = PostsController.new
    @request     = ActionController::TestRequest.new
    @response    = ActionController::TestResponse.new
    @post       = Post.find(1)
  end

  should_be_restful do |resource|
    resource.formats = [:html]
  end

end
```

# Helpers

# Helpers

- params

```
class PostsControllerTest < Test::Unit::TestCase

  def setup
    @controller = PostsController.new
    @request     = ActionController::TestRequest.new
    @response    = ActionController::TestResponse.new
    @post       = Post.find(1)
  end

  should_be_restful do |resource|
    resource.formats = [:html]

    resource.create.params = { :title => "A Great Post", :body => "Lorem ipsum dolor sit amet, consectetu
  end
end
```

# Helpers

- params
- flash

```
class PostsControllerTest < Test::Unit::TestCase

  def setup
    @controller = PostsController.new
    @request     = ActionController::TestRequest.new
    @response    = ActionController::TestResponse.new
    @post        = Post.find(1)
  end

  should_be_restful do |resource|
    resource.formats = [:html]

    resource.create.params = {:title => "A Great Post", :body => "Lorem ipsum dolor sit amet, consectetur"}

    resource.create.flash = /created!/i
  end

end
```

# Helpers

- params
- flash
- redirection

```
class PostsControllerTest < Test::Unit::TestCase

  def setup
    @controller = PostsController.new
    @request     = ActionController::TestRequest.new
    @response    = ActionController::TestResponse.new
    @post       = Post.find(1)
  end

  should_be_restful do |resource|
    resource.formats = [:html]

    resource.create.params = {:title => "A Great Post", :body => "Lorem ipsum dolor sit amet, consectetur"}

    resource.create.flash = /created!/i

    resource.create.redirect = "/"
  end
end
```

# Contexts

```
context "not logged in"

  should_be_restful do |resource|
    resource.denied.actions = :all
    resource.denied.flash   = /must be logged in/i
    resource.denied.redirect = 'login_url'
  end

end

context "logged in" do

  setup do
    login_as(:james)
  end

  should_be_restful do |resource|
    resource.formats = [:html]
  end

end
```



# ActiveRecord

# ActiveRecord

## Associations

- `should_belong_to`
- `should_have_many`
- `should_have_one`
- `should_have_and_belong_to_many`

# ActiveRecord

## Associations

- `should_belong_to`
- `should_have_many`
- `should_have_one`
- `should_have_and_belong_to_many`

## Validations

- `should_require_attributes`
- `should_require_unique_attributes`
- `should_only_allow_numeric_values_for`
- `should_allow_values_for`
- `should_ensure_length_in_range`
- `should_ensure_value_in_range`
- `should_not_allow_values_for`

# Demo Time!

Building and Testing a Blog





**James Golick**

[jamesgolick.com](http://jamesgolick.com)