Red. Green. Refactor.

Phoenix Rails - June 2010

Refactoring?

Refactoring is a disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior. Its heart is a series of small behavior preserving transformations. Each transformation (called a 'refactoring') does little, but a sequence of transformations can produce a significant restructuring. Since each refactoring is small, it's less likely to go wrong. The system is also kept fully working after each small refactoring, reducing the chances that a system can get seriously broken during the restructuring.

http://www.refactoring.com

Requirements

- Passing Tests
- Good Nose
- Timebox
- Measurement

Refactoring Rails

Simplify

```
# FLOG
# ------------------------------
# 10.9: flog total
# 10.9: flog/method average
#
# 10.9: main#active?

def active?
  today = Time.now.strftime('%Y-%m-%d %H:%M')
  today >= self.begin_date.strftime('%Y-%m-%d 06:00') && today <= \
self.end_date.strftime('%Y-%m-%d 23:59')
end
```

Before

Refactoring Rails

Simplify

```
# FLOG
# ----------------------
# 9.0: flog total
# 9.0: flog/method average
#
# 9.0: main#active?

def active?
  now      = Time.now
  earliest = begin_date.strftime('%Y-%m-%d 06:00')
  latest   = end_date.strftime('%Y-%m-%d 23:59')

  now >= earliest && now <= latest
end
```

Refactoring Rails

South Beach Controllers

```ruby
# FLOG
# -------------------------
# 68.0: flog total
# 68.0: flog/method average
#
# 68.0: main#update

# PUT /statuses/1
# PUT /statuses/1.xml
def update
  @status = current_account.statuses.find(params[:id])
  respond_to do |format|
    if @status.update_attributes(params[:status])
      statuses = current_account.statuses.all
      statuses.each do |status|
        unless status.id == @status.id
          if status.position && (status.position >= @status.position)
            status.position = status.position + 1
            status.save
          end
        end
      end
      statuses = current_account.statuses.all
      previousStatusPosition = 0
      statuses.each do |status|
        status.position = previousStatusPosition + 1
        status.save
        previousStatusPosition = status.position
      end
```

Before

Refactoring Rails

South Beach Controllers

```
# FLOG
# -----------------
# 16.1: flog total
# 16.1: flog/method average
#
# 16.1: main#update

# app/views/controllers/statuses_controller.rb
def update
  @status = current_account.statuses.find(params[:id])
  if @status.update_attributes(params[:status])
    flash[:notice] = 'Status was successfully updated.'
    redirect_to(statuses_path)
  else
    render :action => "edit"
  end
end
```

After

Refactoring Rails

Views: Not for logic

```
<% if @panel1_product.max_qty > 0 && @panel1_product.remaining < 20 && \
@panel1_product.available? %>
  <% if @panel1_product.remaining == 1 %>
    <p class="remaining"><%= @panel1_product.remaining %> item left!</p>
  <% else %>
    <p class="remaining"><%= @panel1_product.remaining %> items left!</p>
  <% end %>
<% end %>
```

Before

Refactoring Rails

Views: Not for logic

```
# app/views/products/index.html.erb
<%= display_remaining_items(@featured_product) %>

# app/helpers/products_helper.rb
def display_remaining_items(product)
  return "" unless product.available?
  remaining = pluralize(product.remaining, "item")
  content_tag(:p, :class => "remaining") do
    "#{remaining} left!"
  end
end
```

```
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_
      public_newsletter_messages_url(newsletter, :org_type => 'A', :section_type => '
  end
when 'B'
  if public_link == false
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
      organization_newsletter_messages_url(organization, newsletter, :org_type => 'B'
  else
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
      public_newsletter_messages_url(newsletter, :org_type => 'B', :section_type => '
  end
when 'C'
  if public_link == false
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
      organization_newsletter_messages_url(organization, newsletter, :org_type => 'C'
  else
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
      public_newsletter_messages_url(newsletter, :org_type => 'C', :section_type => '
  end
when 'D'
  if public_link == false
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
      organization_newsletter_messages_url(organization, newsletter, :org_type => 'D'
  else
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
      public_newsletter_messages_url(newsletter, :org_type => 'D', :section_type => '
  end
when 'E'
  if public_link == false
    html << sidebar_link(newsletter.get_messages_title(Organization.find(org_id[org_t
```

Where to Refactor?

- unDRY (flay)
- Complexity (flog)
- Smells (reek/roodi)
- Pair

http://github.com/clayton/phxrails-201006