

Unit 10 Individual Assignment - Caching

Caching with rails helps alleviate expensive round trips to the database by returning only what is changed to the web client in a shorter amount of time. Rails has three types of caching techniques which are page caching, action caching, and fragment caching.

- **Page Caching:** This caching method allows the webserver (could be Apache, could be nginx) to generate the requested page. The request would not go through the Rails stack and would provide very quick responses. Since web servers are serving the page from the system, events such as cache expiration cause issues with this caching method. **Page caching has been removed from Rails 4.**
- **Action Caching:** While page caching has problems with authentication, action caching allows requested web pages to run through the Rails stack. This allows any before filters to run to set any needed restrictions and authentication before serving cached data. **Action Caching is also removed from Rails 4.**
- **Fragment Caching:** Fragment caching approaches the problem of caching dynamically created pages that may require caches for several different parts. Not all caches have the same caching characteristics either. Fragment caching surrounds view logic in cache blocks and serves data from cache stores when new requests come in. An example of fragment caching is having an orders page that updates orders placed in real time. The list of products is the part of the page that we want to keep cached.

This first part updates our page each time an order is placed.

```
<% Order.find_recent.each do |o| %>
  <%= o.buyer.name %> bought <%= o.product.name %>
<% end %>
```

Since the product list is not likely to change in a short time, it is cached.

```
<% cache do %>
  All available products:
  <% Product.all.each do |p| %>
    <%= link_to p.name, product_url(p) %>
  <% end %>
<% end %>
```

Another Rails feature for caching is SQL Caching. Rails caches each result set returned by a query so that if the same query is executed, the cached result set is used instead of querying the database a second time. For example:

```
Class ProductsController < ApplicationController
  Def index
    # Run a query
    @products = Product.all

    # do some stuff

    # Run the same query, which Rails will use
    # the cached result set from earlier
    @products = Product.all
  end
end
```

Website used for information on caching is
http://edgeguides.rubyonrails.org/caching_with_rails.html.