

Rails Caching

Ynon Perek



How Caches Work



→ give me /posts/2 →

←

Let me calculate ... here you go



→ give me /posts/2 →

←

Ah I know this one ... here you go



**“There are only two hard things
in Computer Science: cache
invalidation and naming things.”**

–Phil Karlton

Rails Cache Mechanisms

- ☐ **Page Cache**
- ☐ **Action Cache**
- ☐ **Fragment Cache**
- ☐ **Low Level Cache**
- ☐ **SQL Cache**

Fragment Cache

```
<% @products.each do |product| %>  
  <% cache product do %>  
    <%= render product %>  
  <% end %>  
<% end %>
```


Fragment Cache - Result

- ☐ Creates a new unique cache key:
- ☐ `views/products/1-201505056193031061005000/
bea67108094918eeba42cd4a6e786901`
- ☐ Save the fragment as the value

Fragment Cache Key

view path

Product ID

Updated At

**views/products/1-201505056193031061005000/
bea67108094918eeba42cd4a6e786901**

View Hash

Fragment Cache Invalidation

- ☐ New key is created automatically when:
 - ☐ Model's updated_at is changed
 - ☐ Model id changes
 - ☐ View path changes
 - ☐ Template content changes

Demo

- ☐ Run `bin/rails dev:cache` to turn on caching in development
- ☐ We'll use memcached so need to define it in `config/environments/development.rb`:
- ☐ `config.cache_store = :mem_cache_store`

Demo

```
<tbody>
  <% @tasks.each do |task| %>
    <% cache task do %>
      <tr class="task-line">
        <td><%= task.user&.name %></td>
        <td><%= task.description %></td>
      <% end %>
    </tr>
  <% end %>
</tbody>
</table>
```


Houston, we have a problem

- ☐ **When another user signs in, cache is used and user gets the wrong items**

Solution: Use array of keys

□ `<% cache [current_user, task] do %>`

Caching And Associations

```
class Project < ActiveRecord::Base  
end
```

```
class Todolist < ActiveRecord::Base  
  belongs_to :project, touch: true  
end
```

```
class Todo < ActiveRecord::Base  
  belongs_to :todolist, touch: true  
end
```

```
# This will trigger todo.todolist.touch!,  
# which in turn triggers todo.todolist.project.touch!  
todo.update_attributes!(content: "New stuff")
```


JSON Caching

```
json.cache! ['v1', @person], expires_in: 10.minutes do  
  json.extract! @person, :name, :age  
end
```


Low Level Caching

```
class Product < ApplicationRecord
  def competing_price
    Rails.cache.fetch("#{cache_key}/competing_price",
                      expires_in: 12.hours) do
      Competitor::API.find_price(id)
    end
  end
end
```


Russian Doll Caching

```
<h1>Products</h1>
<% @products.each do |product| %>
  <% cache product do %>
    <%= render product %>
  <% end %>
<% end %>
```

```
<article>
  <h1><%= product.title %></h1>

  <ul>
    <% product.variants.each do |variant| %>
      <% cache(variant) do %>
        <%= render variant %>
      <% end %>
    <% end %>
  </ul>
</article>
```


Tips & Tricks

Cache & Testing

- ☐ **Disable cache when load testing using the null cache store:**
- ☐ **`config.cache_store = :null_store`**

Careful with Page Caching / Action Caching

- ☐ Page cache - save full page result to a file
- ☐ Action cache - Page cache + filters
- ☐ Used mainly for static content

Use Collection Caching

```
<%= render partial: 'products/product', collection: @products, cached: true %>
```


Q & A

Thanks For Listening

- ☐ Ynon Perek
- ☐ www.tocode.co.il
- ☐ ynon@ynonperek.com