Rails Caching

Rails 4 Edition

Let’s start with an HTTP request

```
HTTP/1.1 200 OK
Server: Apache
X-Rack-Cache: miss
ETag: "e6811cdbcedf972c5e8105a89f637d39-gzip"
Status: 200
Content-Type: text/html; charset=utf-8
Expires: Mon, 29 Apr 2013 21:44:55 GMT
Cache-Control: max-age=0, no-cache, no-store
Pragma: no-cache
Date: Mon, 29 Apr 2013 21:44:55 GMT
```

HTTP Caching Headers

Cache-Control:

```
Cache-Control: public,max-age=300
```

```
- Cache the resource for 5 minutes
```

```
Cache-Control: private
```

```
- Don’t cache the resource
```

HTTP Caching Headers

Expires:

```
Expires: Fri, 16 Jul 2014 21:44:55 GMT
```

HTTP Caching Headers

Etag:

```
ETag: "e6811cdbcedf972c5e8105a89f637d39-gzip"
```

- a digest of the resources contents (e.g. MD5 hash) computed by the server

The Rails side

Page Caching

- moved to gem

Action Caching

- moved to gem

Fragment Caching

- still in rails core

Rails App

Scaffold - Server (name, ip)

rake db:setup

development.rb changes

ServersController index call (through Postman) caches list of all the servers

Cache Invalidation

Automatic cache invalidation

helper - **cache_key_for_servers**

- invalidates cache based on updated_at

Cache Invalidation

Other strategies. Cache invalidation at the object level

```
<% Server.all.each do |s| %>
  <% cache(s) do %>
    <%= link_to s.name, server_url(s) %>
  <% end %>
<% end %>
```

s#cache_key returns “model name/id-updated_at”

e.g. `servers/12-20140109142612`

Redis

```
redis-cli

127.0.0.1:6379> keys *
1) "servers/all-23-20140717020612"
…

get "servers/all-23-20140717020612"
...output...
```

Conclusion

Utilize caching for scalability and performance.

Other types of caches :

- Web server (nginx)
- CDN (cloudfront)
- Client Side (localStorage)