

In this lecture, we will discuss...

✧ Cache Revalidation Headers

✧ Custom Logic

- `If-None-Match`
- `If-Modified-Since`



Cache Revalidation Headers

✧ Validate if what we have is **current or stale** using conditional cache validation headers:

- `If-Not-Match` : (Etag)
- `If-Modified-Since` : (Timestamp)



Cache Revalidation Headers

- ✧ If the resource has **not** changed
 - enable server-side to do less processing because client does not need a new copy
 - report the resource has not changed to the client
 - enable client-side to do less processing because nothing has changed

ETag and Last-Modified: Storing

- ✧ Get a current copy of ETag and Last-Modified and store

```
> response=HTTParty.get("http://localhost:3000/movies/12345",  
                        headers:{"Accept"=>"application/json"})  
  
> response.response  
=> #<Net::HTTPOK 200 OK readbody=true>  
  
> etag=response.header["etag"]  
=> "\"dd7543eb8124a81a065c2d0629222e2c\""  
  
> last_modified=response.header["last-modified"]  
=> "Tue, 12 Jan 2016 17:16:35 GMT"
```



GET: Same Action

- ✧ GET request, same payload
- ✧ 200 ID - returned

```
> response=HTTParty.get("http://localhost:3000/movies/12345",  
                        headers:{"Accept"=>"application/json"})  
  
> response.response  
=> #<Net::HTTPOK 200 OK readbody=true>
```

With Headers

- ✧ Add `If-None-Match` header with the `ETag` or `If-Modified-Since` with the `Last-Modified` timestamp
- ✧ `304/NOT_MODIFIED` - returned

```
> response=HTTParty.get("http://localhost:3000/movies/12345",  
                        headers:{"Accept"=>"application/json",  
                                "If-None-Match"=>etag})  
  
> response.response  
=> #<Net::HTTPNotModified 304 Not Modified readbody=true>  
  
> response.body  
=> nil
```

Conditional Logic

✧ Fire only if the caller is not getting 304/NOT_MODIFIED

```
def show
  #   fresh_when(@movie)
  @movie.movie_accesses.create(:action=>"show")
  if stale? @movie
    @movie.movie_accesses.create(:action=>"show-stale")
    #do some additional, expensive work here
  end
end
```

✧ **Note:** `stale?` calls `fresh_when` under the covers

Conditional Logic

```
> response=HTTParty.get("http://localhost:3000/movies/12345",
                        headers:{"Accept"=>"application/json",
                                "If-Modified-Since"=>last_modified})

> response.response
=> #<Net::HTTPNotModified 304 Not Modified readbody=true>

> pp Movie.find("12345").movie_accesses.pluck(:created_at, :action).to_a
[[2016-01-12 18:37:08 UTC, "show"]]
```

✧ `last_modified` value is being set to the **most recent change** in the collection

```
def index
  @movies = Movie.all
  fresh_when last_modified: @movies.max(:updated_at)
end
```



Conditional Logic

- ✧ Provide both `If-Modified-Since` and `If-None-Match` in the header
 - If either fires, our conditional logic will get triggered

```
> response=HTTParty.get("http://localhost:3000/movies/12345",
                        headers:{"Accept"=>"application/json",
                                "If-Modified-Since"=>last_modified,
                                "If-None-Match"=>"123"})

> response.response
=> #<Net::HTTPOK 200 OK readbody=true>
> pp Movie.find("12345").movie_accesses.pluck(:created_at, :action).to_a
[[2016-01-12 18:37:08 UTC, "show"],
 [2016-01-12 18:45:49 UTC, "show"],
 [2016-01-12 18:45:49 UTC, "show-stale"]]
```



Browser Test

- ✧ Chrome → Developer Tools → Network (Preserve Logs)
- ✧ `http://localhost:3000/movies/12345.json` (Status → 200/OK)
- ✧ Hit refresh (Status → 304/Not Modified)
 - `If-Modified-Since` and `If-None-Match` headers were supplied
- ✧ Click disable-cache at the top of the Network tab and hit refresh
 - The conditional headers are not sent to the Rails server and the full response is returned using a 200/OK



Cache Revalidation Headers

Demo

Summary

- ✧ Cache Validation Headers add value and improve performance when it comes to caching

What's Next?

- ✧ Cache-Controls

