Django: Under the Hood
November 3rd, 2016
Aymeric Augustin

debugging performance

Festival international des jardins, Chaumont, 8 mai 2015

👋 I’m Aymeric

django

Core Developer
since 2011

- Time zones
- Python 3
- Transactions
- App loading
- Jinja2

FRACTAL IDEAS

Freelancing
since 2015

Consulting
on Django,
big data, &
architecture

👋 I'm Aymeric

django

Core Developer
since 2011

- Time zones
- Python 3
- Transactions
- App loading
- Jinja2

OTHERWISE

Founder & CTO
since 2015

First collaborative
insurance broker
in France
https://otherwise.fr/ [FR]

response times

Festival international des jardins, Chaumont, 8 mai 2015

Perception of response time

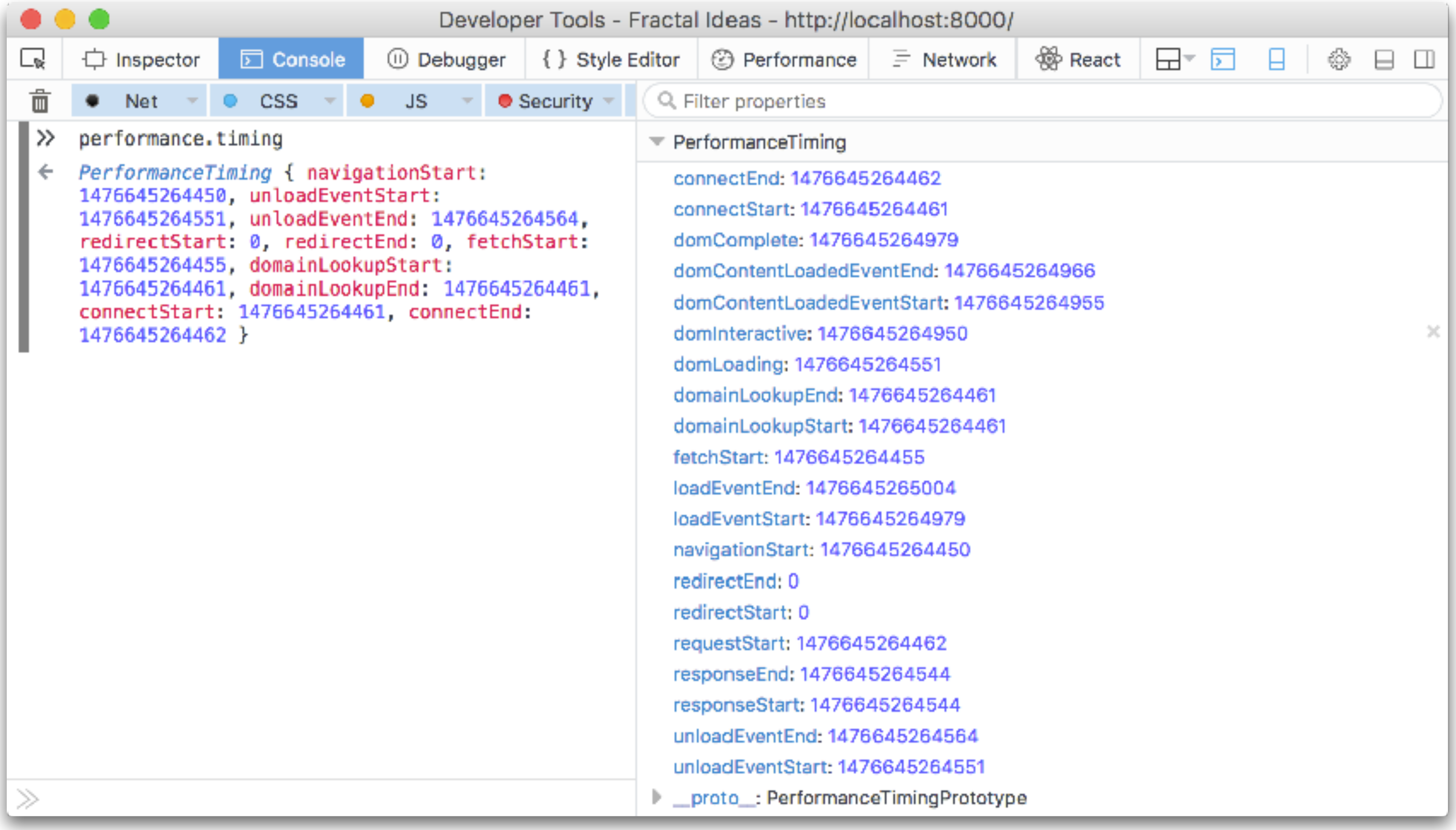
- **fast** = up to **0.1s** = reacting instantaneously
  - just display the result
- **normal** = up to **1s** = not interrupting the user's flow of thought
  - user will notice the delay but no feedback is necessary
- **slow** = up to **10s** = keeping the user's attention focused
  - feedback is important, especially if response time is unpredictable

Source: https://www.nngroup.com/articles/website-response-times/

measuring page load time

Festival international des jardins, Chaumont, 8 mai 2015

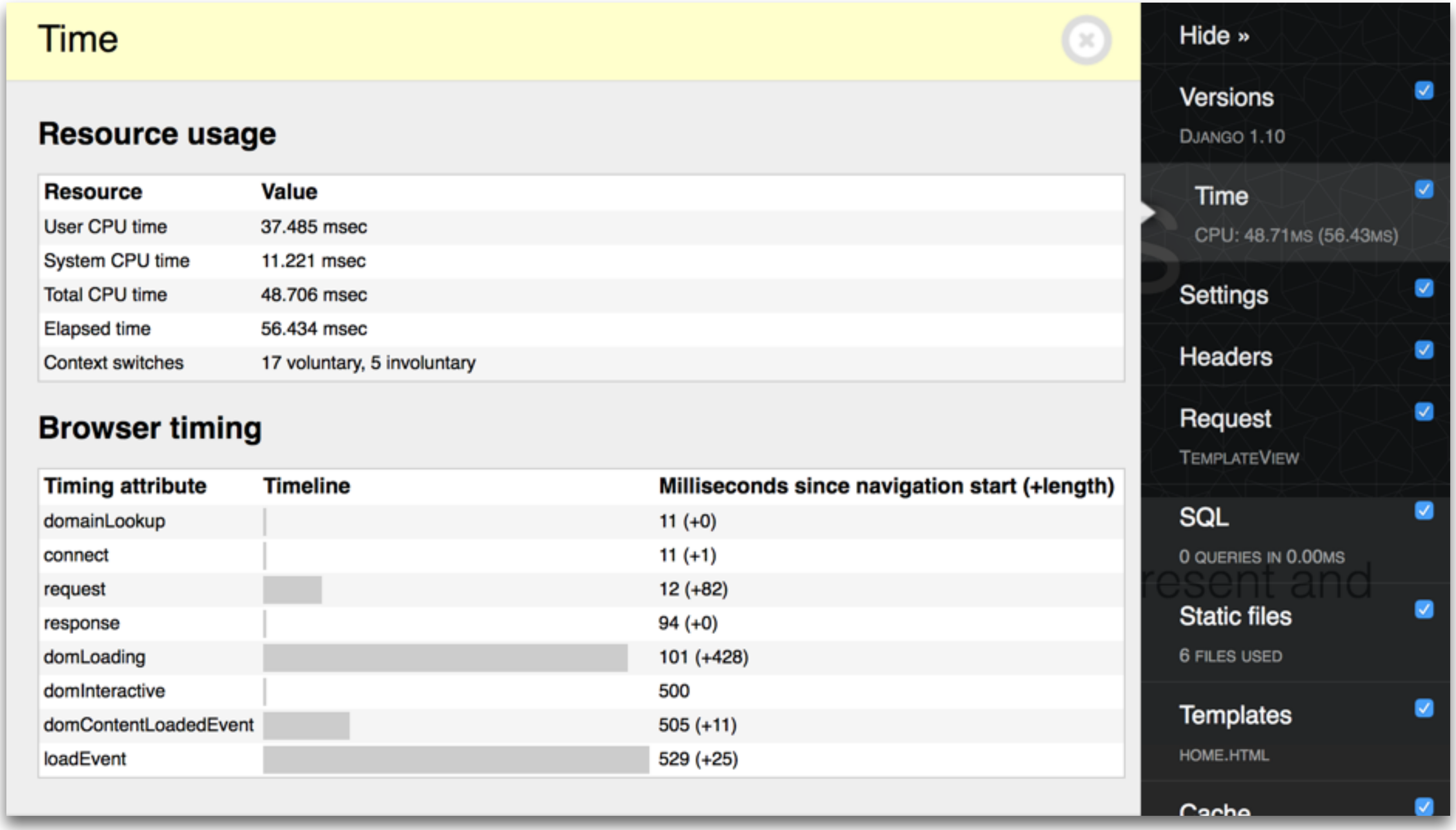
performance.timing



```
>> performance.timing
← PerformanceTiming { navigationStart:
  1476645264450, unloadEventStart:
  1476645264551, unloadEventEnd: 1476645264564,
  redirectStart: 0, redirectEnd: 0, fetchStart:
  1476645264455, domainLookupStart:
  1476645264461, domainLookupEnd: 1476645264461,
  connectStart: 1476645264461, connectEnd:
  1476645264462 }
```

```
▼ PerformanceTiming
    connectEnd: 1476645264462
    connectStart: 1476645264461
    domComplete: 1476645264979
    domContentLoadedEventEnd: 1476645264966
    domContentLoadedEventStart: 1476645264955
    domInteractive: 1476645264950
    domLoading: 1476645264551
    domainLookupEnd: 1476645264461
    domainLookupStart: 1476645264461
    fetchStart: 1476645264455
    loadEventEnd: 1476645265004
    loadEventStart: 1476645264979
    navigationStart: 1476645264450
    redirectEnd: 0
    redirectStart: 0
    requestStart: 1476645264462
    responseEnd: 1476645264544
    responseStart: 1476645264544
    unloadEventEnd: 1476645264564
    unloadEventStart: 1476645264551
  ▶ __proto__: PerformanceTimingPrototype
```

Time panel



Time

Resource usage

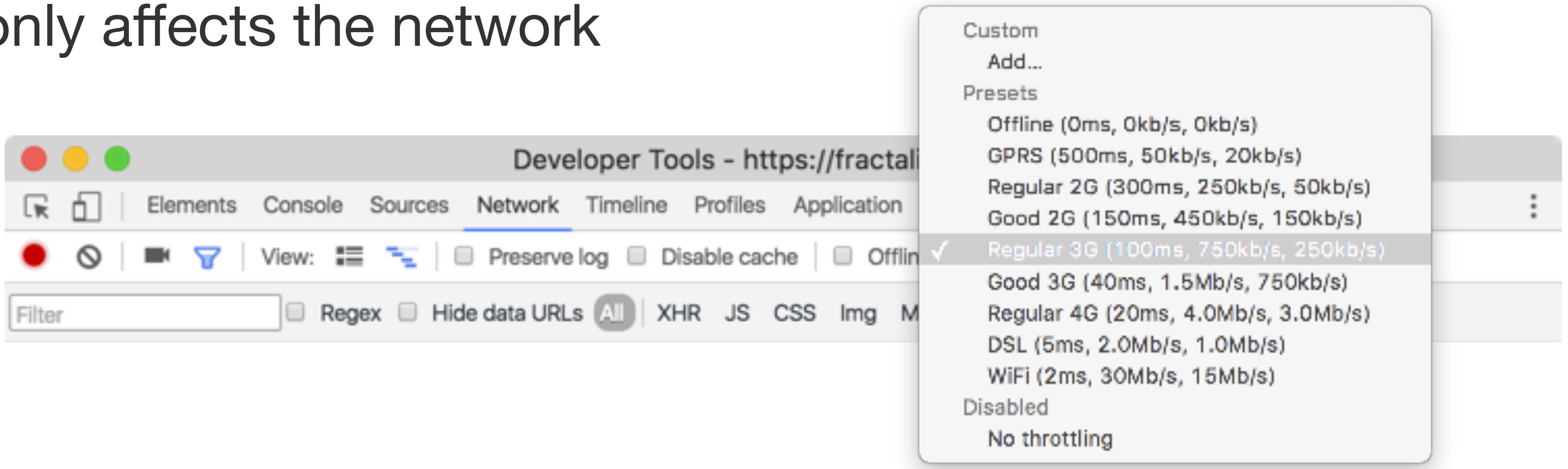
Resource	Value
User CPU time	37.485 msec
System CPU time	11.221 msec
Total CPU time	48.706 msec
Elapsed time	56.434 msec
Context switches	17 voluntary, 5 involuntary

Browser timing

Timing attribute	Timeline	Milliseconds since navigation start (+length)
domainLookup		11 (+0)
connect		11 (+1)
request		12 (+82)
response		94 (+0)
domLoading		101 (+428)
domInteractive		500
domContentLoadedEvent		505 (+11)
loadEvent		529 (+25)

In production

- Google Chrome Developer Tools
    - only affects the network



Developer Tools - https://fractal

Elements Console Sources Network Timeline Profiles Application

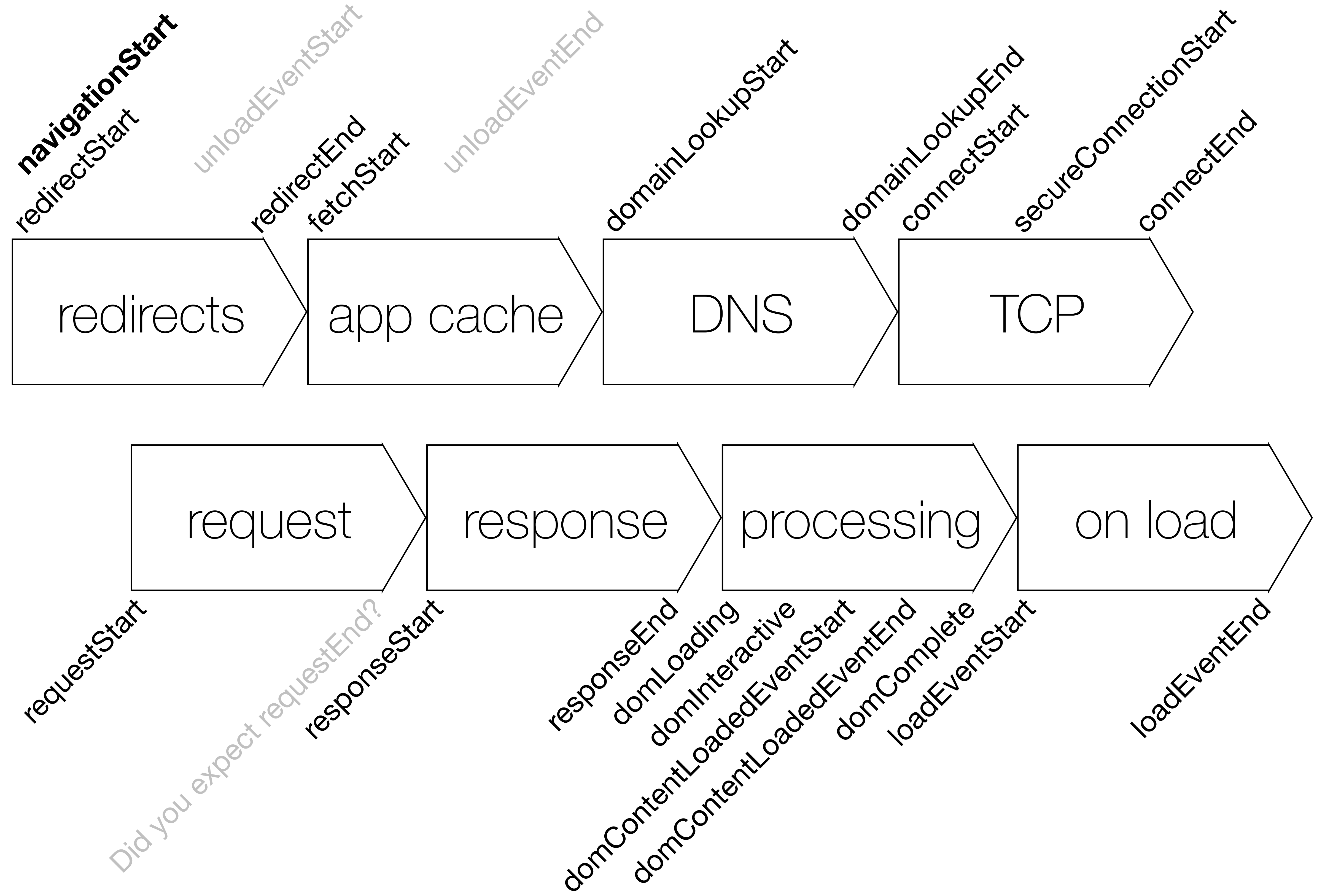
View: Preserve log Disable cache Offlin

Filter Regex Hide data URLs All XHR JS CSS Img M

Custom
Add...
Presets
Offline (0ms, 0kb/s, 0kb/s)
GPRS (500ms, 50kb/s, 20kb/s)
Regular 2G (300ms, 250kb/s, 50kb/s)
Good 2G (150ms, 450kb/s, 150kb/s)
✓ Regular 3G (100ms, 750kb/s, 250kb/s)
Good 3G (40ms, 1.5Mb/s, 750kb/s)
Regular 4G (20ms, 4.0Mb/s, 3.0Mb/s)
DSL (5ms, 2.0Mb/s, 1.0Mb/s)
WiFi (2ms, 30Mb/s, 15Mb/s)
Disabled
No throttling

- Google Analytics Site Speed
    - sampled on 1% of users by default
- Application Performance Monitoring solutions

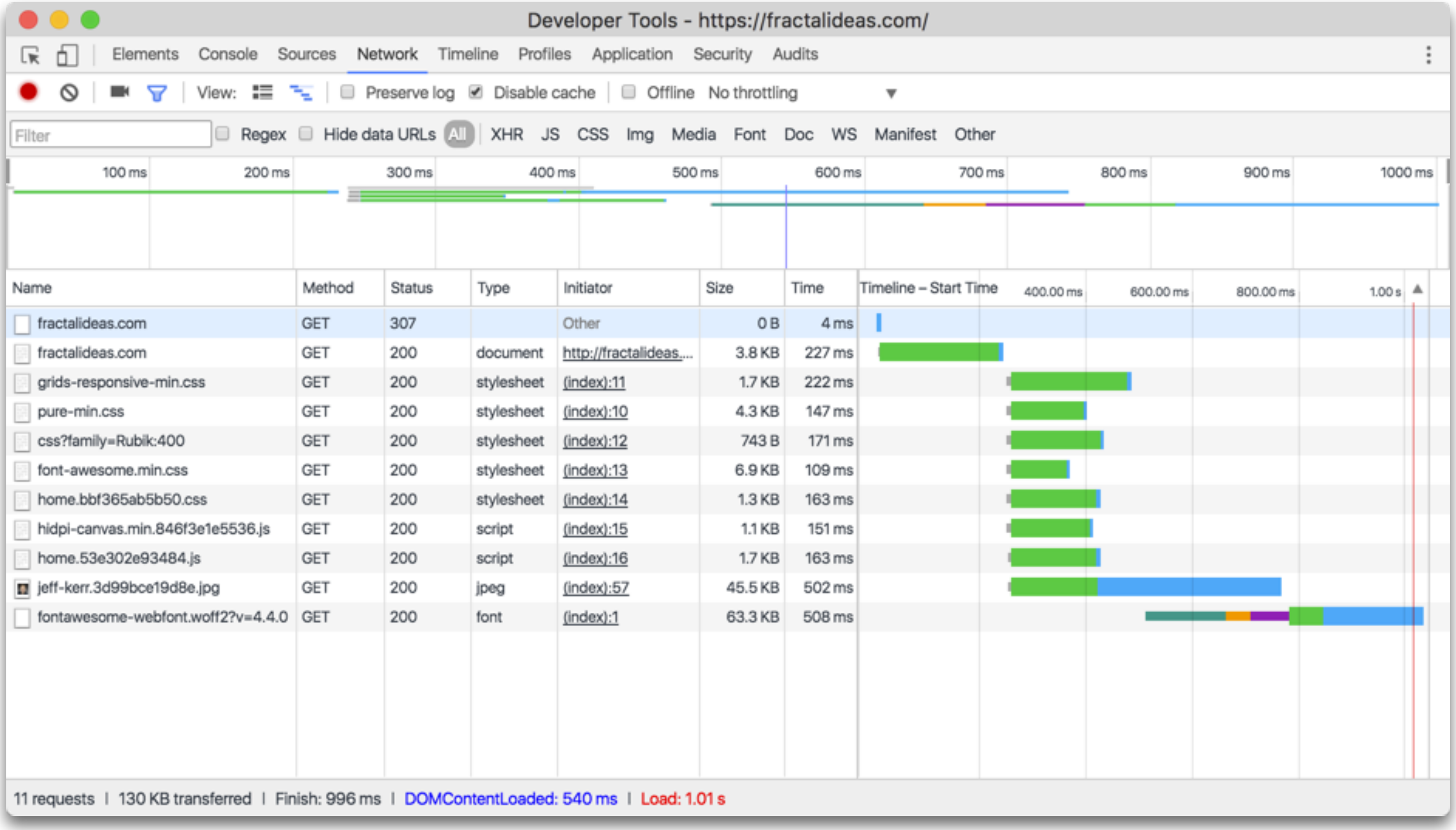
Performance timeline



The backend makes ~15%

of the total page load time.

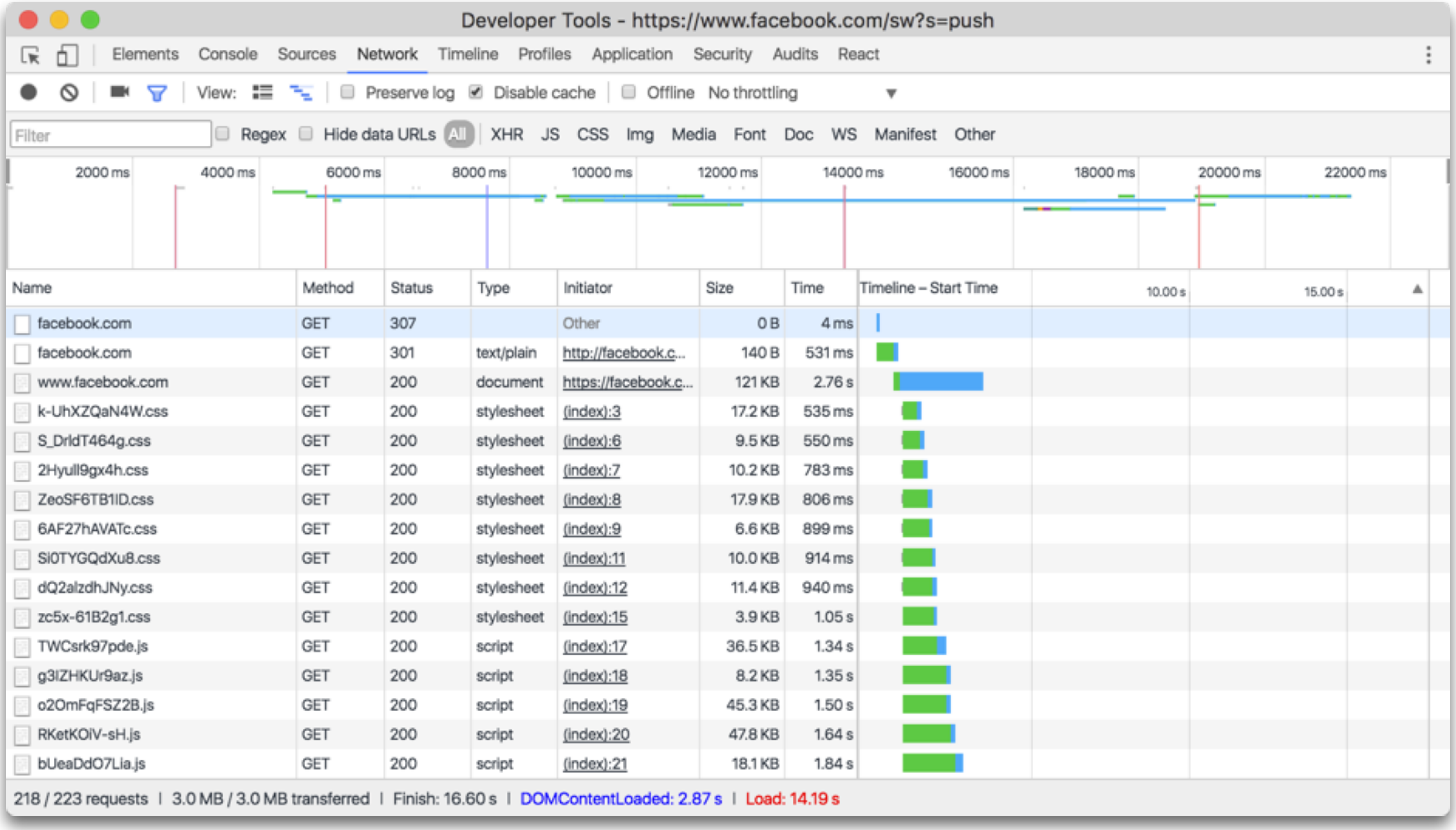
Fractal Ideas - 0.23s / 1.0s = 23%


Developer Tools - https://fractalideas.com/
Elements Console Sources Network Timeline Profiles Application Security Audits
View: Preserve log Disable cache Offline No throttling
Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
100 ms 200 ms 300 ms 400 ms 500 ms 600 ms 700 ms 800 ms 900 ms 1000 ms
Timeline – Start Time 400.00 ms 600.00 ms 800.00 ms 1.00 s

Name	Method	Status	Type	Initiator	Size	Time
fractalideas.com	GET	307		Other	0 B	4 ms
fractalideas.com	GET	200	document	http://fractalideas....	3.8 KB	227 ms
grids-responsive-min.css	GET	200	stylesheet	(index):11	1.7 KB	222 ms
pure-min.css	GET	200	stylesheet	(index):10	4.3 KB	147 ms
css?family=Rubik:400	GET	200	stylesheet	(index):12	743 B	171 ms
font-awesome.min.css	GET	200	stylesheet	(index):13	6.9 KB	109 ms
home.bbf365ab5b50.css	GET	200	stylesheet	(index):14	1.3 KB	163 ms
hidpi-canvas.min.846f3e1e5536.js	GET	200	script	(index):15	1.1 KB	151 ms
home.53e302e93484.js	GET	200	script	(index):16	1.7 KB	163 ms
jeff-kerr.3d99bce19d8e.jpg	GET	200	jpeg	(index):57	45.5 KB	502 ms
fontawesome-webfont.woff2?v=4.4.0	GET	200	font	(index):1	63.3 KB	508 ms

11 requests | 130 KB transferred | Finish: 996 ms | DOMContentLoaded: 540 ms | Load: 1.01 s

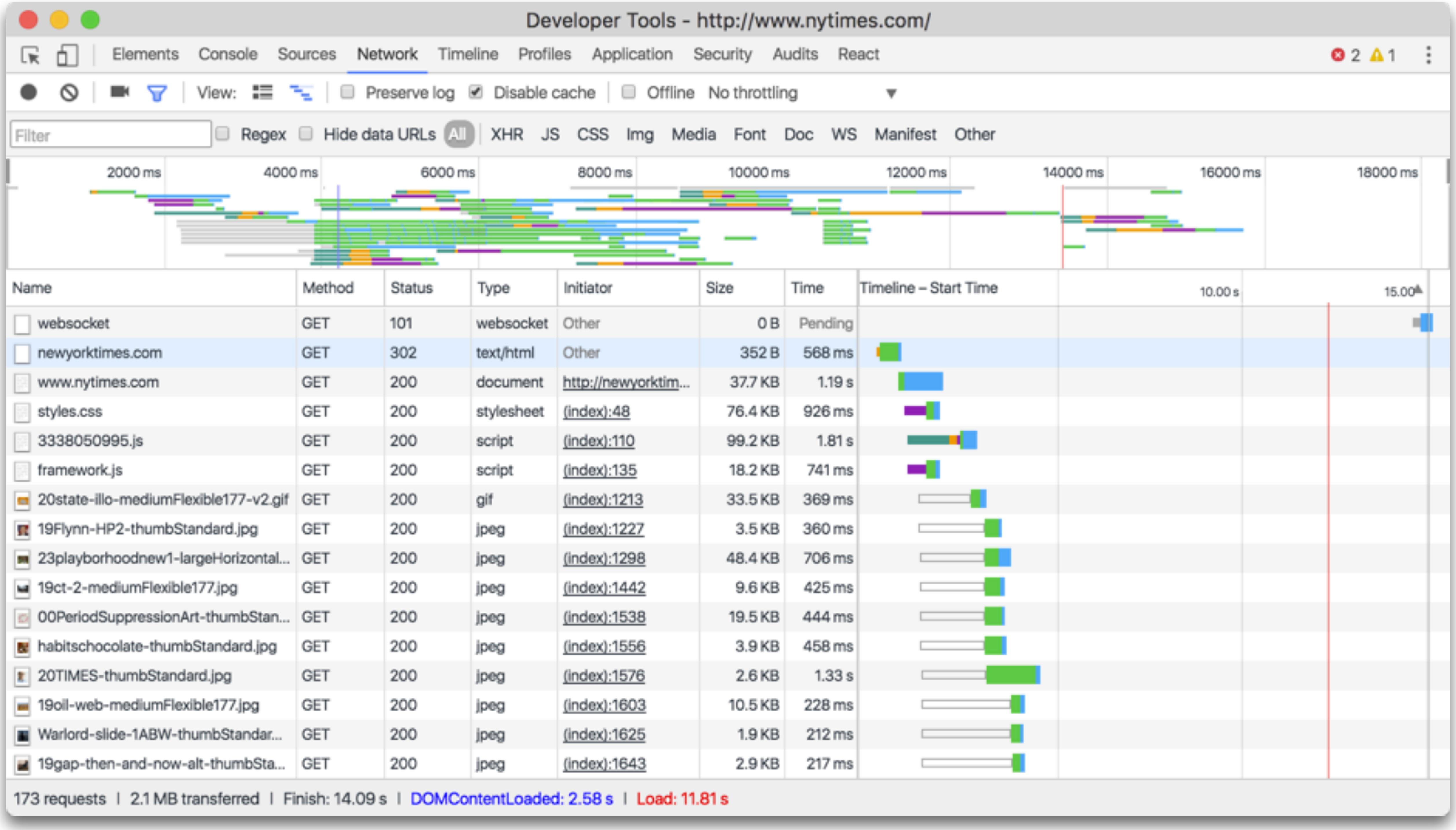
Facebook - 2.8s / 14.2s = 20%



Name	Method	Status	Type	Initiator	Size	Time	Timeline – Start Time
facebook.com	GET	307		Other	0 B	4 ms	
facebook.com	GET	301	text/plain	http://facebook.c...	140 B	531 ms	
www.facebook.com	GET	200	document	https://facebook.c...	121 KB	2.76 s	
k-UhXZQaN4W.css	GET	200	stylesheet	(index):3	17.2 KB	535 ms	
S_DrldT464g.css	GET	200	stylesheet	(index):6	9.5 KB	550 ms	
2Hyull9gx4h.css	GET	200	stylesheet	(index):7	10.2 KB	783 ms	
ZeoSF6TB1ID.css	GET	200	stylesheet	(index):8	17.9 KB	806 ms	
6AF27hAVATc.css	GET	200	stylesheet	(index):9	6.6 KB	899 ms	
SI0TYGQdXu8.css	GET	200	stylesheet	(index):11	10.0 KB	914 ms	
dQ2alzdhJNy.css	GET	200	stylesheet	(index):12	11.4 KB	940 ms	
zc5x-61B2g1.css	GET	200	stylesheet	(index):15	3.9 KB	1.05 s	
TWCsrk97pde.js	GET	200	script	(index):17	36.5 KB	1.34 s	
g3IZHKUr9az.js	GET	200	script	(index):18	8.2 KB	1.35 s	
o2OmFqFSZ2B.js	GET	200	script	(index):19	45.3 KB	1.50 s	
RKetKOiV-sH.js	GET	200	script	(index):20	47.8 KB	1.64 s	
bUeaDdO7Lia.js	GET	200	script	(index):21	18.1 KB	1.84 s	

218 / 223 requests | 3.0 MB / 3.0 MB transferred | Finish: 16.60 s | DOMContentLoaded: 2.87 s | Load: 14.19 s

New York Times - 1.2s / 11.8s = 10%



☐ Preserve log ☑ Disable cache ☐ Offline No throttling

Name	Method	Status	Type	Initiator	Size	Time	Timeline – Start Time
websocket	GET	101	websocket	Other	0 B	Pending	
newyorktimes.com	GET	302	text/html	Other	352 B	568 ms	
www.nytimes.com	GET	200	document	http://newyorktim...	37.7 KB	1.19 s	
styles.css	GET	200	stylesheet	(index):48	76.4 KB	926 ms	
3338050995.js	GET	200	script	(index):110	99.2 KB	1.81 s	
framework.js	GET	200	script	(index):135	18.2 KB	741 ms	
20state-illo-mediumFlexible177-v2.gif	GET	200	gif	(index):1213	33.5 KB	369 ms	
19Flynn-HP2-thumbStandard.jpg	GET	200	jpeg	(index):1227	3.5 KB	360 ms	
23playborhoodnew1-largeHorizontal...	GET	200	jpeg	(index):1298	48.4 KB	706 ms	
19ct-2-mediumFlexible177.jpg	GET	200	jpeg	(index):1442	9.6 KB	425 ms	
00PeriodSuppressionArt-thumbStan...	GET	200	jpeg	(index):1538	19.5 KB	444 ms	
habitschocolate-thumbStandard.jpg	GET	200	jpeg	(index):1556	3.9 KB	458 ms	
20TIMES-thumbStandard.jpg	GET	200	jpeg	(index):1576	2.6 KB	1.33 s	
19oil-web-mediumFlexible177.jpg	GET	200	jpeg	(index):1603	10.5 KB	228 ms	
Warlord-slide-1ABW-thumbStandar...	GET	200	jpeg	(index):1625	1.9 KB	212 ms	
19gap-then-and-now-alt-thumbSta...	GET	200	jpeg	(index):1643	2.9 KB	217 ms	

173 requests | 2.1 MB transferred | Finish: 14.09 s | DOMContentLoaded: 2.58 s | Load: 11.81 s

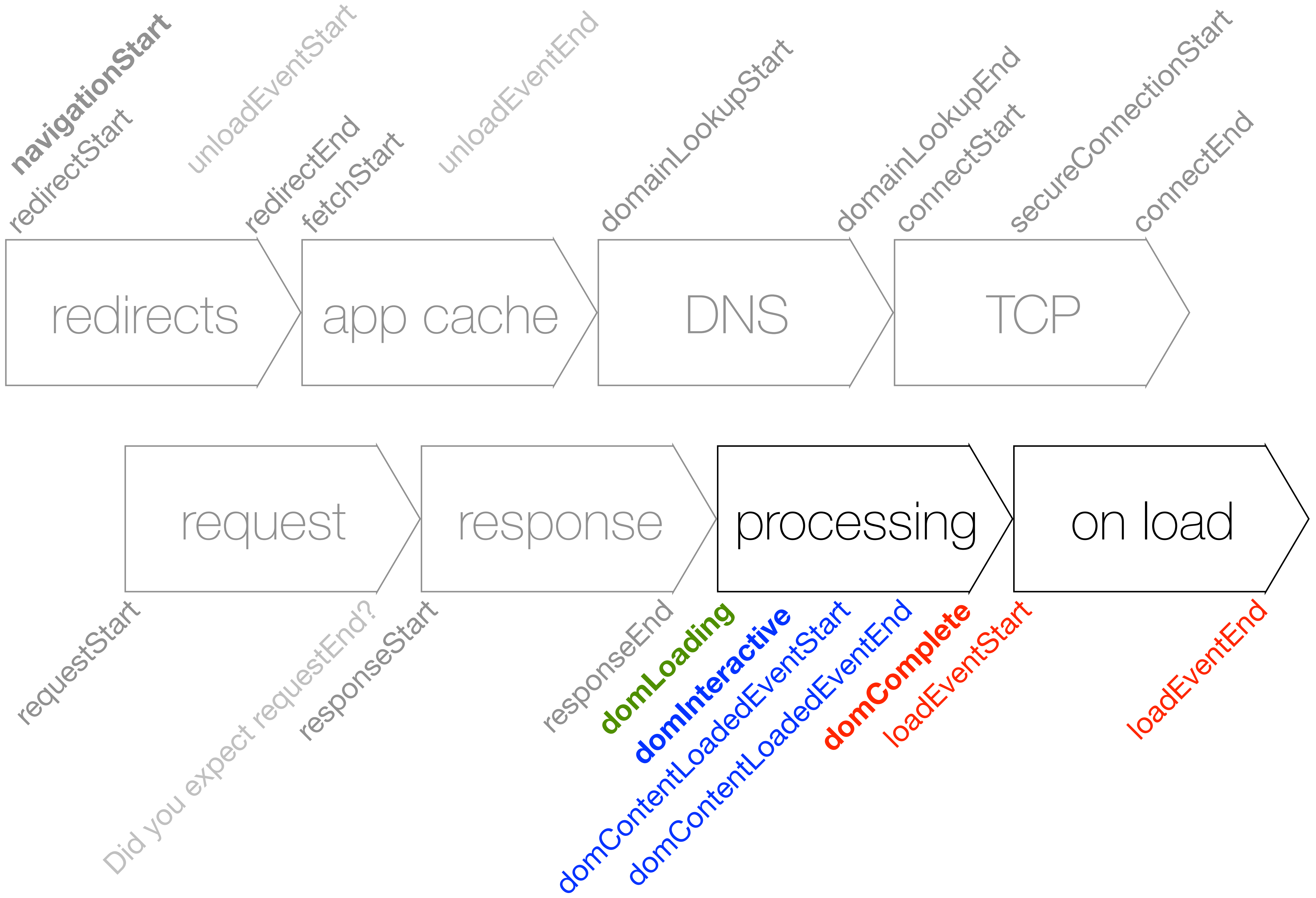
HTTP/1.1 is bad at fetching many resources

- Client-side
  - DNS pre-fetch
  - TCP pre-connect
  - keep-alive & pipelining
  - parallel connections
  - caching

- Server-side
  - concatenation
  - spriting
  - inlining
  - domain sharding
  - allow caching

Let's talk about
the frontend.

Performance timeline



`document.readyState` & page load events

`'loading'`

- **domLoading**
- No event (no JS yet!)
- Parse HTML & build DOM
- Download & run sync JS

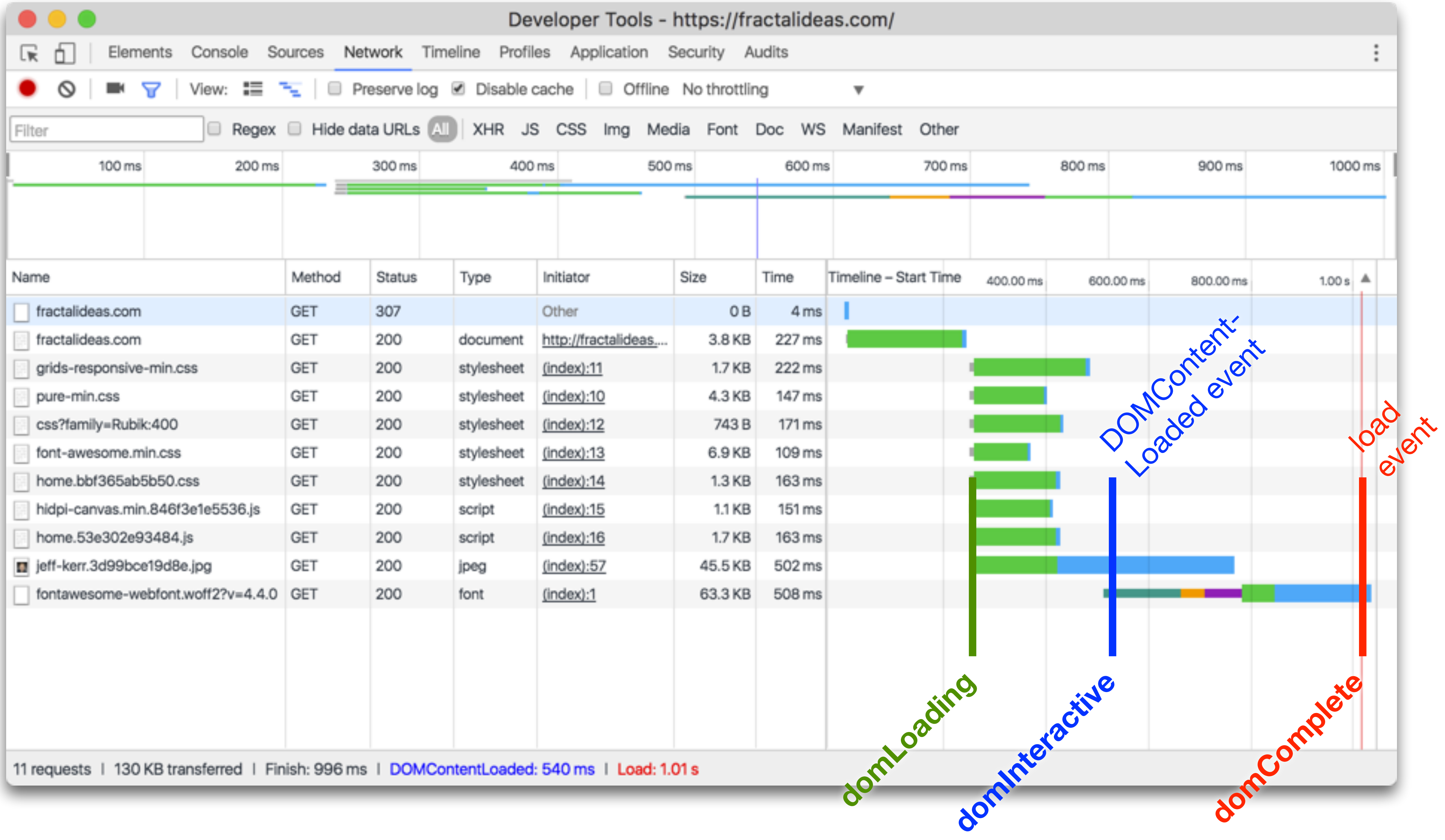
`'interactive'`

- **domInteractive**
- DOMContent-Loaded event
- Download CSS & images
- Parse CSS & build CSSOM

`'complete'`

- **domComplete**
- load event
- Page is fully loaded

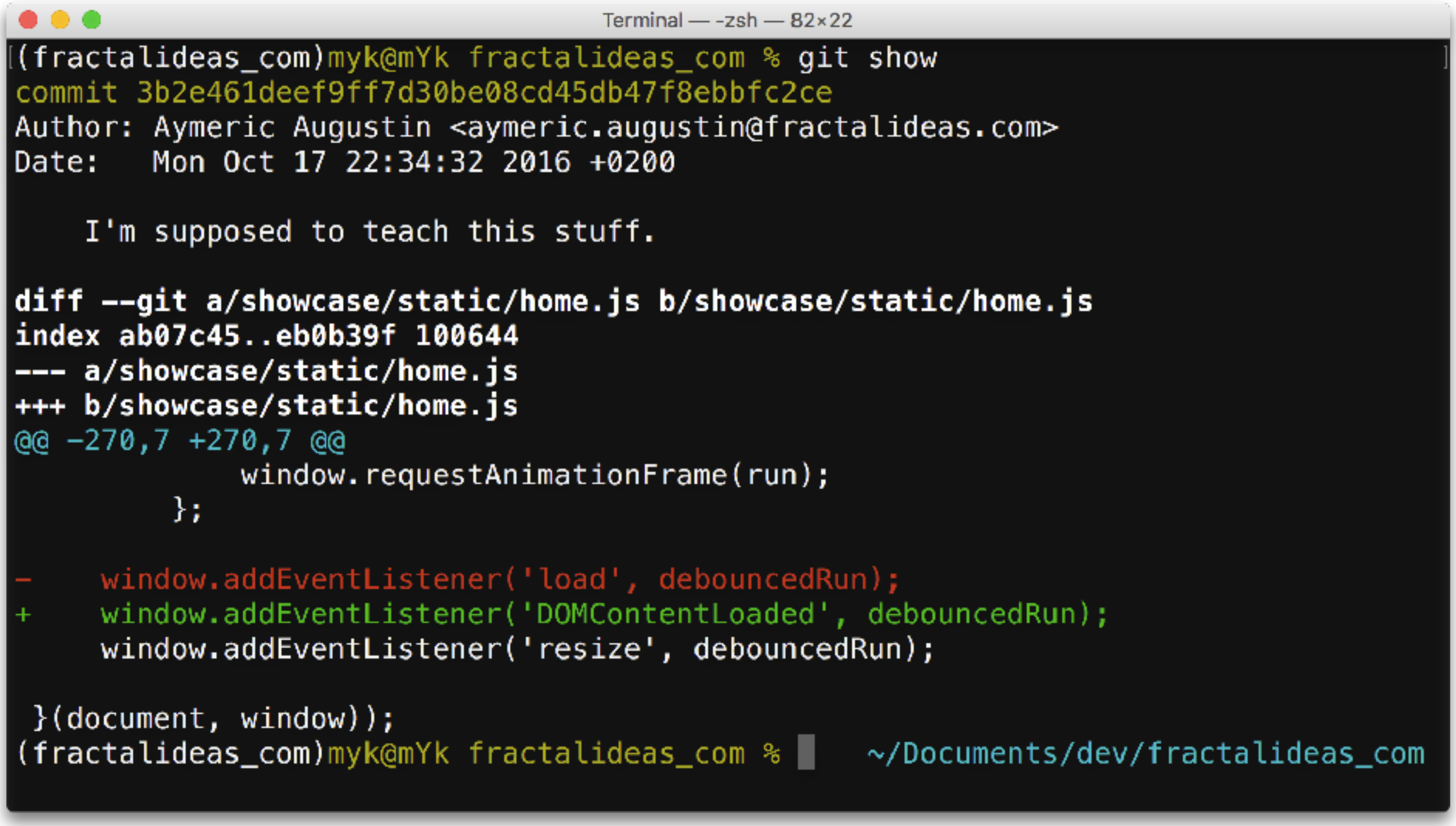
`document.readyState` & page load events



Name	Method	Status	Type	Initiator	Size	Time
fractalideas.com	GET	307		Other	0 B	4 ms
fractalideas.com	GET	200	document	http://fractalideas....	3.8 KB	227 ms
grids-responsive-min.css	GET	200	stylesheet	(index):11	1.7 KB	222 ms
pure-min.css	GET	200	stylesheet	(index):10	4.3 KB	147 ms
css?family=Rubik:400	GET	200	stylesheet	(index):12	743 B	171 ms
font-awesome.min.css	GET	200	stylesheet	(index):13	6.9 KB	109 ms
home.bbf365ab5b50.css	GET	200	stylesheet	(index):14	1.3 KB	163 ms
hidpi-canvas.min.846f3e1e5536.js	GET	200	script	(index):15	1.1 KB	151 ms
home.53e302e93484.js	GET	200	script	(index):16	1.7 KB	163 ms
jeff-kerr.3d99bce19d8e.jpg	GET	200	jpeg	(index):57	45.5 KB	502 ms
fontawesome-webfont.woff2?v=4.4.0	GET	200	font	(index):1	63.3 KB	508 ms

11 requests | 130 KB transferred | Finish: 996 ms | DOMContentLoaded: 540 ms | Load: 1.01 s

Listen to DOMContentLoaded — not load

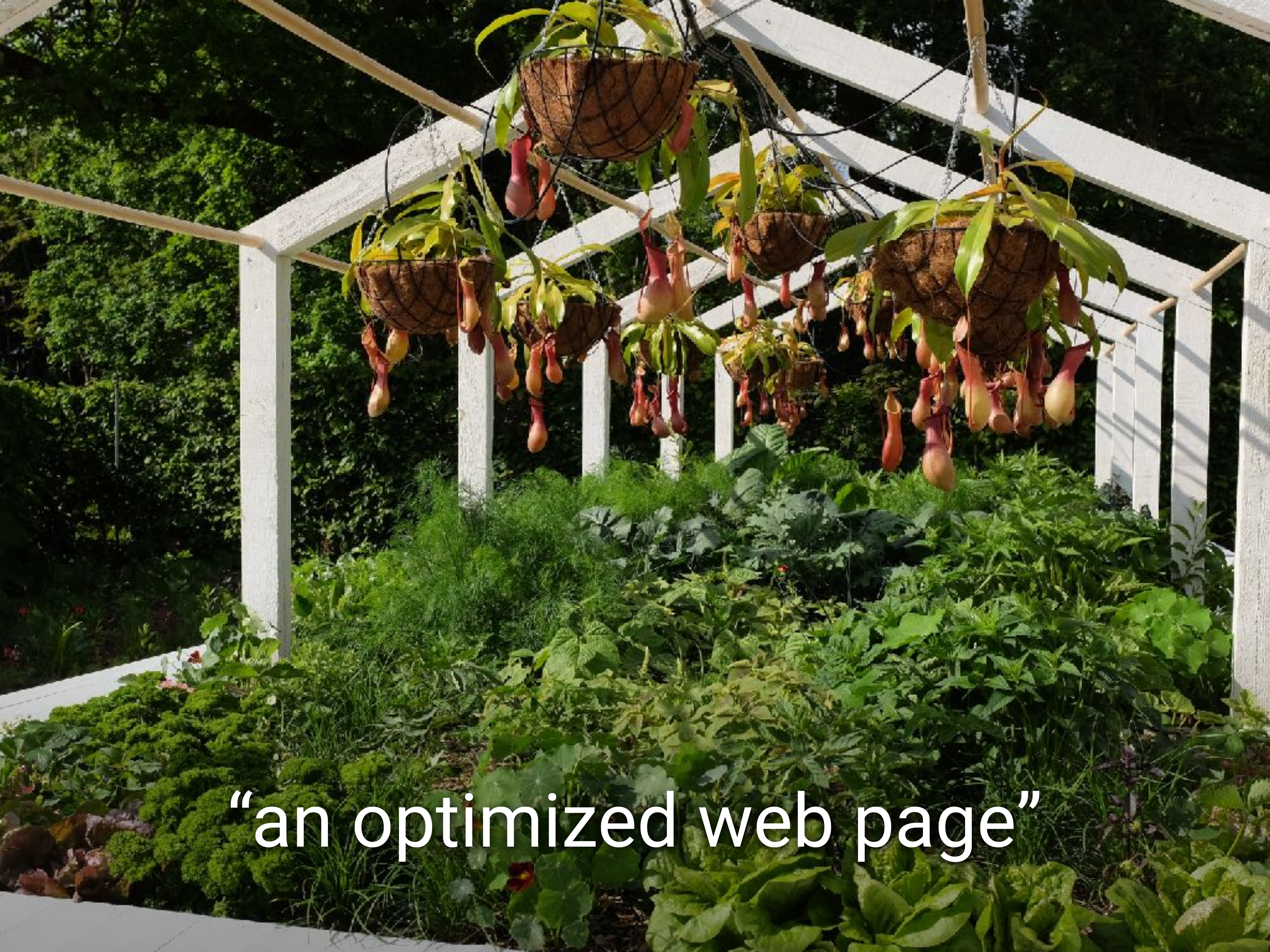


```
(fractalideas_com)myk@mYk fractalideas_com % git show
commit 3b2e461deef9ff7d30be08cd45db47f8ebbfc2ce
Author: Aymeric Augustin <aymeric.augustin@fractalideas.com>
Date:   Mon Oct 17 22:34:32 2016 +0200

    I'm supposed to teach this stuff.

diff --git a/showcase/static/home.js b/showcase/static/home.js
index ab07c45..eb0b39f 100644
--- a/showcase/static/home.js
+++ b/showcase/static/home.js
@@ -270,7 +270,7 @@
             window.requestAnimationFrame(run);
         };

-    window.addEventListener('load', debouncedRun);
+    window.addEventListener('DOMContentLoaded', debouncedRun);
     window.addEventListener('resize', debouncedRun);

 }(document, window));
(fractalideas_com)myk@mYk fractalideas_com %    ~/Documents/dev/fractalideas_com
```

client-side: loading pages

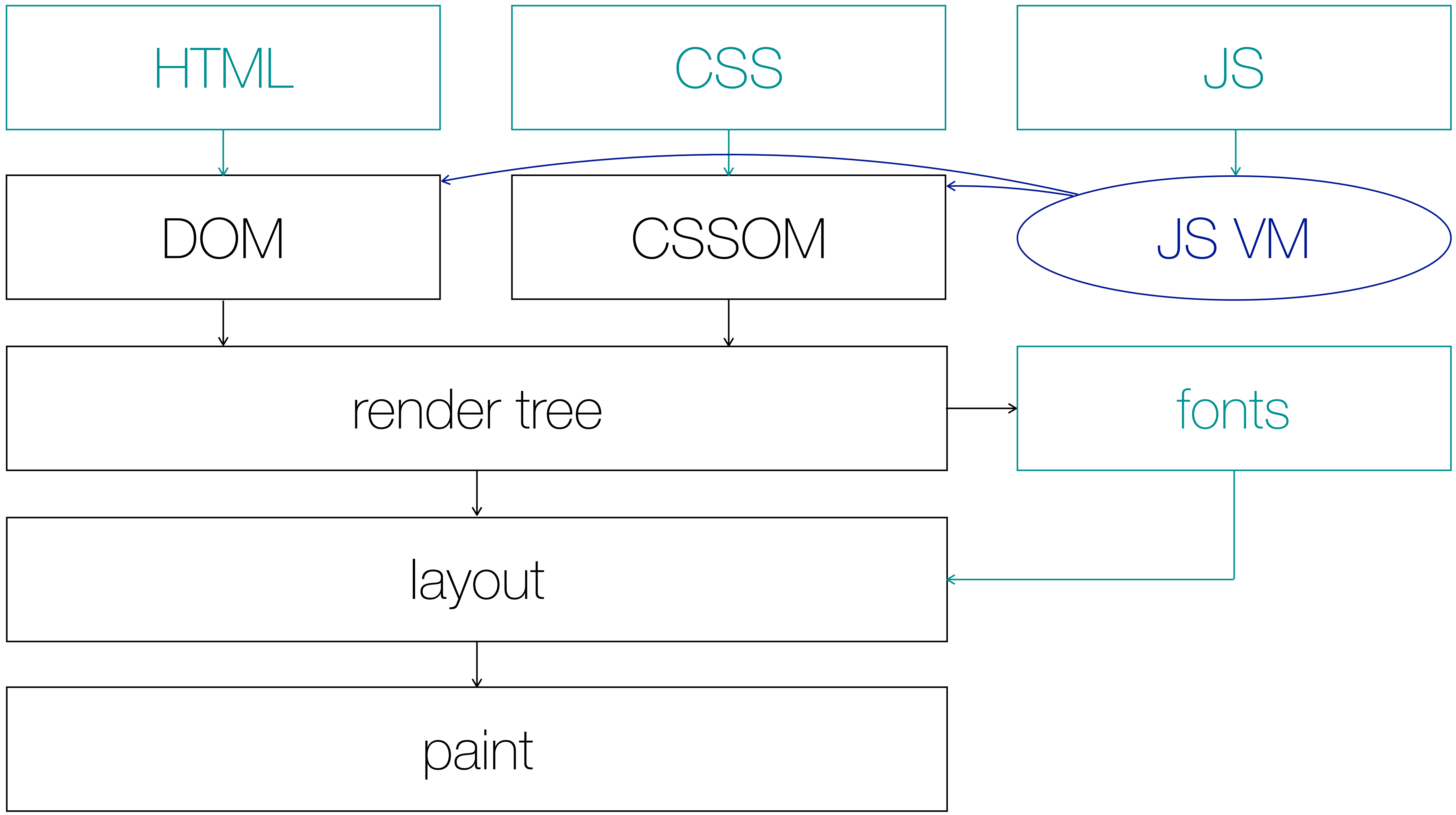
Festival international des jardins, Chaumont, 8 mai 2015

CSS

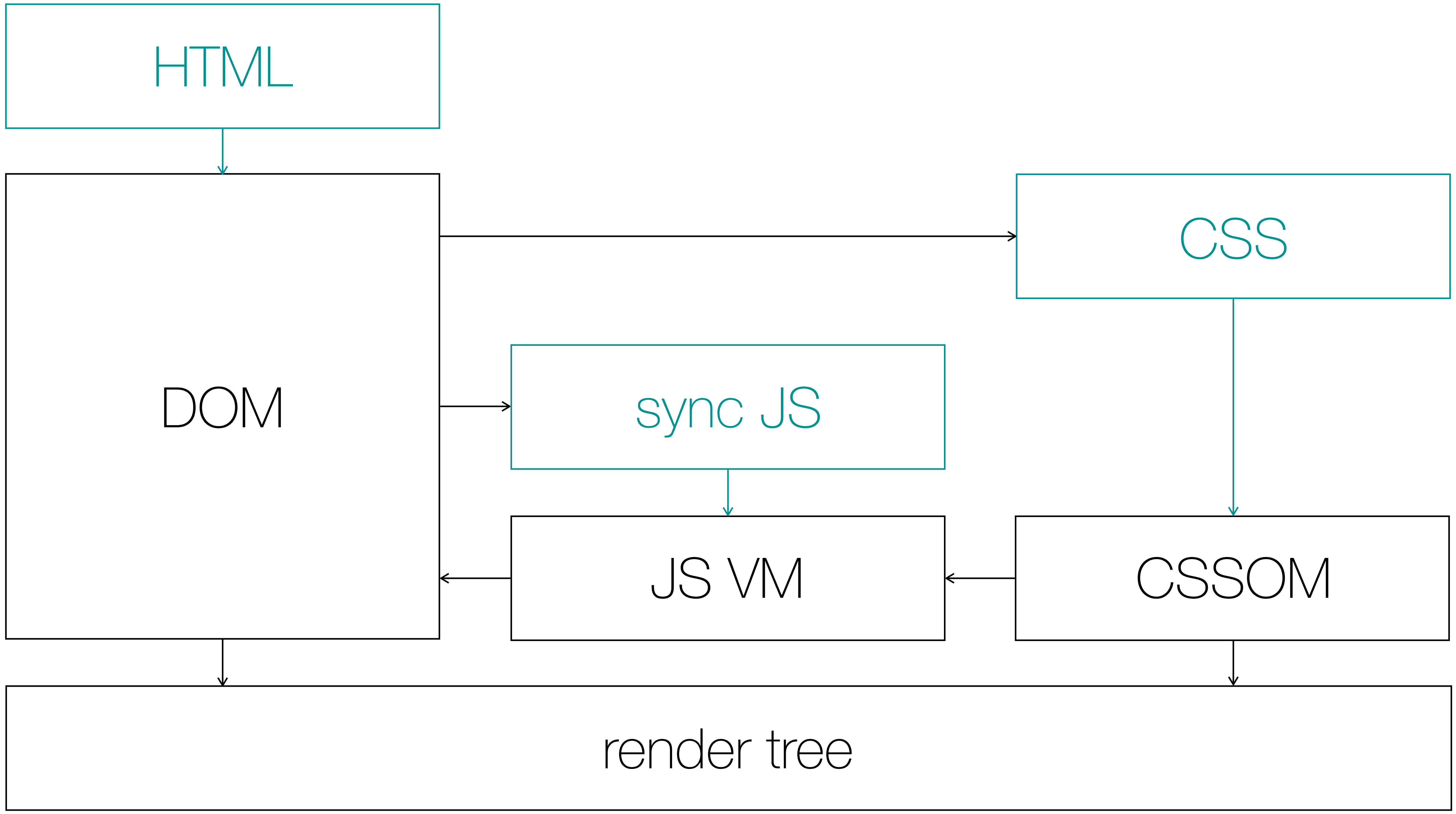


“an optimized web page”

Rendering pipeline



Critical path - visual edition



Critical path - text edition

- Rendering a page requires a DOM and a CSSOM
  - Download and parse HTML
  - Download and parse CSS
- Building the DOM blocks on sync JS
  - Download and execute sync JS
- Executing JS blocks on the CSSOM
  - Wait until CSS is parsed to execute sync JS

Browsers optimize heavily page load time

- Parse HTML incrementally
- Paint while waiting for sync JS
  - After CSS is available
- Paint while waiting for web fonts
  - With a default font — browser dependent
- Preload scanner

Guidelines

1. Optimize HTML load time
2. Optimize CSS load time
    - Unblock first paint
    - Avoid blocking JS execution
3. Avoid sync JS, including inline JS, or put it at the bottom
    - Avoid blocking DOM construction
    - Trigger DOMContentLoaded as early as possible

client-side: aysnc scripts

Festival international des jardins, Chaumont, 8 mai 2015

“The JavaScript tracking snippet”

```
<!-- Google Analytics -->
<script>
(function(i,s,o,g,r,a,m)
{i['GoogleAnalyticsObject']=r;i[r]=i[r]||function()
{(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
Date();a=s.createElement(o),m=s.getElementsByTagName(o)
[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)})
(window,document,'script','https://www.google-
analytics.com/analytics.js','ga');

ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<!-- End Google Analytics -->
```

https://developers.google.com/analytics/devguides/collection/analyticsjs/

Script-injected scripts

```
window.GoogleAnalyticsObject = 'ga'
window.ga = window.ga || function () {
    window.ga.q = window.ga.q || []
    window.ga.q.push(arguments)
}
window.ga.l = 1 * new Date()

var script = document.createElement('script')
script.async = 1
script.src = \
    'https://www.google-analytics.com/analytics.js'

var otherScript = \
    document.getElementsByTagName('script')[0]
otherScript.parentNode.insertBefore(script, otherScript)
```

“Alternative async tracking snippet”

```
<!-- Google Analytics -->
<script>
window.ga=window.ga||function(){(ga.q=ga.q||
[]).push(arguments)};ga.l=+new Date;
ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<script async src='https://www.google-analytics.com/
analytics.js'></script>
<!-- End Google Analytics -->
```

https://developers.google.com/analytics/devguides/collection/analyticsjs/

Async scripts

```
window.ga = window.ga || function () {
    ga.q = ga.q || []
    ga.q.push(arguments)
}
ga.l = +new Date

// plus an async script:
<script async src='https://www.google-analytics.com/
analytics.js'></script>
```

New best practice?

```
<html>
    <head>
        <script> /* Async function queuing */ </script>
        <link rel="stylesheet" href="style.css">
        <script async src="critical.js"></script>
        <style> /* For above-the-fold content */ </style>
    </head>
    <body>
        <script async src="non-critical.js"></script>
    </body>
</html>
```

Credits: Ilya Grigorik

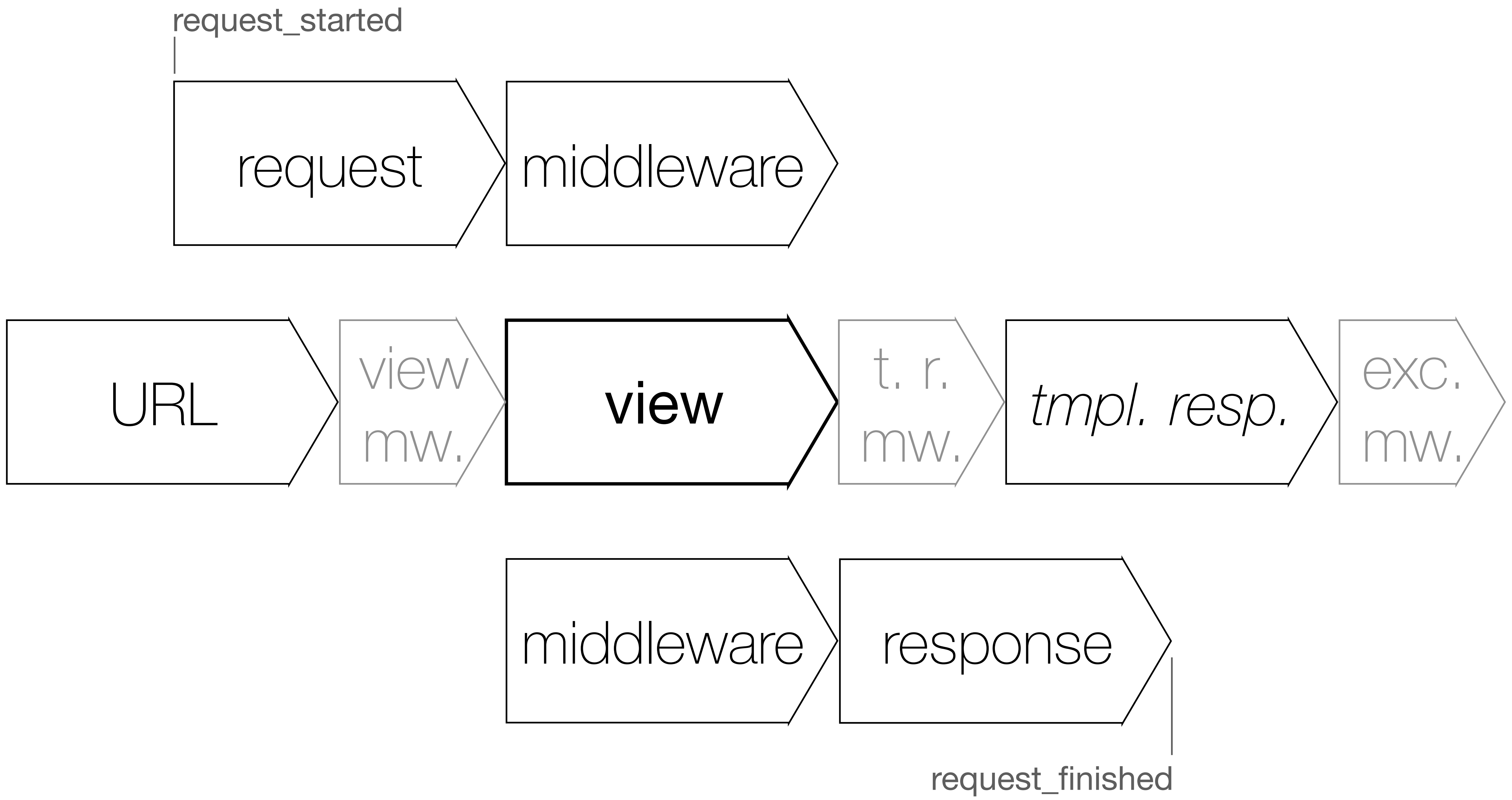
https://developers.google.com/web/fundamentals/performance/

https://www.igvita.com/slides/2012/webperf-crash-course.pdf
https://www.igvita.com/2014/05/20/script-injected-async-scripts-considered-harmful/
https://www.igvita.com/2015/04/10/fixing-the-blank-text-problem/

server-side: serving requests

Festival international des jardins, Chaumont, 8 mai 2015

Request-response cycle



Slow page?

SQL queries!

test setup

Festival international des jardins, Chaumont, 8 mai 2015

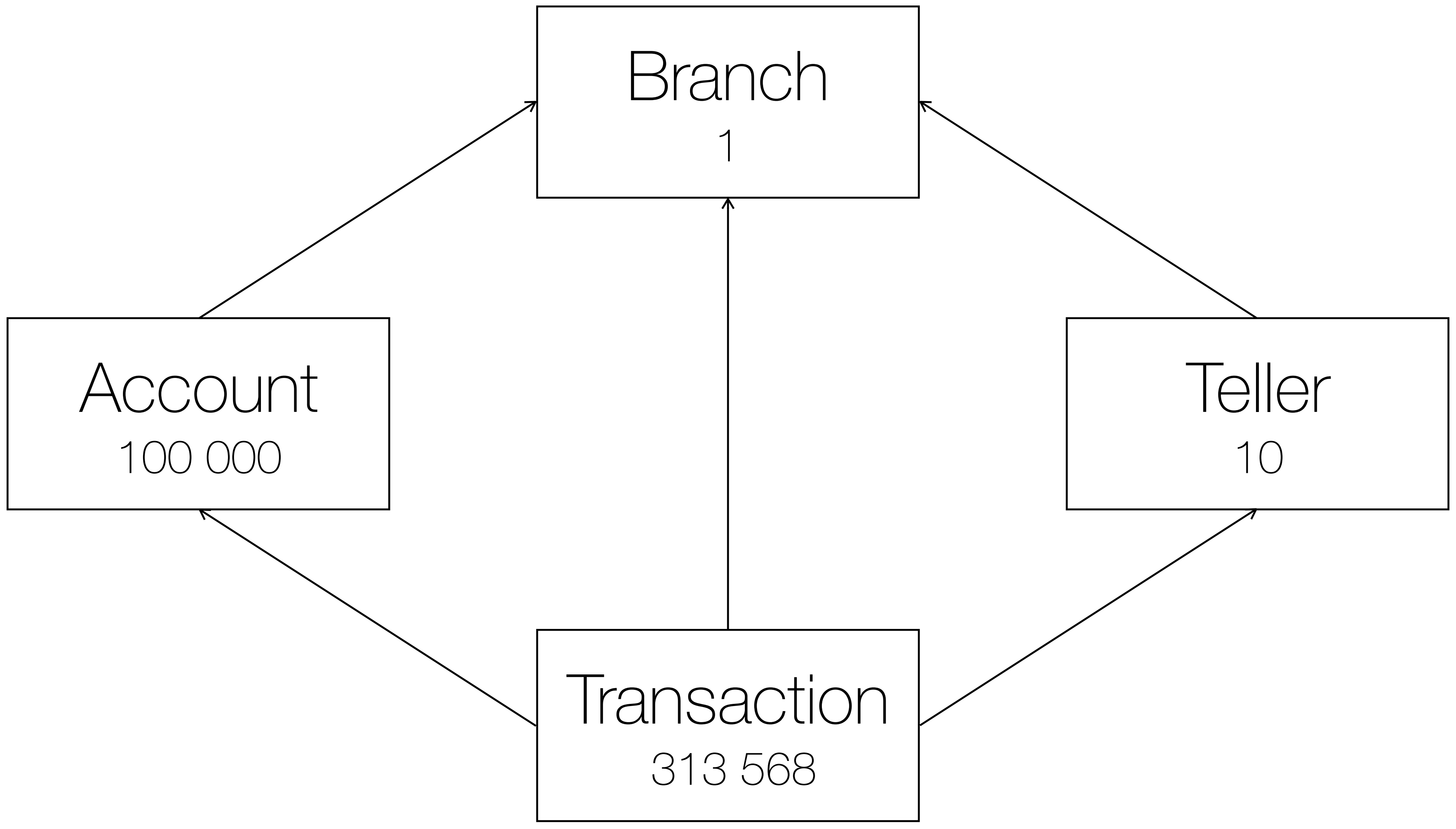
pgbench (1/2)

```
$ pgbench -i --foreign-keys duth16dp
creating tables...
100000 of 100000 tuples (100%) done (elapsed 0.11 s, remaining
0.00 s)
vacuum...
set primary keys...
set foreign keys...
done.
```

pgbench (2/2)

```
$ pgbench -c 30 -T 180 duth16dp
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 1
query mode: simple
number of clients: 30
number of threads: 1
duration: 180 s
number of transactions actually processed: 313568
latency average: 17.221 ms
tps = 1741.538334 (including connections establishing)
tps = 1741.569697 (excluding connections establishing)
```

Database structure



Testing environment

- https://github.com/aaugustin/duth16dp
- https://duth16dp.herokuapp.com/

- Hobby Dyno
- Hobby Basic Postgres

- `pg_dump -O duth16dp | heroku pg:psql`

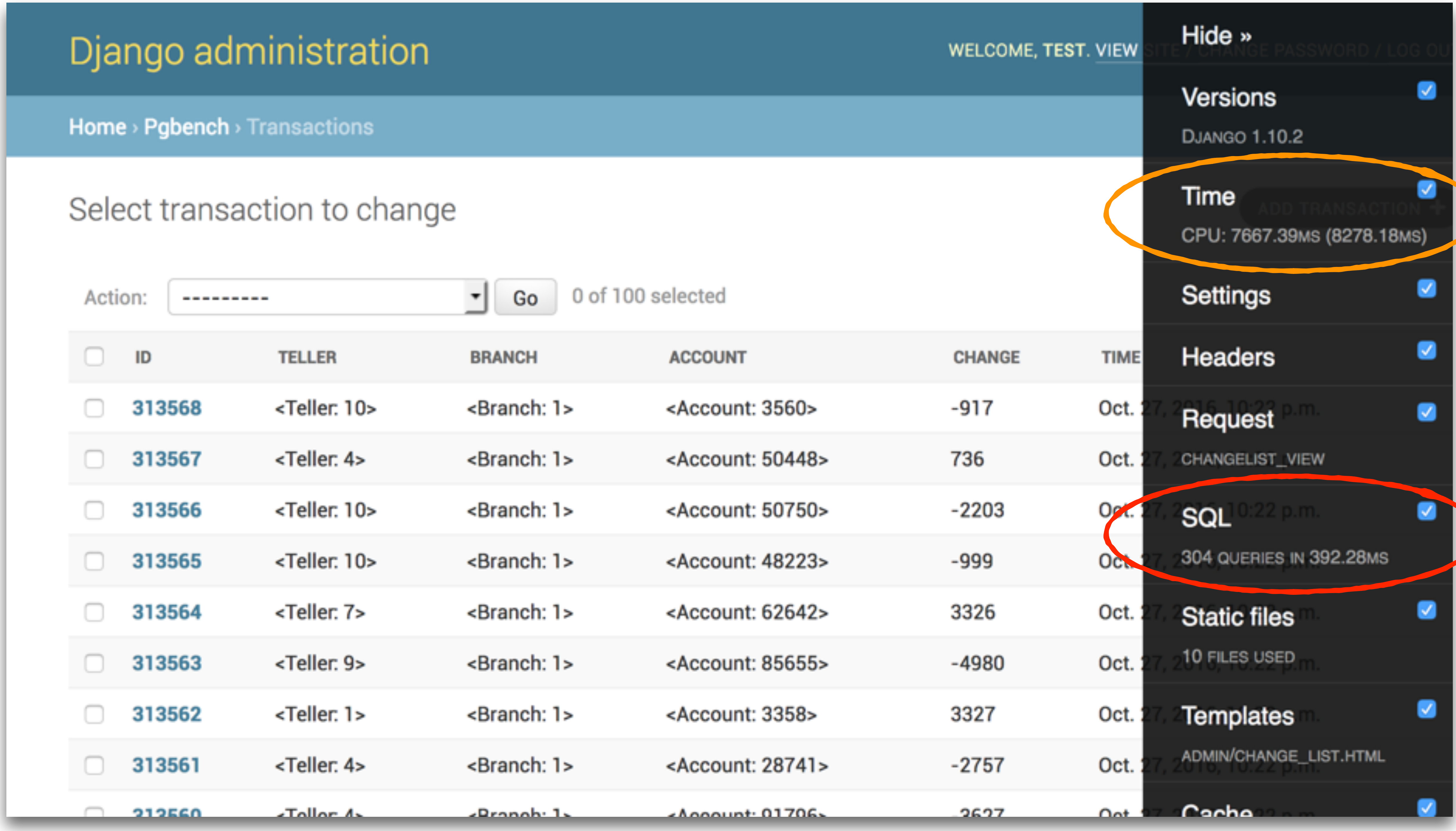
select & prefetch related instances

Festival international des jardins, Chaumont, 8 mai 2015

`select_related()`

https://docs.djangoproject.com/en/stable/ref/models/querysets/#django.db.models.query.QuerySet.select_related

The k * N +1 queries problem



Django administration

WELCOME, TEST. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Pgbench › Transactions

Select transaction to change

Action: ---------- Go 0 of 100 selected

	ID	TELLER	BRANCH	ACCOUNT	CHANGE	TIME
☐	313568	<Teller: 10>	<Branch: 1>	<Account: 3560>	-917	Oct.
☐	313567	<Teller: 4>	<Branch: 1>	<Account: 50448>	736	Oct.
☐	313566	<Teller: 10>	<Branch: 1>	<Account: 50750>	-2203	Oct.
☐	313565	<Teller: 10>	<Branch: 1>	<Account: 48223>	-999	Oct.
☐	313564	<Teller: 7>	<Branch: 1>	<Account: 62642>	3326	Oct.
☐	313563	<Teller: 9>	<Branch: 1>	<Account: 85655>	-4980	Oct.
☐	313562	<Teller: 1>	<Branch: 1>	<Account: 3358>	3327	Oct.
☐	313561	<Teller: 4>	<Branch: 1>	<Account: 28741>	-2757	Oct.
☐	313560	<Teller: 4>	<Branch: 1>	<Account: 91706>	-3627	Oct.

Hide »

Versions ✓
Django 1.10.2

Time ✓
CPU: 7667.39ms (8278.18ms)

Settings ✓

Headers ✓

Request ✓
CHANGELIST_VIEW

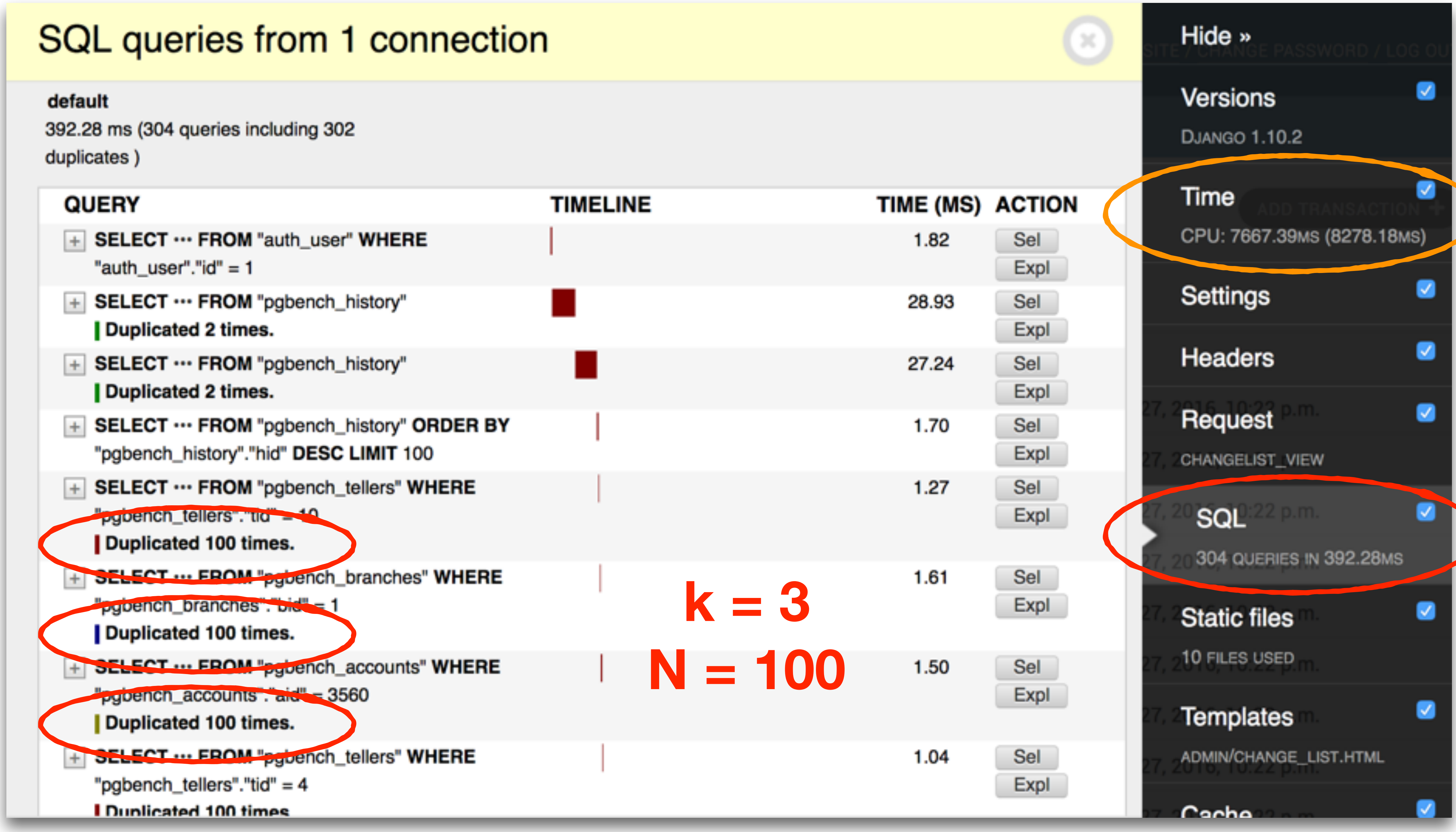
SQL ✓
304 queries in 392.28ms

Static files ✓
10 files used

Templates ✓
ADMIN/CHANGE_LIST.HTML

Cache ✓

The k * N +1 queries problem



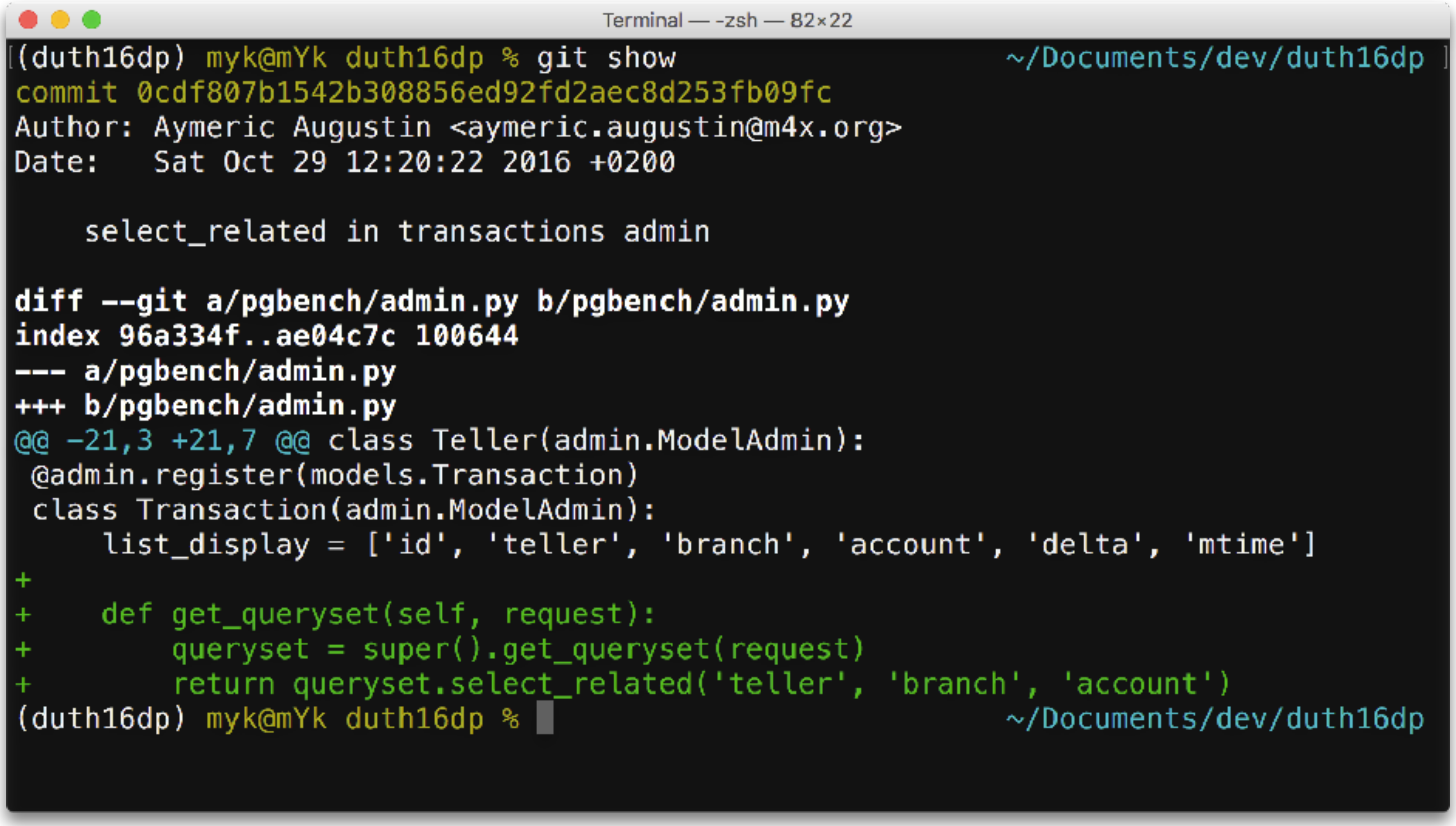
SQL queries from 1 connection

default
392.28 ms (304 queries including 302 duplicates)

QUERY	TIMELINE	TIME (MS)	ACTION
SELECT ··· FROM "auth_user" **WHERE** "auth_user"."id" = 1		1.82	Sel Expl
SELECT ··· FROM "pgbench_history" **Duplicated 2 times.**		28.93	Sel Expl
SELECT ··· FROM "pgbench_history" **Duplicated 2 times.**		27.24	Sel Expl
SELECT ··· FROM "pgbench_history" **ORDER BY** "pgbench_history"."hid" **DESC LIMIT** 100		1.70	Sel Expl
SELECT ··· FROM "pgbench_tellers" **WHERE** "pgbench_tellers"."tid" = 10 **Duplicated 100 times.**		1.27	Sel Expl
SELECT ··· FROM "pgbench_branches" **WHERE** "pgbench_branches"."bid" = 1 **Duplicated 100 times.**		1.61	Sel Expl
SELECT ··· FROM "pgbench_accounts" **WHERE** "pgbench_accounts"."aid" = 3560 **Duplicated 100 times.**		1.50	Sel Expl
SELECT ··· FROM "pgbench_tellers" **WHERE** "pgbench_tellers"."tid" = 4 **Duplicated 100 times.**		1.04	Sel Expl

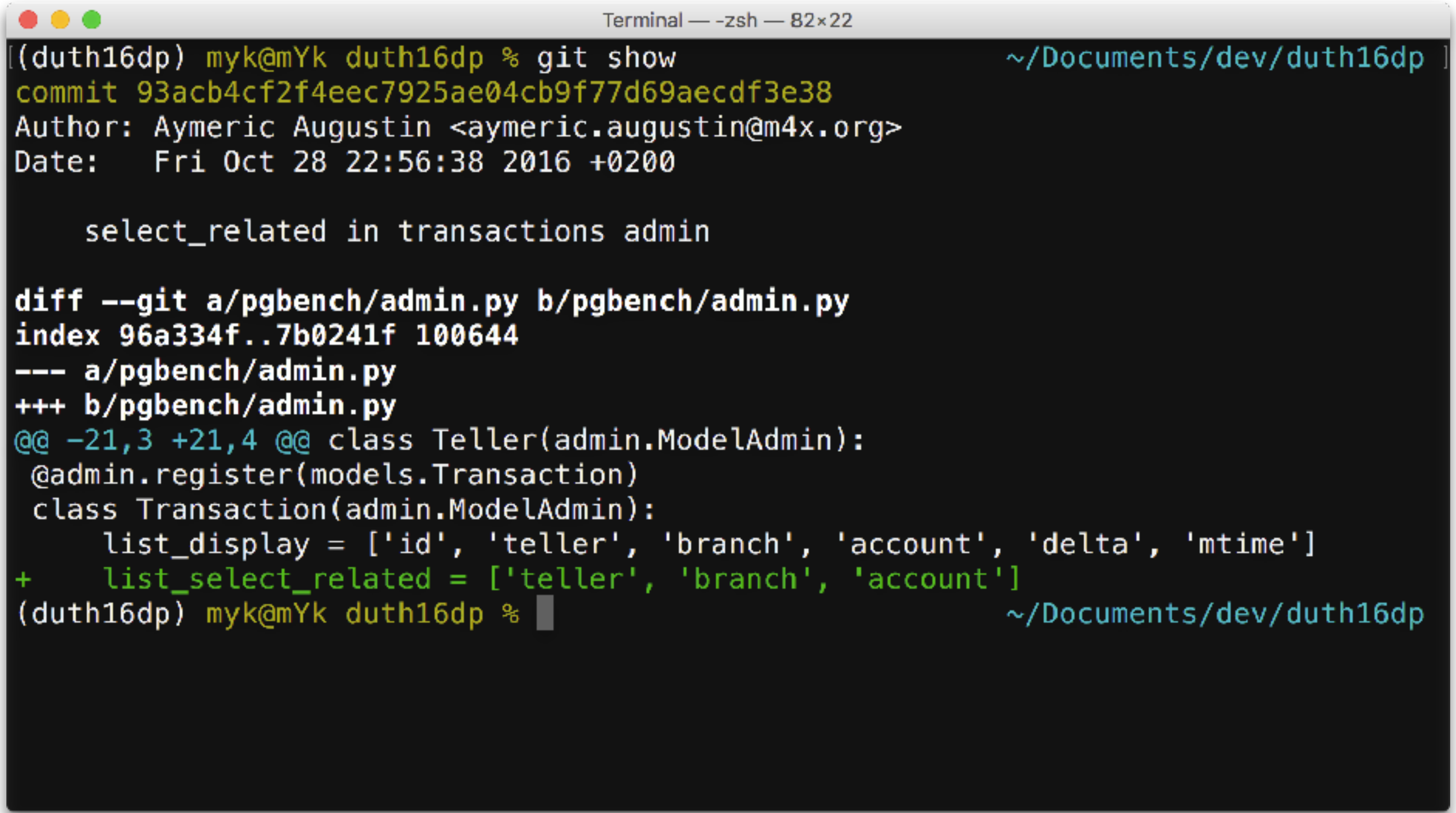
k = 3
N = 100

select_related() to the rescue


Terminal — -zsh — 82×22

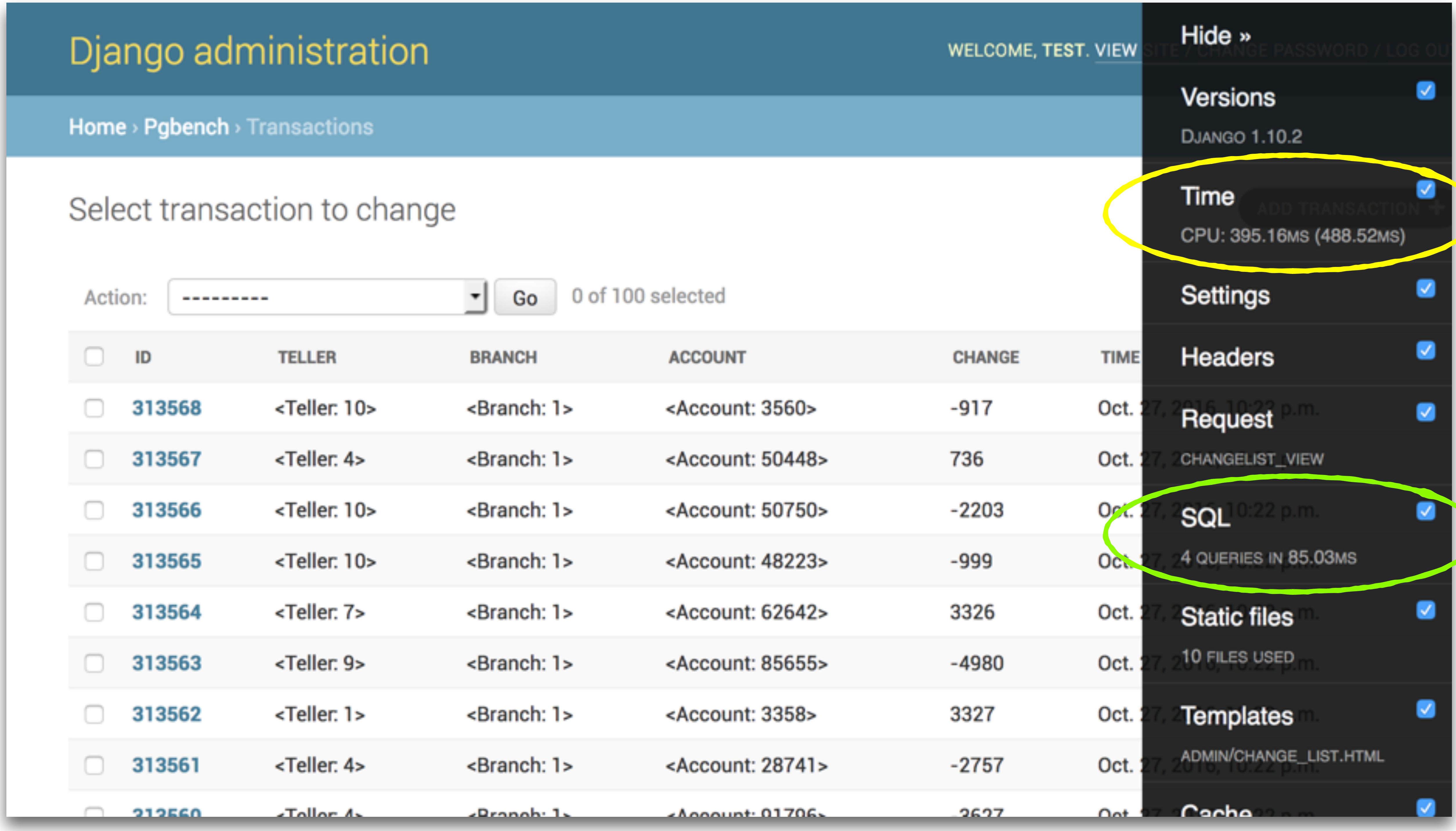
```
(duth16dp) myk@mYk duth16dp % git show                          ~/Documents/dev/duth16dp
commit 0cdf807b1542b308856ed92fd2aec8d253fb09fc
Author: Aymeric Augustin <aymeric.augustin@m4x.org>
Date:   Sat Oct 29 12:20:22 2016 +0200

    select_related in transactions admin

diff --git a/pgbench/admin.py b/pgbench/admin.py
index 96a334f..ae04c7c 100644
--- a/pgbench/admin.py
+++ b/pgbench/admin.py
@@ -21,3 +21,7 @@ class Teller(admin.ModelAdmin):
 @admin.register(models.Transaction)
 class Transaction(admin.ModelAdmin):
     list_display = ['id', 'teller', 'branch', 'account', 'delta', 'mtime']
+
+    def get_queryset(self, request):
+        queryset = super().get_queryset(request)
+        return queryset.select_related('teller', 'branch', 'account')
(duth16dp) myk@mYk duth16dp %                                   ~/Documents/dev/duth16dp
```

select_related() to the rescue



```
(duth16dp) myk@mYk duth16dp % git show                          ~/Documents/dev/duth16dp
commit 93acb4cf2f4eec7925ae04cb9f77d69aecdf3e38
Author: Aymeric Augustin <aymeric.augustin@m4x.org>
Date:   Fri Oct 28 22:56:38 2016 +0200

    select_related in transactions admin

diff --git a/pgbench/admin.py b/pgbench/admin.py
index 96a334f..7b0241f 100644
--- a/pgbench/admin.py
+++ b/pgbench/admin.py
@@ -21,3 +21,4 @@ class Teller(admin.ModelAdmin):
 @admin.register(models.Transaction)
 class Transaction(admin.ModelAdmin):
     list_display = ['id', 'teller', 'branch', 'account', 'delta', 'mtime']
+    list_select_related = ['teller', 'branch', 'account']
(duth16dp) myk@mYk duth16dp %                                   ~/Documents/dev/duth16dp
```

The k * N + 1 queries problem, solved



Django administration

WELCOME, TEST. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Pgbench › Transactions

Select transaction to change

Action: ---------- Go 0 of 100 selected

ID	TELLER	BRANCH	ACCOUNT	CHANGE	TIME
313568	<Teller: 10>	<Branch: 1>	<Account: 3560>	-917	Oct. 27, 2016, 10:22 p.m.
313567	<Teller: 4>	<Branch: 1>	<Account: 50448>	736	Oct. 27, 2
313566	<Teller: 10>	<Branch: 1>	<Account: 50750>	-2203	Oct. 27, 2 10:22 p.m.
313565	<Teller: 10>	<Branch: 1>	<Account: 48223>	-999	Oct. 27, 2
313564	<Teller: 7>	<Branch: 1>	<Account: 62642>	3326	Oct. 27, 2
313563	<Teller: 9>	<Branch: 1>	<Account: 85655>	-4980	Oct. 27, 2016, 10:22 p.m.
313562	<Teller: 1>	<Branch: 1>	<Account: 3358>	3327	Oct. 27, 2
313561	<Teller: 4>	<Branch: 1>	<Account: 28741>	-2757	Oct. 27, 2016, 10:22 p.m.
313560	<Teller: 4>	<Branch: 1>	<Account: 91706>	-3627	Oct. 27,

Hide »

Versions ✓
DJANGO 1.10.2

Time ✓
CPU: 395.16MS (488.52MS)

Settings ✓

Headers ✓

Request ✓
CHANGELIST_VIEW

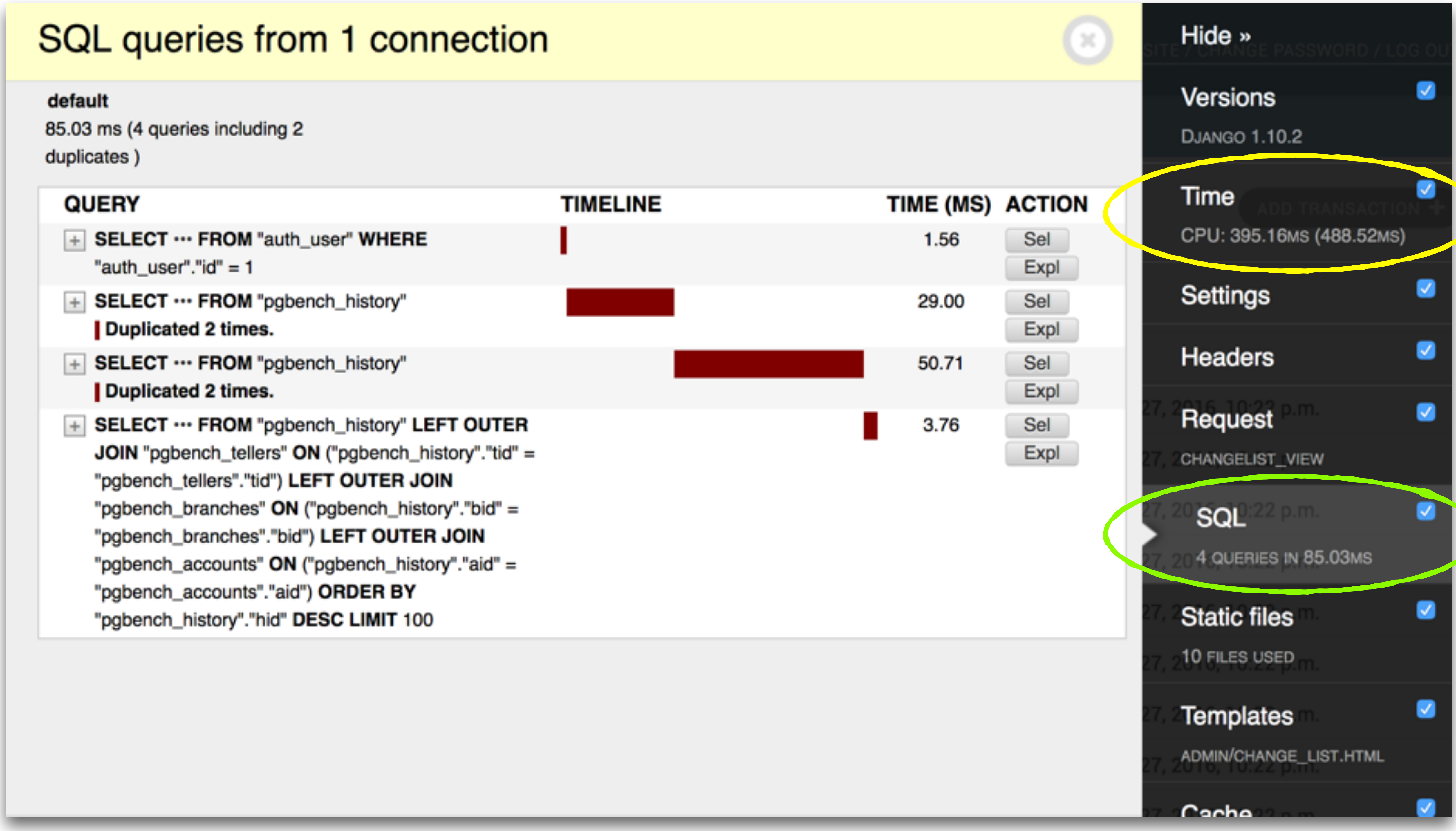
SQL ✓
4 QUERIES IN 85.03MS

Static files ✓
10 FILES USED

Templates ✓
ADMIN/CHANGE_LIST.HTML

Cache ✓

The k * N + 1 queries problem, solved



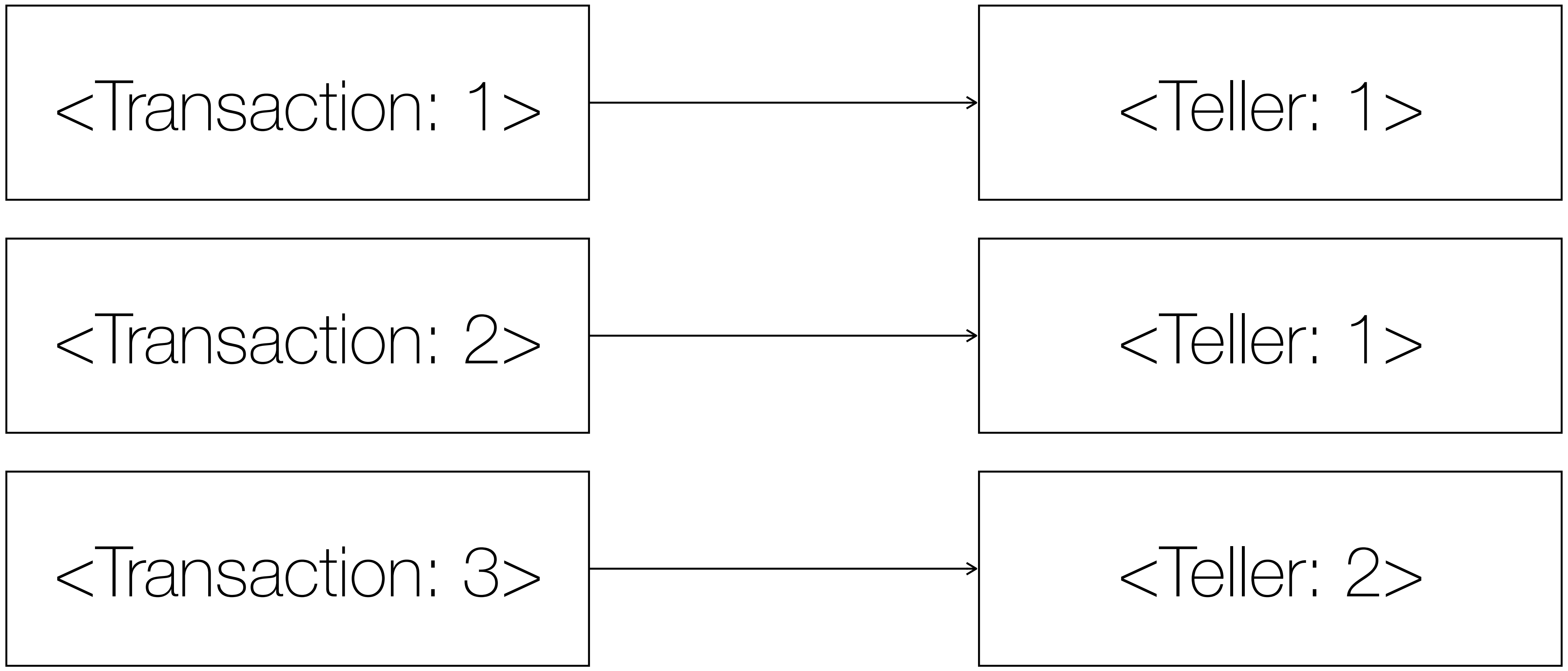
SQL queries from 1 connection

default
85.03 ms (4 queries including 2 duplicates)

QUERY	TIMELINE	TIME (MS)	ACTION
SELECT ··· FROM "auth_user" **WHERE** "auth_user"."id" = 1		1.56	Sel Expl
SELECT ··· FROM "pgbench_history" **Duplicated 2 times.**		29.00	Sel Expl
SELECT ··· FROM "pgbench_history" **Duplicated 2 times.**		50.71	Sel Expl
SELECT ··· FROM "pgbench_history" **LEFT OUTER JOIN** "pgbench_tellers" **ON** ("pgbench_history"."tid" = "pgbench_tellers"."tid") **LEFT OUTER JOIN** "pgbench_branches" **ON** ("pgbench_history"."bid" = "pgbench_branches"."bid") **LEFT OUTER JOIN** "pgbench_accounts" **ON** ("pgbench_history"."aid" = "pgbench_accounts"."aid") **ORDER BY** "pgbench_history"."hid" **DESC LIMIT** 100		3.76	Sel Expl

`select_related()`

JOIN



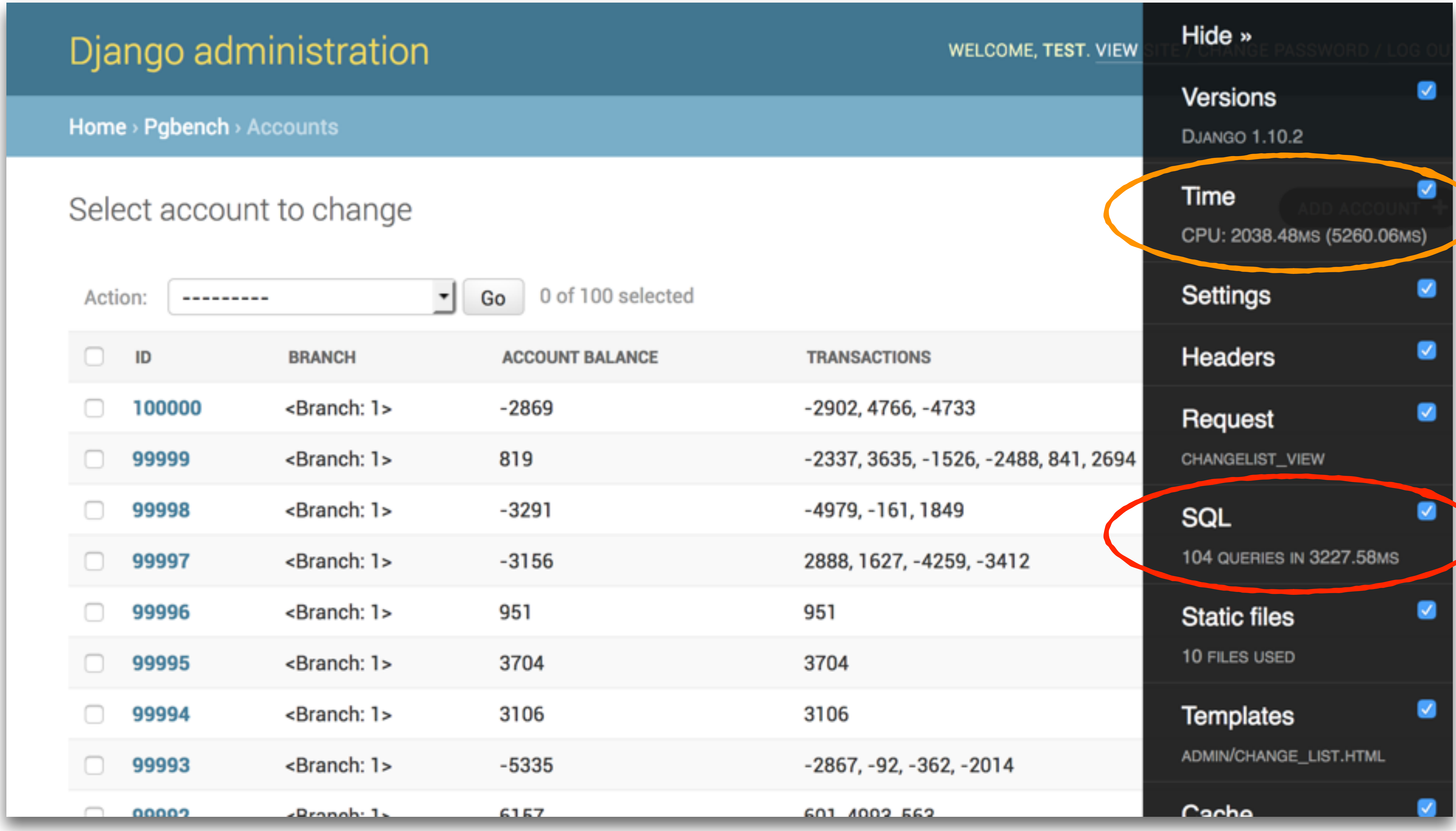
Sprint idea

- Figure out if pathological performance of LEFT OUTER JOIN vs. INNER JOIN still happens these days
- Make the case for treating FKs identically in `select_related` regardless of whether they’re nullable
- Many projects would benefit from a default `select_related` that includes nullable FKs in the admin
- Think about backwards compatibility — no easy answer there
- Search: “site:code.djangoproject.com select_related nullable”

`prefetch_related()`

https://docs.djangoproject.com/en/stable/ref/models/querysets/#django.db.models.query.QuerySet.prefetch_related

The k * N +1 queries problem, again



Django administration

WELCOME, TEST. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Pgbench › Accounts

Select account to change

ADD ACCOUNT

Action: ---------- Go 0 of 100 selected

	ID	BRANCH	ACCOUNT BALANCE	TRANSACTIONS
☐	100000	<Branch: 1>	-2869	-2902, 4766, -4733
☐	99999	<Branch: 1>	819	-2337, 3635, -1526, -2488, 841, 2694
☐	99998	<Branch: 1>	-3291	-4979, -161, 1849
☐	99997	<Branch: 1>	-3156	2888, 1627, -4259, -3412
☐	99996	<Branch: 1>	951	951
☐	99995	<Branch: 1>	3704	3704
☐	99994	<Branch: 1>	3106	3106
☐	99993	<Branch: 1>	-5335	-2867, -92, -362, -2014
☐	99992	<Branch: 1>	6157	601, 4903, 563

Hide »

Versions ✓
DJANGO 1.10.2

Time ✓
CPU: 2038.48MS (5260.06MS)

Settings ✓

Headers ✓

Request ✓
CHANGELIST_VIEW

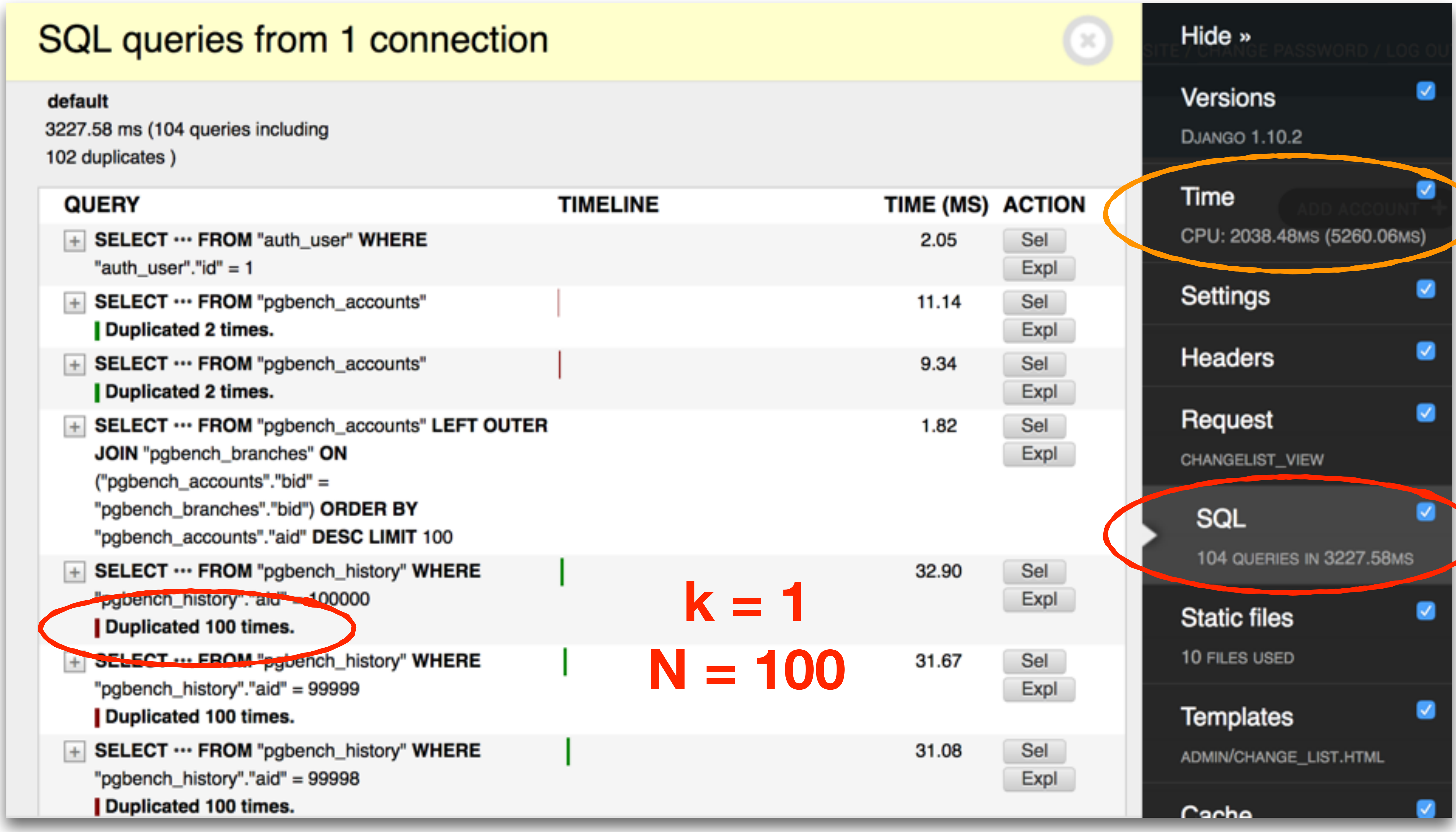
SQL ✓
104 QUERIES IN 3227.58MS

Static files ✓
10 FILES USED

Templates ✓
ADMIN/CHANGE_LIST.HTML

Cache ✓

The k * N +1 queries problem, again



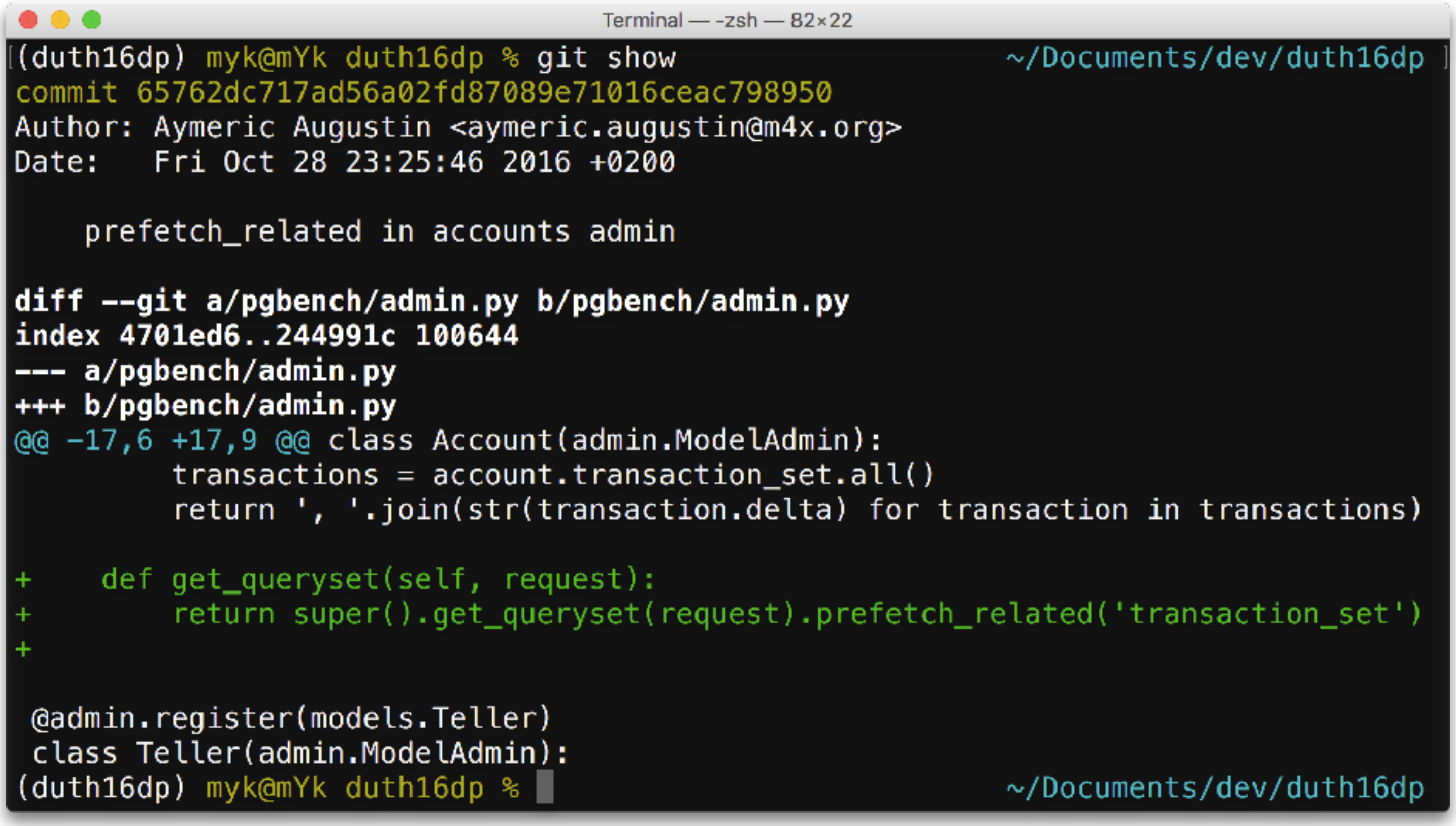
SQL queries from 1 connection

default
3227.58 ms (104 queries including 102 duplicates)

QUERY	TIMELINE	TIME (MS)	ACTION
SELECT ··· FROM "auth_user" WHERE "auth_user"."id" = 1		2.05	Sel Expl
SELECT ··· FROM "pgbench_accounts" Duplicated 2 times.		11.14	Sel Expl
SELECT ··· FROM "pgbench_accounts" Duplicated 2 times.		9.34	Sel Expl
SELECT ··· FROM "pgbench_accounts" LEFT OUTER JOIN "pgbench_branches" ON ("pgbench_accounts"."bid" = "pgbench_branches"."bid") ORDER BY "pgbench_accounts"."aid" DESC LIMIT 100		1.82	Sel Expl
SELECT ··· FROM "pgbench_history" WHERE "pgbench_history"."aid" = 100000 Duplicated 100 times.		32.90	Sel Expl
SELECT ··· FROM "pgbench_history" WHERE "pgbench_history"."aid" = 99999 Duplicated 100 times.		31.67	Sel Expl
SELECT ··· FROM "pgbench_history" WHERE "pgbench_history"."aid" = 99998 Duplicated 100 times.		31.08	Sel Expl

k = 1
N = 100

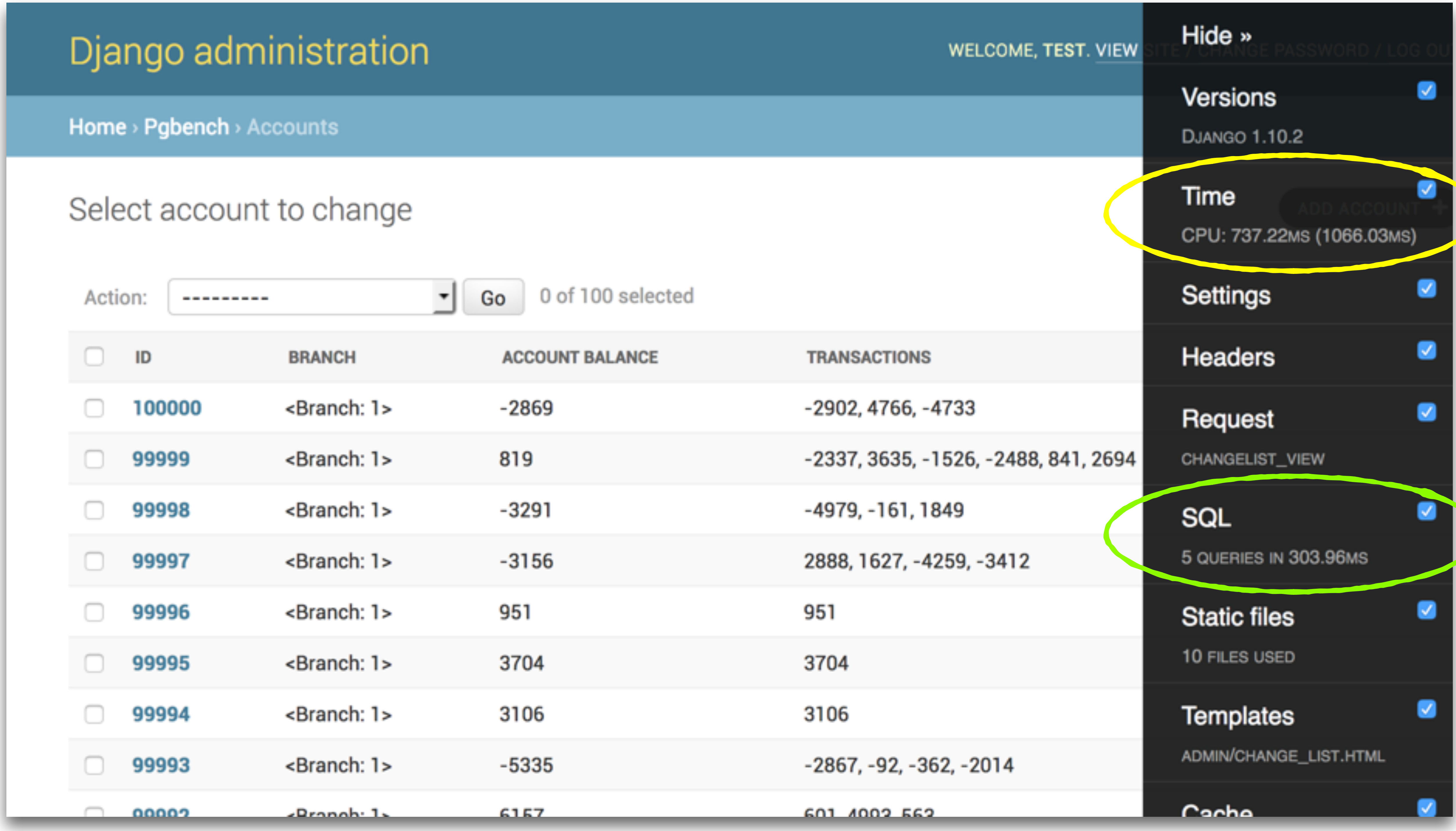
prefetch_related() to the rescue



Terminal — -zsh — 82×22

```
(duth16dp) myk@mYk duth16dp % git show                    ~/Documents/dev/duth16dp
commit 65762dc717ad56a02fd87089e71016ceac798950
Author: Aymeric Augustin <aymeric.augustin@m4x.org>
Date:   Fri Oct 28 23:25:46 2016 +0200

    prefetch_related in accounts admin

diff --git a/pgbench/admin.py b/pgbench/admin.py
index 4701ed6..244991c 100644
--- a/pgbench/admin.py
+++ b/pgbench/admin.py
@@ -17,6 +17,9 @@ class Account(admin.ModelAdmin):
         transactions = account.transaction_set.all()
         return ', '.join(str(transaction.delta) for transaction in transactions)

+    def get_queryset(self, request):
+        return super().get_queryset(request).prefetch_related('transaction_set')
+

 @admin.register(models.Teller)
 class Teller(admin.ModelAdmin):
(duth16dp) myk@mYk duth16dp %                             ~/Documents/dev/duth16dp
```

The k * N + 1 queries problem, solved again



Django administration

WELCOME, TEST. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Pgbench › Accounts

Select account to change

ADD ACCOUNT

Action: ---------- Go 0 of 100 selected

	ID	BRANCH	ACCOUNT BALANCE	TRANSACTIONS
☐	100000	<Branch: 1>	-2869	-2902, 4766, -4733
☐	99999	<Branch: 1>	819	-2337, 3635, -1526, -2488, 841, 2694
☐	99998	<Branch: 1>	-3291	-4979, -161, 1849
☐	99997	<Branch: 1>	-3156	2888, 1627, -4259, -3412
☐	99996	<Branch: 1>	951	951
☐	99995	<Branch: 1>	3704	3704
☐	99994	<Branch: 1>	3106	3106
☐	99993	<Branch: 1>	-5335	-2867, -92, -362, -2014
☐	99992	<Branch: 1>	6157	601, 4993, 563

Hide »

Versions ✓
DJANGO 1.10.2

Time ✓
CPU: 737.22MS (1066.03MS)

Settings ✓

Headers ✓

Request ✓
CHANGELIST_VIEW

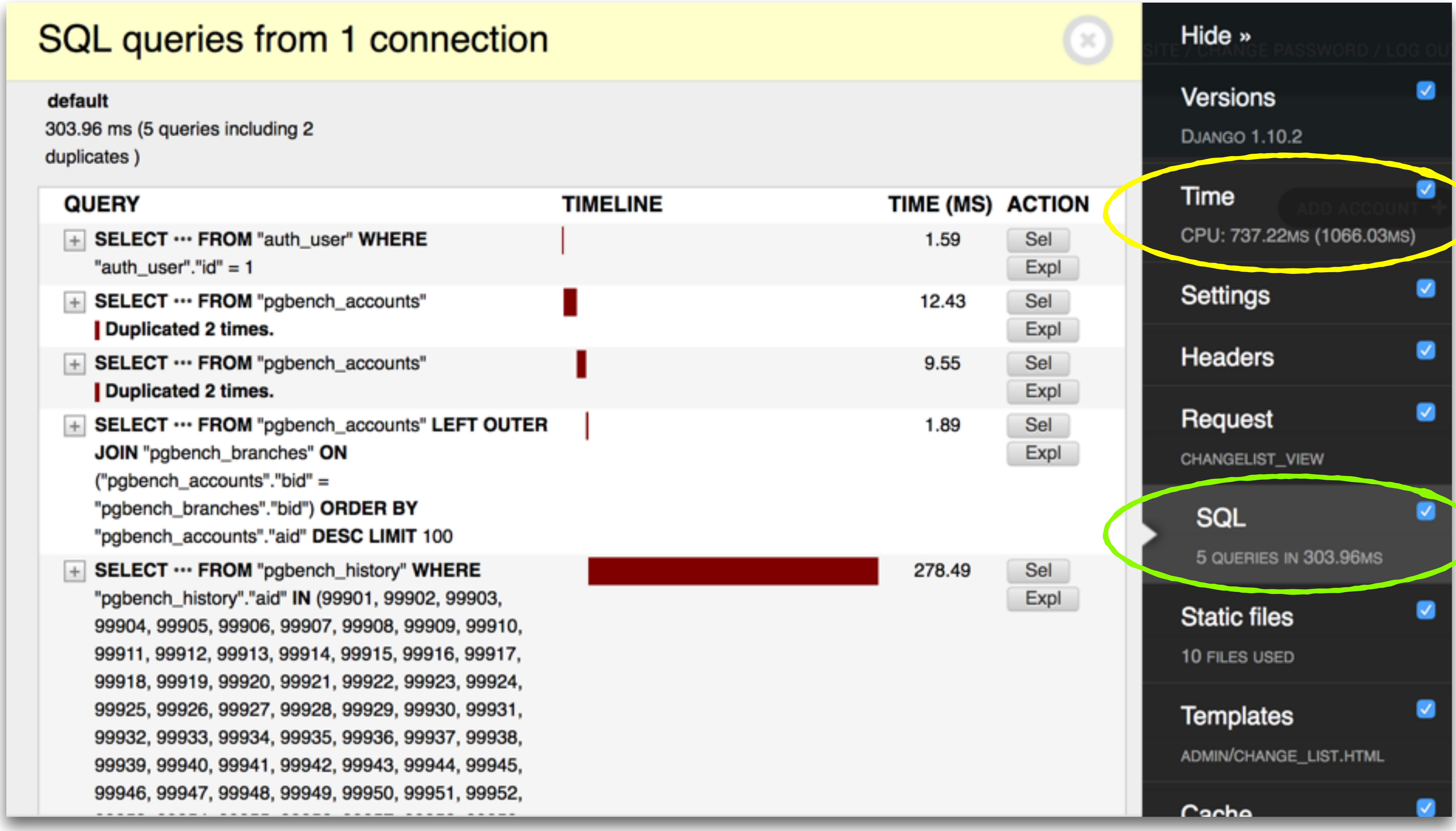
SQL ✓
5 QUERIES IN 303.96MS

Static files ✓
10 FILES USED

Templates ✓
ADMIN/CHANGE_LIST.HTML

Cache ✓

The k * N + 1 queries problem, solved again

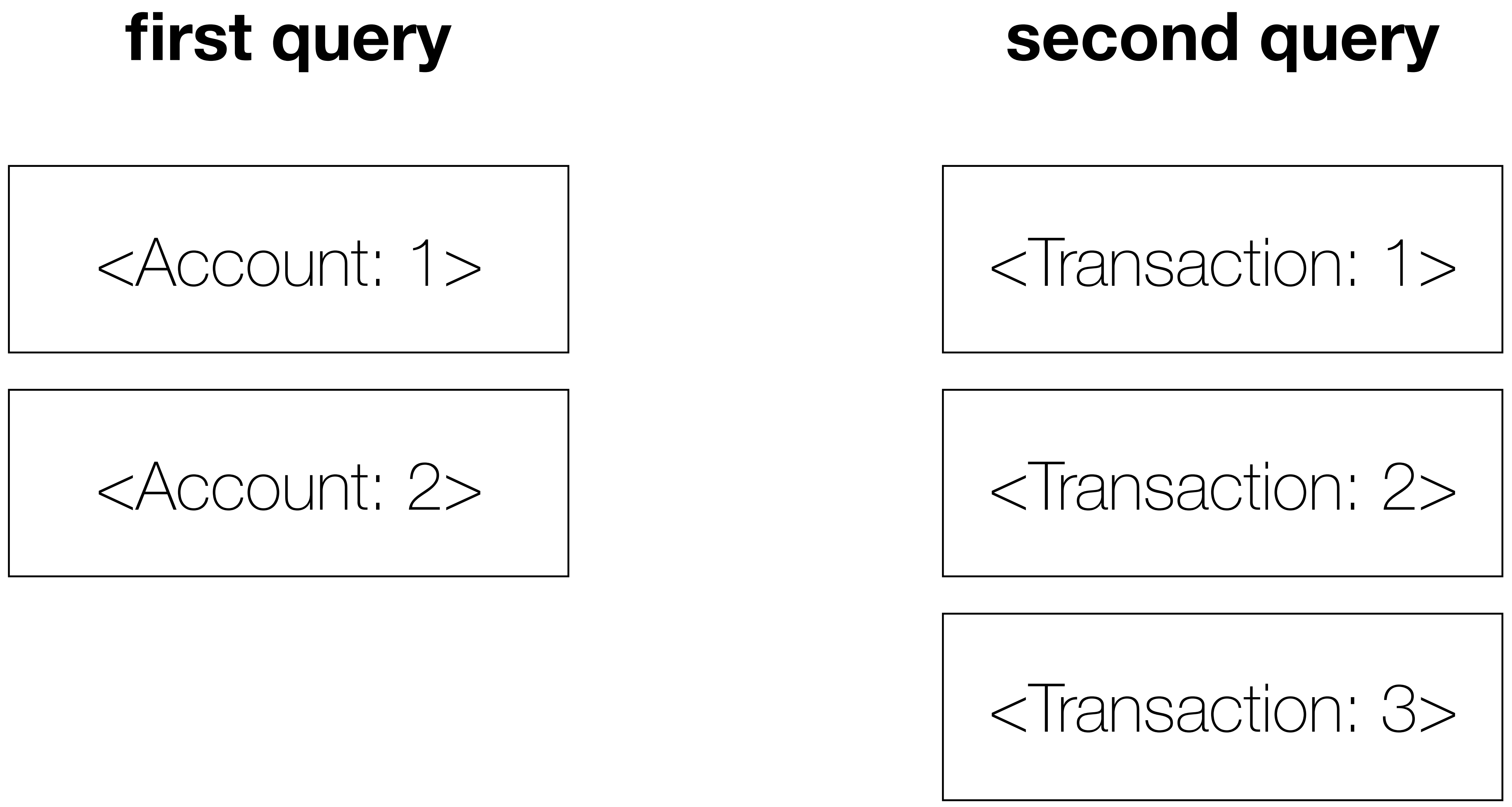


SQL queries from 1 connection

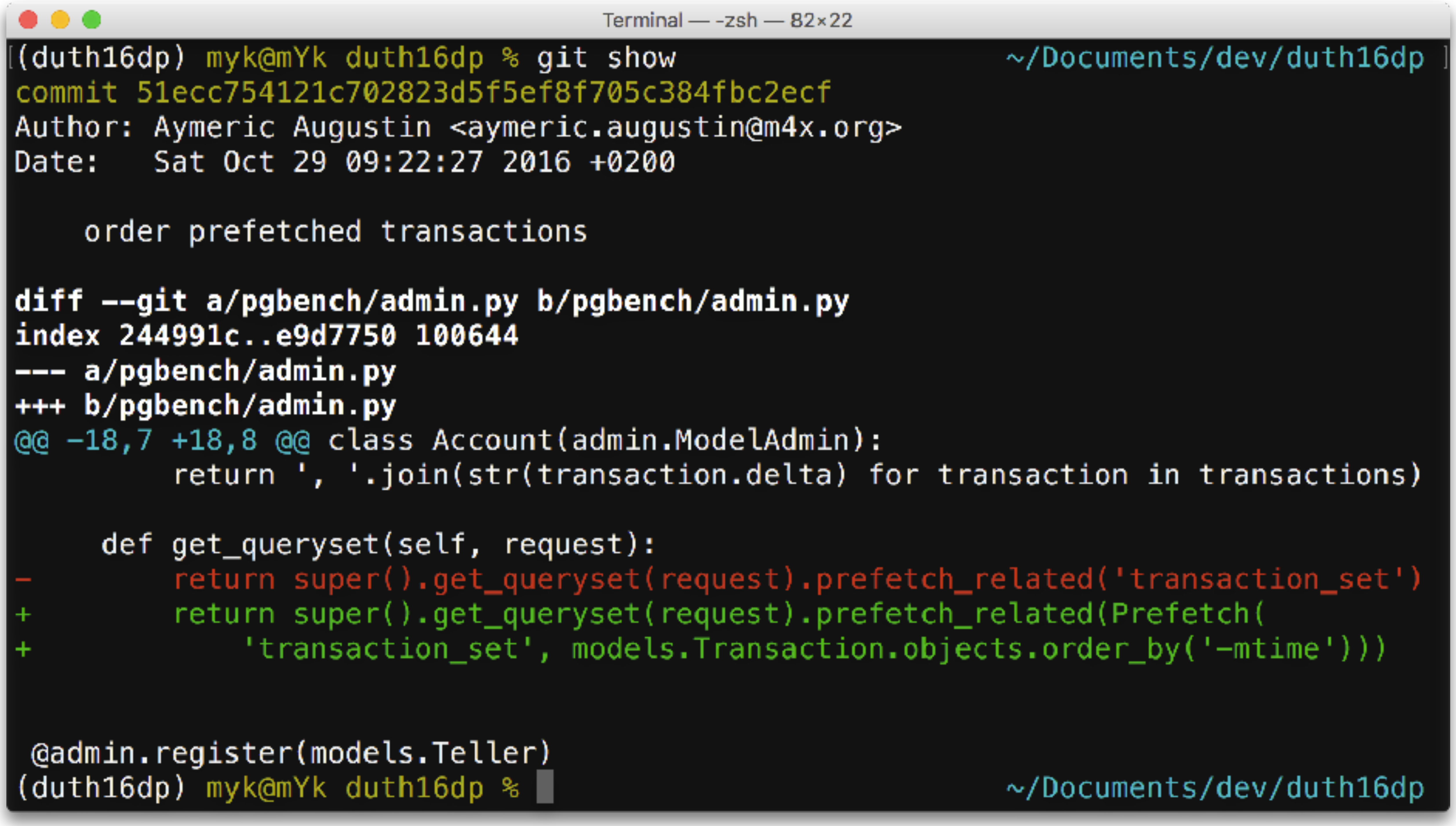
default
303.96 ms (5 queries including 2 duplicates)

QUERY	TIMELINE	TIME (MS)	ACTION
SELECT ··· FROM "auth_user" **WHERE** "auth_user"."id" = 1		1.59	Sel Expl
SELECT ··· FROM "pgbench_accounts" **Duplicated 2 times.**		12.43	Sel Expl
SELECT ··· FROM "pgbench_accounts" **Duplicated 2 times.**		9.55	Sel Expl
SELECT ··· FROM "pgbench_accounts" **LEFT OUTER JOIN** "pgbench_branches" **ON** ("pgbench_accounts"."bid" = "pgbench_branches"."bid") **ORDER BY** "pgbench_accounts"."aid" **DESC LIMIT** 100		1.89	Sel Expl
SELECT ··· FROM "pgbench_history" **WHERE** "pgbench_history"."aid" **IN** (99901, 99902, 99903, 99904, 99905, 99906, 99907, 99908, 99909, 99910, 99911, 99912, 99913, 99914, 99915, 99916, 99917, 99918, 99919, 99920, 99921, 99922, 99923, 99924, 99925, 99926, 99927, 99928, 99929, 99930, 99931, 99932, 99933, 99934, 99935, 99936, 99937, 99938, 99939, 99940, 99941, 99942, 99943, 99944, 99945, 99946, 99947, 99948, 99949, 99950, 99951, 99952,		278.49	Sel Expl

`prefetch_related()`



Prefetch objects



```
(duth16dp) myk@mYk duth16dp % git show                        ~/Documents/dev/duth16dp
commit 51ecc754121c702823d5f5ef8f705c384fbc2ecf
Author: Aymeric Augustin <aymeric.augustin@m4x.org>
Date:   Sat Oct 29 09:22:27 2016 +0200

    order prefetched transactions

diff --git a/pgbench/admin.py b/pgbench/admin.py
index 244991c..e9d7750 100644
--- a/pgbench/admin.py
+++ b/pgbench/admin.py
@@ -18,7 +18,8 @@ class Account(admin.ModelAdmin):
         return ', '.join(str(transaction.delta) for transaction in transactions)

     def get_queryset(self, request):
-        return super().get_queryset(request).prefetch_related('transaction_set')
+        return super().get_queryset(request).prefetch_related(Prefetch(
+            'transaction_set', models.Transaction.objects.order_by('-mtime')))


 @admin.register(models.Teller)
(duth16dp) myk@mYk duth16dp %                                 ~/Documents/dev/duth16dp
```

`Prefetch` objects

- `Prefetch(lookup, queryset=None, to_attr=None)`

- Specify a relation to follow
- Can filter, order, etc. the target queryset
    - Required to filter, order, etc. prefetched querysets
- Can attach the result to an attribute with another name
    - Recommended when the target queryset is filtered

`prefetch_related_objects()`

- Like `prefetch_related`
- Works on any iterable of model instances
- New in Django 1.10

`select_related()`

vs.

`prefetch_related()`

select_related() vs. prefetch_related()

	1/N to 1	1/N to N
`select_related()`	YES	NO
`prefetch_related()`	YES	NO

`select_related()` vs. `prefetch_related()`

- “Generally you’ll want to use `select_related()`”
    - It’s more elegant but it isn’t always faster
- `select_related()` fetches more data
    - Consequences depend on the database schema and content
- `prefetch_related()` makes several queries
    - Transactional consistency isn’t guaranteed (#27403)
- Depends mostly on the latency of database queries

select_related() vs. prefetch_related()

```
from pgbench.models import Transaction
transactions = Transaction.objects.all()


# Many distinct related objects


transactions.select_related('account')
# Database = 1.32s - Total = 11.4s


transactions.prefetch_related('account')
# Database = 0.62s + 0.58s = 1s - Total = 14.5s (+26%)
```

select_related() vs. prefetch_related()

```
from pgbench.models import Transaction
transactions = Transaction.objects.all()


# Few distinct related objects


transactions.select_related('teller')
# Database = 0.87s - Total = 12.6s


transactions.prefetch_related('teller')
# Database = 0.66s + ~0s = 0.66s - Total = 12.3s (-2.4%)
```

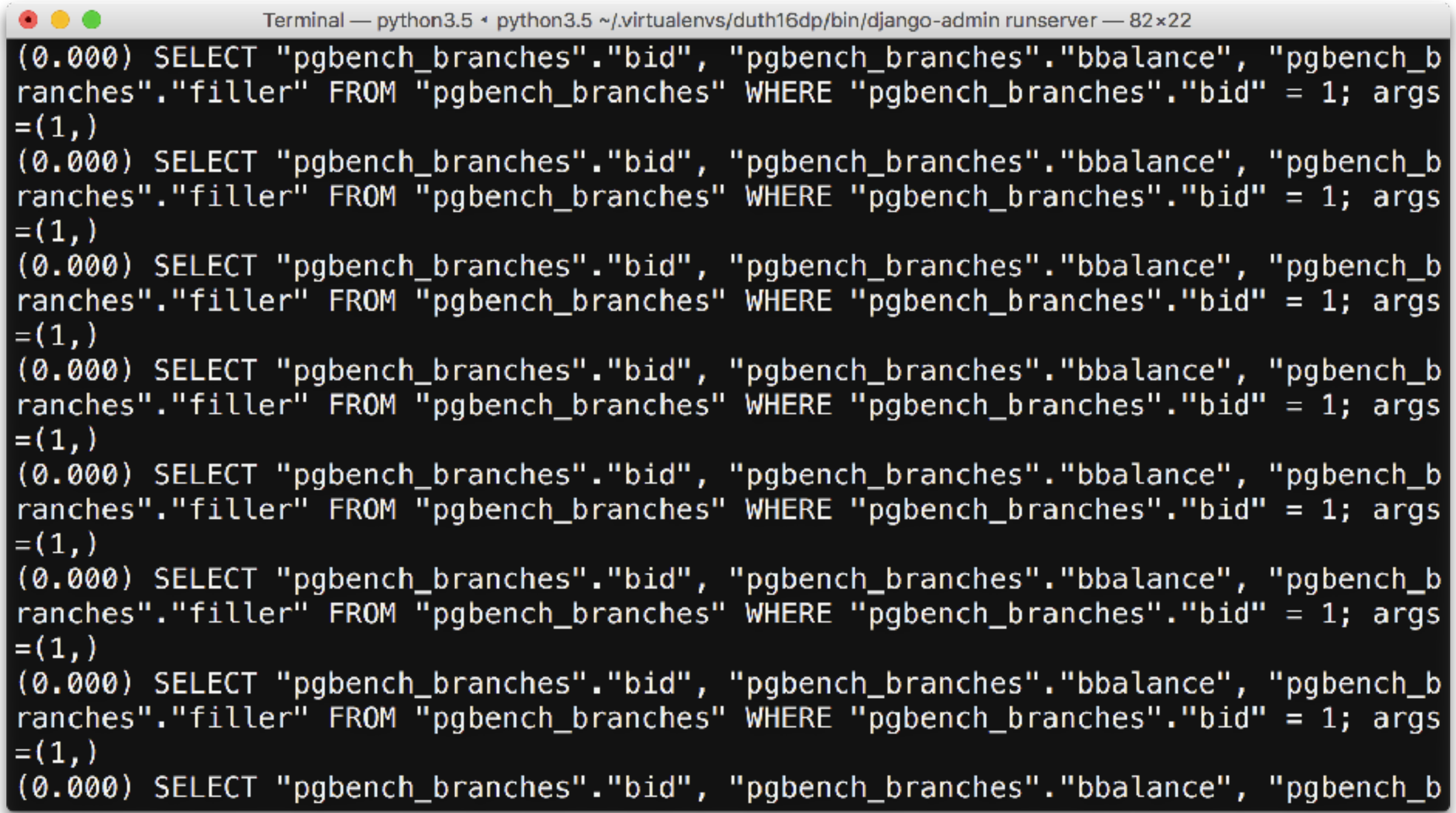
“How do I check query patterns with Django Rest Framework?”

https://www.dabapps.com/blog/api-performance-profiling-django-rest-framework/

Log database queries to the console

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'handlers': {
        'console': {
            'class': 'logging.StreamHandler',
        },
    },
    'loggers': {
        'django.db.backends': {
            'handlers': ['console'],
            'level': 'DEBUG',
        },
    },
}
```

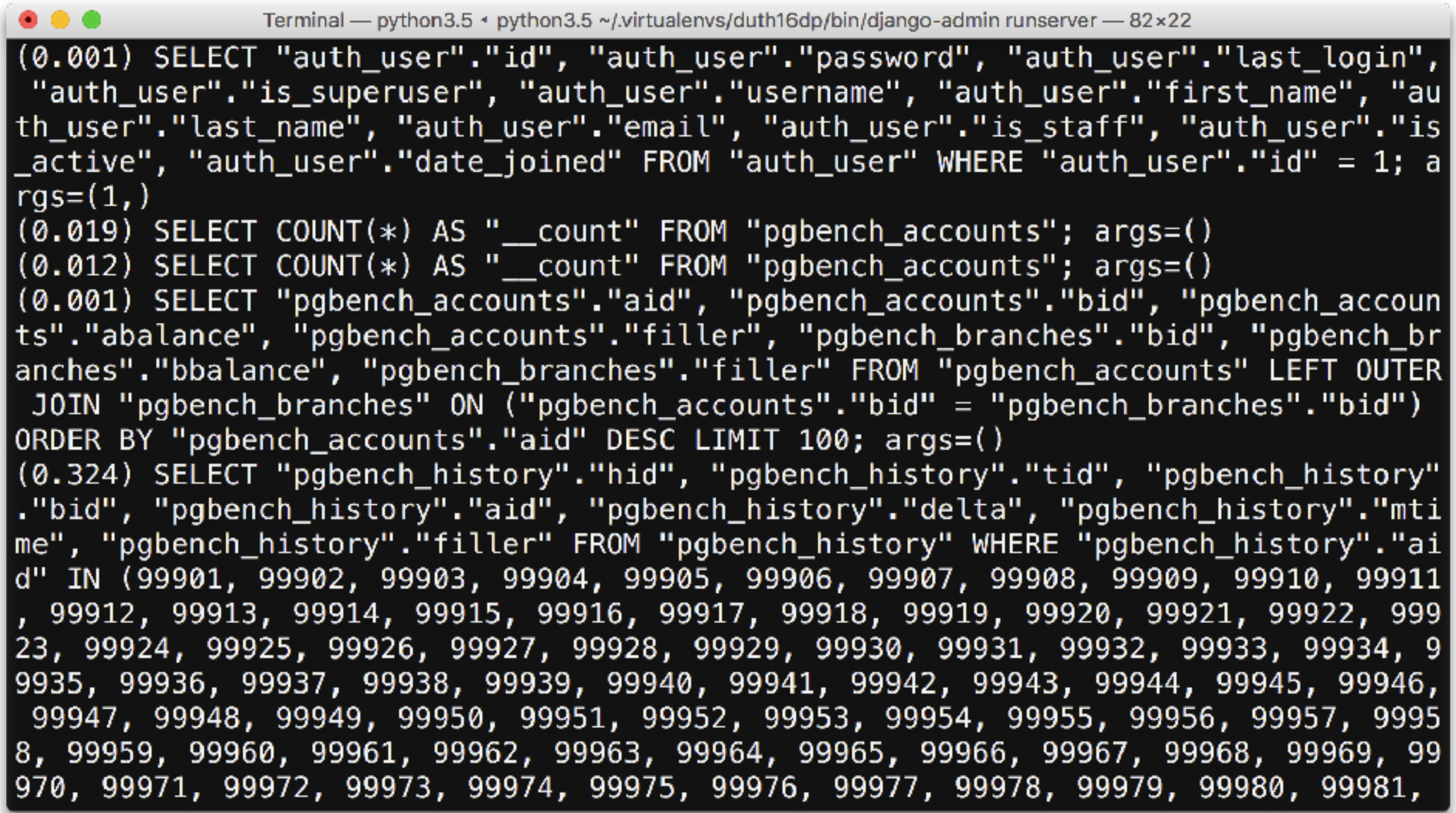
Log database queries to the console



Terminal — python3.5 ◂ python3.5 ~/.virtualenvs/duth16dp/bin/django-admin runserver — 82×22

```
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
ranches"."filler" FROM "pgbench_branches" WHERE "pgbench_branches"."bid" = 1; args
=(1,)
(0.000) SELECT "pgbench_branches"."bid", "pgbench_branches"."bbalance", "pgbench_b
```

Log database queries to the console


Terminal — python3.5 ◂ python3.5 ~/.virtualenvs/duth16dp/bin/django-admin runserver — 82×22

```
(0.001) SELECT "auth_user"."id", "auth_user"."password", "auth_user"."last_login",
 "auth_user"."is_superuser", "auth_user"."username", "auth_user"."first_name", "au
th_user"."last_name", "auth_user"."email", "auth_user"."is_staff", "auth_user"."is
_active", "auth_user"."date_joined" FROM "auth_user" WHERE "auth_user"."id" = 1; a
rgs=(1,)
(0.019) SELECT COUNT(*) AS "__count" FROM "pgbench_accounts"; args=()
(0.012) SELECT COUNT(*) AS "__count" FROM "pgbench_accounts"; args=()
(0.001) SELECT "pgbench_accounts"."aid", "pgbench_accounts"."bid", "pgbench_accoun
ts"."abalance", "pgbench_accounts"."filler", "pgbench_branches"."bid", "pgbench_br
anches"."bbalance", "pgbench_branches"."filler" FROM "pgbench_accounts" LEFT OUTER
 JOIN "pgbench_branches" ON ("pgbench_accounts"."bid" = "pgbench_branches"."bid")
ORDER BY "pgbench_accounts"."aid" DESC LIMIT 100; args=()
(0.324) SELECT "pgbench_history"."hid", "pgbench_history"."tid", "pgbench_history"
."bid", "pgbench_history"."aid", "pgbench_history"."delta", "pgbench_history"."mti
me", "pgbench_history"."filler" FROM "pgbench_history" WHERE "pgbench_history"."ai
d" IN (99901, 99902, 99903, 99904, 99905, 99906, 99907, 99908, 99909, 99910, 99911
, 99912, 99913, 99914, 99915, 99916, 99917, 99918, 99919, 99920, 99921, 99922, 999
23, 99924, 99925, 99926, 99927, 99928, 99929, 99930, 99931, 99932, 99933, 99934, 9
9935, 99936, 99937, 99938, 99939, 99940, 99941, 99942, 99943, 99944, 99945, 99946,
 99947, 99948, 99949, 99950, 99951, 99952, 99953, 99954, 99955, 99956, 99957, 9995
8, 99959, 99960, 99961, 99962, 99963, 99964, 99965, 99966, 99967, 99968, 99969, 99
970, 99971, 99972, 99973, 99974, 99975, 99976, 99977, 99978, 99979, 99980, 99981,
```

other ORM optimizations

Festival international des jardins, Chaumont, 8 mai 2015

Baseline

```python
from collections import defaultdict
balances = defaultdict(lambda: 0)

from pgbench.models import Transaction
txs = Transaction.objects.all()

for tx in txs:
    balances[tx.teller_id] += tx.delta

# Database = 630ms - Total = 6900ms
```

only() and defer()

```python
from collections import defaultdict
balances = defaultdict(lambda: 0)

from pgbench.models import Transaction
txs = Transaction.objects.only('teller_id', 'delta')

for tx in txs:
    balances[tx.teller_id] += tx.delta

# Database = 220ms (-65%) - Total = 7900ms (+15%)
```

Use `only()` or `defer()` when…

- you need model instances
- you don’t need all columns
  - especially columns containing large amounts of data

- Not a common use case
- Consider moving rarely needed data to a separate model

values_list() and values()

```
from collections import defaultdict
balances = defaultdict(lambda: 0)

from pgbench.models import Transaction
txs = Transaction.objects.values_list('teller_id', 'delta')

for teller_id, delta in txs:
    balances[teller_id] += delta

# Database = 160ms (-75%) - Total = 550ms (-92%)
```

Use `values_list()` or `values()` when…

- you don’t need model instances
- you need to manipulate large amounts of data
  - large starts between 1 000 and 10 000 rows :-(
- Common use case: reports
- Huge, easy improvement for queries that return lots of rows

aggregate() and annotate()

```
from django.db.models import Sum
from pgbench.models import Transaction


balances = dict(
    Transaction.objects
        .values_list('teller_id')
        .annotate(Sum('delta'))
)


# Database = 75ms (-88%) - Total = 77ms (-99%)
```

Pro-tip: `print(queryset.query)`

```
SELECT
    "pgbench_history"."tid",
    SUM("pgbench_history"."delta") AS "delta__sum"
FROM
    "pgbench_history"
GROUP BY
    “pgbench_history"."tid"
```

Use `aggregate()` or `annotate()` when…

- you can perform a calculation in the database
- you need to manipulate large amounts of data
    - large starts between 1 000 and 10 000 rows :-(

- Common use case: dashboards
- There's a learning curve

`iterator()`

- Iterates over instances
- Doesn’t cache results

- Not a common use case
- The whole dataset is still fetched from the database at once
- Live references to instances still prevent garbage collection

Baseline

```
txs = Transaction.objects.all()

import tracemalloc
tracemalloc.start()

for tx in txs:
    balances[tx.teller_id] += tx.delta

tracemalloc.get_traced_memory()
# (166775984, 166824045) => current: 159MB, peak: 159MB
```

iterator()

```
txs = Transaction.objects.iterator()

import tracemalloc
tracemalloc.start()

for tx in txs:
    balances[tx.teller_id] += tx.delta

tracemalloc.get_traced_memory()
# (187159, 234230)      => current: 183kB, peak: 229kB
```

Thank you!

Questions?

Festival international des jardins, Chaumont, 8 mai 2015