Laravel Meetup

About things and front end performance.

Last month in Laravel

- Release date
- Scheduler
- Homestead 2.0
- Latest packages

5.OMG Release date

- January 2015

Scheduler

- No more * * * * 1

```
public function schedule(Schedulable $scheduler)
{
    //every day at 4:17am
    return $scheduler
        ->daily()
        ->hours(4)
        ->minutes(17);
}
```

One cron to rule them all

```
* * * * * /usr/bin/rcron php /path/to/artisan scheduled:run 1>> /dev/null 2>&1
```

- Indatus/dispatcher

Latest packages

```
{
    "mascame/arrayer": "1.*",
    "morilog/jalali": "dev-master",
    "robbiep/cloudconvert-laravel": "1.*@dev",
    "deefour/presenter": "~0.1@dev",
    "adityamenon/postcodes-io": "~1.0",
    "bitolaco/silex-eloquent": "*",
    "mlantz/hadron" : "dev-master",
    "howtomakeaturn/csvdumper": "1.0.0"
}
```

Frontend Performance

- International PHP conference talk
- Philip Tellis

Get the most benefit with the least effort

0

Beginning

Start with a slow site

0.1

Start Measuring

webpagetest.org

Web Page Performance Test for

www.everlytic.co.za

From: Amsterdam, NL - **Chrome - Cable**
26/11/2014 10:10:34

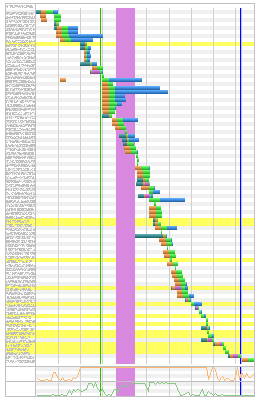
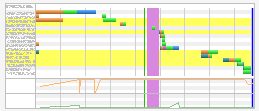
Need help improving?

A	F	A	F	B	X
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

Tester: WPT-PUBLIC01-IE-46.226.57.90
Re-run the test

Raw page data - Raw object data
Export HTTP Archive (.har)
See in ShowSlow
View Test Log

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete Time	Document Complete Requests	Document Complete Bytes In	Fully Loaded Time	Fully Loaded Requests	Fully Loaded Bytes In
First View	9.243s	0.735s	2.893s	3575	612	9.243s	87	2,273 KB	9.787s	88	2,284 KB
Repeat View	4.360s	0.824s	2.192s	2848	612	4.360s	16	130 KB	4.360s	16	216 KB

	Waterfall	Screen Shot	Video
First View (9.243s)			Filmstrip View - Watch Video
Repeat View (4.360s)			Filmstrip View - Watch Video

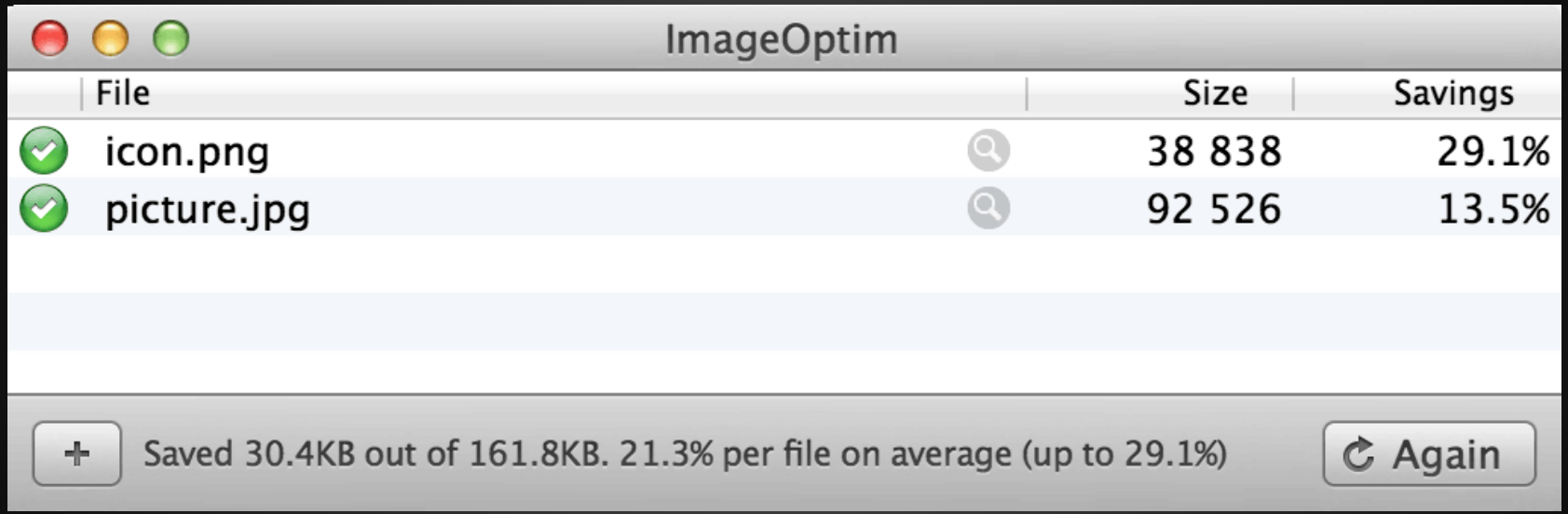
0.2

Enable gzip

0.3

ImageOptim

imageoptim.com


ImageOptim
File
Size
Savings
icon.png
38 838
29.1%
picture.jpg
92 526
13.5%
Saved 30.4KB out of 161.8KB. 21.3% per file on average (up to 29.1%)
Again

0.4

Cache control

YAY

You are now a beginner

1

What the experts do

1.1
CDN

1.2

Split your JS

1.3

Audit your CSS

1.4

Parallelise downloads

1.5

Flush early and often

Get bytes to the client a.s.a.p to avoid TCP slow start

1.6

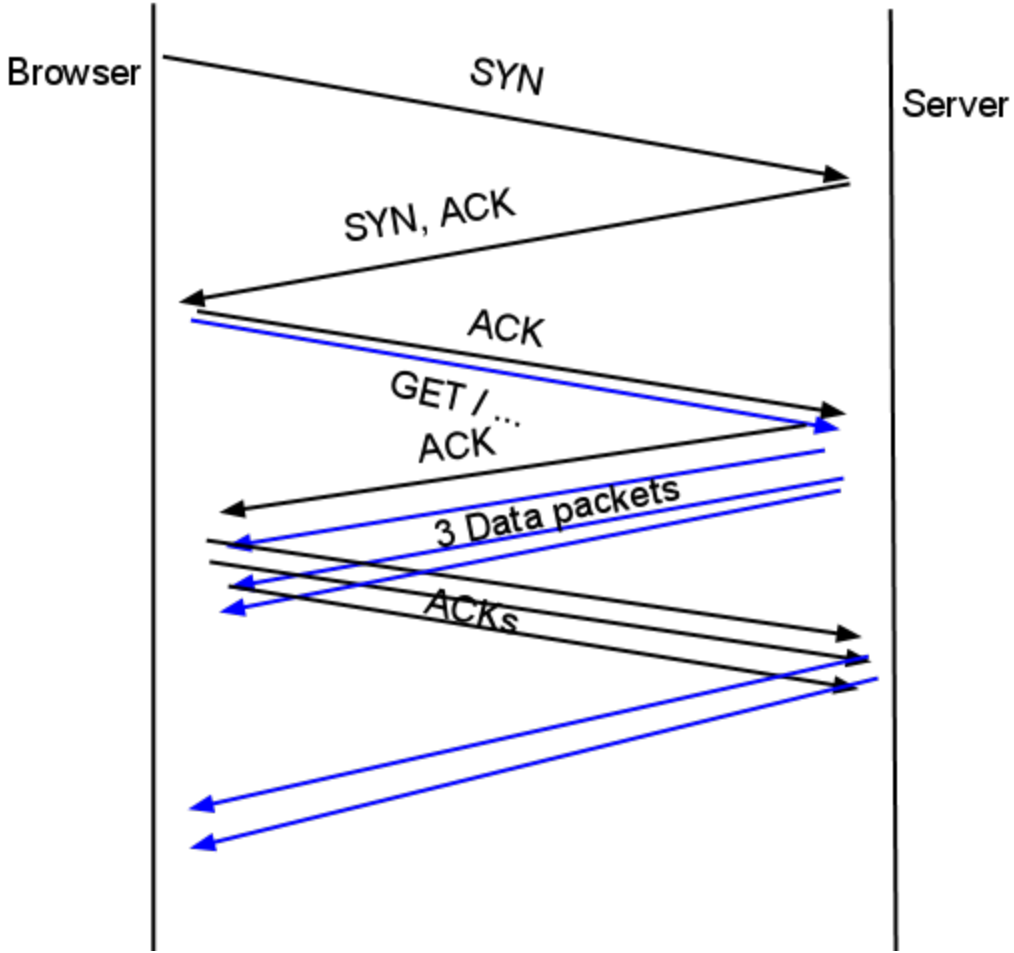
Increase initcwnd

Initial congestion window

1 TCP packet

= 1500 bytes

1.5 kb


Browser
Server
SYN
SYN, ACK
ACK
GET /...
ACK
3 Data packets
ACKs

3 way handshake

- Step 1: Client sends SYN to server - "How are you? My receive window is 65,535 bytes."

- Step 2: Server sends SYN, ACK - "Great! How are you? My receive window is 4,236 bytes"

- Step 3: Client sends ACK, SEQ - "Great as well... Please send me the webpage http://www.example.com/"

- Step 4: Server sends 3 data packets. Roughly 4 - 4.3 kb (3*MSS1) of data
- Step 5: Client acknowledges the segment (sends ACK)
- Step 6: Server sends the remaining bytes to the client

JS files <= 15kb

1.6b

net.ipv4.tcp_slow_start_after_idle=0

1.7

mod_pagespeed || ngx_pagespeed

https://developers.google.com/speed/pagespeed/module

Woot!

You are now in expert land

2

Crazy-Here-Be-Dragons-Land

2.01

Study real user data

2.1

Pre-load assets

2.1b

Pre-render

<link rel=”prerender” href=”url”>

2.1b

Subresource (chromium)

<link rel=”subresource” href=””>

2.1b

DNS Prefetch (FF)

<link rel=”dns-prefetch” href=””>

2.1c

onVisibilityChange

2.2

Post Load

Fetch optimal assets after onload

2.3

Detect broken Accept-Encoding

2.4

Prepare for HTTP/2.0

Multiple assets over the same tcp connection by default

2.5

3rd Party Failures

blackhole.webpagetest.org

2.6

Remove non-crit assets out of the crit path

CrazyLand

fin.