



DUBLIN

DRUPALCON





DUBLIN
DRUPALCON

Stream Your Way to Success

Big Pipe, Refreshless, ESI, AJAX and more!

Fabian Franz (Fabianx) - @fabianfranz

Performance Track

Stream your way to
success!

Big Pipe, Refreshless, ESI, AJAX and
more!

Welcome!

This is a beginner
session!

So if you are not a
beginner ...

... you need to
Leave!

Immediately!

Unless you are here
for the demos.

Or want to know
what crazy things ...

... I have been
working on now.

Then you can
stay! :-)

or for any other reason ;)

So who am I?

I am Fabian Franz

aka Fabianx

Btw. if you wondered: That originally
did stand for Fabian + Linux
I also wanted an -x extension :D

@fabianfranz on
Twitter

I work for Tag 1
Consulting

as a
Senior Performance
Engineer / Technical Lead

And I recently got
appointed to be a

Drupal 7 core maintainer

(a framework manager to be precise)

I was born at the
same day as GNU

(so I am living OSS
history)

which was founded
on September 27, 1983

OMG

That day is today!

I wanted to bring a
cake ...

but that auditorium was
a little too large for it ...

But I'll have some
demos for you later

Just no cake!

But enough about
me

Who are you?

Let me get to know
you

Ah, I know



You all are Pirates!

Eager to conquer the
seas of Performance!

So

Who of you want a
faster site?

Who of you dont?

Ah, nevermind :).

100% of you want a
faster site! - Nice!

But

How do you get
one?

The answer is:

42!

Ah, no wrong track.

No, the answer is:

ACD

(not to be mixed up with ACDC)

So anyone remembers
what that stands for?

No?

Ah, maybe there that
person in the background.

I'll tell you!

Avoid

Cache

Defer

Avoid

Cache

You are here!

Defer



But why defer?

You might know
the problem.

You have build this
huge social network.

Doesn't matter
whats it called.

And you have all this
very complex content

tailored perfectly
to your users needs.

And it all works
really great.

There is just this
one little problem.

Very little problem.

Really not a big
deal

Your site takes 3.5
seconds to load.

That are:

3500 ms (!!!)

Well maybe that is
a big deal

Maybe that even is
a deal breaker

Maybe users are even
leaving your site for it.

So what do you do?

You need a
wonder!

A fast wonder!

So you transform
yourself into Gandalf!

(Other magicians had not been available at
the time of the making of this presentation)



and smoke a Pipe!

(Smoking is not endorsed in anyway; it is purely used for comical purposes)



A BIG PIPE!

(Smoking is not endorsed in anyway; it is purely used for comical purposes)

And you think:

What if I cheat?

What if I give the
user the impression

that the site is
really loading fast.

even though in
reality its not.

And that is:

Perceived
Performance!

And exactly what
Facebook does since ages

And that is what

Drupal & out of the
box

can now do, too!

In 8.1.0 big_pipe entered
core as an experimental
module.

And in 8.2.0 it will be:

Beta Stability!

So I'll now

get myself some
coffee while you

all are probably busy to
try big_pipe out right now.

No?

You want your
money back?

Thats not enough?

Okay, Lets continue
:)

What is streaming?

What is streaming?

- Available since HTTP/1.1 (!) was released in June 1999
- Content-Encoding: chunked
- Transfer-Coding: chunked

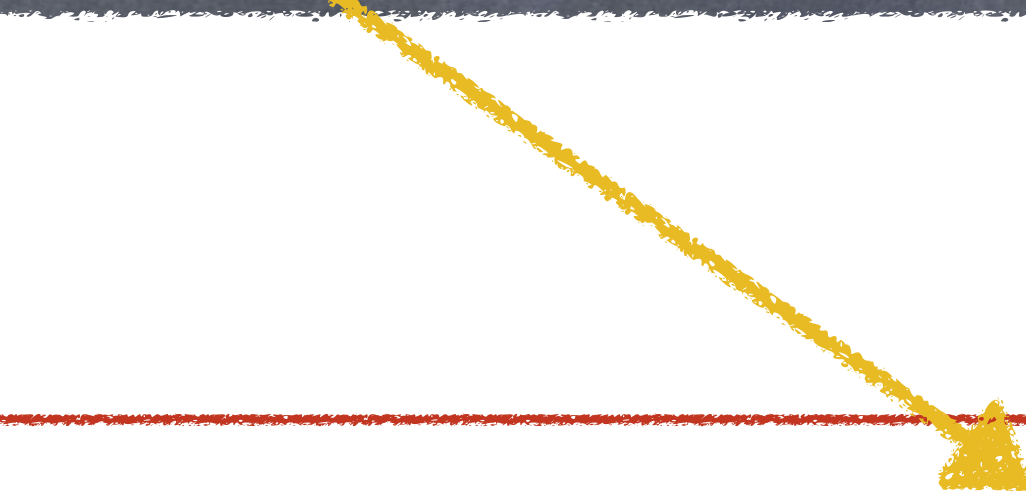
What is streaming?

```
<?php  
echo 'Hello';  
flush();  
  
sleep(10);  
  
echo 'World';
```


What is streaming?

- Previous PHP script: Easiest way to check
- Infrastructure Stack needs to be ready

This needs to be fast!

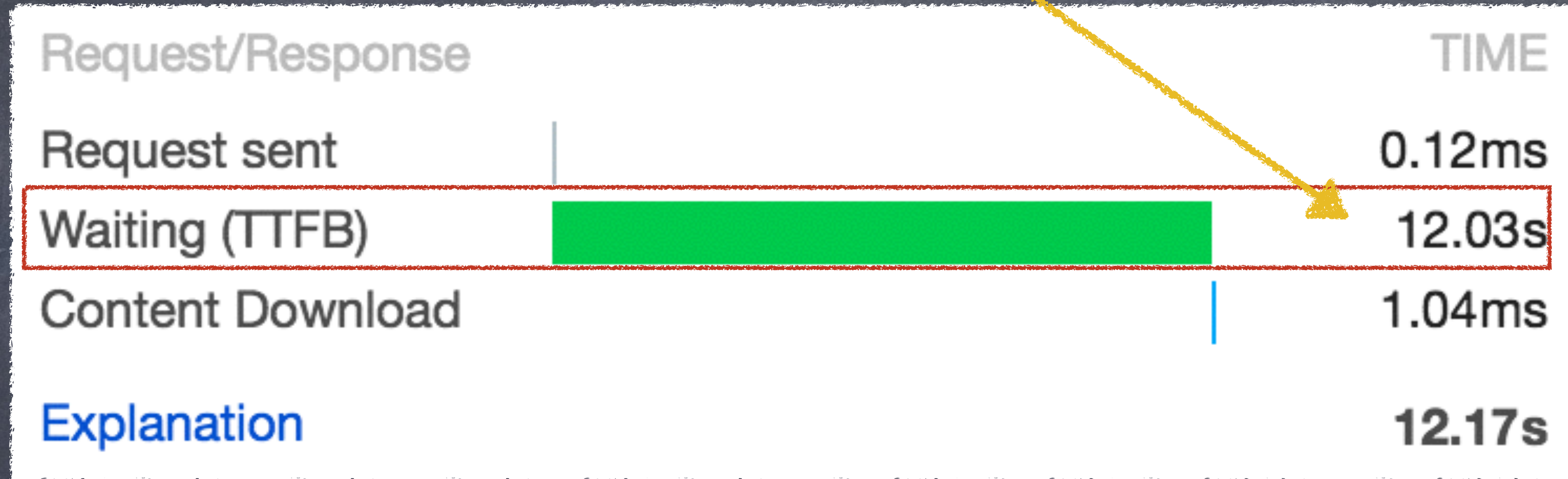


Request/Response	TIME
Request sent	0.07ms
Waiting (TTFB)	0.94ms
Content Download	10.00s
Explanation	10.02s

Ready for streaming!

Previous PHP script: Easiest way to check
Infrastructure Stack needs to be ready

This needs to be fast!



Request/Response	TIME
Request sent	0.12ms
Waiting (TTFB)	12.03s
Content Download	1.04ms
Explanation	12.17s

Not ready!

Previous PHP script: Easiest way to check
Infrastructure Stack needs to be ready

What is streaming?

```
<?php
header('Surrogate-Control: no-store, content="BigPipe/1.0"');
header('X-Accel-Buffering: no');

echo 'Hello';
flush();

sleep(10);

echo 'World';
```

The server needs to
know it uses streaming!

More configuration
is needed.

Server requirements

Server requirements for Streaming

- LAMP: out of the box
- NGINX: out of the box
- Varnish: 3 loc - needed
- Fastly: Out of the box

Now our server is
ready!

Let's get our code
ready, too!

Streaming in Drupal

7 + 8

Streaming in Drupal 8

- use `Symfony\Component\HttpFoundation\StreamedResponse`;

```
return new StreamedResponse(function() {  
    echo 'Hello';
```

```
    flush();  
    sleep(10);
```

```
    echo 'World';  
});
```

Streaming in Drupal 7

- hook_menu:

delivery callback => 'my_callback'

- menu callback:

return function({...});

- my_callback(\$result):

\$result();

If that is so easy ...

Why has no one
used that before?

Because it is not
practical.

Because you usually
don't want to ...

... just send some
"Hello World".

Because you have
the very practical ...

... business need that the
site looks the same ...

... and for the user
nothing changes.

Except that its
faster!

of course!

What to stream?

What to stream?

Nothing?

⇒ Default

What to stream?

Everything?

⇒ Very jumpy

What to stream?

What about every block?

=> Could work, but very empty page at start.

What to stream?

What about specially
configured blocks?

⇒ That works, but what
if its not a block?

What to stream?

What about specially
configured blocks?

⇒ That works, but what
if its not a block?

The solution is:

What to stream?

Dynamic Parts of
the Page!

What to stream?

Dynamic Parts of
the Page!

=> Uncacheable parts of the
page! (or too granular)

But how do you identify
the dynamic parts?

Find the dynamic
parts!

Placeholder it!

- Idea: Placeholder everything that is too dynamic.

Allow that the code can specify that this can be independently rendered.

How does it work?

Shopping Cart
{user} cached

Normal Block
{user.permissions}
cached

{user.permissions} and {url}
cached

How does it work?

Placeholder

Normal Block
{user.permissions}
cached

{user.permissions} and {url}
cached

Placeholder it!

- Put it in ['#attached']
['placeholders'] in a render array.

Placeholder it!

- D8 makes this very easy.
- Cache-Tags, Cache-Contexts and Max-Age

Placeholder it!

- So Drupal & for everything on the page knows:
 - how to invalidate it
 - what to vary it on
 - how long to cache it

Placeholder it!

- Things that are uncacheable:

max-age = 0

- Things that are too granular

cache-contexts = [user]

And now we can ...

... automatically
placeholder it!

Placeholder it!

- We can do that:
 - Bottom-Up approach (D8)
- instead of:
 - Top-Down approach (D7)

Lazy Builders

Lazy Builders

- Independently renderable
=> No outside dependencies!

Lazy Builders (I)

- Autoplaceholder dynamic content:

```
$build = [  
  '#lazy_builder' => ['MyClass::lazyBuild', 'node', 1],  
  '#cache' => [  
    'max-age' => 0,  
  ],  
];
```



Lazy Builders (II)

- Force a placeholder (except for POSTs):

```
$build = [  
  '#lazy_builder' => ['MyClass::lazyBuild', 'node', 1],  
  '#create_placeholder' => TRUE,  
];
```


Lazy Builders (III)

- Force a placeholder always:

```
$lb = [  
  '#lazy_builder' => ['MyClass::lazyBuild', 'node', 1],  
];
```

```
$build['#attached']['placeholders']['{my_unique_placeholder}'] = $lb;  
$build['#markup'] = '<div>{my_unique_placeholder}</div>';
```

Lazy Builder with Dependencies (III)

```
$node = \Drupal::routeMatch->getParameter('node');

$build =
  '#cache' => [
    'contexts' => 'route',
  ],
];

// This is a child item!
$build['lb'] = [
  '#lazy_builder' => ['MyClass::lazyBuild', 'node', $node->id()],
];
```

But we have all of
that!

And its still not
fast !!!

Now we split our
3.5 seconds up

2 seconds for the
content

and 1.5 seconds for the
streamed placeholders

So the UX is now
down to 2 s - at least

But why?

Remainder is
perfectly cacheable

Because there are no
dynamic parts anymore.

In Drupal 8 there is
Dynamic Page Cache

enabled out-of-
the-box.

So we enable it and

...

Like magic!

Down to 35 ms!

and 1.5 seconds of streaming.

Let's take a look at a
Big Pipe Demo!

Demo

The screenshot shows a web browser window with the address bar containing 'tres/node/1'. The page features a blue header with the 'tres' logo and navigation links for 'Home', 'My account', and 'Log out'. The main content area is divided into two columns. The left column contains two sections: 'Most listened to in the past week' with a list of three items (PigPipe, Pink Fluffy Unicorn, and Llama Rock), and 'Currently listening to...' with one item (Dries). The right column features a featured article titled 'The awesomest new band: PigPipe', submitted by 'root' on 09/17/2015. The article includes a painting of a pig in a suit and pipe, followed by two paragraphs of placeholder text.

The browser's address bar shows the URL `tres/node/1`. The page header includes the 'tres' logo and navigation links for 'Home', 'My account', and 'Log out'. The main content area is divided into two columns.

Home

Most listened to in the past week


1. PigPipe — Pink Ponies Pwn!
2. Pink Fluffy Unicorn — Rainbows are for losers
3. Llama Rock — KILL THE KITTENS!

Currently listening to...

Dries — Drupal song

The awesomest new band: PigPipe

Submitted by root on Thu, 09/17/2015 - 17:36

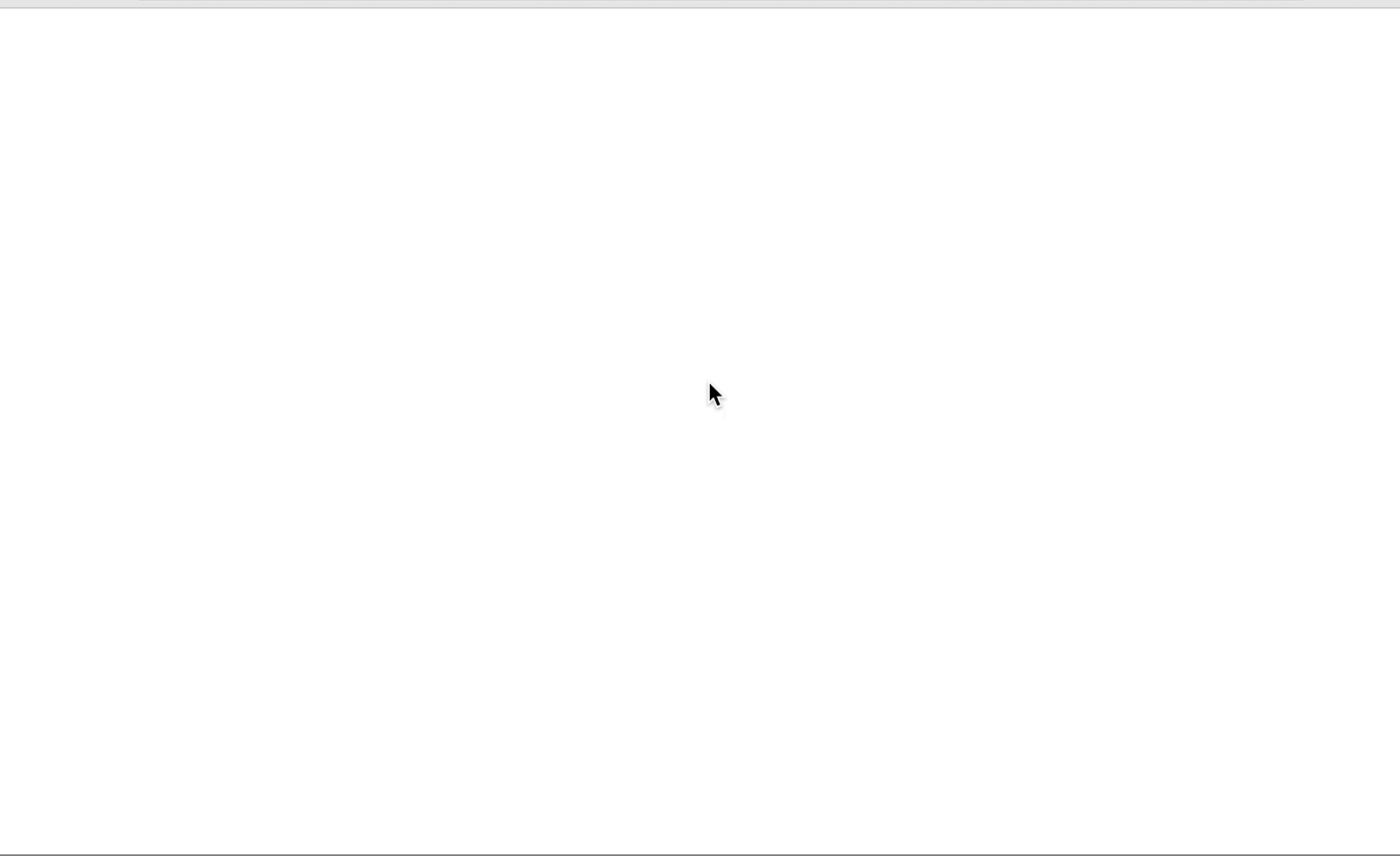


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur. Donec ut libero sed arcu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor. Pellentesque auctor nisi id magna consequat sagittis. Curabitur dapibus enim sit amet elit pharetra tincidunt feugiat nisl imperdiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros. Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.

Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae, sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit, id fringilla sem nunc vel mi. Nam dictum odio

about:blank x

← → ↻ about:blank ☆ S ≡



So if you want a (perceived)
fast site in Drupal 8

you just enable the big_pipe
experimental module and you are
done.

Maybe you need to declare
your custom dynamic parts.

But that's it.

There was once upon a
time,

a Site that had performance
problems ...

... they enabled BigPipe and
the problems have gone away.

So it works.

Ways of Streaming

Idea of placeholders
is not new.

Idea of auto-
placeholdering is.

Another way is: ESI

```
<!-- esi:include '/current-time.php' -->
```


But ESI is easy now,
too!

Thanks to Lazy Builders
+ Placeholders!

Just define a new
ESIPlaceholder Strategy.

Hash the placeholders
in the database.

Define a route: /
esi-fragment/[hash]

And you are done!

Or for AJAX:

Put the placeholders
in drupalSettings

And use JS to replace
the placeholders.

Difference of AJAX/ ESI + BIG PIPE

- ESI/AJAX:

- n requests
- n full bootstraps

11 requests every page load => can
kill your site performance

Difference of AJAX/ ESI + BIG PIPE

- BigPipe:

- 1 request
- Several inlined responses

⇒ way better for cold cache performance

Be creative and
combine!

Future: ESI BigPipe

- Response from the edge / CDN
- Stream the rest later
 - Support in Fastly: "Soon"
 - Varnish: "Patch for v3 available"

What about Drupal 7?

- Just prototype implementations
- But there is major work on core itself!

What about Drupal 7?

Want you to all open up:

[https://www.drupal.org/
node/2754245](https://www.drupal.org/node/2754245)

Review my patch
right now!

Because with reviews,
we can get that in!

Attachments Collectors
solve the "drupal_add_js"
problem.

Needs to be fixed
in Core.

Important base
work

I need your help!

- "Together we can bring back the important parts of the D8 caching system to Drupal 7 - including BigPipe."

How can you help?

- Contact me
- Send your developers
- Sponsor work (as a grant)
- Sprint on Friday

Round of Applause
for:

- Wim Leers

- Acquia

(sponsored Big Pipe
work)

Re

me)

ScreenFlow File Edit Mark Insert Font Actions View Window Help

test 1 | tres

tres/en/node/3

Manage Shortcuts root Edit

My account Log out

tres

Home language

test 1

Home

test 1

View Edit Delete Translate

Submitted by root on Mon, 01/04/2016 - 12:48
asdfsadf

No turbolinks

Add new comment

Subject

Comment *

Text format Basic HTML [About text formats](#)

Restricted HTML
 Allowed HTML tags: <a href hreflang> <cite> <blockquote cite> <code> <ul type> <ol start type> <dl> <dt> <dd> <h2 id> <h3 id> <h4 id> <h5 id> <h6 id>
 Lines and paragraphs break automatically.
 Web page addresses and email addresses turn into links automatically.

Save Preview

Questions ???

Follow me
on Twitter!

Thank you!

- @fabianfranz
- Fabianx
- fabian@tag1consulting.com

Come to the sprints
on Friday!

Please vote for my session!



DUBLIN
DRUPALCON

JOIN US FOR CONTRIBUTION SPRINTS

First Time Sprinter Workshop - 9:00-12:00 - Room Wicklow2A

Mentored Core Sprint - 9:00-18:00 - Wicklow Hall 2B

General Sprints - 9:00 - 18:00 - Wicklow Hall 2A



DUBLIN
DRUPALCON

WHAT DID YOU THINK?

Evaluate This Session

events.drupal.org/dublin2016/schedule

THANK YOU!

