Server-Sent Events
Example from practice

Aurelijus Banelis
Aurelijus Banelis
Software developer
aurelijus.banelis.lt
aurelijus@banelis.lt
Server-Sent Events
Example from practice
WHY
Problem/Context

WHAT
Short intro

HOW
In practice
WHY

Needed real time updates, process while reading
Wanted: Updates from other clients instantly
Problem known as Real time messaging
Options were

- Web sockets
- Auto refresh
- Server-Sent Events
Needed mostly pulling

New thing to try

Server-Sent Events
WHY
Needed real time updates, process while reading

WHAT
Short intro

HOW
In practice
WHAT

new EventSource(...) ... flush()
Easiest to implement in JavaScript and PHP
Easiest to implement in JavaScript and PHP
Easiest to implement in JavaScript and PHP.
```javascript
var source = new EventSource("/stream.php");
source.addEventListener('message', function(e) {
    document.body.innerHTML += e.data + '<br/>';
}, false);
```

```php
header('Content-type: text/event-stream');
for ($i = 0; $i < 10; $i++) {
    foreach (['VilniusPHP', 'VilniusJs'] as $event) {
        print 'data: ' . $event."\n\n"
        ob_flush(); flush();
    }
    sleep(1);
}
```
WHY Needed real-time updates, process while reading

WHAT HTTP-based protocol: EventSource + text/event-stream

HOW In practice
HOW

4 aspects, when using in practice
Reconnect / Keep-alive
Debugging / Buffering
Resource optimisation
Browser support
Reconnects, but...

<table>
<thead>
<tr>
<th>URL</th>
<th>Status</th>
<th>Domain</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET php-tes</td>
<td>200 OK</td>
<td>php-test.local:1380</td>
<td>846 B</td>
</tr>
<tr>
<td>GET stream.</td>
<td>200 OK</td>
<td>php-test.local:1380</td>
<td>0 B</td>
</tr>
<tr>
<td>GET stream.</td>
<td>200 OK</td>
<td>php-test.local:1380</td>
<td>0 B</td>
</tr>
<tr>
<td>GET stream.</td>
<td>200 OK</td>
<td>php-test.local:1380</td>
<td>0 B</td>
</tr>
<tr>
<td>GET stream.</td>
<td>504 Gateway Time-out</td>
<td>php-test.local:1380</td>
<td>183 B</td>
</tr>
</tbody>
</table>

Auto-reconnects if success

Fails if custom error
Reconnect: Solution 1/2

Keep-alive messages

print "\n\n";

"\n\n" is event separator
Catch error and restart

```javascript
function retryEventSource() {
  var source = new EventSource("/stream.php");
  source.addEventListener('error', function(e) {
    source.close();
    retryEventSource(); // Recursive call
  }, false);
}
retryEventSource();
```
Debugging / Buffering

aurelljus@aurelljus-a:~$ curl -k https://php-test.local:13443/stream.php

aurelljus@aurelljus-a:~$ curl -k https://php-test.local:13443/stream.php

No data because of bad formatting or buffering
Debugging / Buffering

No data because of bad formatting or buffering.
Debugging / Buffering
Configure for streaming

```plaintext
fastcgi_buffering off;
proxy_buffering off;

# If previous did not worked
fastcgi_cache off;
proxy_cache off;
ssl_buffer_size 1;
fastcgi_read_timeout 86400;
```

Some examples at: [https://gist.github.com/aurelijusb/19b30fb5795b32946bb538fa5dfc9bf8](https://gist.github.com/aurelijusb/19b30fb5795b32946bb538fa5dfc9bf8)
Easy to consume all resources

Many open connections

In screenshot data visualised by https://github.com/firehol/netdata
Need polyfills for IE

EventSource polyfill - http://www.w3.org/TR/eventsource/

Source: http://caniuse.com/#feat=eventsource
WHY
Needed real time updates, process while reading

WHAT
HTTP based protocol: EventSource + text/event-stream

HOW
Easy for simple cases
Like WebSockets
for real use cases
Conclusion
Change perspective
Change perspective

Request  ↓  Response

Write stream

Read stream
In the context of

| Offline-first | Single page app | Not blocking UI |

Change perspective

Request  
↓
Write stream  
↓
Read stream

Response
Thanks for listening

Best time for questions

Slides already at:
References

- https://github.com/firehol/netdata
- https://github.com/hseeberger/akka-sse
- https://github.com/Yaffle/EventSource
- https://es.auginte.com/

Full animations from presentation

- https://aurelijus.banelis.lt/prezentations/server-sent-events-2016/wanted.mp4
- https://aurelijus.banelis.lt/prezentations/server-sent-events-2016/sse-hello-world.mp4
- https://aurelijus.banelis.lt/prezentations/server-sent-events-2016/sse-buffered.mp4
- https://aurelijus.banelis.lt/prezentations/server-sent-events-2016/sse-real-time.mp4