Service Workers With Streams

Jake Archibald posted an article about combining streams and the cache API in a service worker to further improve rendering times I became curious and wanted to learn more about how it works.

> This approach makes the following assumptions
>
> 1. It works with resources from the same origin
> 2. We only want the content from certain areas to stream, the rest of the content can go through regular fetch (cache then network)
> 3. We have broken the start and end portions of our pages into separate include files
> 4. We precache the start and end of our streamed content

```
var stream = new ReadableStream({
  start(controller) {
  // Get promises for response objects for each page part
  // The start and end come from a cache
  var startFetch = caches.match('/page-start.inc');
  var endFetch = caches.match('/page-end.inc');
  '/page-start.inc'// The middle comes from the network, with a fallbackh
    .catch(() => caches.match('/page-offline-middle.inc'));

  function pushStream(stream) {
    // Get a lock on '/page-middle.inc'// Get a lock on the stream
    var reader = stream.getReader();

    return reader.read().then(function process(result) {
      if (result.done) return;
      // Push the value to the combined stream
      controller.enqueue(result.value);
      // Read more & process
      return reader.read().then(process);
    });
  }
  // Get the start response
  startFetch
    // Push its contents to the combined stream
    .then(response => pushStream(response.body))
    // Get the middle response
    .then(() => middleFetch)
    // Push its contents to the combined stream
    .then(response => pushStream(response.body))
    // Get the end response
    .then(() => endFetch)
    // Push its contents to the combined stream
    .then(response => pushStream(response.body))
    // Close our stream, we're done!
    .then(() => controller.close());
}
```

The problem with this example as written (and I understand that it's an example, not working code) is that the `middle-fetch` portion is locked

The easier way

https://developers.google.com/web/tools/workbox/reference-docs/latest/workbox.streams