Revamping A Web Service With A React Single page a

A what?

iframe: https://tekniikanmaailma.fi

iframe: https://www.youtube.com/embed/3Z9yK3sMDUU?rel=0&controls=0&showinfo=0&autoplay=1

WHY?

- MVP to see if headless WP suits the client's purposes
- New digital magazine
- Building foundations for the whole media group
- Adoption of modern software development practices
- The old theme driven site was looking, and feeling old

How?

- add_filter() & add_action()
- Few thousand lines of custom plugin code
  - URL resolver
  - API customizations
- composer-patches & patch-package for abandonware patches
- Continuous Integration
- nginx, memcached, Puppeteer, ACF to REST API, WP REST API Menus, WP Libre Form, WP_Query Route to REST API, React, CSS Modules, Jest, Redux, Redux-Saga, Kea, React Router, React Modal, React Helmet, styled-components, axios, html-react-parser, localForage


Problems
&
solutions

SEO & SOME

Serving Google and Facebook something other than "Loading..."

```
<!DOCTYPE html>
<html lang="fi">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link rel="icon" type="image/png" href='/assets/logos/favicon.ico'>
    <meta name="theme-color" content="#e4001e"/>
    <meta name="mobile-web-app-capable" content="yes"/>
    <link rel="manifest" href="/assets/manifest.json">
    <title>Tekniikan Maailma</title>

    <!-- Google Tag Manager -->
    <script>
      window.OM_GTM_ID = 'GTM-MTX2862';
      (function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
          new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
        j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
        'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
      })(window,document,'script','dataLayer', window.OM_GTM_ID);
    </script>
    <!-- End Google Tag Manager -->
  <link rel="preload" as="font" crossorigin="crossorigin" href="/98dd9c02671d1c420c2a37ba58992da7.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/0010ec444973fe8afe2f4842f4c13226.ttf">
<link rel="preload" as="font" crossorigin="crossorigin" href="/57a2aa7aa547865b5ae2e315b3bf6508.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/498e3cf531b5195f4980fae4dc9b2e30.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/428706863dcbc2795e9ac4d6fe9da4fd.ttf">
<link rel="preload" as="font" crossorigin="crossorigin" href="/a17af0892b40330f798b561486ce502a.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/bfae5cdc844ff6295b39b28c26990613.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/24dcc221a0b65338e30b409ccf708b00.ttf">
<link rel="preload" as="font" crossorigin="crossorigin" href="/7b8b8bdf456e682e453cbe7896c6a0ee.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/3f498a237808f0784b3e402ce18dc44f.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/9151a9d4d56d7cd492996b5312900c2d.ttf">
<link rel="preload" as="font" crossorigin="crossorigin" href="/fca2e05c00fa8913cbcc6cd9150ea44a.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/9acc01bf034e15f9992b083c212bec1.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/9e8376bb0aa91090fd5f21302fae8f29.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/3947603e4004a8d5d90e3544922c930.ttf">
<link rel="preload" as="font" crossorigin="crossorigin" href="/9ca1c851a8b2d6a4e03620290928 0e0.ttf">
<link rel="preload" as="font" crossorigin="crossorigin" href="/4b5097a75bcc2cd83f71df5652b8567.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/e24473a26d570f9dd12704d07598ffec.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/f39baa82f40e9799b1ba86781ec60a93.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/e935856f331ad512600a6468a20d2e1f.woff2">
<link rel="preload" as="font" crossorigin="crossorigin" href="/7066adfb9daf10d69c5e92127f6c0713.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/1f1ee471d1885c84f0dc4302597ba5cb.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/ec23b998ba38c4edbe817807b8d0f55d.woff2">
<link rel="preload" as="font" crossorigin="crossorigin" href="/711891544b4782266436a81152917d3e.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/2f8ad504372f82ea98d246bb8d9fb178.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/2a4099ed7753b7304110c3ea98a33ecd.woff2">
<link rel="preload" as="font" crossorigin="crossorigin" href="/182211f23e2aeeafa6ebfa3f12e3b5ce.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/8a06e381ad9c952dadf37f61bbb12ff5.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/65dab1a493f0b92114a3e3d2955f9e73.woff2">
<link rel="preload" as="font" crossorigin="crossorigin" href="/4bd1ec4fdc29819984140d496aacb4ec.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/3a22a8f4e669312128bad2ad60bb588a.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/d667997b8bafd5340f7b653a1e76739e.woff2">
<link rel="preload" as="font" crossorigin="crossorigin" href="/cf16a9194e250426b7b5b2c5d17dd840.eot">
<link rel="preload" as="font" crossorigin="crossorigin" href="/7ce269d3005fbb7840c10b5220fee797.woff">
<link rel="preload" as="font" crossorigin="crossorigin" href="/5748517d6bc311a1d417f2d991ba2c25.woff2">
<link rel="shortcut icon" href="/favicon.ico"><link href="/main.b9ba21e09520c2ba976b0d50c3a53848.css" rel="stylesheet"></head>
  <body>
  <!-- Google Tag Manager (noscript) -->
  <noscript>
    <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-MTX2862" height="0" width="0"
      style="display:none;visibility:hidden"></iframe>
  </noscript>
  <!-- End Google Tag Manager (noscript) -->
  <noscript>
    Javascript ei ole päällä selaimessasi, tai selaimesi ei tue javascriptiä. Ole hyvä ja laita javascript päälle selaimesi asetuksista.
  </noscript>
  <div id="app"></div>
  <script type="text/javascript" src="/vendor.1efd718dd38a32b350a9.js"></script><script type="text/javascript" src="/main.7c2e5b0f9a4e28bcab4d.js"></script></body>
</html>
```

The new version isn't server rendered. Crawlers that do not execute JavaScript or wait for async actions to complete get a pretty crude page.

That won't do. Puppeteer to the rescue!

Puppeteer is a Node library for Google Chrome. It allows us to simulate user interaction and run code.

The basic idea is to run a Node server that runs Puppeteer, and route to that server from nginx.

It can be done for all users, users that aren't logged in, or just bots, bots being the easiest to implement.

We run things a bit differently for Puppeteer. We append ?headless=true to the URL and various tweaks around the application happen.

How does the Puppeteer server know when the page is ready?

```
1. const untilAppSaysReady = page.waitForFunction('window.READY_TO_RENDER === true')
2.
3. // Disable image loading to make rendering a lot faster,
```

```
1. [actions.setRendered]: function * ({ payload }) {
2.   if (isHeadless()) {
3.     Array.from(document.querySelectorAll('script')).forEach(script => {
4.       script.remove()
5.     })
6.
7.     window.READY_TO_RENDER = true
8.     console.log('READY_TO_RENDER:', window.READY_TO_RENDER)
9.   }
10. },
11.
12. // elsewhere
13.
14. render () {
15.   return <AsyncComponent whenReady={() => actions.setRendered(true)} />
16. }
```

Now we have full content and meta tags.

But a lot of CSS was missing.

```
/**
 * WTF?

 * In short, styled-components provides API to export styles to HTML string,
 * but the official API (ServerStyleSheet) works only on the server side.
 * Because our SSR is being executed in browser env (eg. client)
 * we have to abuse the secret API to be able to access the styles and then
 * append them to DOM to be served/cached.
 *
 * https://github.com/styled-components/styled-components/issues/1487
```

Prerendered site

(not public so not shown)

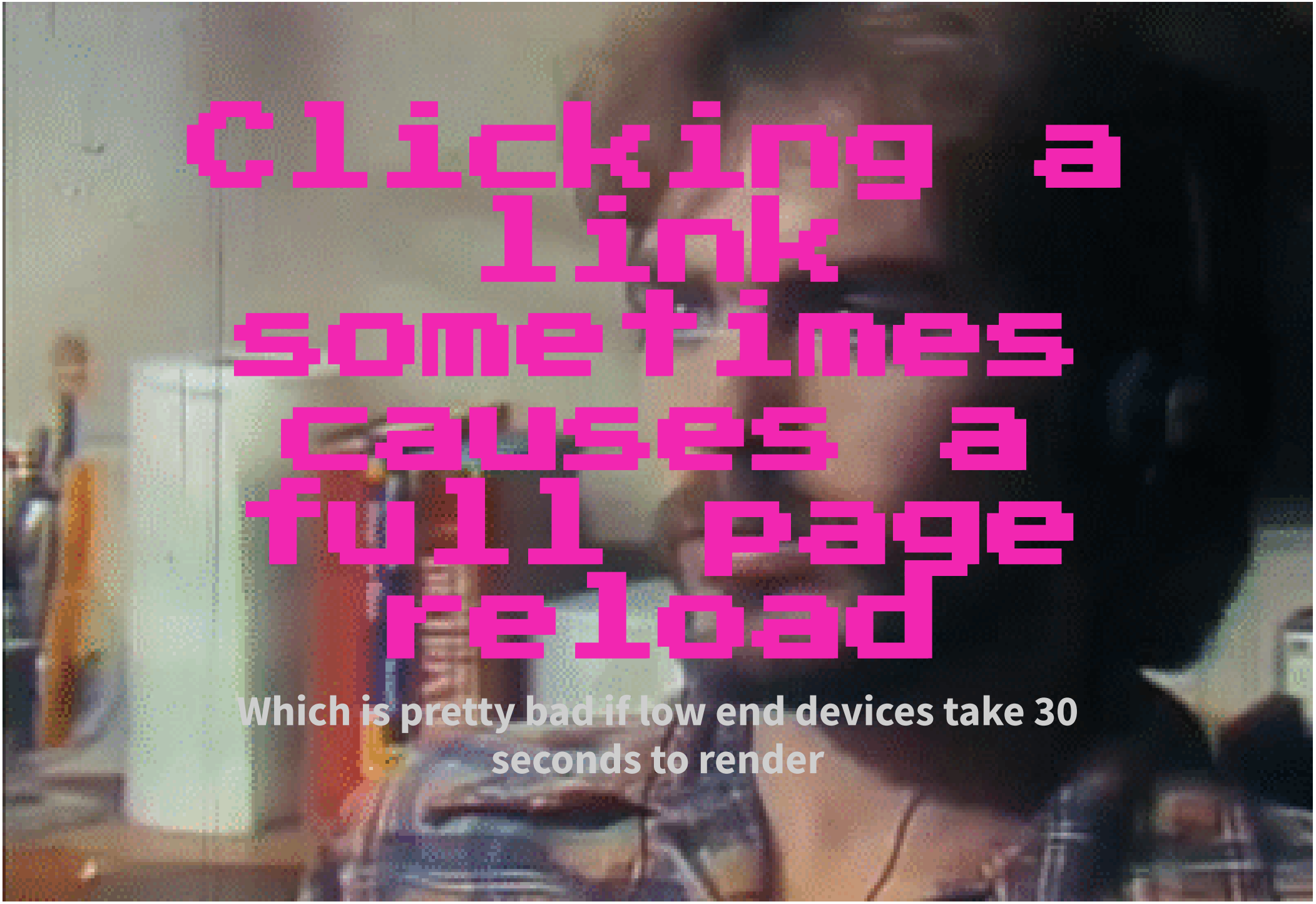
Building the new without breaking the old

The old themebased site had to remain as-is, but we had to make changes to the backend in order to build the new site.

This was surprisingly easy.

```
1. <?php
```

```
 8.     'meta_query' => [
 9.       'relation' => 'AND',
10.       'publish_date' => [
11.         'key' => 'publish_date',
12.         'type' => 'DATETIME',
13.         'compare' => '<=',
14.         'value' => date('Y-m-d H:i:s', $ts),
15.       ],
16.     ],
17.     'order' => 'DESC',
18.     'orderby' => 'meta_value',
19.   ];
20. }
21.
22. function change_digimag_url($menu) {
23.   foreach ($menu["items"] as $k => $v) {
24.     if ($v["title"] === 'Digilehti') {
25.       $terms = get_terms(array_merge([
26.         'hide_empty' => false,
27.         'taxonomy' => 'printmag',
28.         'number' => 1,
29.       ], digimag_query_params()));
30.
31.       if (!empty($terms)) {
32.         $menu["items"][$k] = array_merge($v, [
33.           "object_id" => $terms[0]->term_id,
34.           "url" => get_term_link($terms[0]),
35.         ]);
36.       }
37.     }
38.   }
39.
40.   return $menu;
41. }
42.
43. function order_printmag_by_pubdate($query) {
44.   return array_merge($query, digimag_query_params(strtotime("+ 6 week")));
45. }
46.
47. function remove_unwanted_values($response, $term, $request) {
48.   // Clear `layout` because it's unprocessed and bloated
```


Clicking a link sometimes causes a full page reload
Which is pretty bad if low end devices take 30 seconds to render

A few months ago, we started noticing that sometimes when you clicked a link, it caused a full page reload, instead of seamlessly transitioning.

We tried debugging it, but reproducing it was very hard. The problem got worse and worse.

Finally, our tech lead isolated the problem to Google Tag Manager.

Fixing the problem was easy, locating the problem was the hard part.

3

If strange things happens with Google Tag Manager link click tracking the first thing you should do is to go to your click trigger setup and see if the "wait for tags" and "check validation" checkboxes are enabled (as they are by default) and uncheck them.

They are not necessary in an SPA and (as evidenced) potentially harmful; "wait for tags" adds a delay so that other tags have time to fire before a the link directs a user away from the page (which does not happen within SPAs in any case), "check validation" tests if the link target is a valid URI (which within an SPA it probably isn't by GTMs standards - for example links starting with a hash are not valid etc.).

share improve this answer

answered Apr 24 '16 at 10:30

Eike Pierstorff

24.8k • 3 • 23 • 45

add a comment

We had a similar problem with Frosmo as well.

Cache invalidation

Definitely harder than naming things

Let's do the easy ones first.

- CSS, JS, SVGs etc: Use md5 hash in filename
- ???
- ???

WordPress Transients

- Better class-based API
  - Predictable transient names:
    `omf_om/v1_digimag_issue_md5($params)`
    `$prefix_$route_$path_md5($params)`
  - ~~Prefetch: populate transients just when they expire~~
  - ~~A list of transients, with meta~~
  - ~~Inception-mode (transient within transient)~~
- Cacheproxy endpoint for 3rd party or "native" endpoints
- Automagical transients for all custom API endpoints
- Ugly maintenance class to clean it all up

Unfortunately, because we could only use memcached, the "smart" solution didn't run very well in production.

Memcacheds maximum value size is 1MB, the list of transients quickly got to 4MB, so I tried storing it it wp_options.

PHP could handle the 4MB value without breaking a sweat, but MySQL choked on it in a traffic spike.

So production crashed. A few times.

After that we simplified a bit, and left out the list, prefetch and Inception-mode.

System like this could work wonderfully with Redis, or any LRU-cache with a sensible value size limit.

We ran it over a weekend, and the list size didn't go higher than 4MB, so you're going to have to have a LOT more transients before PHP performance becomes an issue.

from old and new code, and may or may not run. Probably does, but be aware of the mistakes in them.

I will publish a toolkit plugin containing similar

```php
<?php
namespace OM\TM;

/**
 * Better API for Transients API.
 *
 * Totally not threadsafe.
 */
class Transientify {
  const DEFAULT_EXPIRY = 30;
  const BYPASS_KEY = "WHATEVER_YOU_WANT";

  public $expiry;
  public $bypass;
  public $key;

  public function __construct(string $key = null, array $options = []) {
    if (is_null($key)) {
      throw new \Exception('Transient key is required');
    }

    $user = wp_get_current_user();
    $accessLevel = ($user && $user->ID)
      ? \AsteikkoAuth::userGetAccessLevel($user)
      : 0;


    $this->key = "omf_{$accessLevel}_{$key}";

    $this->parseOptions($options);
  }

  public static function getPrefetchList() {
    // Value possibly grows high
    return get_option('omf_transient_prefetch', []);
  }

```

```
1. <?php
2. namespace OM\TM\REST\Routes;
3.
4. class Category extends \OM\TM\RestRoute {
5.   public function __construct() {
6.     parent::__construct("om/v1", "category");
7.
```

```
1. <?php
2. namespace OM\TM;
3.
4. /**
5.  * Base class for creating routes to the API. Abstract so it can't be instantiated.
6.  */
7. abstract class RestRoute extends \WP_REST_Controller {
8.   protected $namespace;
9.   protected $route;
10.   public $routes = [];
11.
12.   public function __construct(string $namespace, string $route) {
13.     $this->setNamespace($namespace);
14.     $this->setRouteBase($route);
15.
16.     // $this->register_routes(); // Call in extending class constructor after createEndpoint
calls
17.   }
18.
19.   public function setNamespace(string $namespace) {
20.     $this->namespace = $namespace;
21.   }
22.
23.   public function setRouteBase(string $route) {
24.     $this->route = $route;
25.   }
```

```
 1. <?php
 2. namespace OM\TM;
 3.
 4. class TransientCleaner {
 5.   public $transientList = [];
 6.
 7.   public function __construct() {
 8.     add_action("save_post", [$this, "maybeCleanTransients"], 10, 3);
 9.
10.     $this->transientList = $this->parseTransientListItems();
11.   }
12.
13.   /**
14.    * Function used with array_filter. Resulting list of transients will be deleted.
15.    * TODO: Finer control. If a post is published with category 3, don't delete transients
16.    * from other categories.
17.    */
18.   public function transientMatchesClearCondition($listItem = []) {
19.     $parsedKey = $listItem["parsedKey"];
20.     $parts = $parsedKey["parts"];
21.     $routeNames = [ // If transient $routeName matches, remove transient.
22.       "frontpage",
23.       "stickies",
24.       "sidebar",
25.       "wp/v2/categories", // cacheproxy
26.       "wp/v2/tags", // cacheproxy
```

localForage

Pretty much every network request made to WP is cached using localForage, which uses IndexedDB

This is great, because it makes the application faster over time, and allows offline use

However, if we were to make changes to the data, the application would break because the cached data isn't compatible with the components.

Storing all requests can also take a lot of disk space.

```
1. // version.js
2. export const current = 'new-feature-broke-stuff'
3.
4. const sampleVersion = {
5.   version: '20270101',
6.   invalidateDiskStorage: false, // Will clear ALL disk saved data
7.   invalidateStores: ['requestCache', 'userData'], // Will clear stores with corresponding names
8. }
9.
10. // Don't go backwards in versions.
11. // If something hits the fan, rollback using git revert,
12. // and add a new version. The version compare logic will fail if you neglect this.
13. export const versions = [
14.   {
15.     version: 'new-feature-broke-stuff',
16.     invalidateStores: ['requestCache'],
17.   },
18.   {
19.     version: '20190225',
20.     invalidateStores: true,
21.   },
22.   // ...
23. ].reverse()
24. export const getVersion = key => versions.find(({ version }) => version === key)
25. export const currentVersion = getVersion(current)
26.
27. export const getVersionIndex = version => versions.findIndex(
28.   ({ version: v }) => v === version
```

By hooking that up to data retrieval functions, the data can be stored on users device.

But that's going to keep growing forever.

```
136. }
137.
138.
139. // CacheDriver.js
140. /**
141.  * LRU cache. Starts cleaning itself when it grows too big.
142.  */
143. class CacheDriver extends DiskStorage {
144.   constructor (store, options = {}) {
145.     const { orderMax = 1000 } = options
146.     super(store)
147.
148.     this.reservedKeys = ['order']
149.     this.orderMax = orderMax
150.
151.     this.store.keys().then(keys => {
152.       if (keys.length > this.orderMax) {
153.         this.clean('10%')
154.       }
155.     }).catch(e => console.log('Unable to check store size', e))
156.
157.     this.clean = this.clean.bind(this)
158.
159.     if (isDevelopment()) {
160.       window.CacheDriver = this
161.     }
162.   }
163.
164.   async getOrder () {
165.     try {
166.       const order = await this.store.getItem('order')
167.
168.       if (order) {
169.         return order
170.       }
171.
172.       return []
173.     } catch (e) {
174.       console.log("Can't get order, private browsing?", e)
175.       return []
176.     }
```

Dealing with fragmented data

I regret not using TypeScript

There's many different formats for the seemingly same data.


```
{ "wp:featuredmedia": [
  {
    "id": 856572,
    "date": "2018-06-18T12:00:38",
    "slug": "img_20180502_100125759",
    "type": "attachment",
    "link":
"http://wpdev.local:8209/img_20180502_100125759/",
    "title": {
      "rendered": "IMG_20180502_100125759"
    },
    "author": 1,
    "acf": [

    ],
    "caption": {
      "rendered": ""
    },
    "alt_text": "",
    "media_type": "image",
    "mime_type": "image/jpeg",
    "media_details": {
      "width": 3456,
      "height": 4608,
      "file": "2018/06/img_20180502_100125759.jpg",
      "sizes": {
        "thumbnail": {
          "file": "img_20180502_100125759-150x150.jpg",
          "width": 150,
```

```
{ "kuva": {
  "ID": 827816,
  "id": 827816,
  "title": "Tasty hamburger with french fries and beer on
wooden table",
  "filename": "istock-638349896.jpg",
  "filesize": 1077278,
  "url": "https://vanha.tekniikanmaailma.fi/wp-
content/uploads/2018/02/istock-638349896.jpg",
  "link": "https://vanha.tekniikanmaailma.fi/miehet-
kuluttavat-paivassa-1-000-kaloria-enemman-kuin-luulevat-
kertoo-tutkimus-naisilla-arvio-heittaa-lahes-yhta-
paljon/tasty-hamburger-with-french-fries-and-beer-on-
wooden-table/",
  "alt": "",
  "author": "666",
  "description": "pikaruoka, roskaruoka, lihava",
  "caption": "Tulevaisuudessa hampurilaispihvisi voi olla
peräisin maatilan sijaan laboratoriosta.",
  "name": "tasty-hamburger-with-french-fries-and-beer-on-
wooden-table",
  "status": "inherit",
  "uploaded_to": 827796,
  "date": "2018-02-19 11:50:01",
  "modified": "2019-02-19 15:27:50",
  "menu_order": 0,
  "mime_type": "image/jpeg",
  "type": "image",
  "subtype": "jpeg",
```

Pretty much everything provided by airesvsg/acf-to-rest-api is in a different format.

We want to use the same components regardless of differences in data.

```
 1. class Article extends Component {
 2.   render() {
 3.     const { featuredImage, acfImage, content } = this.props
 4.
 5.     return (
 6.       <article>
 7.         <Image data={featuredImage} />
 8.         <HTML>{content}</HTML>
 9.         <Image data={acfImage} />
10.       </article>
11.     )
12.   }
13. }
14.
15.
16. export default class ImageModel {
17.   id
18.   title
19.   caption
20.   credit
21.   copyright
22.   mediaType
23.   mimeType
24.   url
```

Authenticating frontend users with WordPress

Cookies?

First, we wanted to use cookie based authentication, because it's easy.

rmccue commented on 5 Mar 2018 • edited

Member

> I'm building a single page app. I'm going to need the nonces as I'm going to have to create posts from the application frontend and modify existing ones. How should I get the nonce? Other authentication methods such as JWT or OAuth aren't going to work for this case.

Fundamentally, cookie authentication is only designed for plugins and themes, and should not be used outside of that. I'd recommend using OAuth personally, but with pre-authorised keys; that is, the user never has to manually authorise the key, so it's essentially invisible to them apart from two redirects.

For single-page apps hosted on the domain that really need to use cookies, I can think of two solutions: either load your single page app via WordPress as a sort of dummy theme (this is how I usually use it, in combination with something like react-wp-scripts), or load the nonce in via a cross-site-secured method.

Other options

JWT is nice in theory, but the there are some gotchas, like token(s) not revoking on password change. We used it briefly.

OAuth

Setting it up was pretty easy with a commercial plugin, although it's documentation was horrid at the time.

https://wp-oauth.com/

Allowed Grant Types

Choosing the correct grant type for your client is important. For security reasons, a single grant type should be used per client. To learn more about which grant type you will need, please visit https://wp-oauth.com/documentation/overview/supported-grant-types/.

Authorization Code ☐
Allows authorization code grant type for this client. This includes the implicit method.

Implicit ☐
Allows implicit method. "Authorization Code" **must** be enabled.

User Credentials ☑
Allows the client to use user credentials to authorize.

Client Credentials ☐
Client can use the client ID and Client Secret to authorize.

Refresh Token ☑
Allows the client to request a refresh token.

Client Information

Client Name

React frontend

Redirect URI

Client I

REDACTED

Client Secret

REDACTED

Advanced Options

Client Credential Assigned User

The "client credential" grant types does not have a user id assigned to it making it hard for an application to perform protected endpoints. The client will then have the same privileges as the selected user.

--- No User ---

Client Scope(s)

*Scopes can be assigned to restrict scopes. This value will also act as the default scope for this client. If you leave this field blank, the default scope will be **"basic"** and the client will have access to all available scopes. If you have multiple scopes, please separate with a single space.*

basic

The docs suggest that we do the following, although "User Credentials" kinda implies client-side usage.

https://wp-oauth.com/docs/general/grant-types/user-credentials/

```
POST
/?oauth=token HTTP/1.1

Headers
Authorization: Basic {base64d_client_id_and_client_secret}
Content-Type: application/x-www-form-urlencoded

Body Request
grant_type=password
&username={users-username}
&password={users-password}
```


what could go wrong?
let's do something else instead

```
1. public function authenticate($request) {
2.   // removed variables extracted from ENV and request to fit this code sample
3.   $authHeader = "Basic " . base64_encode("$clientId:$clientSecret");
4.   $response = wp_remote_post(get_site_url() . '/oauth/token', [
```

(probably removed due to time constraints)

Changing tablet orientation broke "Use offline" feature

(probably removed due to time constraints)

If an user that's viewing the application with an iPad started the application in landscape, and later rotated to portrait, trying to download anything failed silently.

Starting the application in portrait and downloading something worked flawlessly. Only transition from landscape to portrait broke the process.

(probably removed due to time constraints)

This was a serious bug, but how a simple button causes that kind of problems?

Behind the button lies ~~dragons~~ complex logic.

An action (button click) changes Redux state, and triggers a saga which dispatches more actions and starts another saga at the end.


(probably removed due to time constraints)

(probably removed due to time constraints)

Several hours later, I finally figured it out.

```
diff --git a/app/src/containers/App.js b/app/src/containers/App.js
index 77284146..c9c3030e 100644
--- a/app/src/containers/App.js
+++ b/app/src/containers/App.js
@@ -9,6 +9,7 @@ import CookieConsent from 'react-cookie-consent'

 import { is as errorIs, getType, RenderedError, logErrorBoundaryError } from '../lib/errors'
 import application from '../kea/application'
+import downloadLogic from '../kea/downloads'
 import headerLogic from '../kea/header'
```

(probably removed due to time constraints)

By connecting the logic store to a component that will not unmount, the problem disappears.

Integration testing

Can we abuse browsers even more?

We were already using Jest for the few unit tests that we have, so we tried using Puppeteer with Jest.

It works nicely after the third refactor.

```
1. /**
2.  * Function that receives a test running function as a parameter
3.  * and returns a new function that returns a promise that resolves when
4.  * the test function is done.
5.  *
6.  * Async makes things a bit complicated.
7.  */
8. const createTest = fn => () => new Promise(done => fn(global.__BROWSER__, done))
9.
10. const loginTest = createTest((browser) => {
11.   // try/catch/finally blocks omitted to save space
12.   describe('Login', () => {
13.     const url = getFrontendHost()
```

iframe: https://www.youtube.com/embed/sw88Uw8WA8U?rel=0&controls=0&showinfo=0&autoplay=1

Thank you!

My other stuff

- Project: WP Libre Form
- Project: vincit/wordpress
- Project: vincit/wordpress-theme-base
- Twitter: @k1sul1
- GitHub: @k1sul1

Rabbit holes

- wordpress.stackexchange.com/q/295471
- developers.google.com/web/tools/puppeteer/articles/ssr
- stackoverflow.com/q/36769478

Bonus content: CodeSlide navigation in this presentation

The previous slides contained some code. They were made with jamiebuilds/spectacle-code-slide.

It however, did not support my remote controller.

```
 1. import CodeSlideComponent from 'spectacle-code-slide'
 2.
 3. const extractNumberFromString = x => {
 4.   const matches = x.match(/\d.*/gm)
 5.   const value = matches ? matches[0] : null
 6.
 7.   return value ? parseInt(value, 10) : null
 8. }
 9.
10. export default class Presentation extends React.Component {
11.   constructor(props) {
12.     super(props)
13.
14.     this.handleShittyRemote = this.handleShittyRemote.bind(this)
15.     this.guessTheSlide = this.guessTheSlide.bind(this)
16.
17.     this.slideNumber = extractNumberFromString(window.location.hash) // Including it in state
causes rerenders
18.
19.     this.state = {
20.       verticalMode: false,
21.       codeSlideComponent: CodeSlideComponent,
22.     }
23.   }
24.
25.   // Hack: listen to remote keys and move up and down in code slides
26.   handleShittyRemote(e) {
27.     const { slideNumber: number, state } = this
28.     const { verticalMode} = state
29.     const lsKey = `code-slide:${number}`
30.     const currentRange = parseInt(localStorage.getItem(lsKey), 10)
31.
32.
33.     // The "fullscreen" key on the remote
34.     if (e.code === 'Period') {
35.       this.setState({
36.         verticalMode: !verticalMode
```