

CACHING LARGE NAVIGATION MENUS

Benji Fisher

October 4, 2019

**BEFORE WE GET
STARTED**

PUSH THE BIG, RED BUTTON!

CLICK THE DAMNED BUTTON!

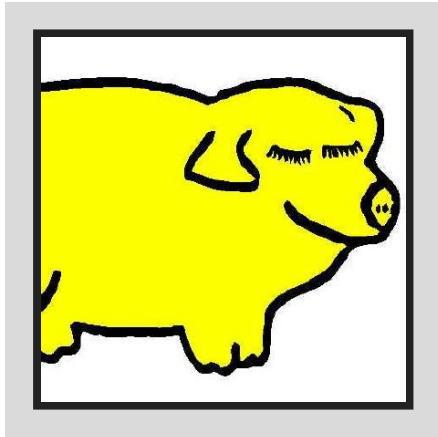


BLOG POST

<https://www.isovera.com/blog/caching-large-navigation-menus-drupal>

INTRODUCTION

ABOUT ME



- Benji Fisher
- [@benjifisher](#) on d.o since 2010-01
- [@benjifisher](#) on GitHub
- [@benjifisher](#) on GitLab
- [@benji17fisher](#) on Twitter

HOOK 42



HOOK42

**BOOK NAVIGATION
NEEDS TO BE
CACHED**

PEGA COMMUNITY DOC PAGE

The screenshot displays a Pega Community Doc page. On the left is a navigation menu with the following items:

- Pega Platform
 - Getting started for experienced development teams >
 - Developing an application >
 - Best practices for application development >
 - Designing a case >
 - Managing application data >
 - Creating the user experience >
 - Channels and interfaces >
 - Styling your application >
 - Forms and fields >
 - Design templates >
 - Sections >
 - Sections - Completing the Create or Save As form >
 - Section form - Completing the Design tab
 - Settings tab of the Section form
 - Section form - Completing the Parameters tab

The main content area shows the current state as 'Published' with a 'Change to Published' button and a warning icon. A blue button labeled 'Update Moderation State' is also present. The article title is 'Settings tab of the Section form'. The content includes a sub-section 'Standard section' and a list of bullet points:

- Allow Section to be edited at run time – When you create a section from a template, select this option to allow run-time editing of the section.
- Specify usage in page template regions – For a section that is in a @baseclass, data- class, or data-portal class, you also can specify the section's usage, such as Work area, Header, or Footer.

Use this option to specify that a list of work area gadgets are available for the section during run-time editing.

A vertical 'feedback' button is located on the right side of the page.

Book navigation is nested several levels

NUMBERS

- 4472 pages in the book
- 2.7 MB rendered (twice) for each page
- 40-50 sec initial load
- 6-9 sec load after caching

After I did this work, the cached page loads in 2-3 sec.

STRATEGY

1. Use the same navigation on all pages
2. Set active trail with javascript
3. Cache the navigation once per book

INTERNAL VS. EXTERNAL CACHE

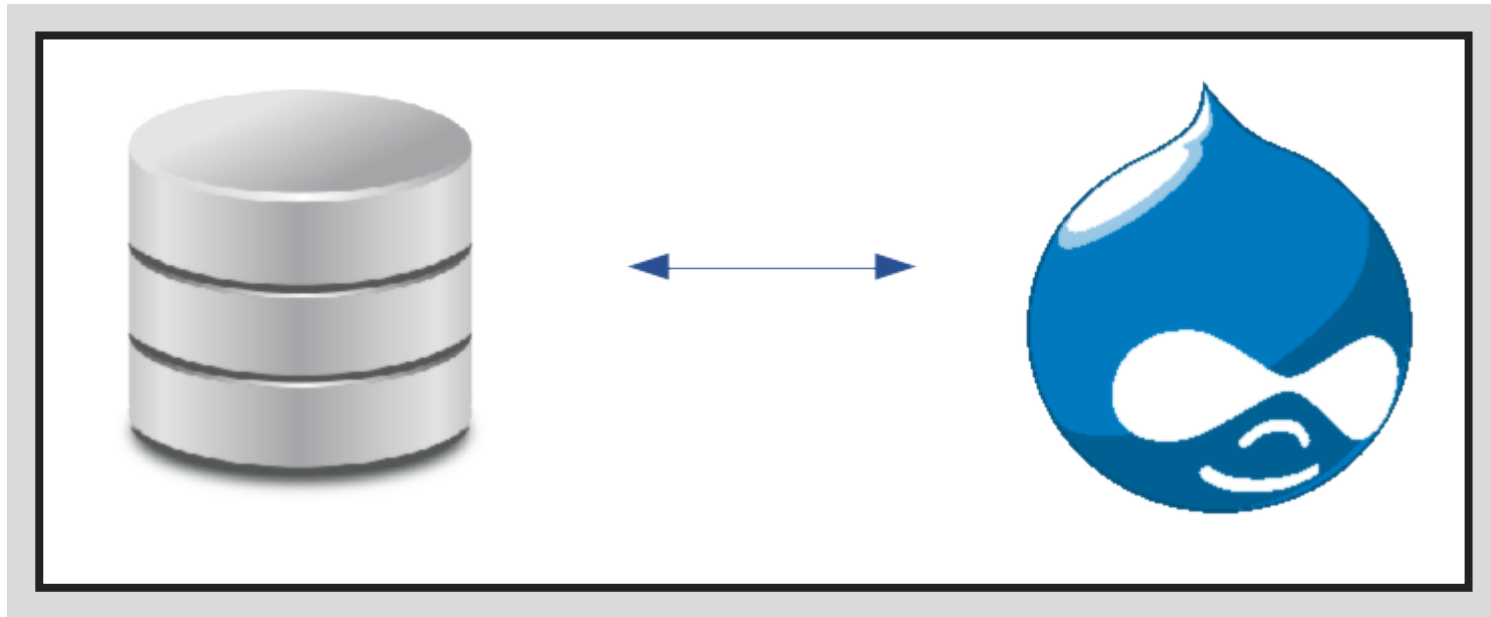
DRUPAL FTW

Caching is one of the strengths of Drupal 8.



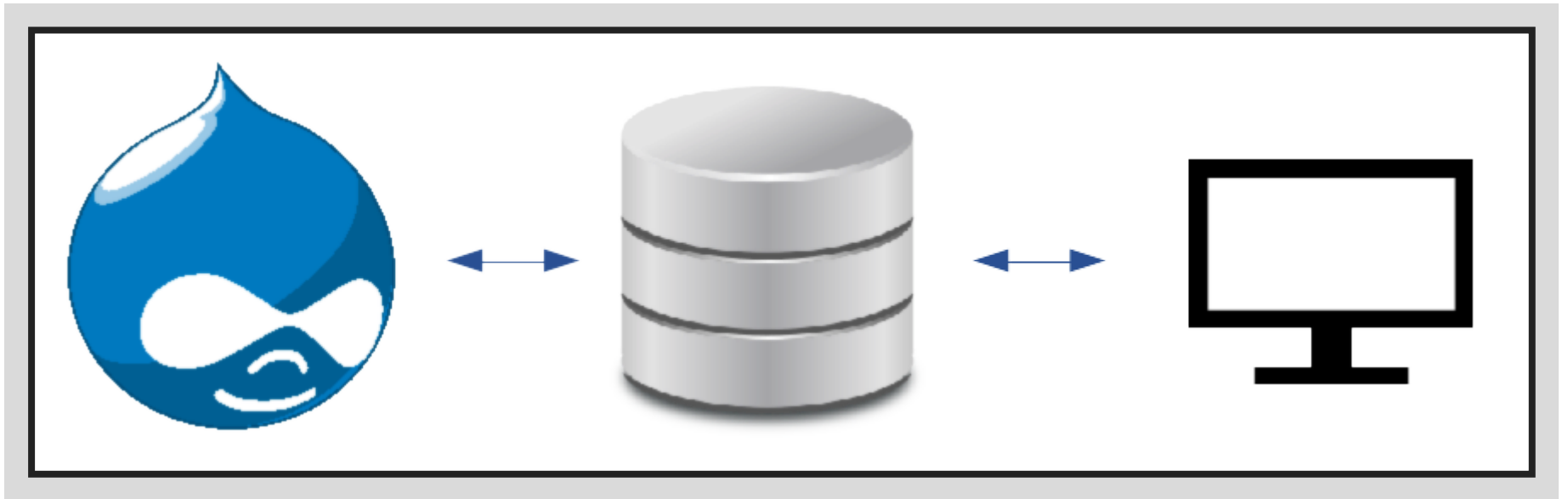
INTERNAL CACHE

Drupal stores bits and pieces in database or memcache or redis.



EXTERNAL CACHE

Whole pages are stored in the database, Varnish, or a CDN.



THE HARD PART

For both internal and external caches, the hard part is knowing when to clear the cache.

TWO HARD THINGS

1. Naming things
2. Off-by-one errors
3. Cache invalidation

TWIG TEMPLATE

TWIG TEMPLATE (SIMPLIFIED)

```
{% if tree %}
  <nav class="c-book-nav" role="navigation" aria-
labelledby="book-label-{{ book_id }}">
    <a href="{{ book_url }}">
      {{ top_book_title }}
    </a>
    {{ tree }}
  </nav>
{% endif %}
```

TWIG TEMPLATE (FULL)

```
{% if tree %}
  <nav class="c-book-nav" role="navigation" aria-
labelledby="book-label-{{ book_id }}">
    {% if top_book_title %}
      {% if not top_book_empty %}
        <a href="{{ book_url }}">
          {% endif %}
          {{ top_book_title }}
          {% if not top_book_empty %}
            </a>
          {% endif %}
        {% endif %}
      {{ tree }}
    </nav>
  {% endif %}
```

DRUPAL CODE

HOOK NODE VIEW

```
function pdn_book_node_view(array &$build, NodeInterface
    $node, EntityViewDisplayInterface $display, $view_mode) {
    if ($view_mode != 'full') {
        return;
    }
    if (empty($node->book['bid']) || !empty($node->in_preview))
    {
        return;
    }
    $book_id = $node->book['bid'];
    $book_node = Node::load($book_id);
    if (!$book_node->access()) {
        return;
    }
}
```

HOOK THEME

```
function pdn_book_theme($existing, $type, $theme, $path) {  
  return [  
    'book_nav' => [  
      'variables' => [  
        'book_id' => 0,  
      ],  
    ],  
  ];  
}
```

PREPROCESS FUNCTION

```
function template_preprocess_book_nav(&$variables) {  
  /** @var \Drupal\book\BookManager */  
  $book_manager = \Drupal::service('book.manager');  
  
  // Get the nested array (tree) of menu links.  
  $book_tree = $book_manager  
    ->bookTreeAllData($variables['book_id']);  
  // Generate a render array from the tree of links.  
  $tree_output = $book_manager  
    ->bookTreeOutput(array_shift($book_tree)['below']);  
  
  $variables['tree'] = $tree_output;  
  $variables['book_url'] = \Drupal::url(  
    'entity.node.canonical',
```


JAVASCRIPT

```
Drupal.behaviors.bookNavExpand = {
  attach: function attach(context) {
    var bookNav = $('<div class="c-book-nav"></div>', context);
    $('<a href="' + context.location.pathname + '">',
bookNav)
      .addClass('active')
      .parentsUntil(bookNav, '.c-book-nav--list-
expandable')
      .addClass('c-book-nav--list-expanded')
      .children('a')
      .addClass('active');
    $('<div class="c-book-nav--list-expanded"></div> > .c-book-nav--list',
context)
      .once('bookNavExpandInit')
      .css('display', 'block');
```

HOOK NODE VIEW (REVIEW)

HOOK NODE VIEW

```
$build['book_nav'] = [  
  '#theme' => 'book_nav',  
  '#book_id' => $book_id,  
  '#weight' => 100,  
  '#cache' => [  
    'keys' => ['pdn_book_nav', $book_id],  
    'contexts' => ['languages'],  
    'tags' => ["node:$book_id"],  
    'max-age' => Cache::PERMANENT,  
  ],  
];
```

CACHE KEYS

```
$build['book_nav'] = [  
  '#cache' => [  
    'keys' => ['pdn_book_nav', $book_id],  
  ],  
];
```

- A unique string to identify “our” cache
- The book ID

This is how we cache once per book.

Without cache keys, any cache data will bubble up.

CACHE CONTEXTS

```
$build['book_nav'] = [  
  '#cache' => [  
    'contexts' => ['languages'],  
  ],  
];
```

If the book is viewed in another language, then the link text will change.

Maybe also the link URLs.

This site is not (yet) multilingual.

CACHE TAGS

```
$build[ 'book_nav' ] = [  
  '#cache' => [  
    'tags' => [ "node:$book_id" ],  
  ],  
];
```

These are saved in the database.

When `node/$book_id` is updated, delete from the cache.

At page level, cache tags are sent in HTTP headers.
Varnish/CDN invalidates based on cache tags.

CACHE MAX AGE

```
$build['book_nav'] = [  
  '#cache' => [  
    'max-age' => Cache::PERMANENT,  
  ],  
];
```

Keep the cached version until I say to clear it.

PEEK AT THE DATABASE

THE `cache_render` TABLE

```
mysql> DESCRIBE cache_render;
```

Field	Type	Null	Key	Default	Extra
<code>cid</code>	<code>varchar(255)</code>	NO	PRI		
<code>data</code>	<code>longblob</code>	YES		NULL	
<code>expire</code>	<code>int(11)</code>	NO	MUL	0	
<code>created</code>	<code>decimal(14,3)</code>	NO	MUL	0.000	
<code>serialized</code>	<code>smallint(6)</code>	NO		0	
<code>tags</code>	<code>longtext</code>	YES		NULL	
<code>checksum</code>	<code>varchar(255)</code>	NO		NULL	

7 rows in set (0.01 sec)

QUERY

```
mysql> SELECT cid, expire, created, tags, checksum
FROM cache_render
WHERE cid LIKE 'pdn_book%'
LIMIT 0,1\G
***** 1. row *****
      cid: pdn_book_nav:704369:[languages]=en:[theme]=pegawww
[user.permissions]=4f64d6e20026c96e963d91bab0192f9824e8cb2e91
      expire: -1
      created: 1543638198.782
      tags: config:system.book.704369 node:704369 rendered
      checksum: 12
1 row in set (0.00 sec)
```

CACHE ID

```
mysql> SELECT cid
FROM cache_render
WHERE cid LIKE 'pdn_book%'
LIMIT 0,1\G
cid:
  pdn_book_nav:
  704369:
  [languages]=en:
  [theme]=pegaww_theme:

  [user.permissions]=4f64d6e20026c96e963d91bab0192f9824e8cb2e9:

1 row in set (0.00 sec)
```

- We specified `pdn_book_nav` in the cache keys
- The book ID also comes from cache keys
- `languages` comes from cache contexts
- `theme` and `permissions` ... see below

CACHE MAX AGE

```
mysql> SELECT expire, created
FROM cache_render
WHERE cid LIKE 'pdn_book%'
LIMIT 0,1\G
```

```
***** 1. row *****
  expire: -1
  created: 1543638198.782
1 row in set (0.00 sec)
```

CACHE TAGS

```
mysql> SELECT tags
FROM cache_render
WHERE cid LIKE 'pdn_book%'
LIMIT 0,1\G
***** 1. row *****
tags:
  config:system.book.704369
  node:704369
  rendered
1 row in set (0.00 sec)
```

PERMISSIONS HASH

Where do permissions come from?

See `sites/default/services.yml`:

```
parameters:
  renderer.config:
    # Renderer required cache contexts:
    #
    # The Renderer will automatically associate these cache
    # contexts with every render array, hence varying every
    # render array by these cache contexts.
    #
    # @default ['languages:language_interface', 'theme',
    'user.permissions']
    required_cache_contexts:
      - 'languages:language_interface'
      - 'theme'
      - 'user.permissions'
```

QUESTIONS