



Photo by Willian Justen de Vasconcellos on Unsplash

“There are only two hard things in Computer Science: cache invalidation and naming things.”

– Phil Karlton

Agenda

- The cache stack — it's not just Drupal
- Drupal caching basics — OK, just Drupal
- Cache invalidation — how does it happen?
- Cache headers — seeing what happened
- Caching gotchas — what might happen
- Tips, tricks and workarounds — make things happen

The cache stack

- MySQL / PHP

The cache stack

- MySQL / PHP
- Redis, Memcache

The cache stack

- MySQL / PHP
- Redis, Memcache
- **Drupal**

The cache stack

- MySQL / PHP
- Redis, Memcache
- Drupal
- **Varnish**

The cache stack

- MySQL / PHP
- Redis, Memcache
- Drupal
- Varnish
- **Content Delivery Network (CDN)**
 - Akamai, Fastly, CloudFlare, etc.

Drupal caching basics

Drupal caching basics

Drupal caching basics

Drupal caching basics

IMPORTANT SAFETY TIP

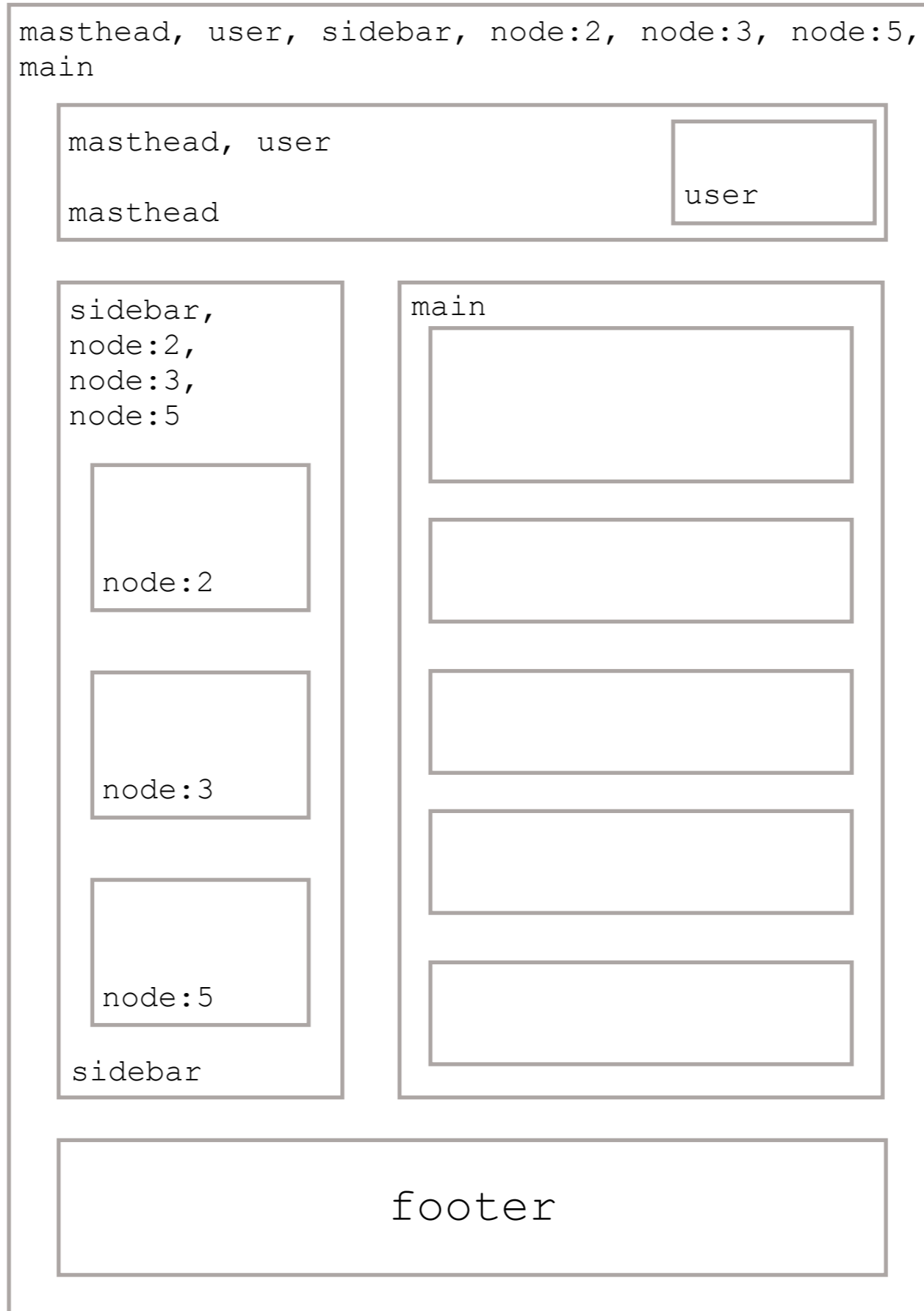
Tags and contexts are arrays.
As the page is being built, tags
complete their parents.
Page contains all the tags and
invalidate the page cache and
If a component doesn't specify
cache is invalidated.

Cacheability Metadata Bubbles Up!



Photo by Sharon McCutcheon on Unsplash

Bubbling in action!



Drupal caching basics

Drupal caching basics


```
'#cache' => [  
  'keys' => ['entity_view', 'node', $node->id()],  
  'contexts' => ['languages'],  
  'tags' => $node->getCacheTags(),  
  'max-age' => Cache::PERMANENT,  
];
```


}

```
$response->addCacheTags ($object->getCacheTags ( ) ) ;  
$response->addCacheContexts ( [ ' language ' ] ) ;
```

}

`Entity::invalidateTagsOnSave()`

- handles normal tags like `node:342786`
- does not handle custom tags

```
Entity::invalidateTagsOnSave()
```

cache_tags.invalidator service

```
$tax_tag = 'taxonomy_term:' . $tid;  
$node_tag = 'node:' . $nid;  
cache_tags.invalidator::invalidateTags([$tax_tag,  
$node_tag]);
```

```
// or, if you can't inject the service  
Cache::invalidateTags([$tax_tag, $node_tag]);
```


Cache invalidation

Cache invalidation

Use the Memcache or Redis modules.

- Store cached data in memory

Views and lists

Big Pipe

Parents and contexts are arrays
As the page is being built, tags are components of their parents
Page contains all the tags and
invalidate the page cache and

If a component doesn't specify
cache is invalidated.



arge

age

n a

Poor cacheability

```
$items['personalized'] = [  
  '#markup' => "Hello $this->account->getDisplayName()",  
  '#cache' => [  
    'cache_tags' => $this->account->getCacheTags()  
  ]  
];
```

Autoplaceholding!

```
$items['personalized'] = [  
  '#lazy_builder' => [  
    'user.toolbar_link_builder:renderDisplayName', []  
  ],  
  '#create_placeholder' => TRUE,  
];
```

```
public function renderDisplayName() {  
  return [  
    '#markup' => "Hello $this->account->getDisplayName()",  
  ]  
}
```

Plan out your caching strategy in advance.

- Am I doing something expensive that's worth caching?
 - If it's in a render array, give it a cache key. If it's an object, make it a cacheable dependency.
- Does this thing vary by permissions, URL, language, etc.?
 - Set appropriate cache contexts for the item.
- What causes this thing to become outdated?
 - Set cache tags on the item.
- When do I need to refresh content?
 - If in doubt, invalidate tags yourself.

Turn it off per <https://www.drupal.org/node/2598914>

Disable twig cache

```
cache: false
```

Test with caching on!





Resources

- <https://www.drupal.org/docs/8/api/cache-api/cache-api>
- <https://webdev.iac.gatech.edu/blog/drupal-8-caching-for-dummies>
- <https://www.lullabot.com/articles/early-rendering-a-lesson-in-debugging-drupal-8>
- <https://pantheon.io/blog/why-we-recommend-redis-drupal-or-wordpress-caching-backend>
- `RenderExampleController.php`