Did you clear the cache?

A Contextual Tour of Rendering Efficiently

WHO ARE WE?

Stephen Lucero

Director of Engineering & DevOps

in /in/srlucero

/u/slucero

A true developer at heart, Stephen brings eight years of Drupal experience to his role as the Director of Engineering and Developer Operations at Forum One. When faced with a challenge or a task, he enjoys finding an elegant solution to solve it and enjoys it even more when he has the opportunity to automate the task in some way.

Throughout his career he's had the privilege of working with a lot of talented individuals who excel in their own varied disciplines. This has provided a lot of influence to embrace a strong multidisciplinary perspective toward projects to achieve the most complete end product.

By embracing the goal of encouraging team members to explore beyond their familiar responsibilities and learn from each other, Stephen has seen significant success both personally and with coworkers to adopt a cross-functional approach toward project development. This has allowed teams to work more cohesively for more opportunity to learn and grow during the build of a more complete and satisfying end result.

FORUM ONE

About Forum One

We’ve been at this for 23 years.

We’ve partnered with more than 1,000 organizations and government agencies.

We’ve helped them better reach their mission on over 2,000 projects.

We are experts in digital strategy, creative, technology, data, and user experience.

We create experiences that make an impact.

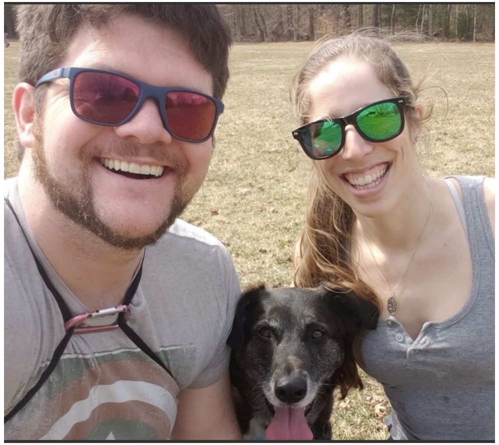
Rob Powell

Drupal Developer

in u/robert-powell-boston

/u/robpowell

- 10 years engineering experience
- 3+ year in Drupal
- When I am not programming or learning more about Drupal, I am heading to the trail head with my family.



About Mediacurrent

Who We Are

Mediacurrent is a full-service digital agency that implements world class open source software development, strategy, and design to achieve defined goals for enterprise organizations seeking a better return on investment.

Overview

- Cache - What is it and why do we need it?
- D8’s Cache API’s special sauce - cacheable metadata
- Dashboard caching exploration
- Caching tips and tricks
- Common gotchas, modules, and resources to save the day

❓ Why Cache?

CACHING
HISTORY
memegenerator

CACHING
CACHING IS EVERYWHERE

Cache Metadata

Max-age

How long a piece of cached data “lives” for.

Cache tags

Granular control to invalidate specific cached entries.

Cache contexts

For variations, i.e. content dependencies on the request details.

SCAMALYTICS DASHBOARDS

Spam ‘em & Scam ‘em

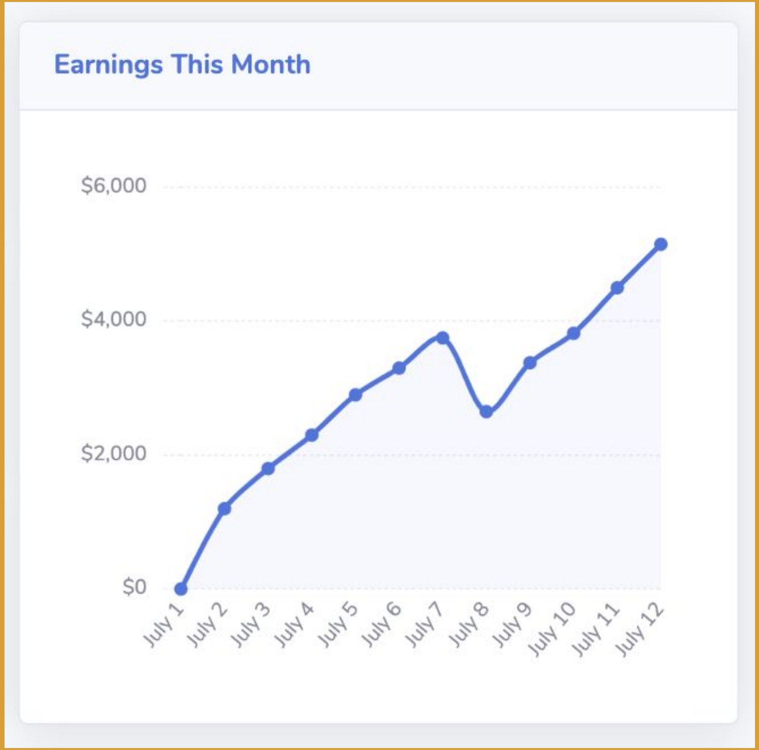
MAX-AGE

Max-Age

- Defaults to `Cache::PERMANENT`
- Defines the maximum length of time a cache may be considered valid and used
- Primary method for time-based expiration
- Used to be the primary method for invalidation
- Much less commonly needed with more advanced invalidation strategies

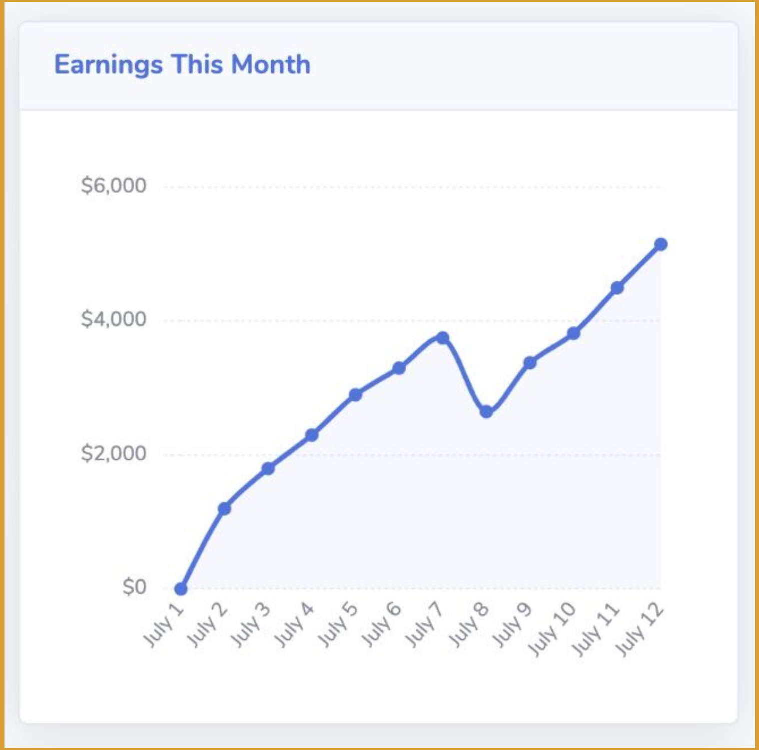
Timed Expiration

- Get a daily update for deposited funds
- Request through an API to our sketchy bank
- API response only updates once per day
- Don't need a new API request every time



Timed Expiration

```
$build = [
  '#cache' => [
    // Cache for 3 hours at a time.
    // 60 seconds * 60 minutes * 3 hours
    ‘max-age’ => 10800,
  ],
];
```



Earnings This Month

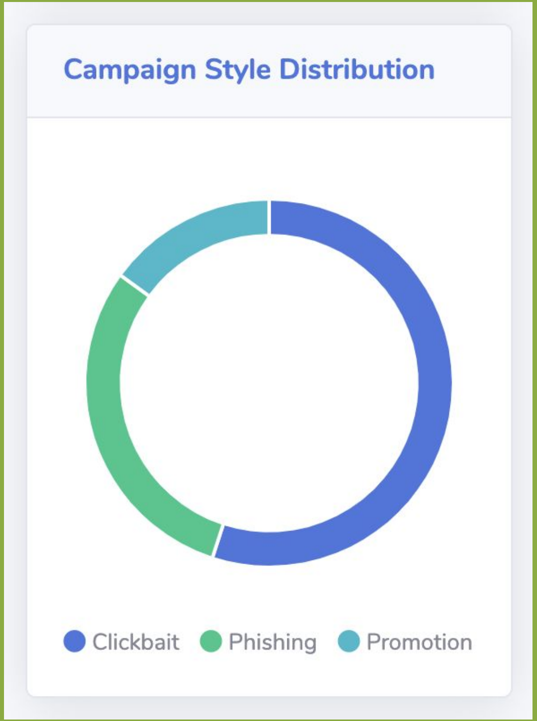
CACHE TAGS

Cache Tags

- Describes dependencies for the content being cached
- Alerts Drupal when a cache record should be invalidated.
- Cache entries tend to have numerous tags per entry
- Commonly structured as a concatenation of entry type and identifier
    - `node:5`
    - `user:3`
    - `config:system.performance`
- Special list tags are also available: `{entity_type}_list`
    - Invalidates when any entity of that type is
        - Created
        - Updated
        - Deleted

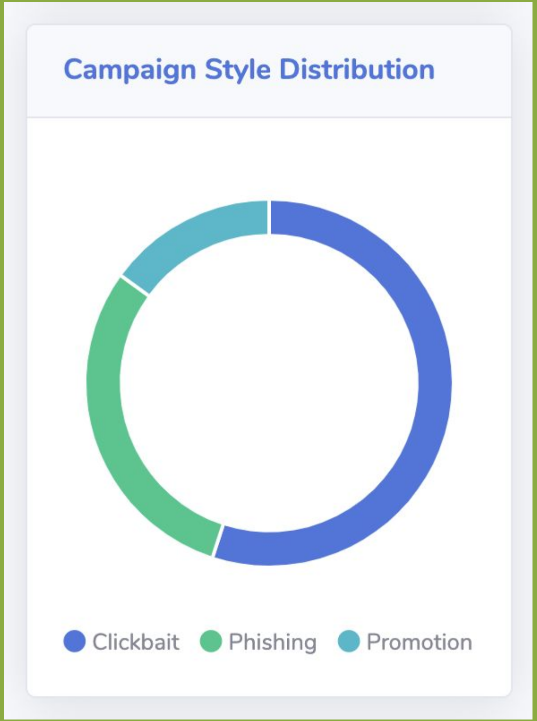
Node Types Widget

- Lists percentage of all Nodes by content type
  - □ Each campaign style is a content type
- Changes when Nodes are:
  - □ Created
  - □ Deleted
  - □ Updated


Campaign Style Distribution

Node Types Widget

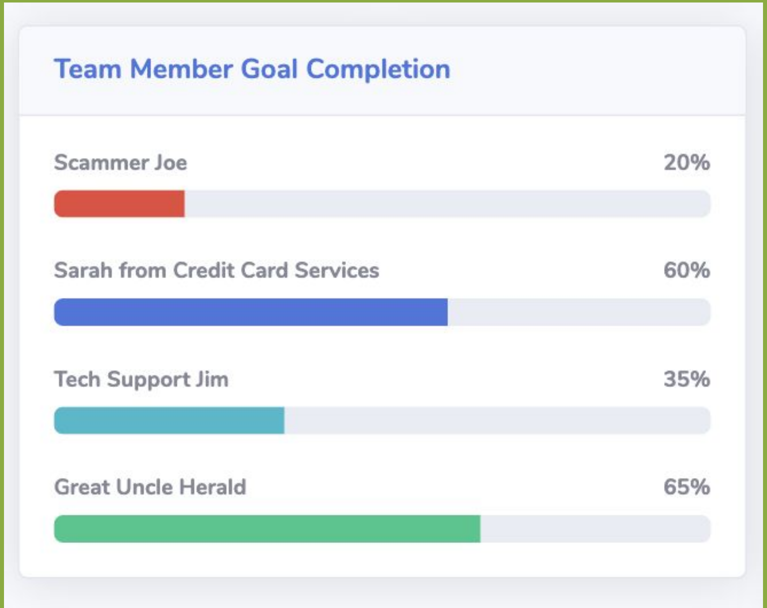
```
$build = [
  '#cache' => [
    'tags' => [
      // Lists of nodes use this special
      // tag to invalidate whenever a
      // node is changed.
      'node_list',
    ],
  ],
];
```



Pro Tip: Target specific bundles with the Handy Cache Tags module

Team Progress Report

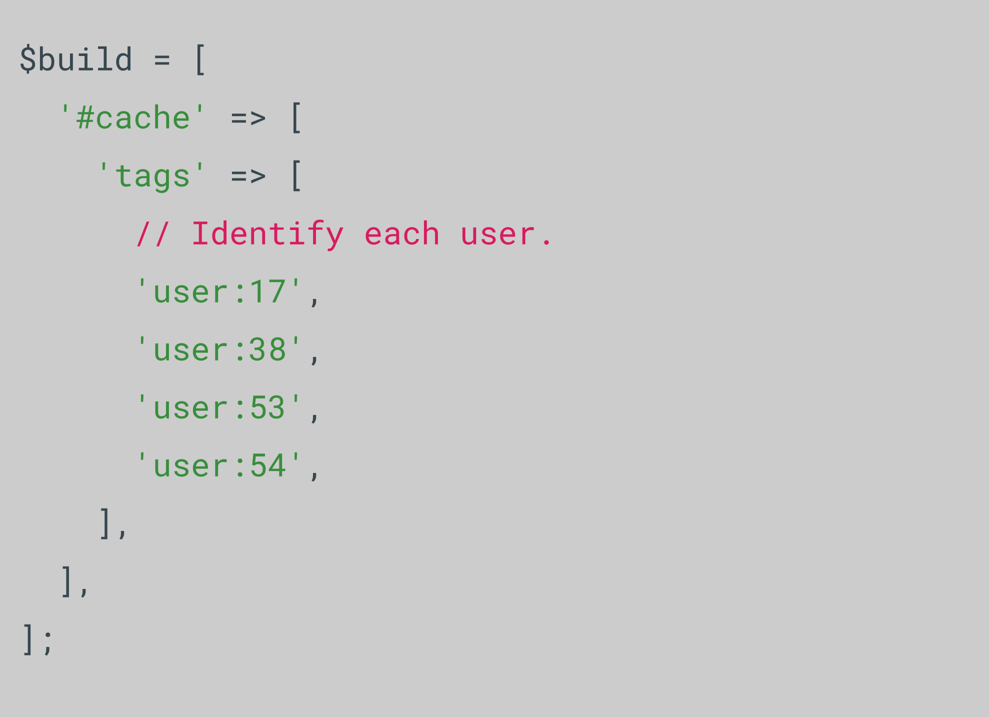
- Tracks completion percentage from user profiles
- Changes when **specific** users are updated



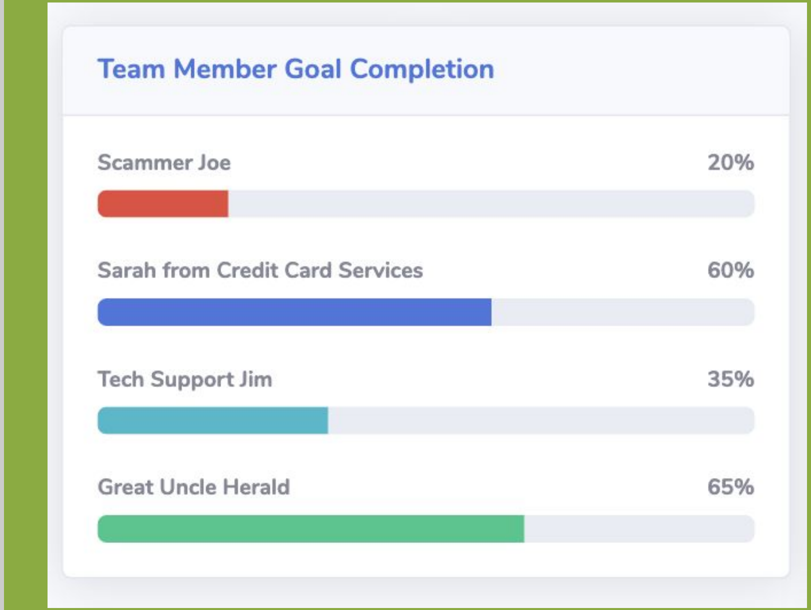
Team Member Goal Completion

Scammer Joe	20%
Sarah from Credit Card Services	60%
Tech Support Jim	35%
Great Uncle Herald	65%

Team Progress Report



```
$build = [
  '#cache' => [
    'tags' => [
      // Identify each user.
      'user:17',
      'user:38',
      'user:53',
      'user:54',
    ],
  ],
];
```



Team Member Goal Completion

Scammer Joe 20%

Sarah from Credit Card Services 60%

Tech Support Jim 35%

Great Uncle Herald 65%

CACHE CONTEXTS

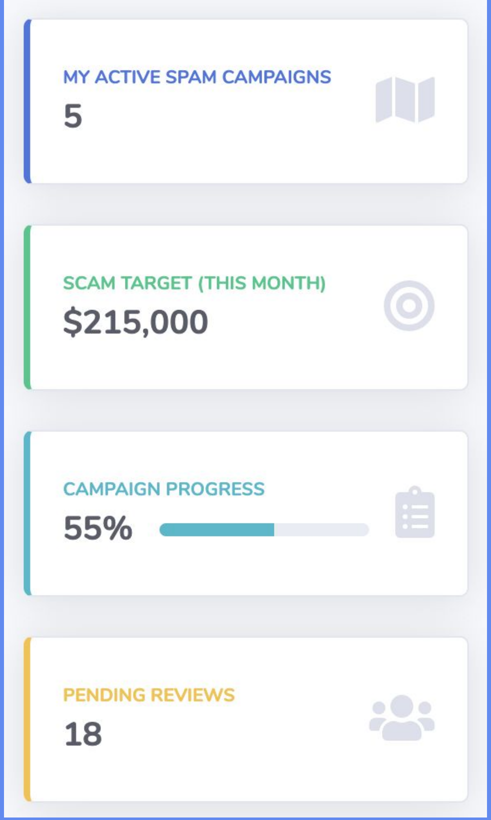
Cache Contexts

- A variation of cache
  - Most notably:
    - Users / theme / route
- Hierarchical in nature
  - User
    - Is_super_user
    - Node_grants
      - :operation
    - .permissions
    - .roles
      - :role

Pro Tip: To find all the caching contexts check for services tagged “name: cache.context”.

Per-User Reports

- Unique details depending on the viewer
- Changes based on the current user



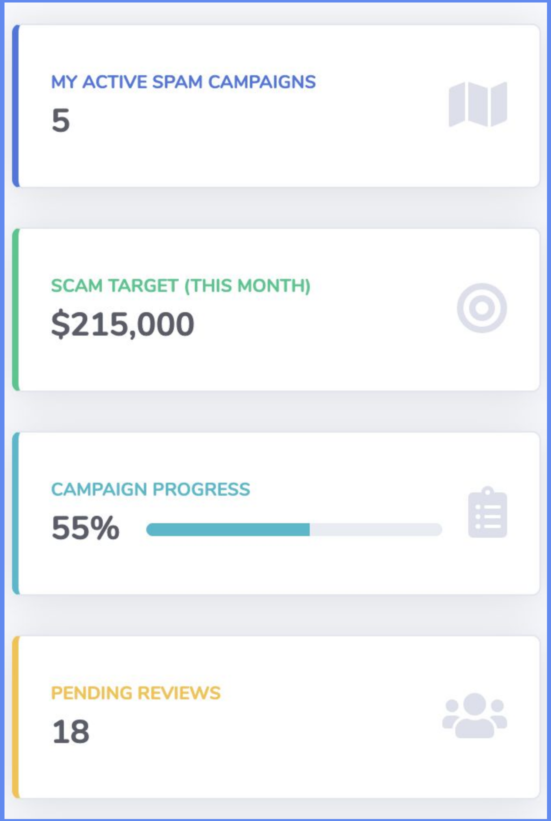
MY ACTIVE SPAM CAMPAIGNS
5

SCAM TARGET (THIS MONTH)
$215,000

CAMPAIGN PROGRESS
55%

PENDING REVIEWS
18

Per-User Reports



MY ACTIVE SPAM CAMPAIGNS

5

SCAM TARGET (THIS MONTH)

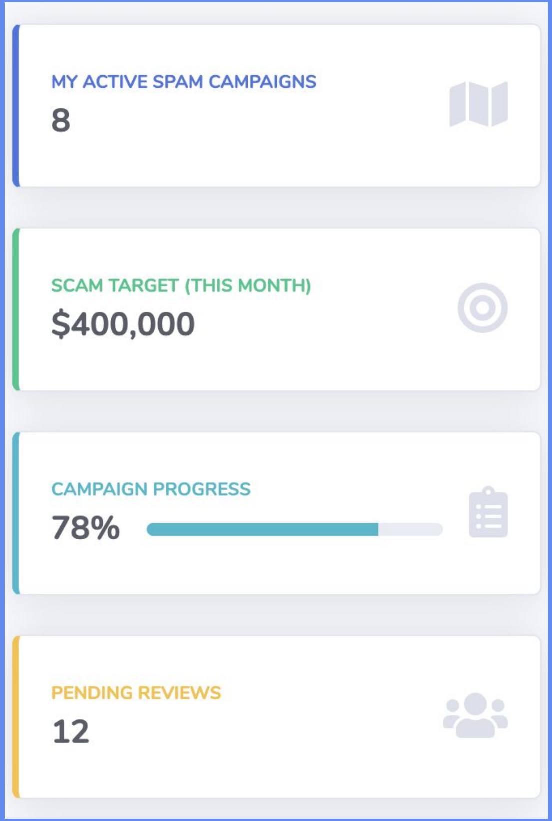
$215,000

CAMPAIGN PROGRESS

55%

PENDING REVIEWS

18



MY ACTIVE SPAM CAMPAIGNS

8

SCAM TARGET (THIS MONTH)

$400,000

CAMPAIGN PROGRESS

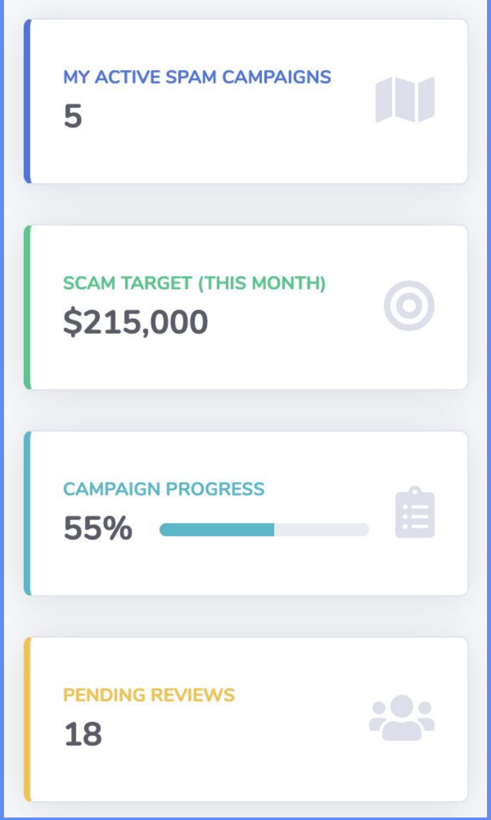
78%

PENDING REVIEWS

12

Per-User Reports

```
$build = [
  '#cache' => [
    ‘tags’ => [
      // Tags for each component.
    ],
    'context' => [
      // Vary content per user.
      'user',
    ],
  ],
];
```



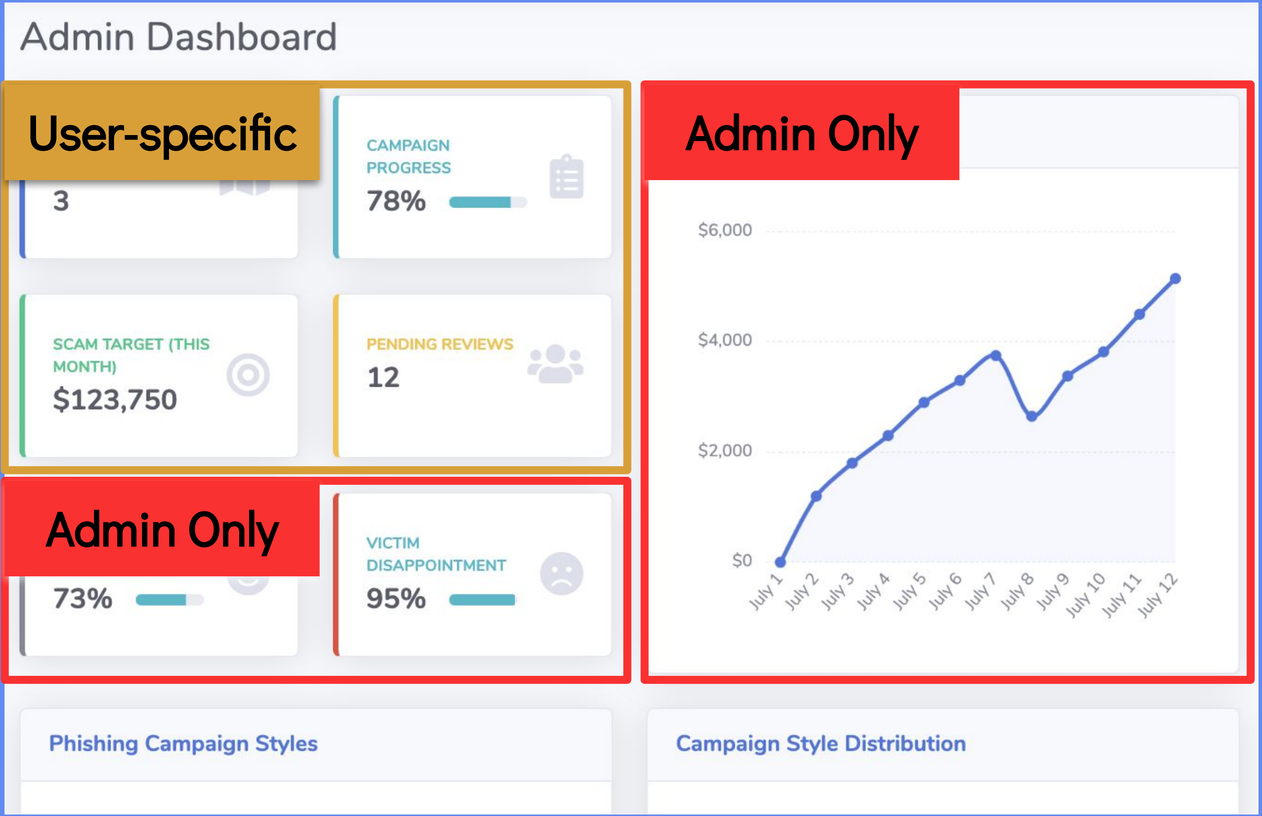
MY ACTIVE SPAM CAMPAIGNS
5

SCAM TARGET (THIS MONTH)
$215,000

CAMPAIGN PROGRESS
55%

PENDING REVIEWS
18

Role-Based Reports



Admin Dashboard

User-specific

3

CAMPAIGN PROGRESS

78%

SCAM TARGET (THIS MONTH)

$123,750

PENDING REVIEWS

12

Admin Only

73%

VICTIM DISAPPOINTMENT

95%

Admin Only

Phishing Campaign Styles

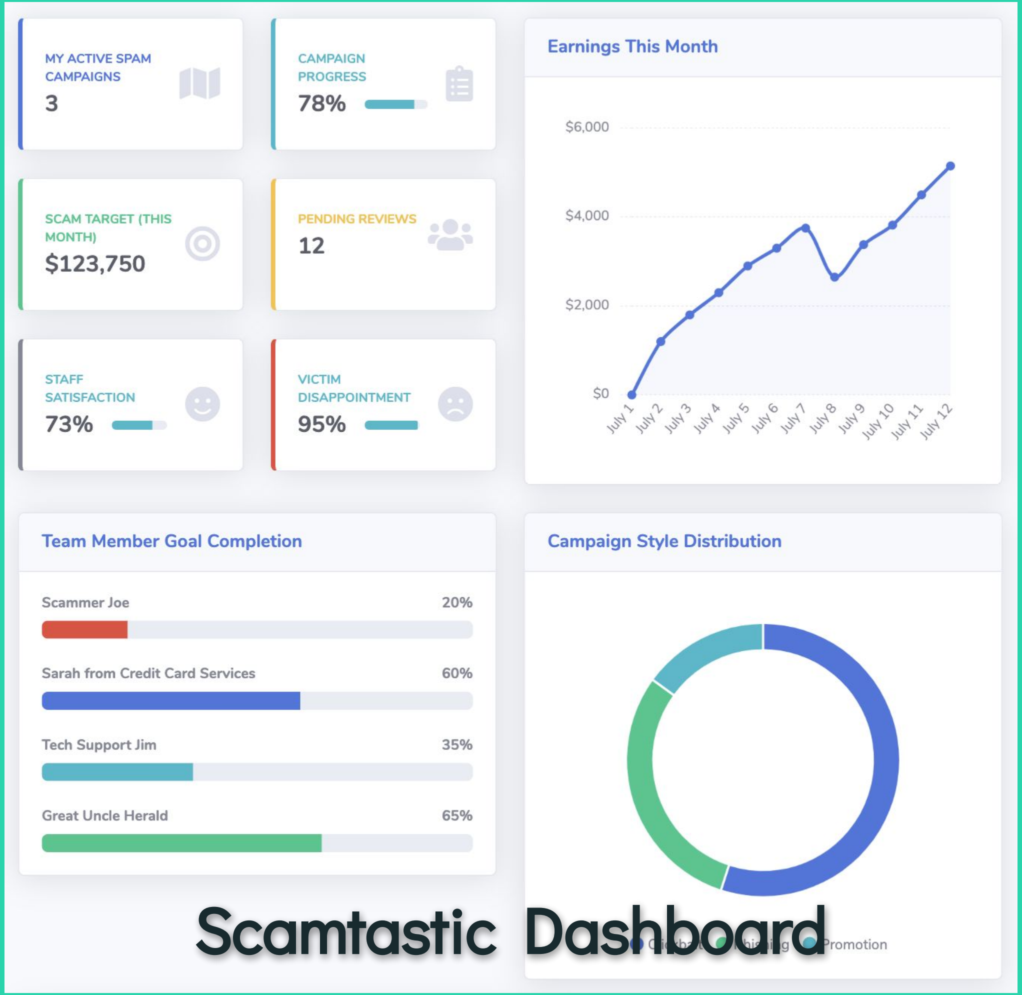
Campaign Style Distribution

Role-Based Reports

- Vary content based on whether a user is assigned the admin role
- Check for boolean presence of a specific role
- Or check for a full combination of roles

```
$build = [
  '#cache' => [
    'context' => [
      // Vary whether a user has
      // the “admin” role.
      'user.roles:admin',
    ],
  ],
];
```

STEPPING BACK



MY ACTIVE SPAM CAMPAIGNS
3

CAMPAIGN PROGRESS
78%

SCAM TARGET (THIS MONTH)
$123,750

PENDING REVIEWS
12

STAFF SATISFACTION
73%

VICTIM DISAPPOINTMENT
95%

Earnings This Month

Team Member Goal Completion

Scammer Joe	20%
Sarah from Credit Card Services	60%
Tech Support Jim	35%
Great Uncle Herald	65%

Campaign Style Distribution

Scamtastic Dashboard

> An HTML page served by Drupal can be considered **one big render tree**, with the root of the tree being the entire page, the first level being the **regions**, the second level the **blocks**, the third level being the **block contents**, and so on.

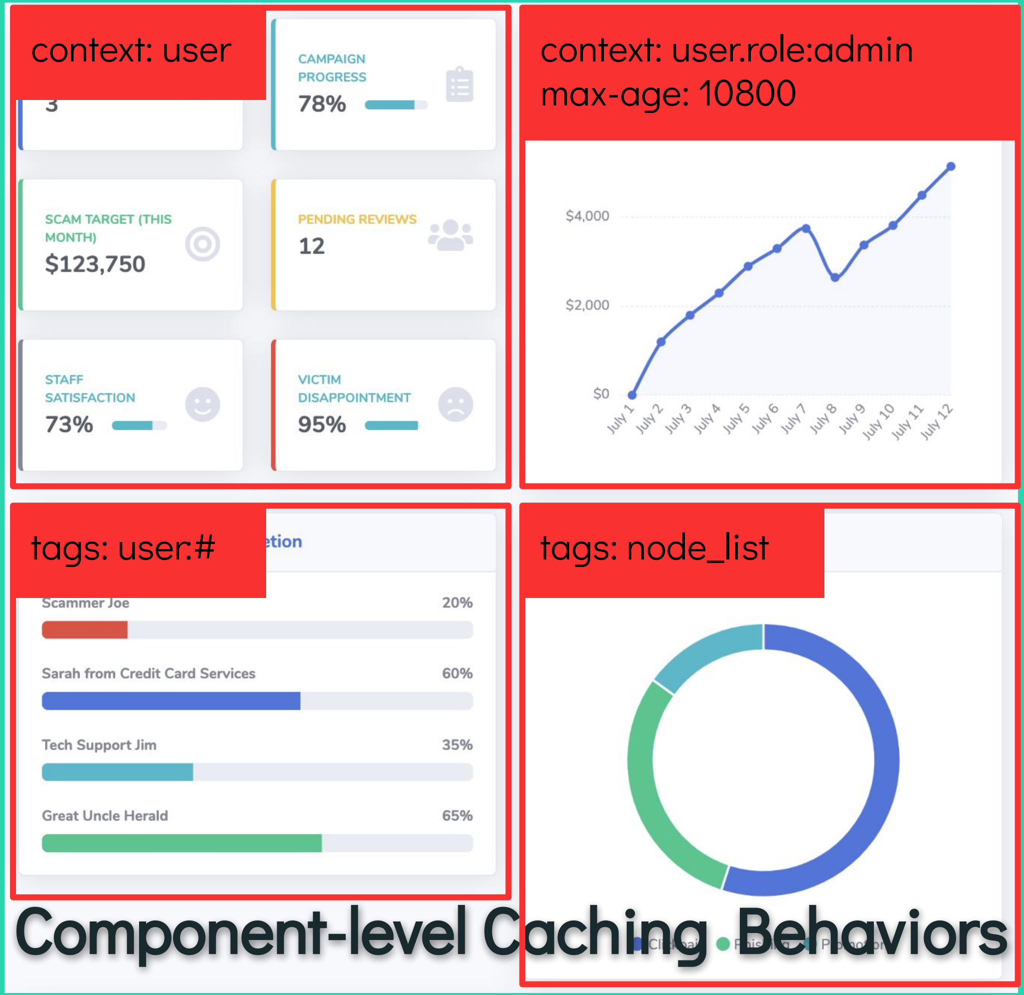
BUBBLING

Bubbled Data

Context: ~~user.role:admin~~
user

Max-age: 10800

Tags: user.#
node_list



context: user

3

CAMPAIGN PROGRESS 78%

SCAM TARGET (THIS MONTH) $123,750

PENDING REVIEWS 12

STAFF SATISFACTION 73%

VICTIM DISAPPOINTMENT 95%

context: user.role:admin
max-age: 10800

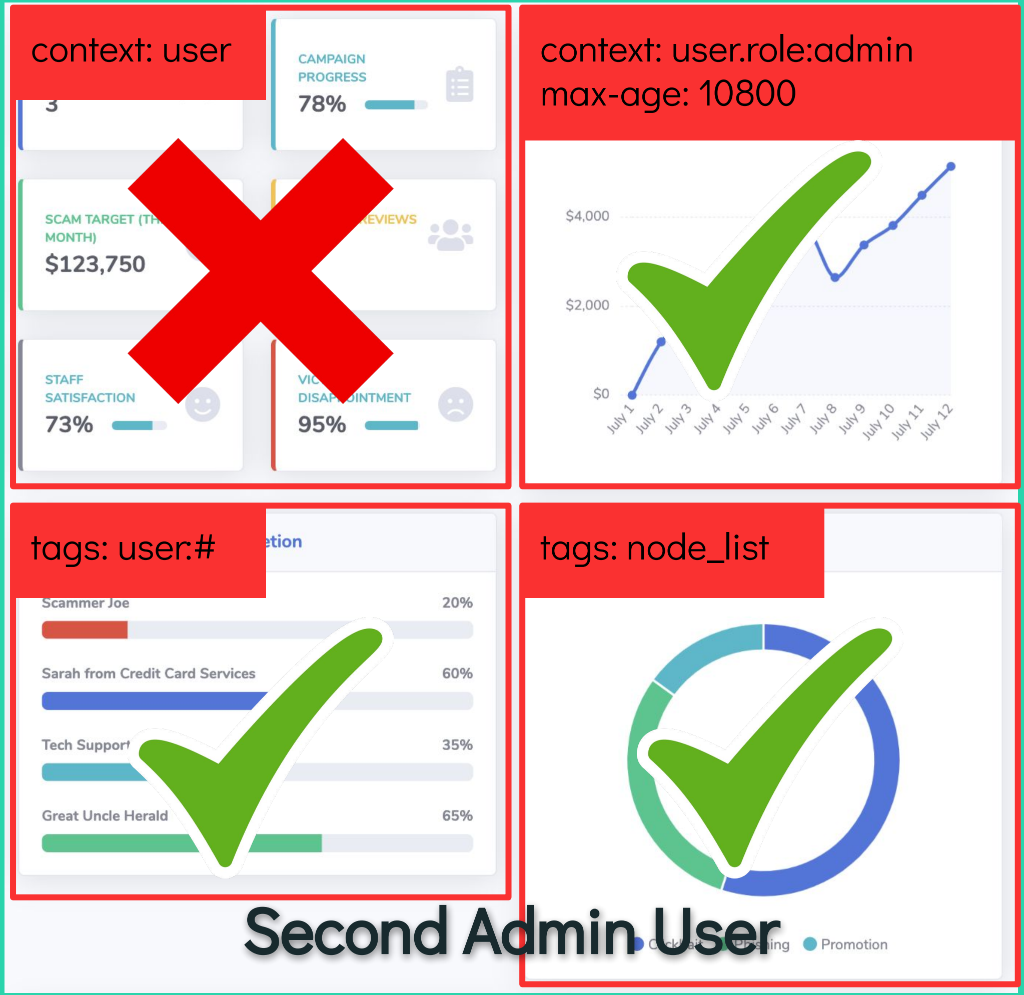
tags: user:#

etion

Scammer Joe	20%
Sarah from Credit Card Services	60%
Tech Support Jim	35%
Great Uncle Herald	65%

tags: node_list

Component-level Caching Behaviors



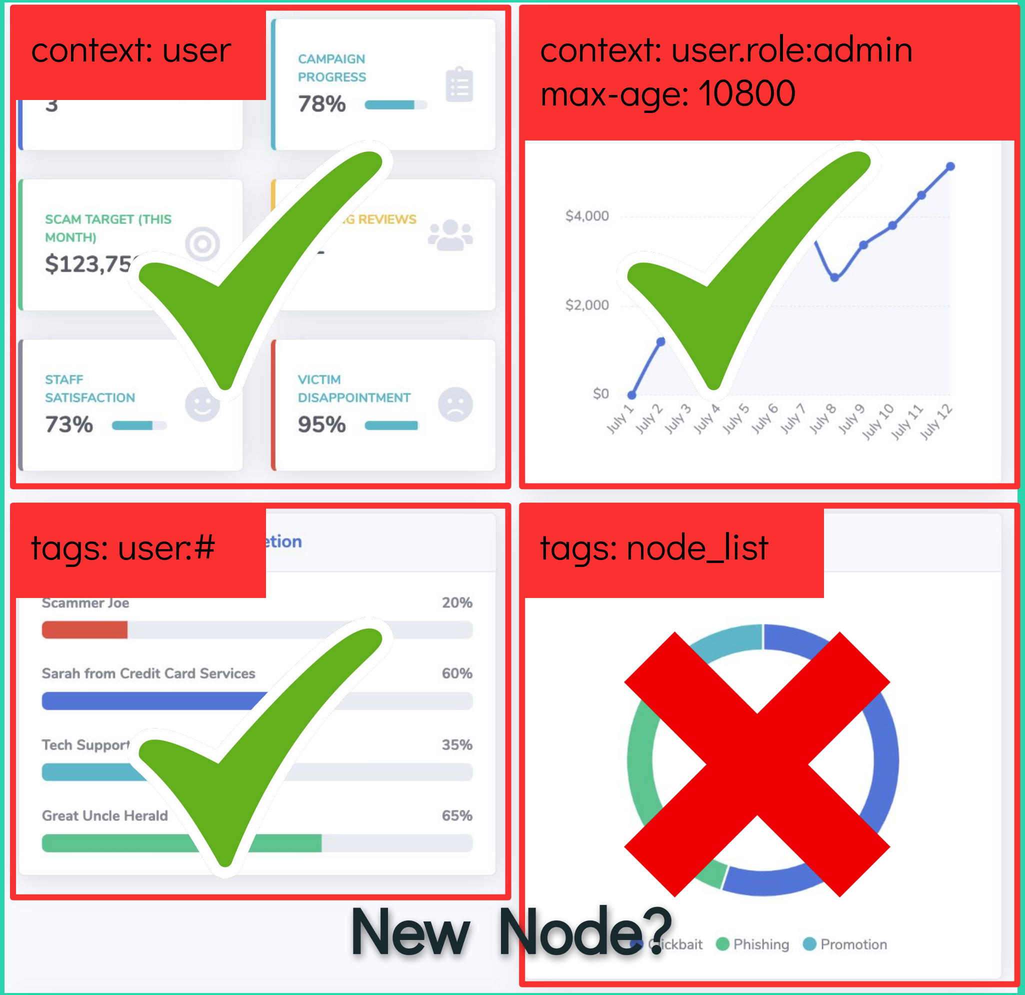
context: user

context: user.role:admin
max-age: 10800

tags: user:#

tags: node_list

Second Admin User



context: user

context: user.role:admin
max-age: 10800

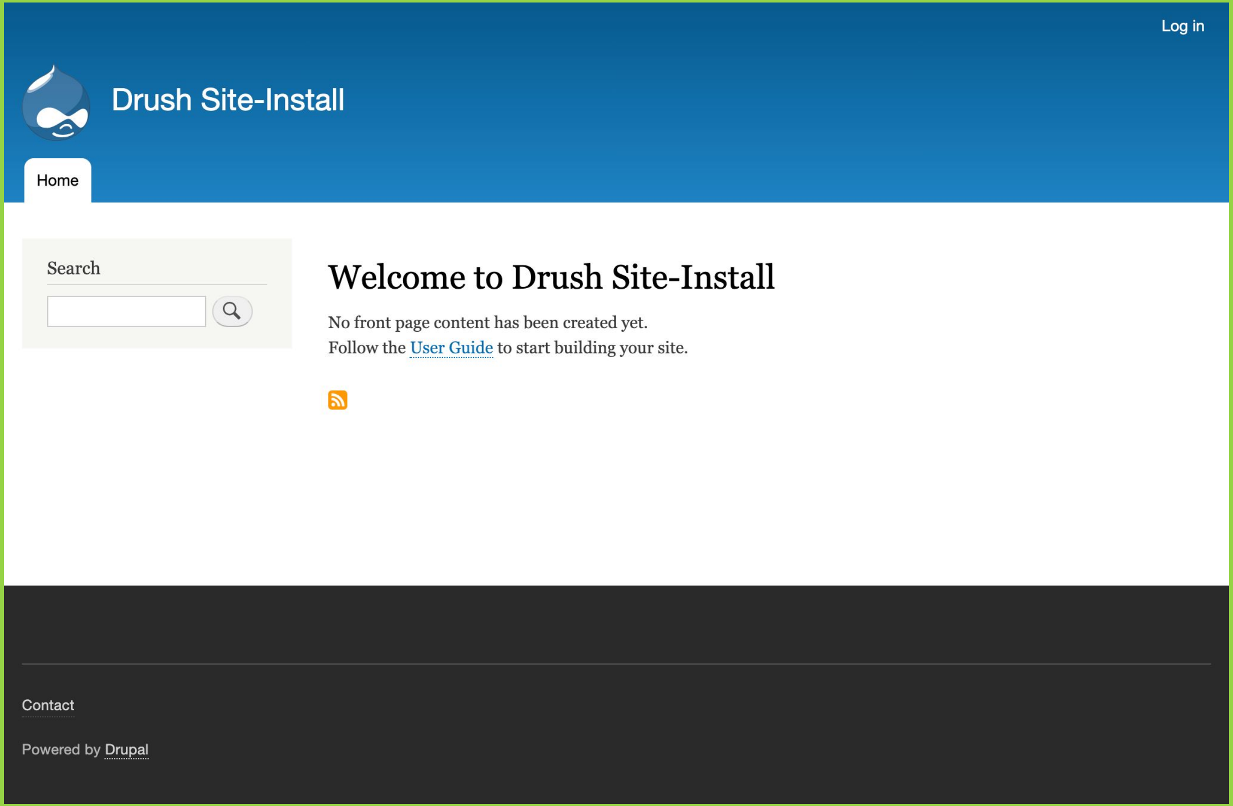
tags: user:#

tags: node_list

New Node?

THE NITTY GRITTY DETAILS

Something Familiar



Log in

Drush Site-Install

Home

Search

Welcome to Drush Site-Install

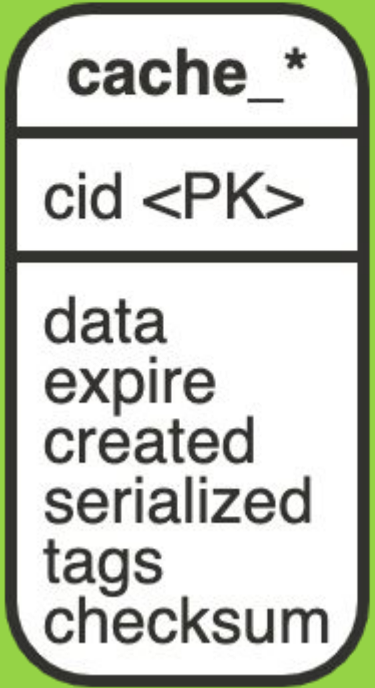
No front page content has been created yet.
Follow the User Guide to start building your site.

Contact

Powered by Drupal

Cache Tables

- All cache tables have the same structure
  - □ cache_bootstrap
  - □ cache_config
  - □ cache_container
  - □ cache_data
  - □ cache_default
  - □ cache_discovery
  - □ cache_dynamic_page_cache
  - □ cache_entity
  - □ cache_menu
  - □ cache_page
  - □ cache_render



cache_*

cid <PK>

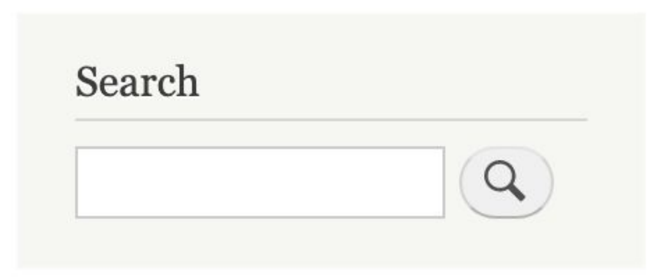
data
expire
created
serialized
tags
checksum

cid: Cache ID

- Each cache entry has to be uniquely identifiable
- Concatenated collection of identifiers
  - Keys for what is being cached
  - Contexts to identify variations

Cache ID

```
entity_view:block:bartik_search:[languages:language_interface]=en:[theme]=bartik:[user.permissions]=55c9d45a6352718ec07b9b9b04cf0514415bef471400a4912e645f80db8808af
```



Keys

- entity_view
- block
- bartik_search

Contexts

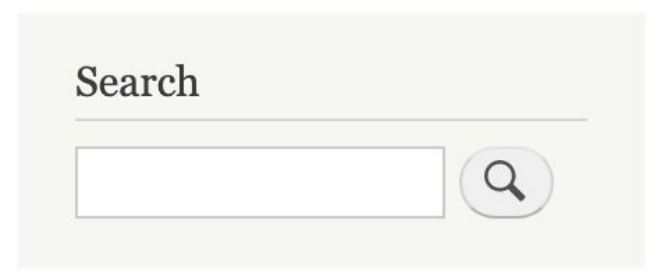
- **[languages:language_interface]**=en
- **[theme]**=bartik
- **[user.permissions]**=55c9d45a6352718ec07b9b9b04cf0514415bef471400a4912e645f80db8808af

tags: Cache Tags

- Space-delimited
- “Targets”
- Bulk identifiers
- Used to invalidate/delete cache entries en masse

Cache Tags

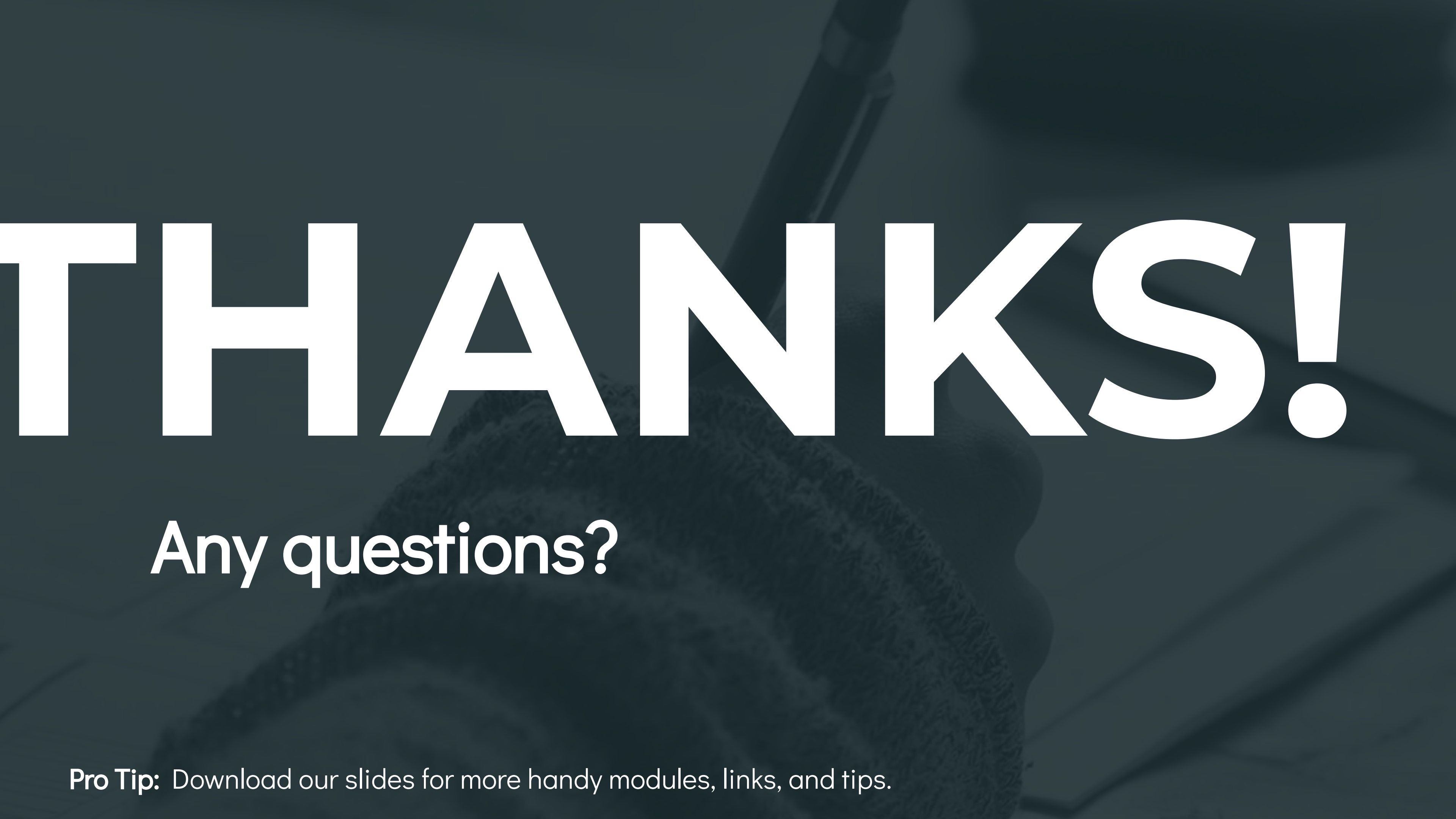
```
block_view
config:block.block.bartik_search
config:search.settings
rendered
```



- block_view
- config:block.block.bartik_search
- config:search.settings
- rendered

⚠ Gotchas

- Invalidation vs deletion
- Unknown content connections
  - Entity references
- Loss of cache metadata in custom templates
  - Render "content" to register bubbleable metadata
- CDN & Purge
- CDN tag hashing
- Expiration doesn't mean timely deletion
  - Expiration checked on lookup



THANKS!

Any questions?

Pro Tip: Download our slides for more handy modules, links, and tips.

Dev & Debugging

- Renderviz module
  - Output element and bubbled cache data in Twig debug output
  - Learning and debugging tool to explore bubbling
- Make sure local caching is not disabled
  - Helpful resources, https://www.dx-experts.nl/blog/2017/drupal-8-development-caching
- Cache debug headers
  - http.response.debug_cacheability_headers container parameter to true, in your services.yml

MODULES & RESOURCES

Handy Dandy Modules

- Views Custom Cache Tags
- Handy Cache Tags
- Renderviz
- Cache Tools

- Purge
- Acquia Purge
- Pantheon Purge

Tomes of Knowledge

- Cacheability of Render Arrays
- Coding with Cache Tags in Drupal 8
- D.O. Cache API Documentation
  - All the things in this section
- API Documentation

Digging Deeper

- Auto-placeholdering
- BigPipe
- Cache bins
    - Custom cache bins
    - Configuration
    - Backends
- Custom contexts
- Response-level Caching
    - Handling redirects and other custom responses

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash
- Dashboard demo theme - SB Admin 2