Caching for Drupal 8: making sense of context, tags and max-age

Drupalcamp NJ 2017
A joke
“There are two hard problems in computer science...”

-Every clever developer, ever
The two hard problems in computer science

- Naming things
- Cache invalidation
- Off by one errors
Naming things

- ContainerFactoryPluginInterface
- EntityDefinitionUpdateManagerInterface
- Node
- (8 classes called 'Node')
- EntityLastInstalledSchemaRepositoryInterface
Drupal\Core\Entity\EntityLastInstalledSchemaRepositoryInterface
Cache Invalidation
A brief history of Drupal caching
Early Drupal 4.? (3.?)

- Offered Page cache to store whole HTML pages
- Not super granular, tricky to invalidate
- Lots of cache_clear_all(). (31 calls in Drupal 4.6)
Drupal 6

- Blocks offered granular caching for the first time
- `BLOCK_CACHE_PER_ROLE`, `BLOCK_CACHE_PER_USER`, `BLOCK_CACHE_GLOBAL`, `BLOCK_NO_CACHE`
- Still no good way to clear caches, much `cache_clear_all()` was had by all. (52 calls)
Drupal 7

• Improved internal static caches, but not more permanent caches

• Even more cache_clear_all(). (79 calls)

• Introduced the render array – markup could have metadata attached to it
function node_save($node) {
    $transaction = db_transaction();
    cache_clear_all();

    try {
        cache_clear_all();
        // Load the stored entity, if any.
        if (!empty($node->nid) && !isset($node->original)) {
            $node->original = entity_load_unchanged('node', $node->nid);
            cache_clear_all();
        }
    }

    cache_clear_all();
    cache_clear_all();
    cache_clear_all();
    field_attach_presave('node', $node);
    cache_clear_all();
    global $user;
    cache_clear_all();
Drupal 8

- Introduces Cache Tags, Cache Contexts
- Added render caching, and cacheability metadata
- R.I.P. cache_clear_all(), DRUPAL_CACHE_PER_PAGE, etc.
What is cacheability metadata
Cacheability metadata

- Determines how something can be cached
- Contains tags, contexts, and max-age
- Cacheability metadata bubbles up through an entire page to control caching
Cache Contexts

- Determines how to vary items according to request
- Similar to D7 block constants, but with many more options
- Organized into a hierarchy, which is auto folded/combined/collapsed to prevent duplication
Cache Contexts

• Examples:
  • cookies, headers, languages, route, session, url, user, etc.
  • user.permissions, user.roles
  • user.roles:anonymous
  • url.query_args:foo, url.path, or just url
Folding/collapsing Cache Contexts

- First item listed is most granular, last is most specific

- For example `url.query_args:foo` only varies when `?foo=bar` is present in URL, but `url` by itself varies every single URL (host, path, query param, etc)

- When both `url.query_args:foo` and `url` are present Drupal only evaluates `url` since it encompasses `url.query_args` as well
More on cache contexts

https://www.drupal.org/docs/8/api/cache-api/cache-contexts
Cache tags

• Cache tags describe the data that your content depends on

• Cache tags describe the data you return on the page that is managed by Drupal

• Cache tags allow caches to be cleared based on corresponding tags

• Can be configured to integrate with CDN, such as Fastly, CloudFlare, KeyCDN, or Varnish
Cache tags

- Cache tags are just strings, generally \texttt{thing:identifier}, but can be anything without spaces.

- Conventions are \texttt{entity_type:id} or \texttt{entity_type_list}.

- Examples: \texttt{node:5}, \texttt{node_list}, \texttt{config:system.performance}.
More on cache tags

https://www.drupal.org/docs/8/api/cache-api/cache-tags
Cache max-age

- Time dependencies
- Controls how long an item may be cached by number of seconds
- Drupal core does not use this, everything is cached permanently and uses cache tags for invalidation
- You should avoid using this as well
Cache max-age

- Expressed as an integer number of seconds that the item is valid for
- 0 means the item is not cacheable, so to disable an item from all caching, set `max-age = 0`
- `Drupal\Core\Cache\Cache::PERMANENT` means an item is cacheable forever, can only be cleared with cache tags
More on cache max-age

https://www.drupal.org/docs/8/api/cache-api/cache-max-age
What is cache bubbling?
Cache bubbling

- All data from each piece of markup, is combined, into blocks, into regions and then into the page

- Duplicates are removed, contexts are folded/collapsed

- All of that information is used to cache the entire page (including CDN)

- Individual components remain cached at the render level
Sample article
Submitted by admin on Sat, 02/04/2017 - 06:44

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sample article
Submitted by admin on Sat, 02/04/2017 - 06:44

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Node: 4
Real world example
(mostly real, anyway)
Site requirement

• Awesome site for fans of sci-fi space stories

• Can enter content about different series/stories

• Want to show related items from same story in a block in the sidebar for each page

• Yes, I know that Views can do this, but if it couldn’t you would really need to know how to do this,

• Trust me
The Inner Light (episode)
Submitted by admin on Wed, 02/01/2017 - 21:30

"The Inner Light" is the 125th episode overall and the 25th episode of the fifth season of the American science fiction television series Star Trek: The Next Generation. The episode was written by freelance writer Morgan Gendel based on his original pitch. Gendel is credited as writer of the story and co-writer of the teleplay with Peter Allan Fields. It is the penultimate episode of the season and was first broadcast on June 1, 1992.

USS Enterprise (NCC-1701-D)
Submitted by admin on Wed, 02/01/2017 - 21:24

The USS Enterprise (NCC-1701-D) (or Enterprise-D, to distinguish it from earlier and later starships with the same name) is a 24th century starship in the Star Trek fictional universe and the principal setting of the Star Trek: The Next Generation television series. [1] The Enterprise-D also appears in the pilot episode of Star Trek: Deep Space Nine ("Emissary"), [2] the series finale of Star Trek: Enterprise ("These Are the Voyages..."), and the movie Star Trek Generations.
Let’s make a block

• Need to get the current node

• Query for other nodes that have the same taxonomy term

• Pass them off to our theme function, and we’re done

• Right?
public function build() {
    $build = [];

    if ($current_node = $this->routeMatch->getParameter('node')) {
        $query = $this->entityQuery->get('node');
        $nids = $query->condition('field_universe.target_id',
                                  $current_node->field_universe->target_id)
                      ->sort('created', 'DESC')
                      ->range(0, 4)
                      ->execute();

        $related_items = Node::loadMultiple($nids);
        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    return $build;
}
The Inner Light (episode)

Submitted by admin on Wed, 02/01/2017 - 21:30

"The Inner Light" is the 125th episode overall and the 25th episode of the fifth season of the American science fiction television series Star Trek: The Next Generation. The episode was written by freelance writer Morgan Gendel based on his original pitch. Gendel is credited as writer of the story and co-writer of the teleplay with Peter Allan Fields. It is the penultimate episode of the season and was first broadcast on June 1, 1992.
William "Bill" Adama is a fictional character portrayed by Edward James Olmos in the re-imagined Battlestar Galactica television series produced and aired by the SyFy cable network. He is one of the main characters, a reimagining of Commander Adama from the 1978 Battlestar Galactica series played by Lorne Greene.
What’s missing?
public function build() {
    $build = [];

    if ($current_node = $this->routeMatch->getParameter('node')) {
        $query = $this->entityQuery->get('node');
        $nids = $query->condition('field_universe.target_id',
                                   $current_node->field_universe->target_id)
                          ->sort('created', 'DESC')
                          ->range(0, 4)
                          ->execute();

        $related_items = Node::loadMultiple($nids);
        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    return $build;
}
public function build() {
    $build = [];

    $cachableMetadata = new CacheableMetadata();
    $cachableMetadata->setCacheContexts(['url.path']);

    if ($current_node = $this->routeMatch->getParameter('node')) {
        // Query and load $related_items from database

        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    $cachableMetadata->applyTo($build);
    return $build;
}
CacheableMetadata

- Drupal\Core\Cache\CacheableMetadata

- Defines a generic class for passing cacheability metadata

- Key methods setCacheContexts(), setCacheTags(), setCacheMaxAge(), and applyTo()
public function build() {
    $build = [];

    $cachableMetadata = new CacheableMetadata();
    $cachableMetadata->setCacheContexts(['url.path']);

    if ($current_node = $this->routeMatch->getParameter('node')) {
        // Query and load $related_items from database

        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    $cachableMetadata->applyTo($build);
    return $build;
}
$build = [
    '#theme' => 'related_items_block',
    '#items' => [ . . . ],
];
$build = [
    '#theme' => 'related_items_block',
    '#items' => [...],
    '#cache' => [
        'contexts' => ['url.path'],
        'tags' => [],
        'max-age' => -1,
    ],
];
William "Bill" Adama is a fictional character portrayed by Edward James Olmos[2] in the re-imagined Battlestar Galactica television series produced and aired by the SyFy cable network. He is one of the main characters, a reimagining of Commander Adama from the 1978 Battlestar Galactica series played by Lorne Greene.
Cool! Everything's fixed, right?
Rey is a fictional character in the Star Wars franchise, portrayed by English actress Daisy Ridley. First appearing as the central character in Star Wars: The Force Awakens,[5][1][6] Rey is a scavenger who was left behind on the planet Jakku when she was a child, and later becomes involved with the Resistance's conflict with the First Order when her solitary life is interrupted by BB-8, the droid of ace Resistance pilot Poe Dameron, and a runaway Stormtrooper named Finn.
Home » Millennium Falcon

Title *

Millennium Falcon

Body (Edit summary)
Millennium Falcon

Submitted by admin on Wed, 02/01/2017 - 20:52

The *Millennium Falcon* is a spaceship in the *Star Wars* universe commanded at one time by Corellian smuggler Han Solo (Harrison Ford) and his Wookiee first mate, Chewbacca (Peter Mayhew). The highly modified Corellian Engineering Corporation (CEC) YT-1300 light freighter first appears in *Star Wars* (1977), and subsequently in *The Empire Strikes Back* (1980), *Return of the Jedi* (1983) and *The Force Awakens* (2015). The ship also makes a brief cameo appearance in *Revenge of the Sith* (2005). Additionally, the *Falcon* appears in a variety of *Star Wars* expanded universe materials, including books, comics, and games; James Luceno's novel *Millennium Falcon* focuses on the titular ship.[2] It also appears in the 2014 animated film *The Lego Movie* in *Lego* form, with Billy Dee Williams and Anthony Daniels reprising their roles of Lando Calrissian and C-3PO, with Keith Ferguson voicing Han Solo.
What’s missing?
public function build() {
    $build = [];

    $cachableMetadata = new CacheableMetadata();
    $cachableMetadata->setCacheContexts(['url.path']);

    if ($current_node = $this->routeMatch->getParameter('node')) {
        // Query and load $related_items from database

        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    foreach ($related_items as $related_item) {
        $cachableMetadata->addCacheableDependency($related_item);
    }

    $cachableMetadata->applyTo($build);
    return $build;
}
CacheableMetadata

- **addCacheableDependency()** takes all the cacheability metadata from another object and adds it to our already established data.

- Uses **CacheableDependencyInterface** to get context, tags, and max-age from the other object.
CacheableDependencyInterface

- Provides a standard interface for getting cache metadata from objects

- Only three methods: `getCacheContexts()`, `getCacheTags()`, and `getCacheMaxAge()`.

- LOTS of things implement this interface: entities, fields, blocks, views, access results, configuration, forms, etc.

- Anything that represents stored data, or represents rendered output – probably implements `CacheableDependencyInterface`
$build = [
    '#theme' => 'related_items_block',
    '#items' => [...],
    '#cache' => [
        'contexts' => ['url.path'],
        'tags' => [],
        'max-age' => -1,
    ],
];
$build = [
    '#theme' => 'related_items_block',
    '#items' => $related_items,
    '#cache' => [
        'contexts' => ['url.path'],
        'tags' => [
            0 => 'node:11',
            1 => 'node:12',
            2 => 'node:4',
            3 => 'node:5',
        ],
        'max-age' => -1,
    ],
];
Destiny

Submitted by admin on Wed, 02/01/2017 - 21:01

Destiny is the massive, flying pyramid which was part of an Ancient experiment to seed the galaxies with Stargates millions of years ago. Their plan to return to the ship after it finished its journey never came about due to their ascension. The ship was sent on auto-pilot, preceded by other automated ships which seeded the habitable worlds along its route with Stargates. It uses an unidentified form of superluminal propulsion, said to be faster than light but "not hyperdrive." Destiny is reached through its on-board Stargate from any other location, using a unique nine-chevron address that acts as a code, dialed on a Stargate with sufficient power supply (only power source sufficient enough so far has been a
Destiny

Submitted by admin on Wed, 02/01/2017 - 21:01

Stargate Universe is set aboard the Ancient ship Destiny, a massive vessel which was part of an Ancient experiment to seed the galaxies with Stargates millions of years ago. Their plan to return to the ship after it finished its journey never came about due to their ascension. The ship was sent on auto-pilot, preceded by other automated ships which seeded the habitable worlds along its route with Stargates. It uses an unidentified form of superluminal propulsion, said to be faster than light but "not hyperdrive." Destiny is reached through its on-board Stargate from any other location, using a unique nine-chevron address that acts as a code, dialed on a Stargate with sufficient power supply (only power source sufficient enough so far has been a
Type *

Ship

Universe *

Stargate Universe
Destiny

Submitted by admin on Wed, 02/01/2017 - 21:01

Stargate Universe

Stargate Universe is set aboard the Ancient ship Destiny, a massive vessel which was part of an Ancient experiment to seed the galaxies with Stargates millions of years ago. Their plan to return to the ship after it finished its journey never came about due to their ascension. The ship was sent on auto-pilot, preceded by other automated ships which seeded the habitable worlds along its route with Stargates. It uses an unidentified form of superluminal propulsion, said to be faster than light but "not hyperdrive." Destiny is reached through its on-board Stargate from any other location, using a unique nine-chevron address that acts as a code, dialed on a Stargate with sufficient power supply (only power source sufficient enough so far has been a
What’s missing?
public function build() {
    $build = [];

    $cachableMetadata = new CacheableMetadata();
    $cachableMetadata->setCacheContexts(['url.path']);

    if ($current_node = $this->routeMatch->getParameter('node')) {
        // Query and load $related_items from database

        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    $cachableMetadata->addCacheableDependency($current_node);

    foreach ($related_items as $related_item) {
        $cachableMetadata->addCacheableDependency($related_item);
    }

    $cachableMetadata->applyTo($build);
    return $build;
}
$build = [
    '#theme' => 'related_items_block',
    '#items' => $related_items,
    '#cache' => [
        'contexts' => ['url.path'],
        'tags' => [
            0 => 'node:11',
            1 => 'node:12',
            2 => 'node:4',
            3 => 'node:5',
        ],
        'max-age' => -1,
    ],
];
$build = [
    '#theme' => 'related_items_block',
    '#items' => $related_items,
    '#cache' => [
        'contexts' => ['url.path'],
        'tags' => [
            0 => 'node:3',
            1 => 'node:11',
            2 => 'node:12',
            3 => 'node:4',
            4 => 'node:5',
        ],
        'max-age' => -1,
    ],
];
Larry Garfield has been building web sites since he was a sophomore in high school, which is longer ago than he’d like to admit. He spent time as a freelance web developer and journalist before settling in as a Drupal developer for the past decade. After 10 years with Palantir.net he recently moved to Platform.sh.
Larry "Crell" Garfield
Submitted by admin on Sat, 02/04/2017 - 08:31

Larry Garfield has been building websites since he was a sophomore in high school, which is longer ago than he'd like to admit. He spent time as a freelance web developer and journalist before settling in as a Drupal developer for the past decade. After 10 years with Palantir.net he recently moved to Platform.sh.

---

The Inner Light (episode)
Submitted by admin on Wed, 02/01/2017 - 21:30

"The Inner Light" is the 125th episode overall and the 25th episode of the fifth season of the American science fiction television series Star Trek: The Next Generation. The episode was written by freelance writer Morgan Gendel based on his original pitch. Gendel is credited as writer of the story and co-writer of the teleplay with Peter Allan Fields. It is the penultimate episode of the season and was first broadcast on June 1, 1992.
What’s missing?
public function build() {
    $build = [];

    $cachableMetadata = new CacheableMetadata();
    $cachableMetadata->setCacheContexts(['url.path']);
    $cachableMetadata->setCacheTags(['node_list']);

    if ($current_node = $this->routeMatch->getParameter('node')) {
        // Query and load $related_items from database

        $build = [
            '#theme' => 'related_items_block',
            '#items' => $related_items,
        ];
    }

    $cachableMetadata->addCacheableDependency($current_node);

    foreach ($related_items as $related_item) {
        $cachableMetadata->addCacheableDependency($related_item);
    }

    $cachableMetadata->applyTo($build);
    return $build;
}
List tags

• List tags are flushed every time a new item is created

• In the format of `ENTITYTYPE_list`
$build = [
    '#theme' => 'related_items_block',
    '#items' => $related_items,
    '#cache' => [
        'contexts' => ['url.path'],
        'tags' => [
            0 => 'node:3',
            1 => 'node:11',
            2 => 'node:12',
            3 => 'node:4',
            4 => 'node:5',
        ],
        'max-age' => -1,
    ],
];
$build = [$
    '#theme' => 'related_items_block',
    '#items' => $related_items,
    '#cache' => [$
        'contexts' => ['url.path'],
        'tags' => [$
            0 => 'node_list',
            1 => 'node:3',
            2 => 'node:11',
            3 => 'node:12',
            4 => 'node:4',
            5 => 'node:5',
        ],
        'max-age' => -1,
    ],
];
Check out the code

- [https://github.com/mikeyp/caching_demo](https://github.com/mikeyp/caching_demo)
- Tags correspond to readme and each step
Tips and tricks for development
Tips and tricks for development

• Need a list of cache contexts? Check core.services.yml
Tips and tricks for development

- Check out `example.settings.local.php` and `development.services.yml`
- Don’t be tempted to disable the render cache!
/**
 * Disable the render cache (this includes the page cache).
 *
 * Note: you should test with the render cache enabled, to ensure the correct
 * cacheability metadata is present. However, in the early stages of
 * development, you may want to disable it.
 *
 * This setting disables the render cache by using the Null cache back-end
 * defined by the development.services.yml file above.
 *
 * Do not use this setting until after the site is installed.
 */

# $settings['cache']['bins']['render'] = 'cache.backend.null';
/**
 * Disable Dynamic Page Cache.
 *
 * Note: you should test with Dynamic Page Cache enabled, to ensure the correct
 * cacheability metadata is present (and hence the expected behavior). However,
 * in the early stages of development, you may want to disable it.
 */

# $settings['cache']['bins']['dynamic_page_cache'] = 'cache.backend.null';
Tips and tricks for development

• Do turn on `debug_cacheability_headers`!
# Local development services.

# To activate this feature, follow the instructions at the top of the
# 'example.settings.local.php' file, which sits next to this file.

parameters:
  http.response.debug_cacheability_headers: true
Tips and tricks for development

• Stub out your blocks and controllers with cacheability metadata before writing any logic
public function build() {
    $build = [];
    $cachableMetadata = new CacheableMetadata();

    // Your logic here...

    $cachableMetadata->applyTo($build);
    return $build;
}
Making your own cache tags
Making your own cache tags

• Do whatever you want

• You should probably namespace it with your module or something

• Make sure you invalidate your own tags when your data changes
$this->state()->set('my_module.some_data', $data);
$this->cacheInvalidator->invalidateTags(['my_module:state:some_data']);

...
$cachableMetadata->setCacheTags(['my_module:state:some_data']);
$cachableMetadata->applyTo($build);
return $build;
Making your own cache contexts
Making your own cache contexts

- Must implement service that starts with `cache_context`, i.e. `cache_context.my_module_form`
- Tag the service with `name: cache.context`
- Must implement `CacheContextInterface` or `CalculatedCacheContextInterface`
- `RequestStackCacheContextBase` may be a good starting point if you depend on the request
class FormCacheContext extends RequestStackCacheContextBase
    implements CalculatedCacheContextInterface {

    public static function getLabel() {
        return t("Form access");
    }

    public function getContext($parameter = NULL) {
        $formAccess = FormAccess::fromRequestStack($this->requestStack);
        if ($parameter === NULL) {
            return FALSE;
        }
        else {
            return $formAccess->hasSubmitted($parameter);
        }
    }

    public function getCacheableMetadata($parameter = NULL) {
        return new CacheableMetadata();
    }
}
$cachableMetadata->setCacheContexts(['my_module_form:' . $parameter]);
$cachableMetadata->applyTo($build);
return $build;
Big pipe?
Big Pipe

- Pioneered at Facebook

- Works well with bits of a page that are very slow, i.e. external APIs, per-user customized content, etc

- Works by rendering already cached or fast parts of the page first, then sending them to the browser

- Slow parts of the page are rendered and sent to the browser when they’re available

- Seriously, go to Facebook and watch it load...
Big Pipe

- Does NOT use AJAX, uses HTTP streaming over single connection
- Can work with auto-placeholdering, or just set max-age to 0.
- Requires a session to be open
public function getCacheMaxAge() {
    return 0;
}

public function build() {
    sleep(5);

    $build = [];

    $cachableMetadata = new CacheableMetadata();
    $cachableMetadata->setCacheContexts(['url.path']);
    $cachableMetadata->setCacheTags(['node_list']);

    // Load related items, etc. . . .

    $cachableMetadata->applyTo($build);
    return $build;
}
The Inner Light (episode)
Submitted by admin on Wed, 02/01/2017 - 13:30

"The Inner Light" is the 125th episode overall and the 25th episode of the fifth season of the American science fiction television series Star Trek: The Next Generation. The episode was written by freelance writer Morgan Gendel based on his original pitch. Gendel is credited as writer of the story and co-writer of the teleplay with Peter Allan Fields. It is the penultimate episode of the season and was first broadcast on June 1, 1992.

USS Enterprise (NCC-1701-D)
Submitted by admin on Wed, 02/01/2017 - 13:24

The USS Enterprise (NCC-1701-D) (or Enterprise-D, to distinguish it from earlier and later starships with the same name) is a 24th century starship in the Star Trek fictional universe and the principal setting of the Star Trek: The Next Generation television series.

[1] The Enterprise-D also appears in the pilot episode of Star Trek: Deep Space Nine ("Emissary"),[2] the series finale of Star Trek:
Questions?
Caching for Drupal 8: making sense of context, tags and max-age

Drupalcamp NJ 2017