

Render API & Cache API

SALUT, MOI C'EST KÉVIN !
ET TOI ?



EN INFORMATIQUE LE PLUS DUR
(C'EST DE NOMMER LES VARIABLES
ET D'INVALIDER LES CACHES).



TU AURAIS UN TRUC POUR
M'APPRENDRE À NOMMER DES
VARIABLES ?



PFF, TROP NAZE, ET POUR
INVALIDER LES CACHES C'EST
PAREIL ? TU NOUS AS MENTI ?



AH, JE PRÉFÈRE !
TU ME REMONTRES CE QU'EST LA
RENDER API ?





⚡ AND THE

LET'S KILL
THE ARRAYS OF DOOM!

University of
Salford
MANCHESTER

@MORTENDK BADCAMP 2012

⚡ AND THE FOMER

LET'S F... KILL THE
F... ARRAYS OF F... DOOM!

University of
Salford
MANCHESTER

@MORTENDK BADCAMP 2012

// Drupal 7.

```
$script_path = drupal_get_path('module', 'dcnantes') . '/js/script.js';
```

// C'est mal :

```
drupal_add_js($script_path);
```

// C'est mieux :

```
$form['#attached']['js'][] = $script_path;
```

```
$content = array(  
  'first_item' => array(  
    '#theme' => 'item_list',  
    '#items' => // Données.  
  ),  
  'second_item' => array(  
    '#markup' => '<p>' . t('Important message') . '</p>',  
  ),  
  ...  
);  
drupal_render($content);  
render($content);
```


Render(able) arrays

- ▶ Disparition de la fonction `theme()`.
- ▶ Tout passe dans un **template**.
- ▶ Utilisation de renderable arrays systématique pour retourner des données à afficher.
- ▶ Toute la page passe par des micro renderable arrays, on peut se le représenter comme le DOM de la page.
- ▶ Objectif : laisser n'importe quel module modifier les données à afficher jusqu'au moment du rendu.

Exemples de render array

```
// Contenu d'une page.  
$page = [  
  '#type' => 'page',  
  'content' => [  
    'system_main' => [...],  
    'another_block' => [...],  
    '#sorted' => TRUE,  
  ],  
  'sidebar_first' => [  
    ...  
  ],  
];
```

```
// Version minimale d'un render array.  
$render_array = [  
  '#markup' => "<p>Some text</p>"  
];
```

Les attributs possibles d'un render array

#type	Valeur correspondant à un RenderElement
#theme	Nom de la <i>clé de thème</i> à utiliser.
#prefix / #suffix	Chaîne à afficher avant / après le rendu du tableau.
#markup	HTML à afficher.
#cache	Attributs pour la cachabilité du tableau.
#pre_render / #post_render	Tableaux de fonctions exécutées avant / après le rendu des données.
#theme_wrappers	Tableau de fonctions de thème à utiliser pour gérer le rendu autour des enfants.

Passer des variables à un template

```
# Exemple de déclaration
function drupal_common_theme() {
  return [
    'item_list' => [
      'variables' => [
        'items' => [],
        'title' => '',
        'list_type' => 'ul',
        'wrapper_attributes' => [],
        'attributes' => [],
        'empty' => NULL,
        'context' => []
      ],
    ],
  ];
}
```

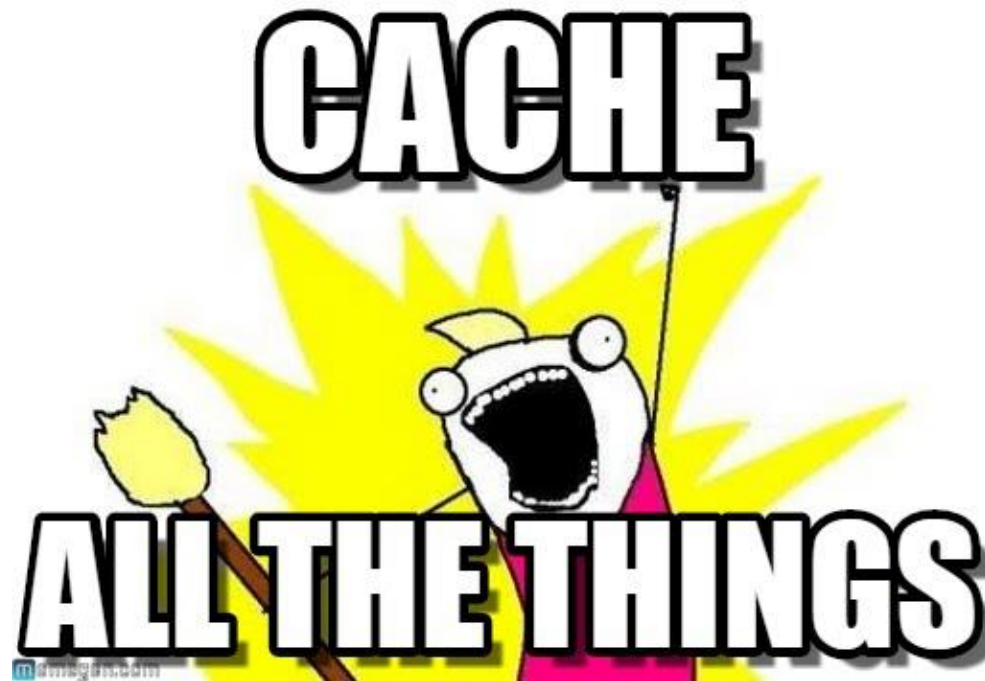
```
# Exemple d'utilisation
public function myMethod() {
  $books_list = [
    "Robinson Crusoé",
    "La guerre des Gaules"
  ];

  return [
    '#theme' => 'item_list',
    '#items' => $books_list,
    '#list_type' => 'ol',
    '#title' => t('Book list'),
  ];
}
```

OK ET SINON TU NOUS REPARLES
DU CACHE ?



Du cache à tous les niveaux



Les types de cache

1. Cache applicatif

- ▶ Cache d'exécution
- ▶ Cache statique

2. Cache d'objets

3. Cache de rendu

4. Cache HTTP

Cache applicatif : le cache d'exécution

- ▶ Durée de vie : Une requête HTTP
- ▶ Fais le tampon en RAM pour les appels suivants
- ▶ Attention à la taille des données mises en cache en multipliant les usages.

anywhere.php

```
function my_overused_function($id) {  
    $cache = &drupal_static(__FUNCTION__, []);  
    if (empty($cache[$id])) {  
        $cache[$id] = do_something_long($id);  
    }  
    return $cache[id];  
}
```

Cache applicatif : le cache statique

```
# anywhere.php
function do_something_long($id) {
    $cid = 'something_long:' . $id;
    $data = NULL;
    if ($cache = \Drupal::cache()->get($cid)) {
        $data = $cache->data;
    }
    else {
        $data = do_the_really_long_thing($id);
        \Drupal::cache()->set($cid, $data);
    }
    return $data;
}
```

- ▶ Durée de vie : Jusqu'à la purge des caches
- ▶ Identique à `cache_get()` `cache_set()`
- ▶ Maintenant dans un Service
- ▶ Stocké dans les bins de cache
- ▶ Base de données, memcache, Redis...

Les types de cache

1. Cache applicatif
 - ▶ Cache d'exécution
 - ▶ Cache statique
2. **Cache d'objets**
3. Cache de rendu
4. Cache HTTP

Cache d'objets

- ▶ Dans Drupal 7 :
 - ▶ Entity Cache
 - ▶ Commerce Entity Cache
 - ▶ Views Content Cache
 - ▶ ...
- ▶ Dans Drupal 8 : Drupal core suffit :)



Les types de cache

1. Cache applicatif
 - ▶ Cache d'exécution
 - ▶ Cache statique
2. Cache d'objets
3. **Cache de rendu**
4. Cache HTTP

Cache de rendu, la nouveauté Drupal 8

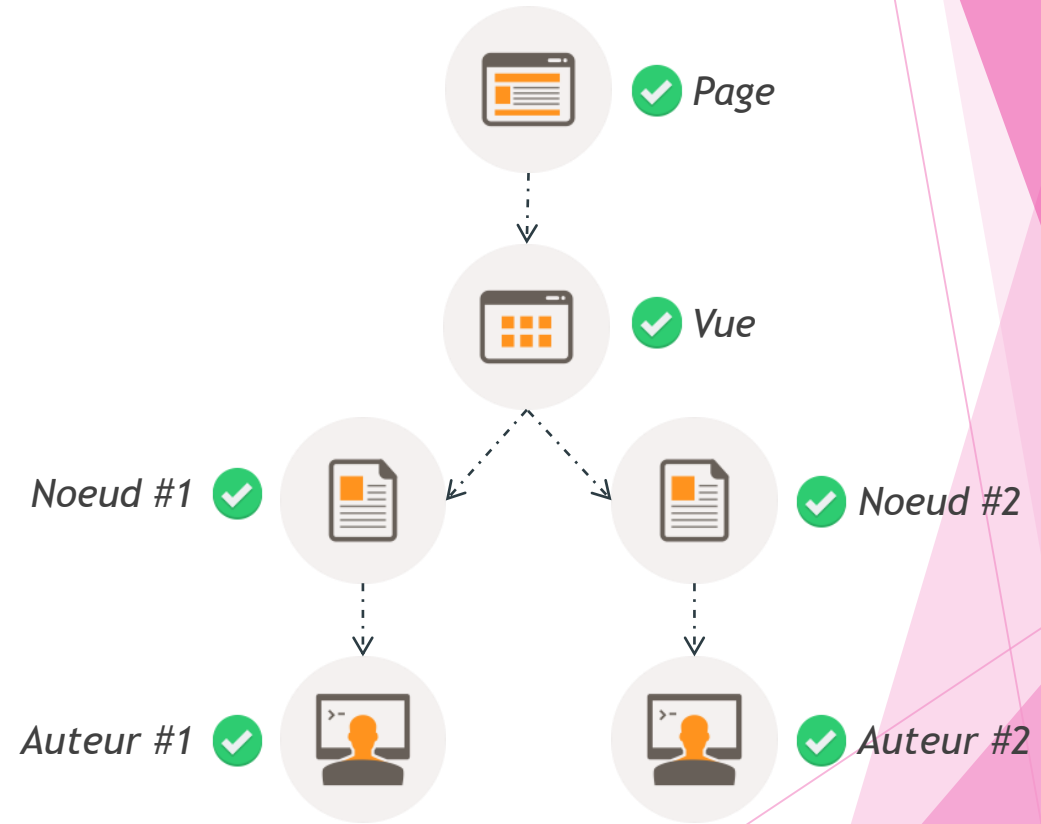


*Retrousssez vos manches,
ça va éclabousser.*

Cache de rendu



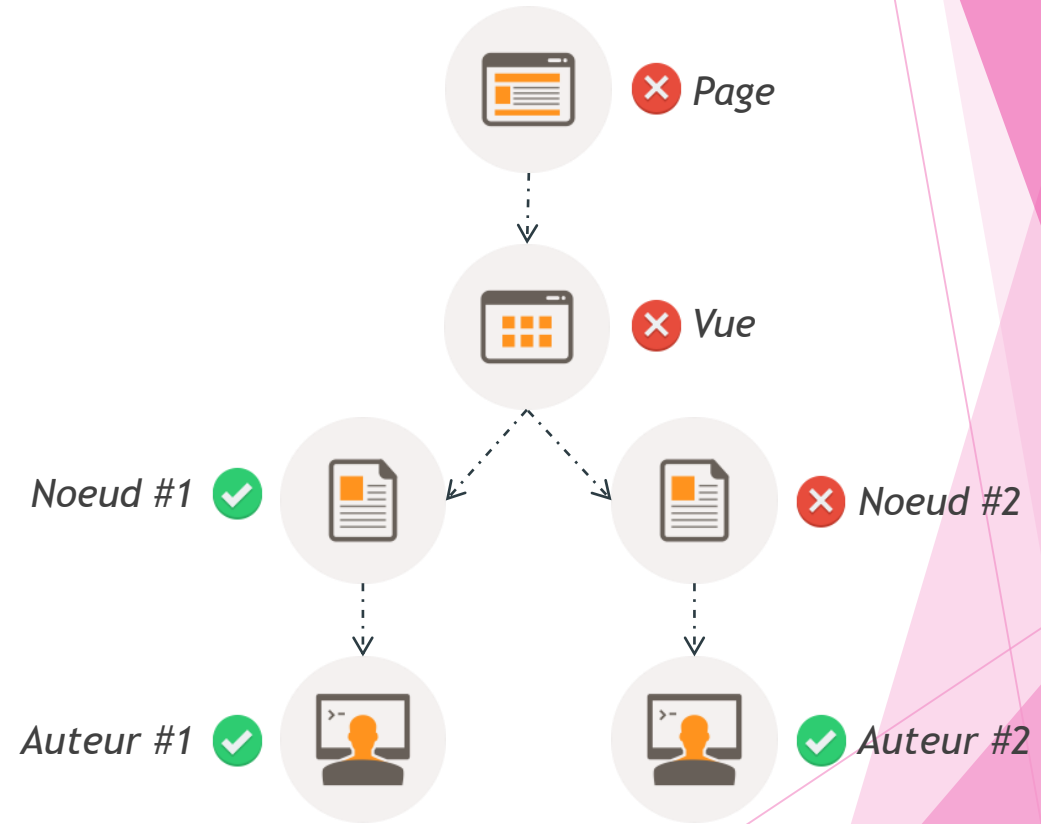
*Affichage de la
liste de noeuds*



Cache de rendu



*Mise à jour du
noeud #2*



Méta-données de cache

```
# Controller.php
return [
  'data' => [
    // Some data.
    '#cache' => [
      'keys' => ['entity_view', 'node', $node->id()],
      'contexts' => ['language'],
      'tags' => ['node:' . $node->id()],
      'max-age' => Cache::PERMANENT,
    ],
  ],
];
```

Méta-données : clés de cache (cache keys)

- ▶ Identifiant du cache pour le stockage / la récupération
- ▶ Devient un préfixe dans toutes les variantes du même contenu pour faciliter l'invalidation

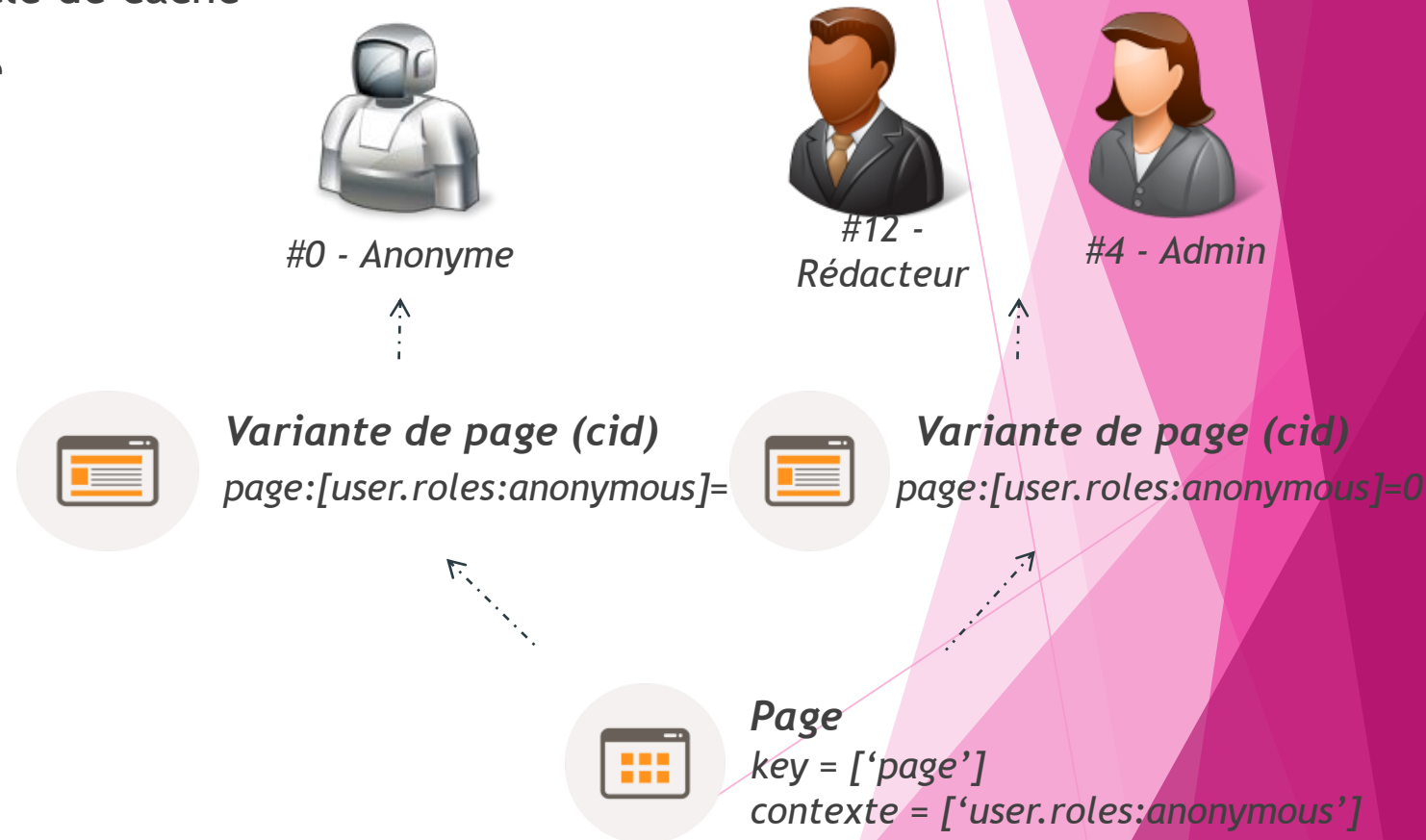
```
'#cache' => [  
  'keys' => [  
    'entity_view',  
    'user'  
  ],  
  // ...  
],
```

CID	DATA
entity_view:block:bartik_account_menu:...	// Rendu du bloc liens utilisateur de Bartik
entity_view:user:1:compact:...	// Rendu du profil de l'utilisateur 1
view:frontpage:display:page_1:...	// Rendu de la page d'accueil

Méta-données : contextes de cache

- ▶ Crée des variations au sein d'une clé de cache
- ▶ Le contexte ajoute une clé résolue dynamiquement

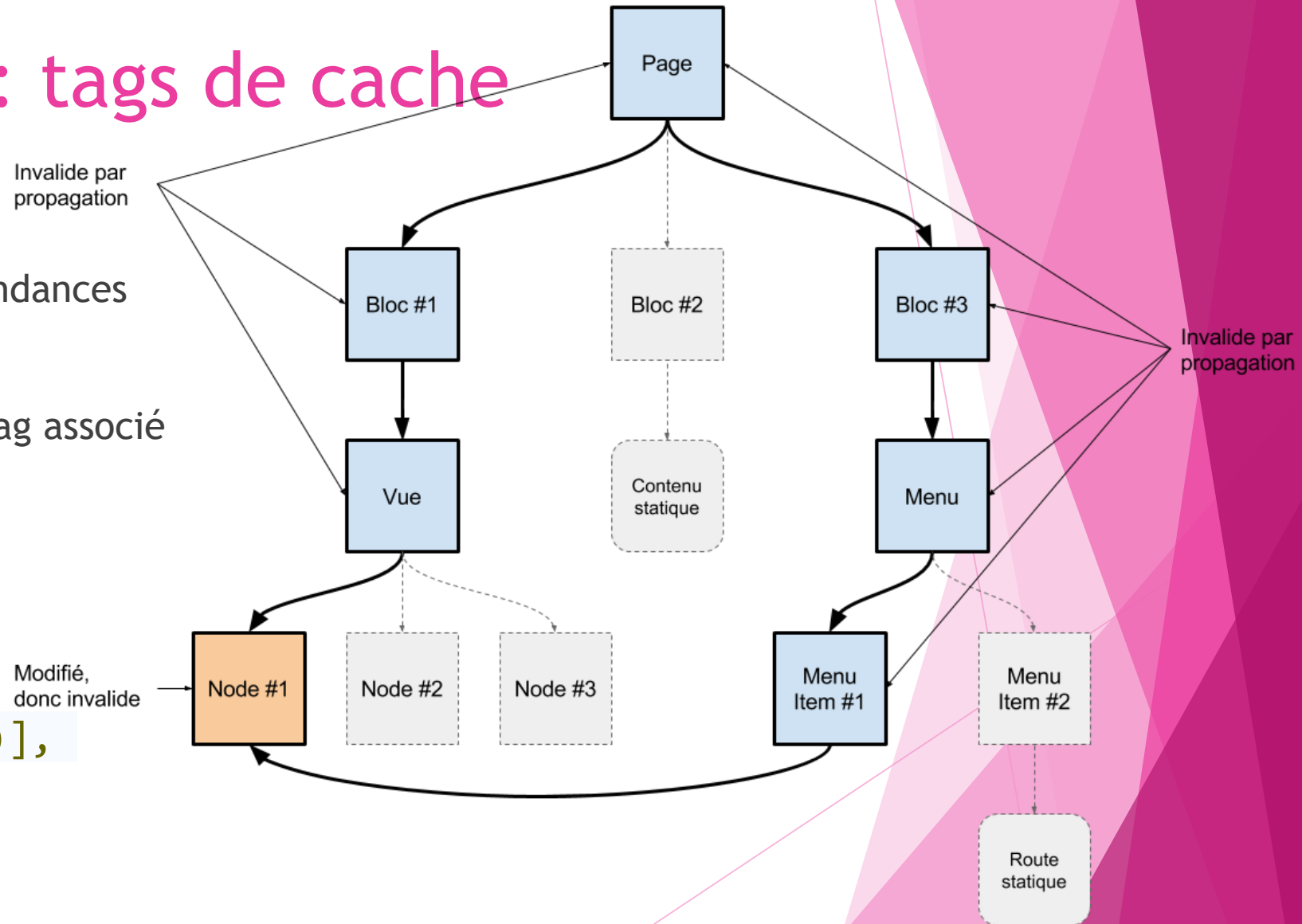
```
'#cache' => [  
  'keys' => 'page',  
  'contexts' => [  
    'user.roles:anonymous',  
  ],  
  // ...  
],
```



Méta-données : tags de cache

- Utilisé pour indiquer des dépendances entre des éléments
- Invalide l'élément lorsque le tag associé est modifié

```
'#cache' => [  
  'tags' => [  
    ['node:' . $node->id()],  
  ],  
  // ...  
],
```



Méta-données : âge d'expiration

- ▶ Expiration des données selon une durée de vie
- ▶ Valeur en secondes
 - ▶ 0 = jamais mis en cache
 - ▶ Cache :: PERMANENT = ne jamais faire expirer selon son âge
- ▶ Exemple : liste d'éléments aléatoire à faire varier toutes les 5 minutes

```
'#cache' => [  
  // ...  
  'max-age' => 300,  
],
```

Les types de cache

1. Cache applicatif
 - ▶ Cache d'exécution
 - ▶ Cache statique
2. Cache d'objets
3. Cache de rendu
4. **Cache HTTP**

Cache HTTP

- ▶ Enfin possible de jouer sur les en-têtes HTTP envoyés au navigateur
 - ▶ Cache-Control, Expires, Etag, Vary, Pragma, etc
- ▶ Les en-têtes sont compilés grâce aux méta-données
- ▶ Permet probablement de s'affranchir de la config du Reverse Proxy ou du CDN.

TROP COOOOOOOL !
ET AVEC DRUPAL 7, ON PEUT LE
FAIRE ?



Une approche similaire par les reverse-proxys

- ▶ Varnish invalidate tag
 - ▶ https://www.drupal.org/project/varnish_tag_invalidate
- ▶ Ajoute un en-tête HTTP avec les tags de la page
- ▶ Une requête PURGE sur un tag permet d'invalider les pages côté Varnish

OK !

UN AUTRE TRUC TROP COOL À ME
DIRE AVANT QUE JE NE PARTE ?



SMALL BALL





100%21:03

The awesomest new band: x

tres/node/1

My accountLog out

tres

Home

Home

Most listened to in the past week

1. PigPipe — Pink Ponies Pwn!

2. Pink Fluffy Unicorn — Rainbows are for losers

3. Llama Rock — KILL THE KITTENS!

Currently listening to...

Dries — Drupal song

The awesomest new band: PigPipe

Submitted by root on Thu, 09/17/2015 - 17:36



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur. Donec ut libero sed arcu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor. Pellentesque auctor nisi id magna consequat sagittis. Curabitur dapibus enim sit amet elit pharetra tincidunt feugiat nisl imperdiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros. Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.

Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae, sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit. id fringilla sem nunc vel mi. Nam dictum odio

ScreenFlow File Edit Mark Insert Font Actions View Window Help

test 1 | tres

tres/en/node/3

Manage Shortcuts root Edit

My account Log out

tres

Home language

test 1

Home

test 1

View Edit Delete Translate

Submitted by root on Mon, 01/04/2016 - 12:48
asdfsadf

No turbolinks

Add new comment

Subject

Comment *

Text format Basic HTML

About text formats

Restricted HTML
Allowed HTML tags: <a href hreflang> <cite> <blockquote cite> <code> <ul type> <ol start type> <dl> <dt> <dd> <h2 id> <h3 id> <h4 id> <h5 id> <h6 id>
Lines and paragraphs break automatically.
Web page addresses and email addresses turn into links automatically.

Save Preview

IMPEC ! BON ET BIEN IL NE ME
RESTE PLUS QU'À APPRENDRE À
NOMMER MES VARIABLES...

