



DUBLIN

DRUPALCON

2016-09-28





Routing in Drupal 9, and lessons learned in Drupal 8

Peter Wolanin
drupal.org/u/pwolanin

Track: Core Conversations

events.drupal.org/dublin2016/sessions/routing-drupal-9-and-lessons-learned-drupal-8

This is a Conversation

If I get through all the slides we should still have time for conversation.

If you want to start talking or disagreeing in the middle - even better!

In either case - you need to use the mic so people not in the room can hear you on the recording!

WSCCI Where is all began

“The Web Services and Context Core Initiative (WSCCI) aims to transform Drupal from a first-class CMS to a first-class REST server with a first-class CMS on top of it. To do that, we must give Drupal a unified, powerful context system that will support smarter, context- sensitive, easily cacheable block-centric layouts and non-page responses using a robust unified plugin mechanism.”

<https://groups.drupal.org/wsccli>

WSCCI Where is all began

Later just the Web Services Initiative - Context replaced by Container

- Symfony2 kernel (subclass), container, routing, session
- Symfony2 request/response objects
- Rest API in core, HTTP client (Guzzle)

Some Goals and Fallout for Drupal 8 Routing

- Convert `hook_menu` into Symfony2 routing (or the like)
- Replace paths with route names for rendering links
- All the hard work from those - including converting all page callbacks to controllers
- “fallout” included new breadcrumb system, new menu link system, conversion of local tasks and actions to plugins

DX Changes

- hook_menu in Drupal 6 / 7 was very overloaded (routing, menu links, local tasks, local actions, contextual links)
- Split finally all the pieces from hook_menu into YAML files
- MM.routing.yml MM.links.menu.yml
- MM.links.task.yml MM.links.action.yml
- MM.contextual.yml

D7 hook_menu

```
$items['node/%node/revisions'] = array(  
  'title' => 'Revisions',  
  'page callback' => 'node_revision_overview',  
  'page arguments' => array(1),  
  'access callback' => '_node_revision_access',  
  'access arguments' => array(1),  
  'weight' => 2,  
  'type' => MENU_LOCAL_TASK,  
  'file' => 'node.pages.inc',  
);
```


D8 node.routing.yml

```
entity.node.version_history:  
  path: '/node/{node}/revisions'  
  defaults:  
    _title: 'Revisions'  
    _controller: '\Drupal\node\Controller\NodeController  
      ::revisionOverview'  
  requirements:  
    _access_node_revision: 'view'  
    node: \d+  
  options:  
    _node_operation_route: TRUE
```

Problems

Discoverability:

Looking at `url()` and seeing path, vs. having to use something like Drupal console to find the path and visit in the browser

Discovering the route from the web page - visiting a page (but not always easy in D7 either)

Compare Generating a URL

Drupal 7: `url('node/1')` or `url('node/' . $nid, $options)`

Drupal 8 usually use `\Drupal\Core\Url`

`Url::fromUri('https://www.drupal.org')`

`Url::fromUri('entity:node/1', $options)`

`Url::fromRoute('entity.node.canonical', ['node' => 1])->toString()`

Performance Win in 8

Performance of access lookups can be much better by having the route parameters and route name explicit - easy to load the route

In Drupal 6 and 7 node menu links had a special hack to shortcut the routing during access checking. This was not extensible to the many entity types we have in Drupal 8. We needed something like route names

DX Wins?

With bare path can't determine which parts are parameters; is the route machine name and parameters better DX?

Reordering of route parameters and changing path, without impacting links - will it be used?

DX Wins?

Ability to match multiple HTTP methods at one path with different routes, or use different controllers for different “page variants”

We can have paths like `/node/{node}` with a `\d+` requirements and a path like `/node/{uuid}` with a `[0-9a-d-]+` requirement that will both be matched and then filtered in PHP

Could We Have Done it Differently?

What would a “path” based Drupal 8 have looked like?

Could we have used paths with slugs as Route names?

```
url('/node/{node}', ['node' => 1])
```

Maybe by 9 we can mostly use `Url::fromUri('entity:foo/1')`?

Is there a more compact format for e.g. Twig filter without access check?

Could We Have Done it Differently?

What about matching different HTTP methods or header-based content?

HTTP methods could possible have been supported by appending some suffix to the path (route name)?

header-based content negotiation is still supported by no longer used by Drupal 8 due to proxy interference

Lessons Learned?

Conversion of path to route created a lot of problems and additional work - extended Drupal 8 release by a year+?

The follow-up work and fallout work seems to have been somewhat ignored or underestimated?

Lessons Learned?

The gains in terms of different controllers for HTTP methods for REST api might have been achieved with a less elegant solution but much less work.

We had to use Drupal algorithms behind Symphony facades, so not clear it was a win to adopt those components - what is the gain?

Routing in Drupal 9.x?

There is a lot of work left if we want to convert e.g. all paths in tests to route names or entity URIs

We will see if real sites use the feature of changing paths

Are modules (panels) going to use the route filtering? What can we remove while retaining the useful features?

Routing in Drupal 9.x or 8.x?

```
Drupal\Core\StackMiddleware\Session::handle  
Drupal\Core\StackMiddleware\KernelPreHandle::handle  
Drupal\page_cache\StackMiddleware\PageCache::pass  
Drupal\page_cache\StackMiddleware\PageCache::handle  
Drupal\Core\StackMiddleware\ReverseProxyMiddleware::handle  
Drupal\Core\StackMiddleware\NegotiationMiddleware::handle  
Stack\StackedHttpKernel::handle  
Drupal\Core\DrupalKernel::handle
```

Start Here



Now find the route?

```
Drupal\Core\Routing\RouteProvider::getRouteCollectionForRequest  
Symfony\Cmf\Component\Routing\NestedMatcher\NestedMatcher::matchRequest  
Symfony\Cmf\Component\Routing\DynamicRouter::matchRequest  
Symfony\Cmf\Component\Routing\ChainRouter::doMatch  
Symfony\Cmf\Component\Routing\ChainRouter::matchRequest  
Drupal\Core\Routing\AccessAwareRouter::matchRequest  
Symfony\Component\HttpKernel\Event\Listener\RouterListener::onKernelRequest  
Drupal\Component\EventDispatcher\ContainerAwareEventDispatcher::dispatch  
Symfony\Component\HttpKernel\HttpKernel::handleRaw  
Symfony\Component\HttpKernel\HttpKernel::handle
```



Routing in Drupal 9.x or 8.x?

Can we reduce the overhead of the router or use less (Symfony) code? Maybe we can do that now? BC?

There are multiple layers of indirection in the critical path to routing - at the least this is not well-documented or clear
<https://www.drupal.org/developing/api/8/routing>

Do we really need to use compiled Symfony routes for matching? Much of the regex and other work in that code is redundant.



DUBLIN
DRUPALCON

JOIN US FOR
CONTRIBUTION SPRINTS
FRIDAY, SEPTEMBER 30

First Time Sprinter Workshop - 9:00-12:00 - Wicklow 2A

Mentored Core Sprint - 9:00-18:00 - Wicklow Hall 2B

General Sprints - 9:00 - 18:00 - Wicklow Hall 2A



DUBLIN
DRUPALCON

WHAT DID YOU THINK?

Evaluate This Session

events.drupal.org/dublin2016/schedule

THANK YOU!

