

TYPO3 Core - Bug #88860

RouteEnhancer with two actions of one extension on one PID not working

2019-07-30 12:52 - Julian Stock

Status:	Rejected	Start date:	2019-07-30
Priority:	Should have	Due date:	
Assignee:		% Done:	0%
Category:	Link Handling, Site Handling & Routing	Estimated time:	0.00 hour
Target version:		Complexity:	
TYPO3 Version:	9	Is Regression:	
PHP Version:	7.2	Sprint Focus:	On Location Sprint
Tags:			

Description

Hey,

I'm trying to migrate/update my old TYPO3 v8 + realurl to TYPO3 v9 + routing.

The "old" way:

1. finanzlexikon.html?tx_news_pi1%5BoverwriteDemand%5D%5Bcategories%5D=1&cHash=391...
2. finanzlexikon.html?tx_news_pi1%5Bnews%5D=1963&cHash=54a...

became to

1. finanzlexikon/a.html
2. finanzlexikon/abgeltungssteuer.html

This is my configuration for realurl:

```
'72' => [
    [
        'GETvar' => 'tx_news_pi1[overwriteDemand][categories]',
        'valueMap' => \ACME\MyRealurl\News\FinanzlexikonCategory::getCategories(),
        'noMatch' => 'bypass'
    ],
    [
        'GETvar' => 'tx_news_pi1[news]',
        'lookupTable' => array(
            'table' => 'tx_news_domain_model_news',
            'id_field' => 'uid',
            'alias_field' => 'title',
            'useUniqueCache' => 1,
            'useUniqueCache_conf' => array(
                'strtolower' => 1,
                'spaceCharacter' => '-',
            ),
            'enable404forInvalidAlias' => true
        ),
    ],
],
```

My (not fully working) configuration for routing:

```
Finanzlexikon:
  type: Extbase
  extension: News
  plugin: Pil
  limitToPages: [72]
  routes:
```

```

- { routePath: '/{news-title}', _controller: 'News::detail', _arguments: { news-title: news
} }
- { routePath: '/{category-name}', _controller: 'News::list', _arguments: { category-name: o
verwriteDemand/categories } }
defaultController: 'News::list'
aspects:
news-title:
type: PersistedAliasMapper
tableName: tx_news_domain_model_news
routeFieldName: path_segment
category-name:
type: PersistedAliasMapper
tableName: sys_category
routeFieldName: slug

```

The result:

1. finanzlexikon.html?tx_news_pi1%5BoverwriteDemand%5D%5Bcategories%5D=1&cHash=391...
2. finanzlexikon.html?tx_news_pi1%5Bnews%5D=1963&cHash=54a...

became to

1. finanzlexikon/a.html
2. finanzlexikon.html?tx_news_pi1%5Bnews%5D=1963&cHash=290...

My conclusion:

It's currently not possible for TYPO3 handling the url rewriting correctly when having two actions of one plugin/controller on one page.

Can anybody confirm that? Technically this has to be possible, because without the RouteEnhancers everything is working as expected (without having nice looking URLs of course).

Thanks for reading!

Related issues:

Related to TYPO3 Core - Bug #87910: ExtbasePluginEnhancer won't work on same ...	New	2019-03-14
--	------------	-------------------

History

#1 - 2019-07-30 12:53 - Julian Stock

- Subject changed from *RouteEnhancer with two actions on one PID not working* to *RouteEnhancer with two actions of one extension on one PID not working*

#2 - 2019-07-30 12:57 - Julian Stock

When changing the defaultController to News::detail

1. finanzlexikon.html?tx_news_pi1%5BoverwriteDemand%5D%5Bcategories%5D=1&cHash=391...
2. finanzlexikon.html?tx_news_pi1%5Bnews%5D=1963&cHash=54a...

becomes to

1. finanzlexikon.html?tx_news_pi1%5BoverwriteDemand%5D%5Bcategories%5D=1&cHash=391...
2. finanzlexikon/abgeltungssteuer.html

It seems that it's currently only possible for one parameter to be rewritten.

#3 - 2019-08-07 13:27 - Julian Stock

- Related to *Bug #87910: ExtbasePluginEnhancer won't work on same page or without page limitation added*

#4 - 2019-08-15 12:25 - Christian Wellinghorst

I have a similar problem (with a custom extension as well as with news) and I'm currently doubting myself - I have done a lot of searching around the core and I can't understand how the example configuration of news could ever work (or any configuration for any extension including routes for different controller actions for that matter).

I found that the URIs are generated in TYPO3\CMS\Core\Routing\PageRouter::generateUri. In here, for extbase extensions, TYPO3\CMS\Extbase\Routing\ExtbasePluginEnhancer::enhanceForGeneration is called. For each route defined for the given plugin, the parameters are verified by TYPO3\CMS\Extbase\Routing\ExtbasePluginEnhancer::verifyRequiredParameters

Here's the point I don't get: This method returns false if there is no default controller set for the route. It also returns false if the given controller and action parameters don't match the default controller and action. Thus, "routePath: '{news-title}'" from the example configuration of the news plugin can *never* work, as the default controller (and action for that matter) in the example configuration is "defaultController: 'News::list'".

If I change the default controller - as Julian did - rewriting for detail page links works, but category page links (e.g.) do not work as they require the list action. If I remove the default controller nothing will work at all.

Finally, if I add MULTIPLE route enhancers (e.g. one for the detail pages, one for pagination) everything works as expected. So my guess it that not multiple parameters are the problem, but mixing multiple routes with different controller actions.

I'm currently using TYPO3 9.5.8 but I didn't see any relevant changes to TYPO3\CMS\Extbase\Routing\ExtbasePluginEnhancer in the 9.5 branch in GIT, so I'm assuming this behaviour happens in all versions (at least within the 9.5 branch).

#5 - 2019-08-16 13:54 - Robert Wolle

In ExtbasePluginEnhancer.php => enhanceForGeneration method the variable \$originalParameters[\$this->namespace] includes no controller or action, which are defined in config.yaml of site configuration. Therefore hasControllerActionValues gives back false and the defaultController is set always.

I tried to dig into, with no results yet.

@Christian: Yes, I can confirm, that all versions of the 9.5 branch behave like that. I tried each version.

Update:

Now I know why it's like that. The method enhanceForGeneration receives the original parameters, which includes all news related parameters (news, category, year, month, etc.), but no controller and no action.

This is what you'll get out of the \$originalParameters:

```
Array
(
    [tx_news_pil] => Array
        (
            [news] => 123
        )
)
```

And this is how it has to be for tx_news related links:

```
Array
(
    [tx_news_pil] => Array
        (
            [news] => 123
            [controller] => News
            [action] => detail
        )
)
```

or

```
Array
(
    [tx_news_pil] => Array
        (
            [@widget_0] => Array
                (
                    [currentPage] => 3
                )
            [controller] => News
            [action] => list
        )
)
```

That's why the "defaultContoller" setting is set (no \$target['controller'] or \$target['action']).

The core seems to generate URL's without controller and action for an extension.

Solution for tx_news:

You have to set "plugin.tx_news.settings.link.skipControllerAndAction = 0" in your typoscript of the news extension. With this setting the controller and the action will be generated in the url.

Solution for extensions in general:

```
plugin.tx_anyextension.features.skipDefaultArguments = 0
```

Summary:

The controller and the action has to be in the url for generating speaking urls for extensions, if you have more than one action. skipDefaultArguments should always be 0.

Have a nice weekend.

#6 - 2019-08-19 09:51 - Christian Wellinghorst

[Robert Lemke](#): thanks for digging into this. It appears I was heavily misguided by the "default controller and action" in TYPO3\CMS\Extbase\Routing\ExtbasePluginEnhancer::verifyRequiredParameters.

For everyone stumbling across this issue (maybe it might help someone thinking the same way as I did): the line

```
$controller = $route->getDefault('_controller');
```

is highly misleading. It (in most cases) does *not* return the default controller/action from the YAML configuration and it obviously *never* returns the controller from the route configuration. It rather returns the parameters passed in the request and (if none are given) falls back to those configured in YAML (defaultController).

Robert's hint is absolutely correct - you should make sure never to use "skipDefaultArguments" (or similar) when using the new routing system.

If a core team member should happen to read this: I did not find a hint for this anywhere when reading about the routing system, so it would be good to mention it here (as it's still supported in extbase).

Furthermore, maybe it's possible to rename or redesign the "getDefault"-method of route objects. I would usually *never* expect to receive parameters passed by URL when calling a "getDefault"-method.

#7 - 2019-08-21 12:19 - Julian Stock

Robert Wolle wrote:

[...]

Solution for tx_news:

You have to set "plugin.tx_news.settings.link.skipControllerAndAction = 0" in your typoscript of the news extension. With this setting the controller and the action will be generated in the url.

[...]

Thanks, that solved my problem!

Christian Wellinghorst wrote:

[...]

I would usually never expect to receive parameters passed by URL when calling a "getDefault"-method.

[...]

I'm totally with you!

#8 - 2019-09-09 11:04 - Christian Wellinghorst

Hoping that some day this ticket might be read by someone with a good overview of the routing system as well as the ability and will to make some adjustments... ;-)

I found another (in my opinion) major drawback regarding this: in earlier versions of typo (using RealURL) if you used "skipDefaultArguments" and used the "redirect"-method within a controller (with those default arguments) no parameters were appended to the URL. This appears to be not possible anymore. I'm having the problem within an extension - i just used to redirect to a list view. Now, if I want to keep the URL "clean", i would have to build the URL myself - previously, this was not necessary.

I really like Typo3 in many ways and I admire the work the community is doing, but with every major version and every major feature it just seems that a lot of those new features are not thought through well enough and it takes forever to adjust them to a level where they really *work* well (just think of how long it took for extbase / fluid to get to a level where you could call them "usable"...).

Something similar seems to be happening to the routing system now. Some aspects are well thought through and work well, other things have just been completely left out. Considering the fact that there *was* a pretty well-working system like RealURL (which of course also had it's upsides and downsides like any system), the goal should have been to keep up with the features realurl offered and maybe introduce some improvements. Of course it's a hard task, but right now, we're left with a concept where some things just don't work at all (like leaving "skipDefaultArguments" enabled),

where no autoconfiguration is possible (at least there is nothing in the core which would allow that - you have to write your own functions), where there's no real standard for slug generation and what not.

Please don't get me wrong - I don't want to say that the work the core team is doing is bad in any way, I'm just suggesting a change of mind regarding the roadmaps for future Typo versions - maybe just put *less* new features and overhauls into new versions but take more time for single features so that the first "stable" version can really be considered "finished".

#9 - 2019-09-09 11:07 - Susanne Moog

Without looking further into it currently, I just wanted to leave the official docs link in here:

<https://docs.typo3.org/m/typo3/reference-coreapi/master/en-us/ApiOverview/Routing/AdvancedRoutingConfiguration.html#extbase-plugin-enhancer>
(at least the skipDefaultArguments is documented)

#10 - 2020-01-12 12:46 - Susanne Moog

- *Sprint Focus set to On Location Sprint*

#11 - 2020-01-20 11:43 - Oliver Hader

Christian Wellinghorst wrote:

[Robert Lemke](#): thanks for digging into this. It appears I was heavily misguided by the "default controller and action" in TYPO3\CMS\Extbase\Routing\ExtbasePluginEnhancer::verifyRequiredParameters.

For everyone stumbling across this issue (maybe it might help someone thinking the same way as I did): the line [...]

is highly misleading. It (in most cases) does *not* return the default controller/action from the YAML configuration and it obviously *never* returns the controller from the route configuration. It rather returns the parameters passed in the request and (if none are given) falls back to those configured in YAML (defaultController).

If a core team member should happen to read this: I did not find a hint for this anywhere when reading about the routing system, so it would be good to mention it here (as it's still supported in extbase).

Furthermore, maybe it's possible to rename or redesign the "getDefault"-method of route objects. I would usually *never* expect to receive parameters passed by URL when calling a "getDefault"-method.

I fail to understand how you conclude from having defaultController in the configuration to access it as `_controller` - I think that was a wrong assumption you did during debugging the scenario, please don't blame it on the core team. The `_controller` argument is a concept of Symfony and the part on "parameters passed by URL" is just wrong.

In case you'd opt for having the behavior of `getDefault` to be changed, feel free to open a corresponding request at the Symfony project (please see <https://symfony.com/doc/current/cmfbundle/book/routing.html#getting-the-controller-and-template> as reference).

#12 - 2020-01-20 22:24 - Oliver Hader

Basically, multiple Extbase routes are working, see

<https://github.com/HofUniversityIndieOne2019/book-store-project/blob/master/config/sites/book-store/config.yaml> as an example.

Based on the initial report and its configuration examples - there are some remarks:

```
routes:
  - { routePath: '/{news-title}', _controller: 'News::detail', _arguments: { news-title: news } }
  - { routePath: '/{category-name}', _controller: 'News::list', _arguments: { category-name: overwriteDemand/categories } }
```

Both parameters are mapped to corresponding database records. However, only the first route will be applied during route resolving in case mapped results are not distinct.

Imagine there's a category "News" and a news entity with "News" as well - thus `finanzlexikon/news.html` could be resolved to two controller-action pairs, in Symfony routing the first wins.

#13 - 2020-01-20 22:24 - Oliver Hader

- *Status changed from New to Rejected*