

TYPO3.Flow - Bug # 43588

Status:	Closed	Priority:	Should have
Author:	Bastian Waidelich	Category:	MVC - Routing
Created:	2012-12-04	Assigned To:	Bastian Waidelich
Updated:	2012-12-10	Due date:	
PHP Version:	5.4		
Has patch:	No		
Complexity:			
Affected Flow version:	(any)		
Subject:	Route defaults can be omitted when creating URIs		
Description	<p>The intended behavior is following:</p> <p>When generating an URI using the UriBuilder, a route only matches the specified routeValues if they are all equal to the defaults of the route. So given the following route:</p> <pre>- uriPattern: 'foo' defaults: bar: 'baz'</pre> <p>(@package, @controller etc left out intentionally to keep it simple)</p> <p>following should be the case:</p> <pre>Router::resolve(array('bar' => 'baz')); // resolves to "foo" Router::resolve(array('bar' => 'xyz')); // does NOT resolve Router::resolve(array()); // does NOT resolve</pre> <p>But currently this is the behavior:</p> <pre>Router::resolve(array('bar' => 'baz')); // resolves to "foo" Router::resolve(array('bar' => 'xyz')); // does NOT resolve Router::resolve(array()); // resolves to "foo"</pre> <p>Meaning: you can omit default values (if they are not part of the uriPattern).</p> <p>This works fine in most cases, but it makes it quite hard to get everything right, because too many routes match and it's hard to comprehend.</p> <p>I already work on a fix, but it has two drawbacks I'd like to get your feedback for:</p> <ol style="list-style-type: none">1. It's a breaking change and you'll have to check all your routes after applying the fix.. <p>I did that with a quite large project and most routes still worked out of the box. Except for the pagination routes:</p> <pre>- name: 'Articles with optional pagination' uriPattern: 'articles(/page/{--paginate.currentPage})' defaults: '@package': 'My.Package' '@controller': 'Article'</pre>		

```
'@action': 'index'  
'--paginate':  
  currentPage: '0'  
'@package': "  
'@subpackage': "  
'@controller': "  
'@action': 'index'
```

This route wouldn't resolve if `$routeValues['--paginate']['@action']` is not set to "index" from now on, so I had to create an additional route:

```
-  
name: 'Articles with pagination'  
uriPattern: 'articles/page/{--paginate.currentPage}'  
defaults:  
  '@package': 'My.Package'  
  '@controller': 'Article'  
  '@action': 'index'  
  '--paginate':  
    '@package': "  
    '@subpackage': "  
    '@controller': "  
    '@action': 'index'
```

```
-  
name: 'Articles'  
uriPattern: 'articles'  
defaults:  
  '@package': 'My.Package'  
  '@controller': 'Article'  
  '@action': 'index'
```

2. @action can't be omitted

The other drawback is the fallback behavior of the UriBuilder:

Currently, if you don't specify `@package` the package of the current request is used. The same for `@subpackage` and `@controller` (that's why `<f:link.action action="foo" />` links to the `fooAction` of the current controller.

For `@action` it's different currently: If you don't specify it, the `@action` is not set in the `$routeValues`.

Because of the current behavior a route specifying the `@action` will still resolve. With the fix that wouldn't be the case anymore..

I see two solutions:

A) Make the `actionName` argument of `UriBuilder::uriFor()` required (as well as the action arguments of `uri.action`, `link.action` and `form`).

B) Fallback to "index" (or detect the first action of the controller class) I'm adding this for completeness, but this would be quite error prone IMO..

Related issues:

duplicates TYPO3.Flow - Bug # 43589: Route defaults can be omitted when creat...

Resolved

2012-12-04

History

#1 - 2012-12-10 13:24 - Karsten Dambekalns

- Status changed from Accepted to Closed

Closing this duplicate