

TYPO3.Flow - Bug # 50395

Status:	Accepted	Priority:	Should have
Author:	Marco Falkenberg	Category:	MVC - Routing
Created:	2013-07-24	Assigned To:	Bastian Waidelich
Updated:	2013-10-14	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: Git master			
Subject:	Route cache caches routes for non dispatchable requests		
Description			
<p>The route cache caches all routes, even if the dispatchers cannot find the proper controller or action. On the one hand, that leads to a messy caching directory, if someone calls unavailable routes repeatedly. On the other the cached routes "block" potential identity routes, which occurs in my case.</p> <p>Assume the the following routes configuration:</p> <pre>- uriPattern: 'users/{user}' defaults: '@controller': 'User' '@action': 'index' routeParts: 'user': objectType: 'MyPackage\Model\User' uriPattern: '{name}' # Fallback route - uriPattern: '{@controller}/{@action}{.{@format}}' defaults: '@format': 'html'</pre> <p>If you now call /users/horst and the user horst is not present for now, the fallback routes resolves the request with controller users and action horst and a caching entry is made. The dispatcher then throws the appropriate exception and a 404 shows up.</p> <p>If you then create horst and call the url again, the 404 shows up (again), because the cached fallback route is used.</p>			

History

#1 - 2013-10-08 17:46 - Bastian Waidelich

- Status changed from New to Needs Feedback

The routing cache kicks in as soon as a route matches/resolves successfully. As the routing is (kind of) decoupled from the MVC dispatching, I don't know how to solve this otherwise..

In general I would consider it bad practice to have to use a fallback route with **{@controller}** placeholder.. This comes with Flow in order to get started quickly, but usually you want to explicitly configure the routes for supported controllers of your app.

But even then the problem is the same with **{@action}**.. Do you have a suggestion on how to solve this?

#2 - 2013-10-14 10:41 - Marco Falkenberg

Mmmm... certainly the decoupled setting is well-chosen by design and so i have to deal with this drawback. Maybe i choose to replace the fallback route and in case of the users i define some routes like this: (What do you think?)

```
-  
uriPattern: 'user/new'  
defaults:  
  '@controller': 'User'  
  '@action': 'new'  
  
-  
uriPattern: 'user'  
defaults:  
  '@controller': 'User'  
  '@action': 'index'  
  
-  
uriPattern: 'users/{user}/{@action}'  
defaults:  
  '@controller': 'User'  
  '@action': 'show'  
routeParts:  
  'user':  
    objectType: 'MyPackage\Model\User'  
    uriPattern: '{name}'
```

#3 - 2013-10-14 11:09 - Bastian Waidelich

- Status changed from Needs Feedback to Accepted
- Assigned To set to Bastian Waidelich

Hi Marco,
thanks for your feedback.

We're currently working on automatic redirect-support for routes that might partially solve your problem.

I thought about the issue again during the weekend and maybe there is a solution...:
If we don't persist the cache directly but on `__sleep()` we could intercept that in the 404 handling.. I'll give this a shot asap.

#4 - 2013-10-14 12:00 - Marco Falkenberg

Hi Bastian.

Sounds good to me. Thanks for your infos!

Cheers

