

TYPO3.Flow - Bug # 6902

Status:	Resolved	Priority:	Should have
Author:	Robert Lemke	Category:	Object
Created:	2010-03-17	Assigned To:	Karsten Dambekalns
Updated:	2010-10-20	Due date:	
PHP Version:			
Has patch:			
Complexity:			
Affected Flow version:			
Subject:	\$objectName is not meaningful enough in serializeObjectAsPropertyArray()		
Description			
<p>The method ObjectSerializer::serializeObjectAsPropertyArray() used \$objectName as the key for its object registry to cache objects it already has serialized. From looking at the code this might be not clear enough because there can be two instances of the same type which need to be serialized.</p> <p>Instead of \$objectName it might be better to use an SPLObjectStorage as the internal registry and change the signature to</p> <pre>public function serializeObjectAsPropertyArray(\$object)</pre>			

Associated revisions

Revision b78ada4e - 2010-03-18 12:13 - Karsten Dambekalns

[~TASK] FLOW3 (MVC): Fixed doc comment in RequestHandlerInterface, getPriority() is not expected to throw LogicException.

[+BUGFIX] FLOW3 (Object): The ObjectSerializer now uses a unique hash for all objects, fixes #6902.

History

#1 - 2010-03-17 20:48 - Andreas Förthner

Hi,

the object name is only used for top-level objects. For all objects below that we use the current object hash as key. The mentioned problem occurs only if the top-level object (the one with the @scope session) is ambiguous, i.e. is not a singleton. Does it make sense to store a non-singleton in the session as a top-level object? What would you expect to get, if you call the object manager for a prototype that lies in the session? Which one of those that might be there would you get?

I think the parameter might be a bit misleading as the object name is often the object hash, but an spl_object_storage should not be needed here.

Tell me if I'm wrong ;-)

Greets Andi

#2 - 2010-03-17 21:21 - Karsten Dambekalns

- Status changed from New to Accepted
- Assigned To set to Karsten Dambekalns
- Target version set to 1.0 alpha 8

- Start date set to 2010-03-17

The object serializer might be useful at some point even outside of @scope session use, thus we should stay on the safe side here. The name parameter is not needed, thus the signature could be simplified anyway. Clean code and all... :)

I'd also skip the SplObjectStorage, as we need to have all that in array form anyway. Rather use spl_object_hash() on "top-level" objects as is done for nested objects.

#3 - 2010-03-18 11:42 - Karsten Dambekalns

Ok, seemingly tricky...

Part I: The ObjectSerializer relied on the "object name" stored to detect top-level objects. When the name was registered with the object manager, it concluded it must be a top-level object. Well, easily solved by adding a flag instead - seems safer, too.

Part II : When getting back @scope session objects from the session the static object container relied on the object name to check if the (concluded top-level object) should be put into the object registry. Can be solved by getting the class name from the object and checking for it having session scope.

#4 - 2010-03-18 13:00 - Karsten Dambekalns

- Status changed from Accepted to Resolved

- % Done changed from 0 to 100

Applied in changeset r3991.