z2-Environment - Bug #2137

Bad invalidation of good resource in concurrency situation

02.08.2022 15:46 - Henning Blohm

Status:	Resolved	**Start date:**	02.08.2022
Priority:	Urgent	**Due date:**	
Assignee:	Henning Blohm	**% Done:**	0%
Category:	z2-core	**Estimated time:**	0.00 hour
Target version:	2.10		
origin:			

Description

The code in com.zfabrik.impl.resources.NamespaceImpl._getResourceHandleImplNonNull(String) may run into a concurrency situation, where a collected resource handle leads to a repeated tick() and invq adding and then invalidation of a concurrently just newly created resource.

Example

- Threads T1, T2.
- Resource name N

Flow:

1. T1: lookup for N returns value holder with null resource handle.
2. T1: Goes into looping = true
3. T1: leaves synchronized block
4. T2: runs tick() and enters synchronized block
5. T2: Creates and adds new resource handle
6. T2: leaves synchronized block
7. T1: enters synchronized block again
8. T1: finds that it is looping and the value holder for N is not in the invQ and hence puts it into invQ (although the resource handle is NOT null anymore!!!)
9. T1: invalidates the resource for N right away

Acceptance Criteria

- When looping for pending invalidations make sure not to accidentally consider a valid non-null handle as to be invalidated

Test Cases

- com.zfabrik.test.core.resources.tests.Invalidations.concurrentLookupOfCollectedResource: Concurrent lookup and GCing on weakly held resource.

History

#1 - 03.08.2022 11:47 - Henning Blohm

- Description updated

#2 - 03.08.2022 14:53 - Henning Blohm

- Status changed from In Progress to Feedback

#3 - 02.12.2022 10:10 - Henning Blohm

- Target version changed from 2.9.2 to 2.10

#4 - 14.05.2023 18:39 - Henning Blohm

- Status changed from Feedback to Resolved