

EDIT - bug #4962

task # 3336 (In Progress): Improve overall dataportal page loading performance

improve portal/classification/{taxon_uuid}/childNodesAt/{rank_uuid} web service

06/03/2015 02:00 PM - Andreas Kohlbecker

Status:	Resolved	Start date:	
Priority:	New	Due date:	
Assignee:	Andreas Müller	% Done:	0%
Category:	cdmlib	Estimated time:	0:00 hour
Target version:	Release 5.43	Found in Version:	
Severity:	normal		
Description			
the @portal/classification/{taxon_uuid}/childNodesAt/{rank_uuid} @-service is one of the slowest service endpoints.			
for example in the case of Bellis perennis: [classification/314a68f9-8449-495a-91c2-92fde8bcf344/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6/portal/classification/314a68f9-8449-495a-91c2-92fde8bcf344/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6] it takes about 2,5 seconds on a cdmsserver with 22 instances(see #4925)			
with fauna europaea this web service is extremely slow: http://cdm.biodiv.naturkundemuseum-berlin.de/cdm/cdmfauneur2/portal/classification/e66c33cc-47d8-4215-9946-7790d0d4889e/childNodesAt/b0785a65-c1c1-4eb4-88c7-dbd3df5aaad1.json about 15s as of 2015 July 14th			
this is a problem in cdmlib-services or in cdmlib-persistence.			
for documentation on the behaviour of this method see http://dev.e-taxonomy.eu/trac/wiki/CdmClassificationRankSpecificRootnodes			
performance improvements that have been applied in the past to this method			
<ul style="list-style-type: none">• f989e26 avoiding LEFT OUTER JOIN by splitting query in two• ba84f3f performance improvement for listChildNodes of a taxo			
Related issues:			
Related to EDIT - bug #7045: portal/classification/{uuid}/childNodesOf/{uuid}...		Feedback	
Related to EDIT - feature request #9271: Improve E+M dataportal performance		In Progress	

History

#1 - 07/14/2015 03:09 PM - Andreas Kohlbecker

- Keywords changed from performance, to performance, FaunaEuropaea,

#2 - 11/19/2015 11:14 PM - Andreas Kohlbecker

results of more detailed performance measures. In the following log the time (ms) of specific parts of the whole process are printed out, in the dao method two queries are run sequentially to fetch all nodes:

- service.listRankSpecificRootNodes() : total time of the service method, this includes the subprocesses
 - dao.listRankSpecificRootNodes() - query[x].list() : time of the first or second queries
 - dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() : execution of the bean initializer
- create JSON Object 392 : time taken by the JSONView for serialization

```
2015-11-19 22:59:50,975 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChildNodesAtRank() /portal/classification/f65ce83e-3856-40ad-b8e2-ba046f984f07/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 1275
```

```

dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 4840
dao.listRankSpecificRootNodes() - query[1].list() 1101
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.listRankSpecificRootNodes() 9426
create JSON Object 392
2015-11-19 23:00:14,704 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChildNodesAtRank() /portal/classification/f65ce83e-3856-40ad-b8e2-ba046f984f07/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 937
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 4067
dao.listRankSpecificRootNodes() - query[1].list() 1156
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.listRankSpecificRootNodes() 8285
create JSON Object 190
2015-11-19 23:00:37,157 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChildNodesAtRank() /portal/classification/f65ce83e-3856-40ad-b8e2-ba046f984f07/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 932
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 3338
dao.listRankSpecificRootNodes() - query[1].list() 1098
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.listRankSpecificRootNodes() 7424
create JSON Object 93

```

The bean initializer clearly dominates in these results. Therefore the next steps to improve this should be to

1. reduce or avoid the use of the initialization strategy in the controller.
2. let the Dao method return the TaxonNodeDTO directly instead of constructing it in the view. This DTO class is by the way named TaxonNodeDao which is wrong and thus should be changed.

#3 - 11/20/2015 02:06 PM - Andreas Kohlbecker

- Status changed from New to Resolved

- Target version changed from cdmllib - Next Major Release to Release 3.12

Improvements implemented:

- f36a078 [r26575 commenting performance measure print-out via Systemerr]* 256c8f8 r26574] avoiding bean initialization in getChildNodesAtRank() as much as possible - performance improvement

The attempt to avoid the use of an initialization strategy by adding JOIN FETCH statements to the hql statements in the dao but keeping the default initialization strategy '\$' already leads to a significant speed up:

before

- 7030.257 [ms](#)* 7965 [ms]

after

- 2898.261 [ms](#)* 5120 [ms]

further improvements can not be applied. First of all because of the name cache strategies which require a big part of the object graph to be loaded, secondly because of the TaxonNodeComparator which also needs detailed and atomized information on the taxon name and the authorship cache. I created a new ticket for this major topic [#5387](#)

#4 - 11/20/2015 02:23 PM - Andreas Kohlbecker

I think we can close this ticket since no further improvement is possible without solving [#5387](#)

#5 - 01/08/2016 01:32 PM - Andreas Kohlbecker

the commits which removed the init strategy introduced a LIE, so i am re-adding the init strategy for taxonNames: 77db7b5 r26928

#6 - 01/11/2016 01:29 PM - Andreas Kohlbecker

2b1209a r26947 : using LEFT FETCH JOIN for ranks and sec fixed problems and improves performance (all times in ms):

tested with euro+med_

##before

```

2016-01-11 10:55:55,975 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChild

```

```
dNodesAtRank() /portal/classification/314a68f9-8449-495a-91c2-92fde8bcf344/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 252
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 455
dao.listRankSpecificRootNodes() - query[1].list() 1
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.pageRankSpecificRootNodes() - Collections.sort(results, taxonNodeComparator) 66
service.listRankSpecificRootNodes() 808
```

```
## Using LEFT JOIN FETCH for n.rank and t.sec
2016-01-11 10:57:38,471 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChildNodesAtRank() /portal/classification/314a68f9-8449-495a-91c2-92fde8bcf344/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 279
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 340
dao.listRankSpecificRootNodes() - query[1].list() 1
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.pageRankSpecificRootNodes() - Collections.sort(results, taxonNodeComparator) 45
service.listRankSpecificRootNodes() 695
```

tested with fauna europaea:

```
##before
2016-01-11 11:02:07,901 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChildNodesAtRank() /portal/classification/7d85033b-2f6b-49d3-a9b7-073f2756f78d/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 2747
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 1215
dao.listRankSpecificRootNodes() - query[1].list() 1811
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.pageRankSpecificRootNodes() - Collections.sort(results, taxonNodeComparator) 211
service.listRankSpecificRootNodes() 10016
```

```
## Using LEFT JOIN FETCH for n.rank and t.sec
2016-01-11 11:03:24,429 INFO [eu.etaxonomy.cdm.remote.controller.ClassificationPortalListController] - getChildNodesAtRank() /portal/classification/7d85033b-2f6b-49d3-a9b7-073f2756f78d/childNodesAt/af5f2481-3192-403f-ae65-7c957a0f02b6.json
dao.listRankSpecificRootNodes() - query[0].list() 2847
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 958
dao.listRankSpecificRootNodes() - query[1].list() 1834
dao.listRankSpecificRootNodes() - defaultBeanInitializer.initializeAll() 0
service.pageRankSpecificRootNodes() - Collections.sort(results, taxonNodeComparator) 175
service.listRankSpecificRootNodes() 5854
```

#7 - 11/02/2017 08:42 AM - Andreas Kohlbecker

- Related to bug #7045: portal/classification/{uuid}/childNodesOf/{uuid} service slow added

#8 - 11/02/2017 09:13 AM - Andreas Kohlbecker

- Description updated

- Private changed from Yes to No

#9 - 11/02/2020 09:51 PM - Andreas Müller

- Related to feature request #9271: Improve E+M dataportal performance added

#10 - 03/23/2022 10:27 AM - Andreas Müller

- Target version changed from Release 3.12 to Release 5.44

Please review and move back to 3.12 when done.

Also check if new implementation is needed (follow up ticket)

#11 - 03/23/2022 10:39 AM - Andreas Müller

See also [#7045](#) which indicates that the problem is not fully solved yet.

#12 - 04/12/2023 10:48 AM - Belen Escobari

- Target version changed from Release 5.44 to Release 5.37

#13 - 04/12/2023 11:13 AM - Andreas Müller

- Target version changed from Release 5.37 to Release 5.44

#14 - 02/14/2024 11:45 PM - Andreas Müller

- Target version changed from Release 5.44 to Release 5.43

#15 - 02/26/2024 05:13 PM - Andreas Müller

Andreas Müller wrote in [#note-11](#):

See also [#7045](#) which indicates that the problem is not fully solved yet.

Not sure if this is true. [#7045](#) is about another method (childnodes of nodes in the tree/siblings of nodes in the tree path).