EDIT - feature request #6554

The details views are newly created with every change of the focus

04/03/2017 04:07 PM - Katja Luther

Status:	In Progress	**Start date:**	
Priority:	Highest	**Due date:**	
Assignee:	Katja Luther	**% Done:**	0%
Category:	taxeditor	**Estimated time:**	0:00 hour
Target version:	Release 5.45		
Severity:	normal		

Description

With every focus change the details view is created completely new. this leads to performance problems.

===

We should test if it is not possible to reuse certain precomputed details views or details view elements to improve performance

Related issues:

Related to EDIT - bug #6809: Name Editor throws double selection change events	**Closed**
Related to EDIT - feature request #7040: [MASTER] Improve UI performance	**New**
Related to EDIT - feature request #7838: Optimize details view creation for s...	**Closed**
Related to EDIT - bug #10186: Problems with session handling in taxeditor	**Closed**
Related to EDIT - task #10187: Use only 1 service call to initialize suppleme...	**New**

Associated revisions

Revision 8da50b71 - 09/25/2018 12:00 PM - Patrick Plitzner

ref #6554 Avoid double destruction of details view sections

Revision 3b7513fa - 09/25/2018 03:12 PM - Patrick Plitzner

ref #6554 Avoid re-rendering of details view for same selection

Revision 9ef8582f - 09/26/2018 04:00 PM - Patrick Plitzner

ref #6554 Cache form section for taxon and derived unit details view

Revision 93246c1b - 09/26/2018 06:16 PM - Patrick Plitzner

ref #6554 Fix class cast exception

Revision 2e1e89cd - 09/27/2018 10:19 AM - Patrick Plitzner

ref #6554 Cache collection for entity collection sections

Revision 59916c08 - 09/27/2018 10:20 AM - Patrick Plitzner

ref #6554 Avoid double setInput() on managedForm

Revision 81215eb9 - 09/27/2018 11:00 AM - Patrick Plitzner

ref #6554 set cached collection to null when entity is null

Revision 18b043ef - 09/27/2018 11:01 AM - Patrick Plitzner

ref #6554 Implement update() for DeterminationDetailElement

Revision 90524380 - 09/27/2018 04:43 PM - Patrick Plitzner

ref #6554 Evaluate expansion state at widget creation time

Conflicts:

eu.etaxonomy.taxeditor.store/src/main/java/eu/etaxonomy/taxeditor/view/e4/details/DetailsViewerE4.java

Revision 1685391e - 09/27/2018 06:20 PM - Patrick Plitzner

ref #6554 Fix checkbox selection in update() method

Revision 25fe366b - 09/27/2018 06:21 PM - Patrick Plitzner

ref #6554 Implement update() for taxon details view sections

Revision 143dd6ba - 09/27/2018 06:24 PM - Patrick Plitzner

ref #6554 Rename ISelectable interface to avoid ambiguity for checkboxes

Revision 515cd3ec - 09/28/2018 10:07 AM - Patrick Plitzner

ref #6554 Cache NameDetailElement (experimental)

Revision ed60b90e - 09/28/2018 10:49 AM - Patrick Plitzner

ref #6554 Evaluate expansion state at widget creation time

Revision 69955d6c - 09/28/2018 01:25 PM - Patrick Plitzner

ref #6554 Revert caching of NameDetailElement

Revision ae2b6d7b - 09/28/2018 01:53 PM - Patrick Plitzner

ref #6554 Cache preferred terms in TermManager

Revision 9d0d72ef - 09/28/2018 01:54 PM - Patrick Plitzner

ref #6554 Cache preferred terms in TermManager

Revision 69957587 - 09/28/2018 02:25 PM - Patrick Plitzner

ref #6554 Evaluate expansion state for remaining elements

Revision ac4c5087 - 10/01/2018 09:49 AM - Patrick Plitzner

ref #6554 Pull up isCached() to abstract data viewer class

Revision bb6c8049 - 10/01/2018 10:17 AM - Patrick Plitzner

ref #6554 Evaluate expansion state for missing sections

Revision e12facbf - 10/01/2018 04:23 PM - Patrick Plitzner

ref #6554 Avoid unnecessary refreshs

Revision 15aa0a1c - 10/17/2018 11:37 AM - Patrick Plitzner

ref #6554 Fix initial expansion state

Revision 362c07ab - 10/17/2018 01:36 PM - Patrick Plitzner

ref #6554 Set initial expansion state

Revision cb7b8f2b - 02/21/2019 09:11 AM - Patrick Plitzner

ref #6554 Reduce details view rendering

- enhance check against previous selection
- remove unused methods

Revision 0db555e2 - 02/07/2024 11:55 AM - Katja

ref #6554: avoid ws for rank combo

Revision 66b8cde4 - 02/07/2024 11:55 AM - Katja

ref #6554: add dto comparator

Revision 4d1aeb6c - 02/07/2024 12:16 PM - Katja

ref #6554: add old dto combo method to formFactory

History

#1 - 04/04/2017 02:21 PM - Andreas Müller

- Tracker changed from bug to feature request

- Subject changed from the details view are newly created with every change of the focus to The details views are newly created with every change of the focus

- Description updated

#2 - 07/13/2017 11:27 AM - Patrick Plitzner

- Related to bug #6809: Name Editor throws double selection change events added

#3 - 10/27/2017 11:12 PM - Patrick Plitzner

- Related to feature request #7040: [MASTER] Improve UI performance added

#4 - 09/26/2018 11:34 AM - Patrick Plitzner

The performance loss is separated in two areas:

1. SWT/UI rendering
   - can be optimized by caching the UI widgets
2. cdm service layer calls
   - can be optimized by reducing service calls
   - We should also watch out for preference queries that contain service calls like e.g. PreferencesUtil.getGlobalLanguage()

#5 - 09/26/2018 11:35 AM - Patrick Plitzner

- Status changed from New to In Progress

- Assignee changed from Katja Luther to Patrick Plitzner

- Target version changed from Unassigned CDM tickets to Release 5.4

#6 - 09/28/2018 02:45 PM - Patrick Plitzner

Another bottleneck is

```
org.hibernate.collection.internal.PersistentSet.size()
org.hibernate.collection.internal.PersistentSet.iterator()
```

in some cases taking >90% of the rendering time of the supplemental data view

#7 - 09/28/2018 04:17 PM - Andreas Müller

Patrick Plitzner wrote:

> Another bottleneck is
>
> ```
> org.hibernate.collection.internal.PersistentSet.size()
> org.hibernate.collection.internal.PersistentSet.iterator()
> ```
>
> in some cases taking >90% of the rendering time of the supplemental data view

That is interesting. What exactly is it doing. Loading all data or only 1 request per Set to get the size?

I discussed this with Cherian already if we should have something like a count cache for persistend collections.

Or is it possible to combine the initialization or only the size-request for all supplemental data and not run it instance for instance?

#8 - 09/28/2018 04:18 PM - Andreas Müller

The count cache might be also something for the data portals

#9 - 10/22/2018 11:14 AM - Patrick Plitzner

- Related to feature request #7838: Optimize details view creation for section expansion added

#10 - 10/22/2018 11:14 AM - Patrick Plitzner

- Target version changed from Release 5.4 to Release 5.5

#11 - 01/30/2019 08:30 AM - Patrick Plitzner

- Target version changed from Release 5.5 to Release 5.6

#12 - 02/19/2019 03:41 PM - Andreas Müller

- Target version changed from Release 5.6 to Reviewed Next Major Release

#13 - 11/18/2019 03:02 PM - Andreas Müller

- Assignee changed from Patrick Plitzner to Katja Luther

#14 - 01/05/2023 05:45 PM - Andreas Müller

- Related to bug #10186: Problems with session handling in taxeditor added

#15 - 01/05/2023 05:45 PM - Andreas Müller

- Related to task #10187: Use only 1 service call to initialize supplemental data added

#16 - 02/07/2024 09:45 AM - Katja Luther

Patrick Plitzner wrote in #note-4:

> The performance loss is separated in two areas:
>
> 1. SWT/UI rendering
>   - can be optimized by caching the UI widgets
> 2. cdm service layer calls
>   - can be optimized by reducing service calls
>   - We should also watch out for preference queries that contain service calls like e.g. PreferencesUtil.getGlobalLanguage() -> this is already solved by using db preference cache

#17 - 02/07/2024 06:01 PM - Andreas Müller

- Tags set to performance

#18 - 02/15/2024 12:11 AM - Andreas Müller

- Target version changed from Reviewed Next Major Release to Release 5.45