

EDIT - bug #9683

Auto-initialize root bean

06/28/2021 09:35 AM - Andreas Müller

| | | | |
|--|----------------|--------------------------|-----------|
| Status: | Closed | Start date: | |
| Priority: | Highest | Due date: | |
| Assignee: | Andreas Müller | % Done: | 100% |
| Category: | cdmlib | Estimated time: | 0:00 hour |
| Target version: | Release 5.25 | Found in Version: | |
| Severity: | normal | | |
| Description | | | |
| Currently only related objects are auto-initialized by the AbstractBean.beanAutoInitializers . The root bean itself is not initialized this way. This leads to a situation where loading e.g. a field unit without any property path will throw an exception when fieldUnit.getTitleCache() is called outside the initializing transaction. | | | |
| Related issues: | | | |
| Related to EDIT - task #9678: Remove Derived- and FieldUnitFacadeCacheStrategies | | Closed | |
| Related to EDIT - bug #7874: TeamOrPersonBase entity can become unusable due ... | | Closed | |

Associated revisions

Revision 5348613b - 06/28/2021 09:39 AM - Andreas Müller

fix #9683 autoinitalize root bean

Revision bcbe5633 - 06/28/2021 11:28 AM - Andreas Müller

ref #9683 add loadProxy method to services and adapt test to work with current auto initialize settings

Revision c1e0cb15 - 06/29/2021 09:30 PM - Andreas Müller

ref #9683. ref #9664 fix tests in AgentServiceImplTest

Revision 61c33dfd - 07/01/2021 03:01 PM - Andreas Müller

ref #9683 adapt method name and javadoc

History

#1 - 06/28/2021 09:39 AM - Andreas Müller

- Status changed from New to Resolved
- % Done changed from 0 to 50

Applied in changeset [cdmlib|5348613baa59038220d4efd1e4a851e3ccdb6c58](#).

#2 - 06/28/2021 09:42 AM - Andreas Müller

- Related to task #9678: Remove Derived- and FieldUnitFacadeCacheStrategies added

#3 - 06/28/2021 09:45 AM - Andreas Müller

- Assignee changed from Andreas Müller to Andreas Kohlbecker

#4 - 06/28/2021 11:27 AM - Andreas Müller

- Related to bug #7874: TeamOrPersonBase entity can become unusable due to replacement of the title caches on using getters added

#5 - 06/28/2021 02:10 PM - Andreas Kohlbecker

- Status changed from Resolved to Feedback
- Assignee changed from Andreas Kohlbecker to Andreas Müller

I think the new method, which by the way is only used for test classes, is irritating:

```
public T loadProxy(int id);
```

The method name suggests that a proxy method is returned always. This is not the case as explained in the java doc of `org.hibernate.Session.load(Class<T> theClass, Serializable id)` : "This method **might** return a proxied instance that is initialized on-demand, when a non-identifier method is accessed."

This additional method, which is similar to other `find*(..)` methods makes the `Cdm*Dao-API` unclear as there are already many similar methods.

Can't you use `getSession().load()` in the test class directly or do you expect that the `loadProxy()` will be used in non test classes frequently? In the latter case I suggest renaming it to just

```
public T load(int id);
```

#6 - 06/28/2021 03:06 PM - Andreas Müller

- Assignee changed from Andreas Müller to Andreas Kohlbecker

Andreas Kohlbecker wrote:

I think the new method, which by the way is only used for test classes, is irritating:

```
public T loadProxy(int id);
```

The method name suggests that a proxy method is returned always. This is not the case as explained in the java doc of `org.hibernate.Session.load(Class<T> theClass, Serializable id)` : "This method **might** return a proxied instance that is initialized on-demand, when a non-identifier method is accessed."

This additional method, which is similar to other `find*(..)` methods makes the `Cdm*Dao-API` unclear as there are already many similar methods.

Can't you use `getSession().load()` in the test class directly or do you expect that the `loadProxy()` will be used in non test classes frequently? In the latter case I suggest renaming it to just

```
public T load(int id);
```

Yes, there is a need for this method if you want to attach an object to another one without initializing the object and without invoking cascade during save afterwards. This is for performance reasons and it is e.g. needed for imports.

We may discuss the label for the method. I first thought about `loadWithoutInitializing` but it is a bit long but maybe semantically better.

Purely "load(int)" has many disadvantages as all `service.load(...)` methods do explicitly initialize the return object(s) so it is very unexpected that only `load(int)` does not initialize (while eg. `load(uuid)` or `load(int, null)` do initialize).

Also the `service.loadProxy(int)` **will** always return a proxy except for the situation where it is used within another transaction. This is because it will only return a non-proxy if the object is already in the session which can only be the case if used within another transaction.

#7 - 06/28/2021 03:08 PM - Andreas Müller

My comment was about the method in service. Maybe in dao it is slightly different as it is always used within a session. As this is not API we could more easily rename it to `loadWithoutInitializing` as long method names are maybe not so relevant here.

#8 - 06/30/2021 03:09 PM - Andreas Kohlbecker

- Status changed from Feedback to Resolved

- Assignee changed from Andreas Kohlbecker to Andreas Müller

Ok, thank you for the explanation. I would prefer having the same method names in the DAP and service layer. We can keep `loadProxy(int)` if the java doc of the methods are comprehensive enough to fully understand the implications of these methods. You could use the text of your comment above for extending the java doc.

#9 - 07/01/2021 02:57 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

- % Done changed from 50 to 90

I finally decided to rename the methods (in service and persistence) to `loadWithoutInitializing` as this is semantically more correct. Also I slightly adapted the javadoc.

So I guess we can close the ticket or are there any other open issues?

#10 - 07/15/2021 09:02 AM - Andreas Kohlbecker

- *Status changed from Feedback to Closed*
- *Assignee changed from Andreas Kohlbecker to Andreas Müller*
- *% Done changed from 90 to 100*