

EDIT - bug #9422

Improve performance during Preferences startup

01/25/2021 05:26 PM - Andreas Müller

Status:	Closed	Start date:	
Priority:	Highest	Due date:	
Assignee:	Katja Luther	% Done:	0%
Category:	taxeditor	Estimated time:	0:00 hour
Target version:	Release 5.19	Found in Version:	
Severity:	critical		

Description

During startup the following function is called

CdmPreferenceCache.getAllTaxEditorDBPreferences by PreferenceUtil.updateDBPreferences.

This code loads the terms very unefficiently if terms are explicitly defined in the local or DB preferences by calling

```
List<DefinedTermBase> definedTermBases = termService.load(uuidList, null);
List<TermDto> dtos = new ArrayList<>();
for (DefinedTermBase<?> term: definedTermBases){
    dtos.add(TermDto.fromTerm(term));
}
```

So there is 1 or more webservice call(s) for EACH term as dtos.add requires lazy loading.

A simple solution could be to at least load all DTOs serverside within 1 service call for all termtypes or atleast for each termtype.

Another more dangerous solution could be to use property path, but this is error prone if things change or if a term types needs more loading then defined in the property path.

An even better solution (maybe for the future/another ticket) could be to load the preferences asynchronously and / or using a persisted cache via ehcache. However, these solutions are more complex.

===

In general we should find similar such issues e.g. by switching on CachingHttpInvokerProxyFactoryBean.measureDuration (by using VMarg -Dremoting.httpinvoker.measureDuration=true and setting logging to CachingHttpInvokerProxyFactoryBean to info) which shows you each httpinvoker call and its duration. During development this (or a similar functionality should always be used !!).

===

A last issue:

for CdmPreferenceCache.getAllTaxEditorDBPreferences() the code

```
PrefKey key = CdmPreference.NewKey(PreferenceSubject.NewTaxEditorInstance(), PreferencePredicate.AvailableDistributionStatus);
```

```
if (get(key) != null){
    if (!PreferencesUtil.getOverrideForPreference(PreferencePredicate.AvailableDistributionStatus.getKey()) || !get(key).isAllowOverride()){
        //get terms for the uuids... and add them to the termManager as preferred terms
        ITermService termService = CdmStore.getService(ITermService.class);
        List<UUID> uuidList = new ArrayList<>();
        if (get(key).getValue() != null){
            String[] uuidArray = findBestMatching(key).getValue().split(";");
            for (String uuidString:uuidArray){
                try {
                    uuidList.add(UUID.fromString(uuidString));
                } catch (Exception e) {
                    logger.warn("Preference loading failed", e);
                }
            }
        }
    }
}
```

```

    }
}

List<DefinedTermBase> definedTermBases = termService.load(uuidList, null);
List<TermDto> dtos = new ArrayList<>();
for (DefinedTermBase<?> term: definedTermBases){
    dtos.add(TermDto.fromTerm(term));
}
CdmStore.getTermManager().setPreferredTermsByType(dtos, TermType.PresenceAbsenceTerm);
}
}

```

is very redundantly used 3x. This should be deduplicated. Such c&p within the same class almost always shows that code deduplication should be applied. We should try to avoid this in future.

Related issues:	
Related to EDIT - bug #9423: Improve language handling for TermType.getMessage...	New
Related to EDIT - feature request #9427: Code cleaning in CdmPreferenceCache	Closed
Related to EDIT - feature request #9428: Asynchronous loading of terms	New

Associated revisions

Revision f8326308 - 01/26/2021 12:39 PM - Katja Luther

ref #9422: get termDtos from server

History

#1 - 01/25/2021 06:00 PM - Andreas Müller

- Priority changed from New to Highest

#2 - 01/26/2021 08:31 AM - Andreas Müller

- Related to bug #9423: Improve language handling for TermType.getMessage() and others added

#3 - 01/26/2021 08:34 AM - Andreas Müller

- Description updated

#4 - 01/26/2021 12:40 PM - Katja Luther

- Status changed from New to In Progress

Now the method to get dtos from serverside is used instead of getting the terms and create dtos on editor side.

#5 - 01/28/2021 11:51 AM - Andreas Müller

- Status changed from In Progress to Feedback

- Target version changed from Release 5.21 to Release 5.19

The first fix went into 5.19, correct?

So I think we should split the ticket and set this one to resolved?

#6 - 01/28/2021 11:55 AM - Katja Luther

- Status changed from Feedback to Closed

yes, the first fix (term loading at start) is fixed for 5.19. So I create a new ticket for the remaining issues and close this ticket.

#7 - 01/28/2021 12:12 PM - Katja Luther

- Related to feature request #9427: Code cleaning in CdmPreferenceCache added

#8 - 01/28/2021 12:18 PM - Katja Luther

- Related to feature request #9428: Asynchronous loading of terms added

#9 - 07/09/2021 03:12 PM - Andreas Müller

- Description updated