EDIT - bug #7021

CREATE permission not sufficient to save new TaxonName entity

10/17/2017 05:57 PM - Andreas Kohlbecker

Status:	Resolved	**Start date:**	
Priority:	Highest	**Due date:**	
Assignee:	Andreas Müller	**% Done:**	50%
Category:	cdmlib	**Estimated time:**	0:00 hour
Target version:	Release 4.11		
Severity:	critical	**Found in Version:**	

Description

Saving a newly created name entity fails if the authenticated use is only having the TAXONNAME.[CREATE,READ] authority.

in the cdm-vaadin:eu.etaxonomy.cdm.service.CdmStore.saveBean(..) method the the bean is saved by doing a merge:

```
public EntityChangeEvent saveBean(T bean) {

        Type changeEventType;
        if(bean.getId() > 1){
            changeEventType = Type.MODIFIED;
        } else {
            changeEventType = Type.CREATED;
        }

        Session session = getSession();
        logger.trace(this._toString() + ".onEditorSaveEvent - session: " + session.hashCode());

        if(txNonConversational == null || (conversationHolder != null && !conversationHolder.
isTransactionActive())){
            // no running transaction, start one ...
            startTransaction();
        }

        // merge the changes into the session, ...
        T mergedBean = mergedBean(bean);
        session.flush();
        commitTransction();

        return new EntityChangeEvent(mergedBean.getClass(), mergedBean.getId(), changeEventType);
}
```

The session.flush() after the merge causes a scheduleUpdate() which in fact is requiring the authenticated user being granted with the UPDATE authority. Below is the according stack trace:

```
eu.etaxonomy.cdm.database.PermissionDeniedException: [UPDATE] not permitted for 'andreas' on Taxon
Name[uuid:b93e9a49-5016-48d0-93ef-38c12ba3886e', toString:'TaxonName#2343<b93e9a49-5016-48d0-93ef-
38c12ba3886e>']
    at eu.etaxonomy.cdm.persistence.hibernate.CdmSecurityHibernateInterceptor.checkPermissions(Cdm
SecurityHibernateInterceptor.java:158)
    at eu.etaxonomy.cdm.persistence.hibernate.CdmSecurityHibernateInterceptor.onFlushDirty(CdmSecu
rityHibernateInterceptor.java:116)
    at org.hibernate.event.internal.DefaultFlushEntityEventListener.invokeInterceptor(DefaultFlush
EntityEventListener.java:348)
    at org.hibernate.event.internal.DefaultFlushEntityEventListener.handleInterception(DefaultFlus
hEntityEventListener.java:325)
    at org.hibernate.event.internal.DefaultFlushEntityEventListener.scheduleUpdate(DefaultFlushEnt
ityEventListener.java:276)
    at org.hibernate.event.internal.DefaultFlushEntityEventListener.onFlushEntity(DefaultFlushEnti
tyEventListener.java:143)
    at org.hibernate.event.internal.AbstractFlushingEventListener.flushEntities(AbstractFlushingEv
entListener.java:216)
    at org.hibernate.event.internal.AbstractFlushingEventListener.flushEverythingToExecutions(Abst
```

```
ractFlushingEventListener.java:85)
    at org.hibernate.event.internal.DefaultFlushEventListener.onFlush(DefaultFlushEventListener.ja
va:38)
    at org.hibernate.internal.SessionImpl.flush(SessionImpl.java:1282)
    at eu.etaxonomy.cdm.service.CdmStore.saveBean(CdmStore.java:206)
```

Related issues:

Related to EDIT - bug #4307: User with permission group 'Editor' cannot creat...	**Feedback**
Has duplicate EDIT - bug #6886: Entity creation for users having only CREATE ...	**Duplicate**
Copied to EDIT - bug #7022: TaxonName.protectedAuthorshipCache should initial...	**Rejected**

Associated revisions

Revision 7631d065 - 10/18/2017 11:23 AM - Andreas Kohlbecker

fix #7021 excluding not protected cache fields from modification check in CdmSecurityHibernateInterceptor

History

#1 - 10/17/2017 05:57 PM - Andreas Kohlbecker

- Target version changed from Unassigned CDM tickets to Release 4.11

#2 - 10/17/2017 09:42 PM - Andreas Kohlbecker

This problem is caused by the cache fields which are empty at the time when the merge happens. In turn of the flush these fields are being filled (in case the cache field is not protected) the CdmSecurityHibernateInterceptor detects the modification of these fields and requires the user to be granted for UPDATE:

```
        if (isModified(currentState, previousState, propertyNames, exculdeMap.get(baseType(cdmEntity)))) {
            // evaluate throws EvaluationFailedException
            //if(cdmEntity.getCreated())
            checkPermissions(cdmEntity, Operation.UPDATE);
```

The isModified method can exclude specific fields from the check. Cache fields should be excluded as long they are not set to protected via their according "protectedCache" flag.

#3 - 10/18/2017 11:23 AM - Andreas Kohlbecker

- Status changed from New to Resolved

- % Done changed from 0 to 50

Applied in changeset cdmlib|7631d065e97a577e932260bf2546ab1c2b161fe2.

#4 - 10/18/2017 11:33 AM - Andreas Kohlbecker

- Assignee changed from Andreas Kohlbecker to Andreas Müller

This issue is solved now by excluding un-proteced cache fields from the modification check in the CdmHibernateSecurityInterceptor.

But there is another issue: The default value of TaxonName.protectedAuthorshipCache is true. Since this cache is protected a user with the authority TAXONNAME.[CREATE] can not create and persist a new TaxonName without setting this field to false prior doing the flush. I will create a new ticket for this issue ...

Andreas Müller please review my implementation

#5 - 10/18/2017 11:40 AM - Andreas Kohlbecker

- Copied to bug #7022: TaxonName.protectedAuthorshipCache should initially be false added

#6 - 10/23/2017 02:41 PM - Andreas Kohlbecker

- Has duplicate bug #6886: Entity creation for users having only CREATE may fail in long running conversations added

#7 - 06/13/2018 04:10 PM - Andreas Kohlbecker

- Related to bug #4307: User with permission group 'Editor' cannot create new authorteam via wizzard added

#8 - 01/24/2024 08:55 AM - Andreas Müller

- Description updated