

## EDIT - bug #6859

### DerivedUnitFacade.newFacadeFrom() fails when occurrenceUuid references a FieldUnit

07/31/2017 09:39 AM - Andreas Kohlbecker

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	New	<b>Due date:</b>	
<b>Assignee:</b>	Andreas Müller	<b>% Done:</b>	0%
<b>Category:</b>	cdmlib	<b>Estimated time:</b>	0:00 hour
<b>Target version:</b>	Unassigned CDM tickets	<b>Found in Version:</b>	
<b>Severity:</b>	normal		
<b>Description</b>			

#### History

##### #1 - 07/31/2017 09:40 AM - Andreas Kohlbecker

Hallo Andreas, Patrick

Wir (AK + ich) hatten uns das ja vorhin angeschaut, dass Walters jetzige Bilder Fehler schmeißen.

Das liegt v.a. an der Methode

```
private DerivedUnitFacade newFacadeFrom(UUID occurrenceUuid, HttpServletResponse response, List<String>
extendedInitStrategy)

throws IOException {

    List<String> initStrategy = new ArrayList<String>(initializationStrategy);

    if(extendedInitStrategy != null && extendedInitStrategy.size() > 0){

        initStrategy.addAll(extendedInitStrategy);

    }

    SpecimenOrObservationBase<?> sob = service.load(occurrenceUuid, null);

    if(sob instanceof DerivedUnit){

        try {

            return service.getDerivedUnitFacade(((DerivedUnit)sob), initStrategy);

        } catch (DerivedUnitFacadeNotSupportedException e) {

            logger.error(e); //TODO ...

        }

    } else {

        HttpStatusMessage.UUID_REFERENCES_WRONG_TYPE.send(response);

    }

    return null;

}
```

Die Frage ist hier, ob der instanceof check wirklich notwendig ist oder abgeschwächt werden könnte. Im hiesigen Fall gibt es ja sowohl MediaSpecimen, die da nicht abgedeckt sind, als auch FieldUnits, die abgefragt werden und die inzwischen ja auch von der DerivedUnitFacade abgedeckt werden (genau, wir sollten die dann wohl mal umbenennen J).

Letzteres sollte nach dem neuen Import zwar kein Problem mehr sein, in Zukunft sollten wir das aber besser händeln, da eigentlich nichts dagegen spricht auch FieldUnits als IndividualsAssociations zu speichern.

Die Methode wird an mehreren Stellen aufgerufen an denen man dann tlws. Checks einbauen muss, ob die Verwendung Sinn macht (können wir mal

mündlich besprechen).

Meine Frage jetzt erstmal, ob dieser Check grundsätzlich überhaupt sein muss aus eurer Sicht. Vielleicht übersehe ich ja was.

Viele Grüße,

Andras M.

**#2 - 07/31/2017 09:40 AM - Andreas Kohlbecker**

Hallo,

ich verstehe das nicht ganz. Es ist doch genau anders herum: FieldUnit wird hier nicht abgedeckt, aber MediaSpecimen als Subklasse von DerivedUnit schon.

Wenn es darum geht, eine DerivedUnitFacade zu erstellen, dann sollte als übergebene occurrence FieldUnit sowohl als DerivedUnit möglich sein, was es in diesem Fall nicht ist. Der Check wäre somit unnötig, weil es ja eh schon eine SpecimenOrObservation ist.

Oder habe ich was übersehen?

Gruß,  
Patrick

**#3 - 07/31/2017 09:41 AM - Andreas Kohlbecker**

Hallo,

dem was Patrick geschrieben hat schließe ich mich weitgehend an, aber wenn an die Konstruktoren der DerivedUnitFacade betrachtet wird klar dass eine Weiche benötigt wird die nach DerivedUnit und FieldUnit unterscheidet:

- public static DerivedUnitFacade NewInstance(SpecimenOrObservationType type)
- public static DerivedUnitFacade NewInstance(SpecimenOrObservationType type, FieldUnit fieldUnit)
- public static DerivedUnitFacade NewInstance(SpecimenOrObservationType type, FieldUnit fieldUnit, DerivedUnitFacadeConfigurator config)
- public static DerivedUnitFacade NewInstance(DerivedUnit derivedUnit)
- public static DerivedUnitFacade NewInstance(DerivedUnit derivedUnit, DerivedUnitFacadeConfigurator config)
- private DerivedUnitFacade(SpecimenOrObservationType type, FieldUnit fieldUnit, DerivedUnitFacadeConfigurator config)

es gibt keinen Konstruktor der eine SpecimenOrObservation instanz akzeptiert!

Andreas