

4. ALBERT-Anwender*innen-Treffen

- Oliver Kant, Viktoria Gerlach, Stefan Lohrum, Alexey Sharenkov, Martin Weigelt, Julia Boltze

Mapping in ALBERT (1)

```
<field name="autor" type="texttype" indexed="true" stored="true" multiValued="true"/>  
<field name="name" type="texttype" indexed="true" stored="false" multiValued="true"/>  
<field name="person" type="string" indexed="true" stored="true" multiValued="true" />  
<field name="book_url" type="string" indexed="false" stored="true" multiValued="true"/>
```

```
<fieldType name="texttype" class="solr.TextField" positionIncrementGap="100">  
  <analyzer>  
    <charFilter class="solr.MappingCharFilterFactory" mapping="mapping-FoldToUmlaut.txt"/>  
    <tokenizer class="solr.StandardTokenizerFactory" />  
    <!-- Due to a solr issue, the full class name needs to be used for the ICUFoldingFilterFactory  
         (https://issues.apache.org/jira/browse/SOLR-6188)-->  
    <filter class="org.apache.lucene.analysis.icu.ICUFoldingFilterFactory"/>  
    <filter class="solr.LowerCaseFilterFactory" />  
  </analyzer>  
</fieldType>
```

Mapping in ALBERT (2)

isbn=020a

reference=773atdgxz

edition=250ab:533abcdefn

fussnote=501a:505t:515a:516a:518a:525a:530a:535a:538a:546a:550a:555a

formerTitle=780iat:247a, first

autor=100abcd:110abndc:111andce

name=script(name.bsh) * Gets names for search field

```
List<String> result =  
record.getSubfieldValues("100abcd:110abndc:111andce:700abcd:710abndc:711andce:720a:909a:910a:911a:508a:511a");
```

person=script(person.bsh) * Gets persons and organisations for facet

```
List<String> result = record.getSubfieldValues("100abc:110abndc:111andce:700abc:710abndc:711andce:720a:508a:511a");
```

Mapping in ALBERT (3)

```
<disp:facetDefinition facetName="person"/>
```

```
<disp:facetDefinition facetName="language"  
    valueMappingPrefix="facet.language."/>
```

```
<disp:detailsField fieldKey="formerTitle"/>
```

```
<disp:detailsField fieldKey="language"  
    valueMappingPrefix="facet.language."  
    listSeparator=", "/>
```

Mapping in ALBERT (4)

```
<disp:detailsField fieldKey="series_statement"  
    valueMappingPrefix="detail.hit.link."  
    hitFieldProcessorClass="de.kobv.albert.frontend.base.display.processors.fields.IDSearchProcessor"  
    listSeparator="newLine"  
    escapeXml="false"  
    anchorSplitRegex="@A@"  
    action="uid.do"  
    linkClass="hitlink">  
    <disp:actionParam key="query" value="ID"/>  
</disp:detailsField>
```

Mapping in ALBERT (4)

```
<disp:detailsField fieldKey="series_statement"  
    valueMappingPrefix="detail.hit.link."  
    hitFieldProcessorClass="de.kobv.albert.frontend.base.display.processors.fields.IDSearchProcessor"  
    listSeparator="newLine"  
    escapeXml="false"  
    anchorSplitRegex="@A@"  
    action="uid.do"  
    linkClass="hitlink">  
    <disp:actionParam key="query" value="ID"/>  
</disp:detailsField>
```

Mapping in ALBERT (5)

Dazu hier die Erklärungen:

Folien:

1) Mapping in ALBERT (1)

Das was sie dort sehen sind Beispiele für definierte Solr-Index-Felder. Die haben einen Namen und können weitere Parameter haben. Wichtig zu wissen ist, dass diese Felder durchsuchbar sein können (aber nicht müssen), anzeigbar sein können (aber nicht müssen) oder mehrfach belegt sein dürfen (oder auch nicht).

2) Mapping in ALBERT (2)

Dort sehen sie Beispiele für ein Mapping von MARC auf die o.g. Solr-Felder. Heisst, hier wird festgelegt, welches MARC-Feld in welches Solr-Index-Feld geschrieben wird. Es gibt auch hier verschiedene Möglichkeiten, die eine Auswirkung darauf haben, wie die Solr-Felder gefüllt werden.

So gibt es zum Beispiel einen Unterschied, ob wir definieren `formerTitle=780iat` oder `formerTitle=780i:780a:780t`, wobei `formerTitle` der Name eines Solr-Feldes ist und die Zahlen plus Buchstaben die MARC-Felder repräsentieren. Näheres dazu in dem Word-Dokument.

Vielleicht noch die Anmerkung, das wir auch komplexeres Mapping definieren können, bspw `name=script(name.bsh)`, bedeutet dass ein extra Javaprogramm aufgerufen wird, was das Feld Name auf eine ganz bestimmte Art und Weise füllt.

3) Mapping in ALBERT (3)

Hier wiederum wird jetzt im ALBERT-Frontend die Anzeige definiert. Da wird dann bspw. ein Facettenfeld bzw. ein Detail-Anzeigefeld auf Basis der gefüllten Solr-Felder definiert. Zusätzlich gibt es da Parameter, die gesetzt werden können (aber nicht müssen), die eine Auswirkung auf die Darstellung der Anzeige haben. Bspw. `listSeperator=","` könnte auch ein Doppelpunkt sein.

Mapping in ALBERT (4) zeigt ein Beispiel, dass es auch noch komplizierter geht.

All das sollte zeigen, dass wir verschiedene Stellschrauben haben, Daten zu verarbeiten, zu indexieren und anzuzeigen. Es gibt aber natürlich auch Grenzen.