MetaDataManagement Documentation

Release 1.0.75

Robert Birkelbach, Anne Gärtner, René Reitmann

11.12.2018

Inhaltsverzeichnis:

1 User-Rollen **1**
- 1.1 Public User . . . 1
- 1.2 Data Provider . . . 1
- 1.3 Publisher . . . 1
- 1.4 Admin . . . 1

2 Metadatenabgabe (Data Provider) **3**
- 2.1 Allgemeines . . . 3
  - 2.1.1 Der Datenaufnahmeprozess im FDZ des DZHW . . . 3
  - 2.1.2 Die eigenständige Abgabe von Metadaten . . . 3
- 2.2 Vorbereitende Schritte . . . 4
  - 2.2.1 Registrierung . . . 4
  - 2.2.2 Neues Projekt anlegen . . . 5
- 2.3 Notwendige Schnittstellen und Dateien . . . 6
  - 2.3.1 Eingabemasken . . . 6
  - 2.3.2 Excel-Tabellen . . . 6
  - 2.3.3 Anhänge . . . 9
  - 2.3.4 Die korrekte Anordnung der Dateien im Ordner . . . 9
- 2.4 Die Abgabe von Metadaten für die einzelnen Ebenen . . . 9
  - 2.4.1 Studie (study) . . . 9
  - 2.4.2 Erhebungen (surveys) . . . 14
  - 2.4.3 Erhebungsinstrumente (instruments) . . . 19
  - 2.4.4 Fragen (questions) . . . 22
  - 2.4.5 Questions (manuell) . . . 23
  - 2.4.6 Questions (Zofar) . . . 25
  - 2.4.7 Datensätze (dataSets) . . . 25
  - 2.4.8 Variablen (variables) . . . 28
  - 2.4.9 Publikationen (relatedPublications) . . . 32
- 2.5 Die Freigabe eines neuen Projekts . . . 32
- 2.6 Anhang . . . 34
  - 2.6.1 Checkliste für Abgabe der Metadaten . . . 34

3 Javadoc **35**
- 3.1 eu.dzhw.fdz.metadatamanagement.common.domain . . . 35
  - 3.1.1 AbstractRdcDomainObject . . . 35
  - 3.1.2 Counter . . . 36
  - 3.1.3 I18nString . . . 36

3.1.4 Period . . . 37
3.1.5 Person . . . 37
3.1.6 Resolution . . . 37
3.2 eu.dzhw.fdz.metadatamanagement.datasetmanagement.domain . . . 38
3.2.1 DataSet . . . 38
3.2.2 DataSetAttachmentMetadata . . . 39
3.2.3 DataSetTypes . . . 41
3.2.4 Format . . . 41
3.2.5 SubDataSet . . . 42
3.3 eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain . . . 42
3.3.1 Instrument . . . 42
3.3.2 InstrumentAttachmentMetadata . . . 44
3.3.3 InstrumentAttachmentTypes . . . 45
3.3.4 InstrumentTypes . . . 46
3.4 eu.dzhw.fdz.metadatamanagement.ordermanagement.domain . . . 46
3.4.1 Customer . . . 46
3.4.2 Order . . . 47
3.4.3 OrderClient . . . 48
3.4.4 OrderState . . . 48
3.4.5 OrderedStudy . . . 48
3.4.6 Product . . . 49
3.5 eu.dzhw.fdz.metadatamanagement.projectmanagement.domain . . . 50
3.5.1 Configuration . . . 50
3.5.2 DaraUpdateQueueItem . . . 50
3.5.3 DataAcquisitionProject . . . 51
3.5.4 Release . . . 52
3.5.5 Requirements . . . 52
3.6 eu.dzhw.fdz.metadatamanagement.questionmanagement.domain . . . 53
3.6.1 ImageType . . . 53
3.6.2 Question . . . 53
3.6.3 QuestionImageMetadata . . . 55
3.6.4 QuestionTypes . . . 57
3.6.5 TechnicalRepresentation . . . 57
3.7 eu.dzhw.fdz.metadatamanagement.relatedpublicationmanagement.domain . . . 58
3.7.1 RelatedPublication . . . 58
3.8 eu.dzhw.fdz.metadatamanagement.studymanagement.domain . . . 60
3.8.1 DataAvailabilities . . . 60
3.8.2 Study . . . 60
3.8.3 StudyAttachmentMetadata . . . 62
3.8.4 StudyAttachmentTypes . . . 63
3.8.5 SurveyDataTypes . . . 63
3.8.6 SurveyDesigns . . . 64
3.9 eu.dzhw.fdz.metadatamanagement.surveymanagement.domain . . . 64
3.9.1 DataTypes . . . 64
3.9.2 Population . . . 65
3.9.3 Survey . . . 65
3.9.4 SurveyAttachmentMetadata . . . 67
3.9.5 SurveyResponseRateImageMetadata . . . 68
3.10 eu.dzhw.fdz.metadatamanagement.variablemanagement.domain . . . 69
3.10.1 AccessWays . . . 69
3.10.2 DataTypes . . . 70
3.10.3 Distribution . . . 71
3.10.4 FilterDetails . . . 72
3.10.5 FilterExpressionLanguages . . . 72

3.10.6 GenerationDetails . . . 73
3.10.7 Missing . . . 73
3.10.8 RelatedQuestion . . . 74
3.10.9 RuleExpressionLanguages . . . 75
3.10.10 ScaleLevels . . . 75
3.10.11 Statistics . . . 76
3.10.12 StorageTypes . . . 77
3.10.13 ValidResponse . . . 80
3.10.14 Variable . . . 81

4 Indizes und Tabellen **85**

KAPITEL 1

User-Rollen

Im Metadatenmanagementsystem (MDM) gibt es mehrere User-Rollen, für die jeweils nur bestimmte Teile der Dokumentation von Interesse sind. Im folgenden werden daher die Rollen aufgeführt und kurz erklärt.

1.1 Public User

Student, Researcher

1.2 Data Provider

DZHW interne/externe DatengeberIn, siehe *Metadatenabgabe (Data Provider)*.

1.3 Publisher

DZHW-FDZ MitarbeiterIn

1.4 Admin

Weist User-Rollen zu.

KAPITEL 2

Metadatenabgabe (Data Provider)

2.1 Allgemeines

2.1.1 Der Datenaufnahmeprozess im FDZ des DZHW

Das FDZ des DZHW stellt Daten quantitativer und qualitativer Erhebungen aus dem Feld der Hoch-schul- und Wissenschaftsforschung zur Verfügung. Ein fester Bestandteil dieser Arbeit ist zunächst der Prozess der Datenaufnahme, welcher im FDZ des DZHW durch ein selbst entwickeltes System, das Metadatenmanagementsystem (MDM) unterstützt wird. Das Besondere am MDM ist, dass Informationen über die eigentlich erhobenen Forschungsdaten, also Metadaten, erfasst werden. Für die strukturierte Aufnahme der Metadaten sind sieben unterschiedliche Ebenen im MDM vorgesehen: *Studie*, *Erhebungen*, *Erhebungsinstrumente*, *Fragen*, *Datensätze*, *Variablen* und *Publikationen*.

Innerhalb der Aufnahme von Forschungsdaten wird anhand der Metadaten auf diesen Ebenen erfasst, welcher Studie sowie Erhebung die Daten angehören, welche Erhebungsinstrumente genutzt wurden, welche Fragen darin gestellt wurden, welche Datensätze existieren, welche Variablen sich darin befinden und welche Veröffentlichungen bereits mit den Daten realisiert worden sind. Dadurch, dass alle Ebenen miteinander verknüpft sind, wird eine umfassende Durchsuchbarkeit aller verfügbaren Daten, die vom FDZ des DZHW über das MDM verwaltet werden, ermöglicht. Das System ist über die Website https://metadata.fdz.dzhw.eu zu erreichen.



Abb. 2.1: Darstellung der verschiedenen Ebenen im MDM, Ebene *Studie* aktiv

2.1.2 Die eigenständige Abgabe von Metadaten

Wenn Sie Ihre Daten im FDZ des DZHW abgeben möchten, erfassen Sie die projektbezogenen Metadaten selbst und können diese teilweise eigenständig in das MDM hochladen.

Die Abgabe der Metadaten ist innerhalb der einzelnen Ebenen unterschiedlich komplex, sodass die Daten für jede der sieben Ebenen separat erfasst und hochgeladen werden müssen. Hierfür hat das FDZ des DZHW feste Strukturen

entwickelt, die im weiteren Verlauf dieser Anleitung für jede Ebene detailliert erläutert werden. Diesen Vorgaben ist unbedingt Folge zu leisten, damit ein erfolgreicher Upload der einzelnen Metadaten gewährleistet werden kann.

Aktuell können einige Metadaten bereits über eine Eingabemaske im MDM direkt eingetragen werden. Für die übrigen Ebenen müssen Excel-Tabellen ausgefüllt werden. Zusätzlich ist auf manchen Ebenen das Erstellen von weiteren Anhängen (vgl. Kapitel 2.5) vorgesehen. Die Excel-Tabellen und eventuelle Anhänge müssen für jede Ebene dann entweder in einer bestimmten Ordnerstruktur ins MDM hochgeladen werden oder zunächst dem FDZ zur weiteren Bearbeitung geschickt werden. Tabelle 1 zeigt eine erste Übersicht über das Vorgehen der Metadateneingabe auf jeder Ebene, detaillierte Erläuterungen werden in den nachstehenden Kapiteln folgen.

Tabelle 1: Vorgehen bei der Metadateneingabe pro Ebene

Ebene	Metadaten eingeben	Zusätzliche Dateien	Metadaten abgeben
Studie	Eingabemaske	Anhänge (PDF)	Eingabemaske ausfüllen
Erhebungen	Eingabemaske	Anhänge (PDF)	Eingabemaske ausfüllen
Erhebungs- instrumente	Excel-Tabelle/ Eingabemaske	Anhänge (PDF, Excel)	Ordner ins MDM laden/ Eingabemaske ausfüllen
Fragen	Excel-Tabelle (oder Zofar)	Fragebilder (.png)	Im Vorlage-Ordner speichern
Datensätze	Excel-Tabelle	Anhänge (PDF)	Ordner ins MDM laden
Variablen	mind. 1 Excel-Tabelle	mind. 1 Datensatz (Stata)	Im Vorlage-Ordner speichern
Publikationen	•	•	Informationen dem FDZ schicken

2.2 Vorbereitende Schritte

2.2.1 Registrierung

Sie müssen sich zunächst auf https://metadata.fdz.dzhw.eu registrieren, um die Berechtigung für das Anlegen eines Projektes sowie das Hochladen von Metadaten zu erhalten. Dies können Sie über die Sidebar links auf der Website erledigen:

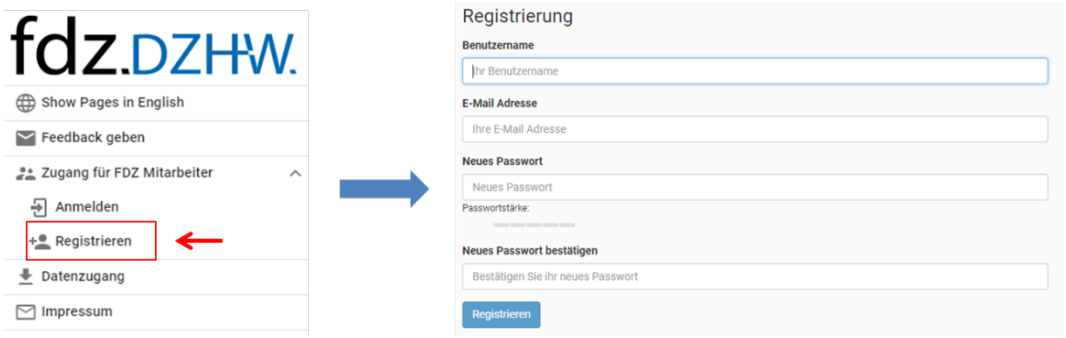


Abb. 2.2: Registrierung im MDM

Da das FDZ Ihre Registrierungsanfrage zunächst bestätigen muss, kann es etwas dauern, bis Sie freigeschaltet sind. Sie bekommen dann eine Bestätigung per Email.

2.2.2 Neues Projekt anlegen

Nach erfolgreicher Registrierung können Sie ihr Projekt im MDM anlegen, indem Sie sich in der Sidebar links anmelden:



Abb. 2.3: Anmeldung im MDM

Nach erfolgreicher Anmeldung erscheint in der Sidebar das folgende Feld:

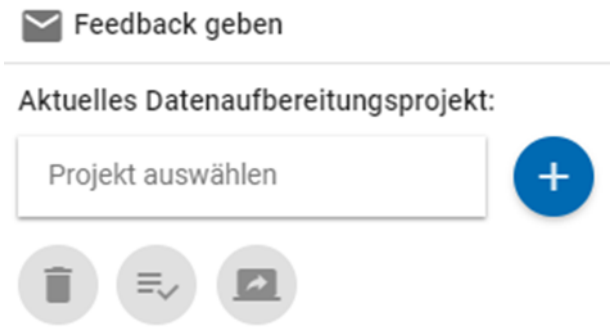


Abb. 2.4: Bereich für Verwaltung von Projekten im MDM

Über den Plus-Button können Sie ein neues Projekt anlegen (vgl. Abb. 2.5). Als Projektname müssen Sie eine bestimmte ID angeben, welche das FDZ zuvor speziell für Ihr Projekt vergeben hat und Ihnen mitteilen muss (z. B. „gra2005“ für das Absolventenpanel 2005). Das Projekt ist dadurch im MDM angelegt, aber noch nicht automatisch freigegeben. Solange das Projekt noch nicht freigegeben wurde, können Sie es jederzeit über den Mülleimer-Button ganz links (vgl. Abb. 2.5) wieder löschen.

Sie können nun beginnen, Ihr Projekt mit Metadaten zu füllen. Wie genau dies funktioniert, wird im Folgenden zunächst prinzipiell erläutert, ehe die konkret geforderten Metadaten in den einzelnen Ebenen in Kapitel 4 detailliert beschrieben werden.

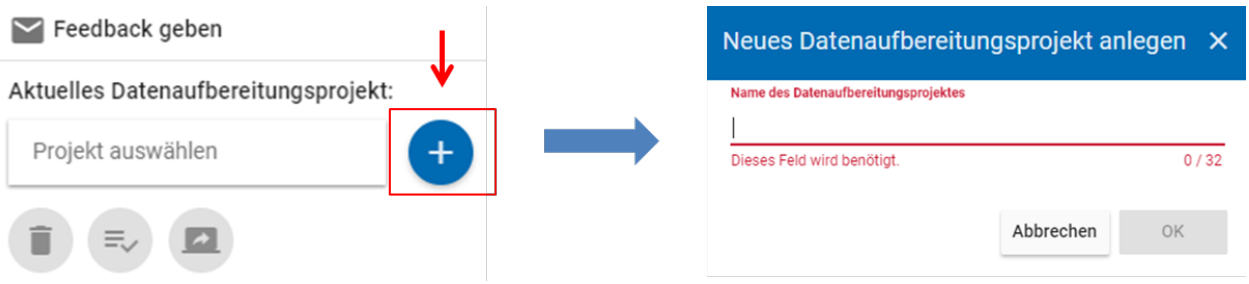


Abb. 2.5: Ein neues Projekt im MDM anlegen

2.3 Notwendige Schnittstellen und Dateien

Grundsätzlich gibt es zwei verschiedene Möglichkeiten Metadaten einzutragen bzw. hochzuladen: Eingabemasken und Excel-Tabellen. Eingabemasken ermöglichen eine komfortable Abgabe der Metadaten direkt auf der Website, während Excel-Tabellen zunächst ausgefüllt und im Anschluss daran entweder in einer bestimmten Ordnerstruktur (vgl. Kapitel 3.4) abgelegt und im MDM hochgeladen oder dem FDZ zur weiteren Bearbeitung zugeschickt werden müssen.

2.3.1 Eingabemasken

Für die Ebenenen Studie, Erhebungen und Instrumente steht bereits die Nutzung von Eingabemasken im MDM zur Verfügung. Jede Maske umfasst verschiedene Felder, welche mit den einzutragenden Informationen beschriftet sind. Einige Felder sind verpflichtend auszufüllen und deshalb mit einem Sternchen versehen. Sie werden beim Speichern der Eingaben automatisch darauf hingewiesen, wenn noch Felder offen sind, die nicht leer bleiben dürfen. Die Bedienung der Eingabemasken ist weitgehend intuitiv und an vielen Stellen selbsterklärend. Im Rahmen der relevanten Ebenen Studie, Erhebungen und Instrumente wird die Handhabung der jeweiligen Eingabemasken dann konkret gezeigt (s. Kapitel 4.1 und 4.2).

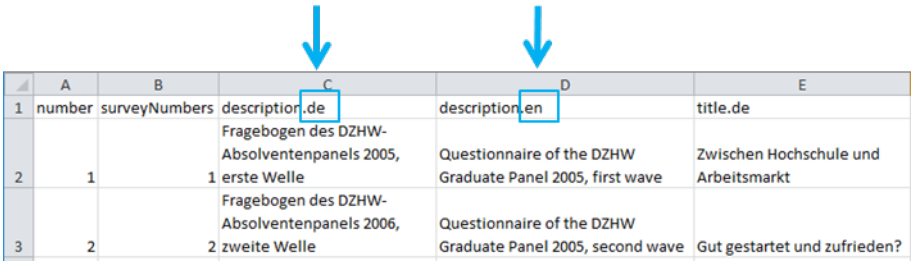
2.3.2 Excel-Tabellen

Für die weiteren Ebenen können Sie bereits vorbereitete Excel-Tabellen verwenden, in denen verschiedene Metadaten spaltenweise erfasst werden. Diese Excel-Tabellen finden Sie im Vorlage-Ordner der jeweiligen Ebene (vgl. Kapitel 3.4). Welche Metadaten Sie an welcher Stelle in der Excel-Tabelle eintragen müssen, erkennen Sie an den bereits vorgegebenen Spaltenüberschriften in der ersten Zeile der Tabelle. Dort stehen die Bezeichnungen der jeweiligen Metadaten, welche auf den ersten Blick etwas kryptisch erscheinen, innerhalb der noch folgenden Beschreibung der einzelnen Ebenen jedoch näher erläutert werden. Beispiele für Spaltenüberschriften in Abb. 2.6: *number* = Nummer des Instruments (Spalte A), *surveyNumbers* = Nummern der zugehörigen Erhebungen (Spalte B).

	A	B	C	D	E
1	number	surveyNumbers	description.de	description.en	title.de
2	1	1	Fragebogen des DZHW-Absolventenpanels 2005, erste Welle	Questionnaire of the DZHW Graduate Panel 2005, first wave	Zwischen Hochschule und Arbeitsmarkt
3	2	2	Fragebogen des DZHW-Absolventenpanels 2006, zweite Welle	Questionnaire of the DZHW Graduate Panel 2005, second wave	Gut gestartet und zufrieden?

Abb. 2.6: Spaltenüberschriften, Excel-Vorlage für die Ebene *Erhebungsinstrumente*

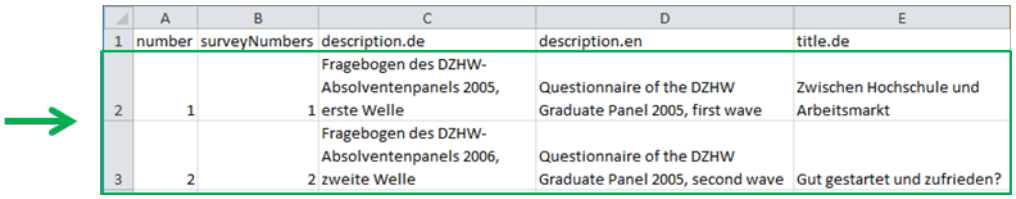
Die Suffixe „.de“ und „.en“ sind Teil vieler Spaltenüberschriften und weisen auf die Sprache des einzutragenden Metadatums hin. Beispiel in Abb. 2.7: Beschreibung des Instruments auf Deutsch (Spalte C), Beschreibung des Instruments auf Englisch (Spalte D).



	A	B	C	D	E
1	number	surveyNumbers	description.de	description.en	title.de
2	1	1	Fragebogen des DZHW-Absolventenpanels 2005, erste Welle	Questionnaire of the DZHW Graduate Panel 2005, first wave	Zwischen Hochschule und Arbeitsmarkt
3	2	2	Fragebogen des DZHW-Absolventenpanels 2006, zweite Welle	Questionnaire of the DZHW Graduate Panel 2005, second wave	Gut gestartet und zufrieden?

Abb. 2.7: Sprachensuffixe, Excel-Vorlage für die Ebene *Erhebungsinstrumente*

Ab der zweiten Zeile sind die Tabellenvorlagen leer und hier nur für das Beispiel bereits gefüllt (vgl. Abb. 2.8). Dort können Sie Ihre Inhalte entsprechend der Spaltenüberschriften eintragen. Inhaltliche Hilfen hierfür finden Sie in den Erklärungen zu den einzelnen Ebenen.



	A	B	C	D	E
1	number	surveyNumbers	description.de	description.en	title.de
2	1	1	Fragebogen des DZHW-Absolventenpanels 2005, erste Welle	Questionnaire of the DZHW Graduate Panel 2005, first wave	Zwischen Hochschule und Arbeitsmarkt
3	2	2	Fragebogen des DZHW-Absolventenpanels 2006, zweite Welle	Questionnaire of the DZHW Graduate Panel 2005, second wave	Gut gestartet und zufrieden?

Abb. 2.8: Beispiele für den Inhalt der einzelnen Metadaten, Excel-Vorlage für die Ebene *Erhebungsinstrumente*

Bitte beachten Sie außerdem:

- Die Excel-Tabellen enthalten je nach Ebene unterschiedlich viele Tabellenblätter, die Sie bearbeiten müssen.
- Die Anzahl der Spalten pro Tabellenblatt variiert, sodass ein seitliches Scrollen oftmals notwendig ist.
- Es gibt Metadaten, die Sie ausfüllen *müssen*, und solche, die Sie ausfüllen *können*. Die konkreten Ausfüllanweisungen finden Sie in Kapitel 4.

Kontrolliertes Vokabular

Für einige Metadaten gibt es ein sogenanntes „kontrolliertes Vokabular“, d. h. dort können nur bestimmte Inhalte in die Spalten eingetragen werden. In diesen Spalten wird Ihnen in jeder Zelle eine Auswahl der möglichen Antworten anhand eines Drop-Down-Menüs angeboten und Sie müssen diese nur auswählen.

Häufig ist es der Fall, dass aus dem kontrollierten Vokabular einer bestimmten Zelle automatisch der Inhalt der nächsten Zelle folgt. Für dieses Szenario sind die Excel-Vorlagen vorbereitet, d. h. in den betreffenden Spalten sind über mehrere Zeilen bereits Formeln hinterlegt, die die nächste Zelle automatisch füllen und Ihnen viel Tipparbeit ersparen (vgl. Abb. 2.9). Die Vorlagen sind für alle Fälle vorbereitet, so dass die Formeln auch in höher nummerierten Zeilen stehen werden, die Sie voraussichtlich nicht mehr benötigen (bis einschließlich Zeile 20 der Excel-Vorlage für die Ebenen Instrumente sowie Datensätze, bis einschließlich Zeile 2000 der Excel-Vorlage für die Fragen- und Variablenebene). Zum leichteren Erkennen sind die betreffenden Zeilen in den Vorlagen bereits grau hinterlegt.

Da Formeln in solchen Zellen, die nicht mit Inhalt befüllt sind (vgl. Abb. 2.10), einen fehlerhaften Upload der Excel-Tabelle hervorrufen, müssen die überflüssigen Formeln aus den nicht benötigten Zeilen herausgelöscht werden. Dies können Sie erledigen, indem Sie die nicht benötigten der grau eingefärbten Zeilen bis einschließlich der Zeile 20 (für Instrumente und Datensätze) bzw. der Zeile 2000 (für Fragen- und Variablenebene) markieren und über das Menü per „Blattzeilen löschen“ komplett entfernen (vgl. Abb. 2.11).

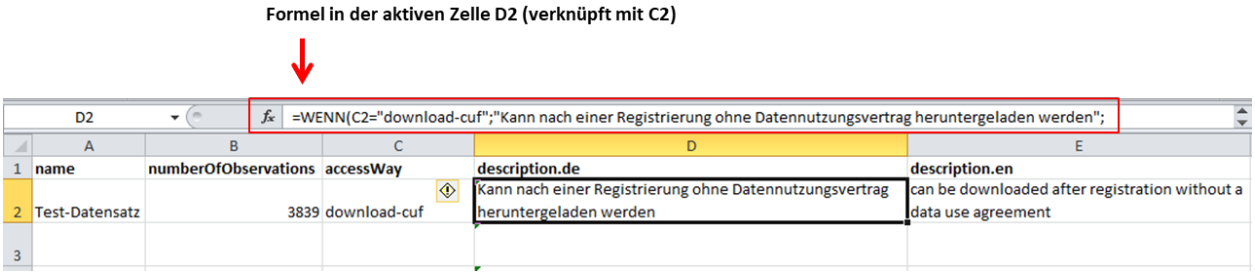


Abb. 2.9: Beispiel für eine Formel in einer befüllten Zelle in der Excel-Vorlage für die Ebene *Datensätze*

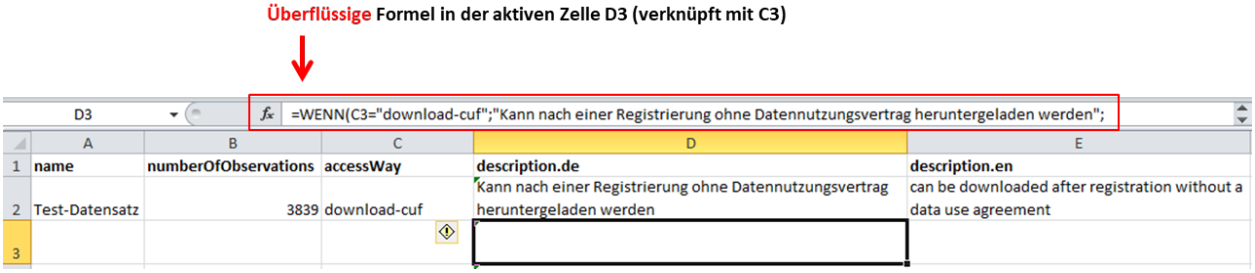


Abb. 2.10: Beispiel für eine Formel in einer leeren Zelle in der Excel-Vorlage für die Ebene *Datensätze*

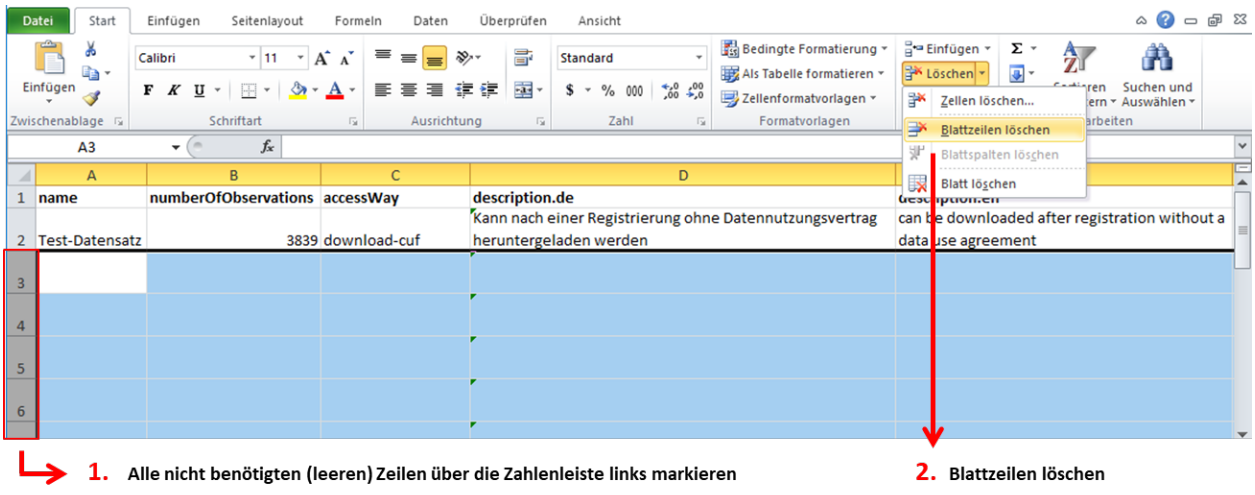


Abb. 2.11: Beispiel für das Löschen nicht benötigter Formeln aus der Excel-Vorlage für die Ebene *Datensätze*

2.3.3 Anhänge

Für einige Ebenen können verschiedene Anhänge entweder direkt über die Eingabemasken oder innerhalb der festgelegten Ordnerstruktur (vgl. Kapitel 3.4) im MDM hochgeladen werden. Zu den Anhängen zählen z. B. der Daten- und Methodenbericht auf der Studienebene sowie Fragebögen oder Codierlisten auf Instrumentenebene. Diese Dokumente müssen als Dateien im PDF- oder Excel-Format vorliegen (Details dazu finden Sie innerhalb der einzelnen Ebenen in Kapitel 4) und zudem nach bestimmten Richtlinien benannt werden. Für die Anhänge im PDF-Format gilt es darüber hinaus zu beachten, dass dokumenteigene Metadaten wie Autor und Titel aus der PDF-Datei gelöscht werden. Dies können Sie im PDF-Dokument über „Datei“ -> „Eigenschaften. . . “ erledigen.

Für das Hochladen der Metadaten über Excel-Tabellen gilt, dass die Anhänge im MDM in der Reihenfolge dargestellt werden, in der sie in der Excel-Tabelle eingetragen wurden. Genaue Informationen dazu finden Sie in den Erläuterungen für die einzelnen Ebenen.

2.3.4 Die korrekte Anordnung der Dateien im Ordner

Für einen erfolgreichen Upload der Metadaten über Excel-Tabellen müssen Sie sämtliche Dateien ihrer zugehörigen Ebene entsprechend in den vom FDZ vorbereiteten Vorlage-Ordner ablegen, welcher nach der jeweiligen Ebene benannt ist. Dieser Ordner sowie auch seine Unterordner sind mit englischen Begriffen betitelt. So heißt z. B. der Ordner der Instrumentenebene „instruments“ und der Ordner der Datensatzebene „dataSets“. Der Unterordner, welche alle Anhänge enthält, heißt unabhängig von der Ebene immer „attachments“. Die Reihenfolge der Dateien im Ordner sowie in den Unterordnern spielt keine Rolle. Da meist eine Sortierung nach alphabetischer Reihenfolge voreingestellt ist, steht der „attachments“-Ordner an erster Stelle, während die Excel-Tabelle erst danach folgt. Die vorliegende Ordnerstruktur am Beispiel der Instrumentenebene zeigt Abb. 2.12.

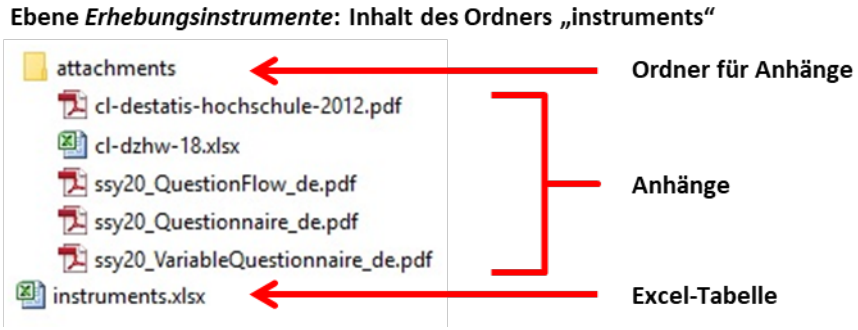


Abb. 2.12: Ordnerstruktur am Beispiel der Instrumentenebene

2.4 Die Abgabe von Metadaten für die einzelnen Ebenen

2.4.1 Studie (study)

Übersicht

Anhand der Informationen, die Sie bzgl. Ihrer Studie an das MDM liefern, wird dort später eine Übersichtsseite erstellt, die im Folgenden am Beispiel des Absolventenpanels 2005 dargestellt wird:

Eine neue Studie anlegen

Nachdem Sie ein neues Projekt erstellt haben (vgl. Kapitel 2.2), können Sie nun innerhalb des Projektes eine Studie anlegen. Dazu finden Sie im Reiter „Studien“ unten rechts auf der Seite einen orangefarbenen Plus-Button. Wenn Sie mit dem Mauszeiger über diesen Button fahren, erscheinen links davon zwei weiße Buttons (vgl. Abb. 2.14).

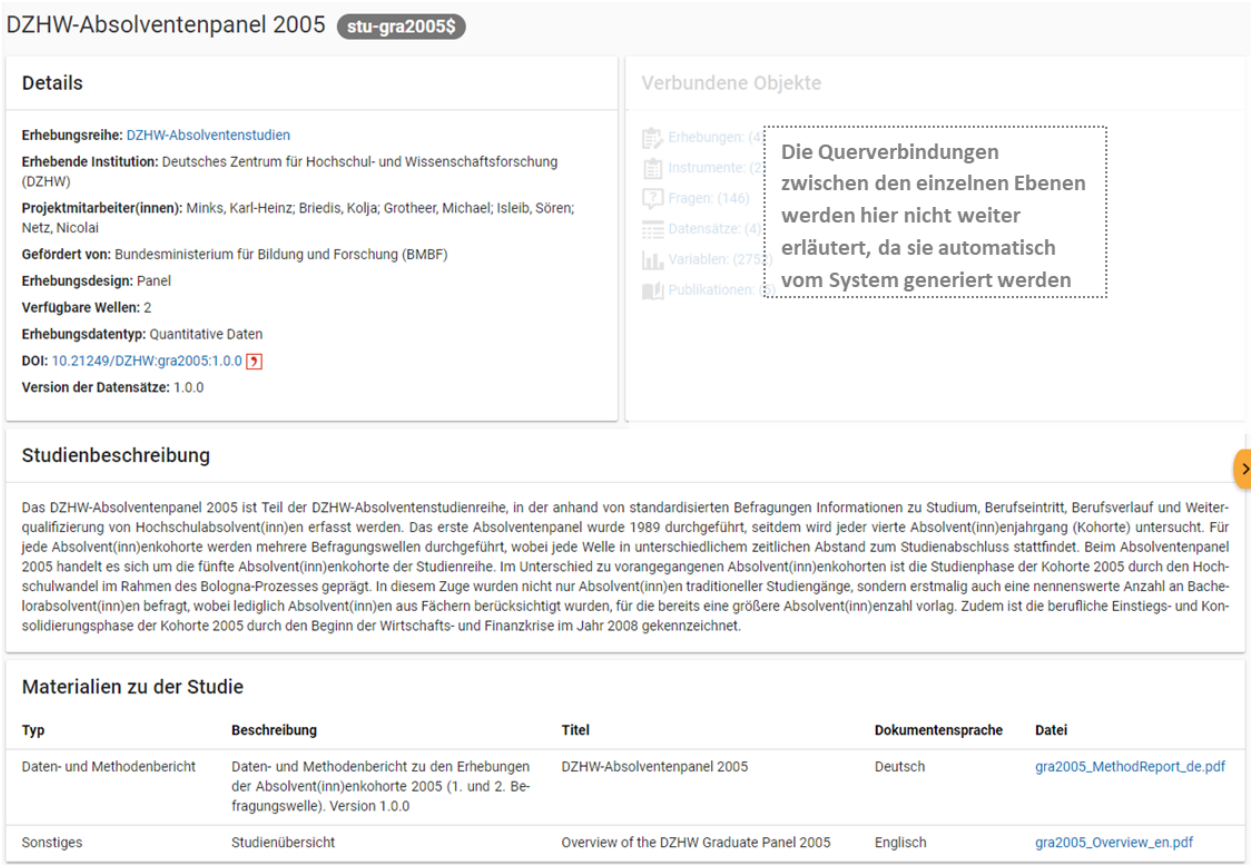


Abb. 2.13: Studienübersicht im MDM am Beispiel des Absolventenpanels 2005

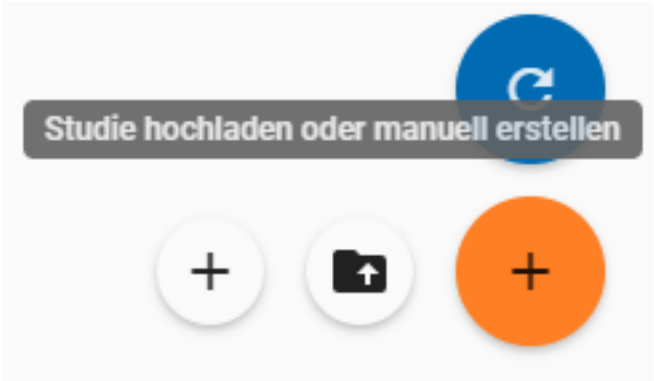


Abb. 2.14: Optionen für das Anlegen einer Studie

Über den weißen Ordner-Button können Studieninformationen weiterhin mittels Ordner, der eine Excel-Tabelle und Anhänge enthält, hochgeladen werden, wobei diese Variante hier nicht weiter erläutert wird, da sie von der Eingabemaske abgelöst wurde. Mit einem Klick auf den weißen Plus-Button öffnet sich die Eingabemaske, in der Sie Ihre Informationen zur Studie ablegen können.

Eingabemaske

Die Eingabemaske auf Studienebene besteht aus den vier Abschnitten „Details“, „Studienbeschreibung“, „Projektmitarbeiter(innen)“ sowie „Materialien zu der Studie“. Der Abschnitt „Details“ ist der umfangreichste und wird im Folgenden aufgrund der Veranschaulichung mit bereits eingetragenen Informationen dargestellt (hier beispielhaft: 21. Sozialerhebung):

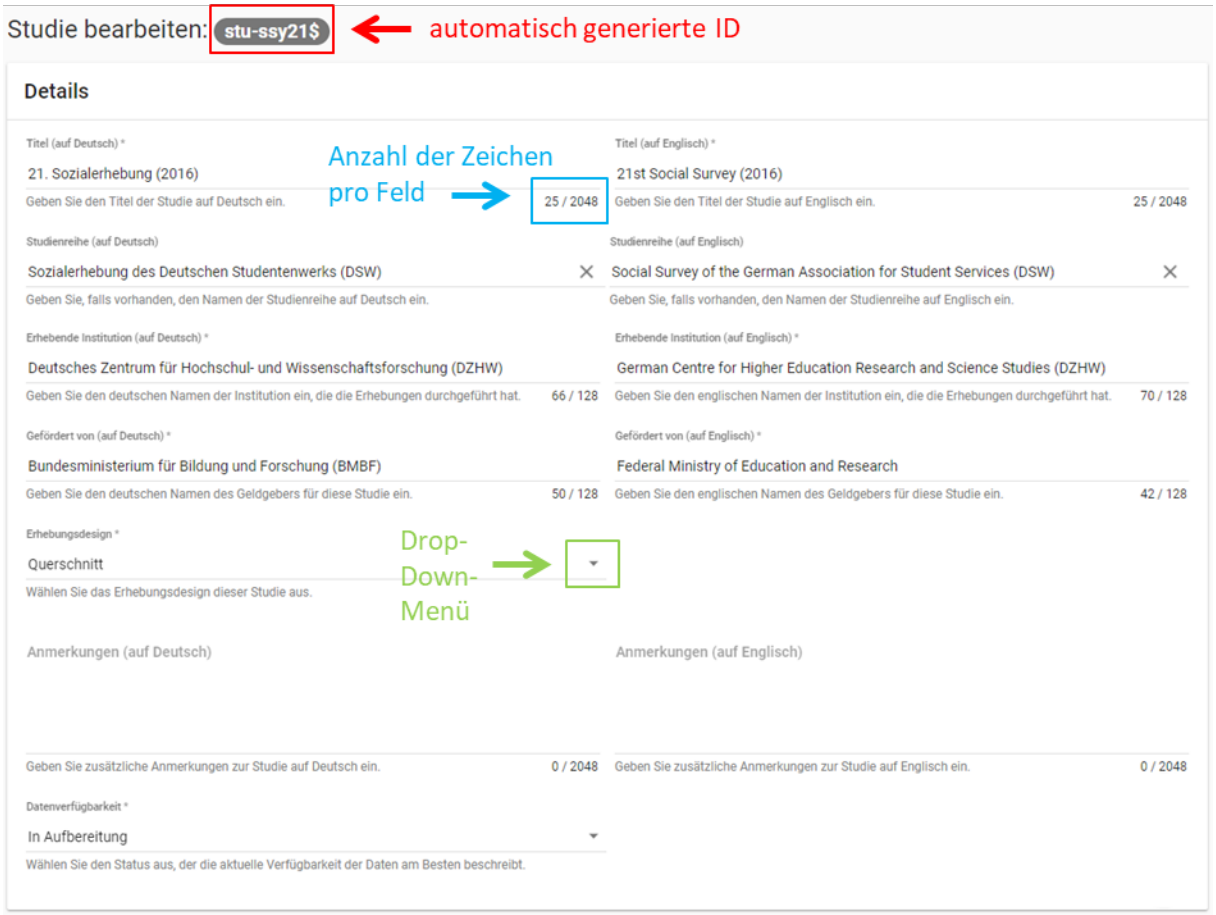


Abb. 2.15: Eingabemaske auf Studienebene, Abschnitt „Details“ am Beispiel der 21. Sozialerhebung

Nach dem Öffnen der Eingabemaske erscheint ganz oben die aus ihrem Projektnamen automatisch generierte ID für die Studienseite (s. rotes Kästchen in Abb. 2.15). Einige Felder, die Sie frei ausfüllen können, verfügen über einen Zeichenzähler, der Sie darüber informiert, wie viele Zeichen Sie dort insgesamt eintragen dürfen und wie viele Zeichen Sie bereits eingetragen haben (s. blaues Kästchen in Abb. 2.15). Außerdem finden Sie teilweise Drop-Down-Menüs vor, in denen Sie aus vorgegebenen Alternativen auswählen können (s. grünes Kästchen in Abb. 2.15).

Im zweiten Abschnitt der Eingabemaske müssen Sie eine Beschreibung Ihrer Studie sowohl auf Deutsch als auch auf Englisch eingeben. Für ein Beispiel ist im Folgenden die Beschreibung der 21. Sozialerhebung abgebildet:

Im dritten Abschnitt der Eingabemaske geben Sie die Mitarbeiter(innen) Ihres Projekts ein. Für die Eingabe weiterer Personen klicken Sie einfach auf den blauen Plus-Button (s. Abb. 2.17). Wenn mindestens zwei Personen eingetragen sind, erscheinen die Pfeil-Buttons als aktiv (Farbwechsel von grau zu blau). Dann können Sie die Reihenfolge der Per-

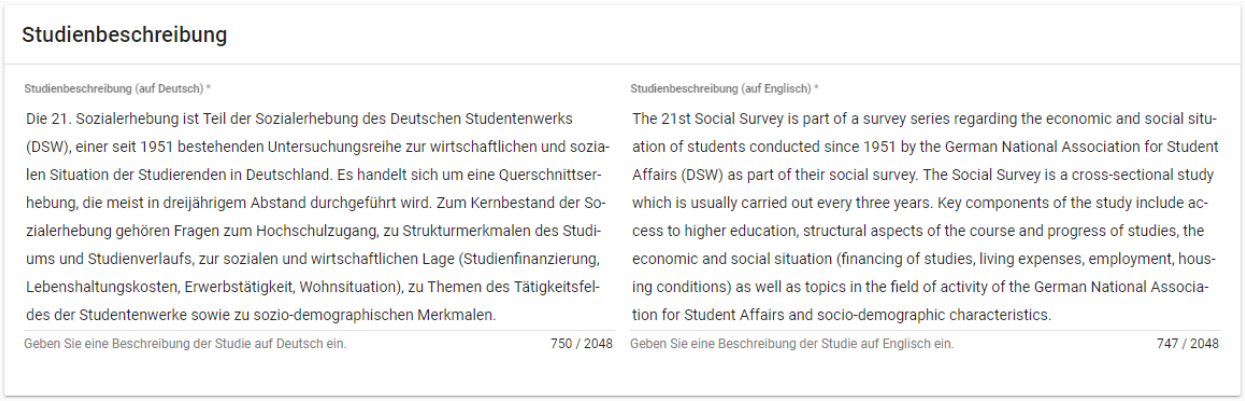


Abb. 2.16: Eingabemaske auf Studienebene, Abschnitt „Studienbeschreibung“ am Beispiel der 21. Sozialerhebung

sonen ändern, indem Sie die Namen nach oben oder unten verschieben. Links neben den bereits aufgeführten Personen erscheint in jeder Zeile ein blauer Button mit einem Mülleimer-Symbol, mit dem Sie den jeweiligen Namen wieder löschen können. Mit dem orangefarbenen Save-Button unten rechts können Sie Ihre Eingaben jederzeit abspeichern. Dies müssen Sie spätestens jetzt tun, da Sie ansonsten den letzten Abschnitt der Eingabemaske („Materialien zu der Studie“) nicht bearbeiten können.

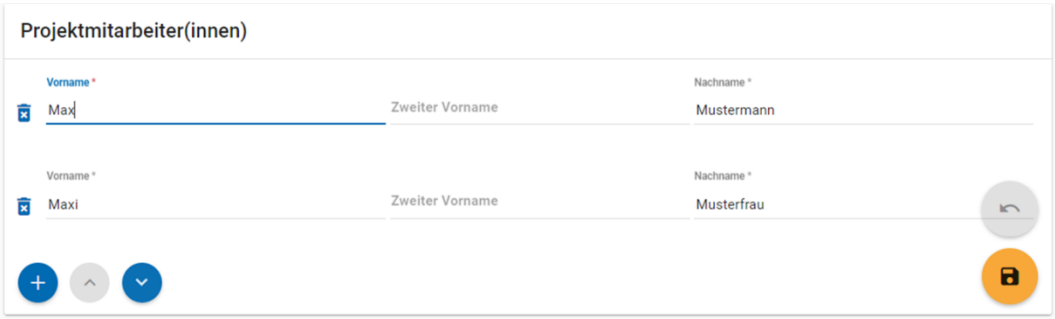


Abb. 2.17: Eingabemaske auf Studienebene, Abschnitt „Projektmitarbeiter(innen)“

Im vierten und letzten Abschnitt der Eingabemaske können Sie Materialien zur Studie ablegen. Dazu klicken Sie auf den blauen Plus-Button (s. Abb. 2.18), woraufhin sich ein Dialog öffnet, in dem Sie eine Datei hochladen und diese näher beschreiben können. Die hier relevanten Materialien sind momentan der deutsch- und englischsprachige Daten- und Methodenbericht (DMB) sowie eine englischsprachige *study overview*.[1] Die Eingaben müssen Sie anschließend über den orangefarbenen Save-Button abspeichern. Mit den Pfeil-Buttons können Sie dann ggf. die Reihenfolge bereits eingegebener Materialien verändern. Wenn Sie eine geänderte Reihenfolge beibehalten möchten, müssen Sie erneut speichern.

Editieren und historisieren

Falls Sie Ihre Informationen auf Studienebene nicht in einem Vorgang eingeben und hochladen können oder möchten, ist es immer möglich, dass Sie Ihre bisherigen Eingaben abspeichern und zu einem späteren Zeitpunkt weiter bearbeiten. Hierfür wird Ihnen im Reiter „Studien“ am rechten Rand neben Ihrer Studie ein Stift-Button angezeigt, über den Sie wieder in die Eingabemaske gelangen (s. Abb. 2.19).

Ebenso können Sie ältere Versionen Ihrer abgespeicherten Eingaben wiederherstellen, indem Sie im Bearbeitungsmodus den Historisierungs-Button (blauer Pfeil-Button über dem Save-Button unten rechts auf der Seite) verwenden (s. Abb. 2.20).

[1] Bitte beachten Sie, die dokumenteigenen Metadaten der PDF-Dateien vorab zu löschen (vgl. Kapitel 3.3).

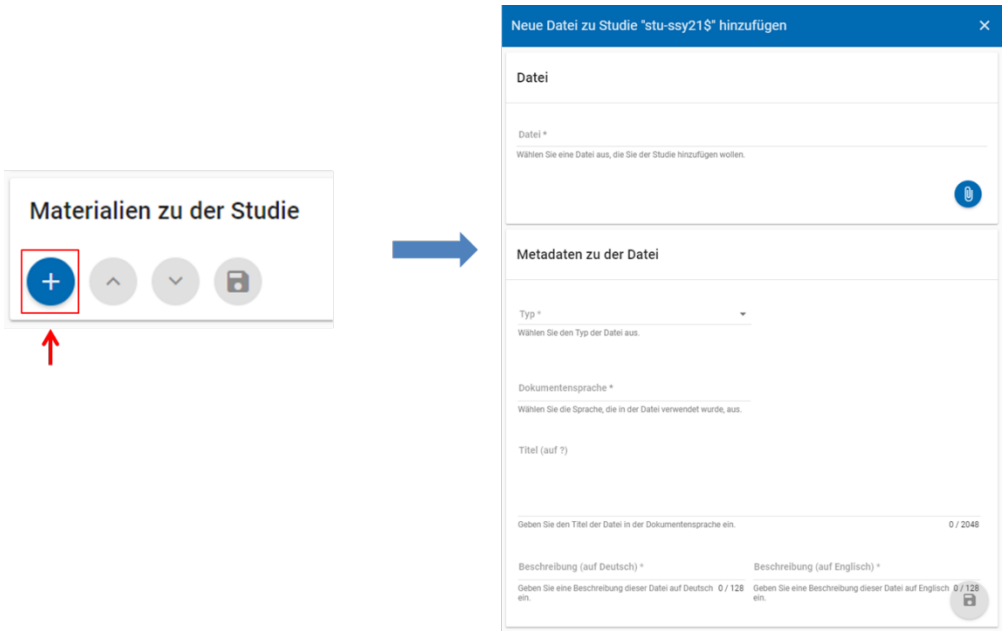


Abb. 2.18: Eingabemasken auf Studienebene, Abschnitt „Materialien zu der Studie“

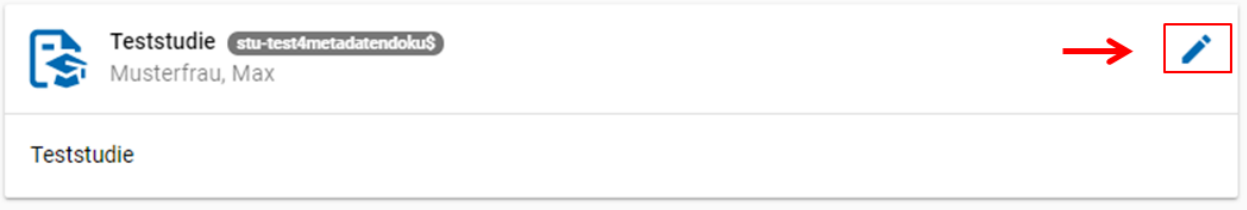


Abb. 2.19: Weitere Bearbeitung einer bereits abgespeicherten Studie

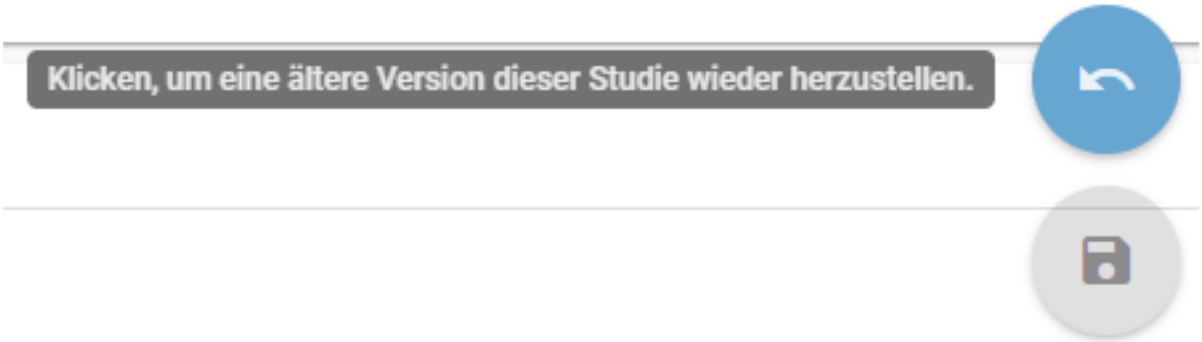


Abb. 2.20: Ältere Versionen einer Studie wiederherstellen

Bei einem Klick auf den Historisierungs-Button öffnet sich ein Dialog, der die verschiedenen Versionen der Studie anzeigt (s. Abb. 2.21). Zudem sind der Name des Nutzers, der die entsprechende Version der Studie gespeichert hat, sowie das Änderungsdatum sichtbar. Durch Klicken auf die Version wird diese wiederhergestellt, aber nicht automatisch als aktuelle Version gespeichert. Dies müsste über einen Klick auf den Save-Button erfolgen. Zu beachten ist, dass Materialien zur Studie nicht historisiert werden.

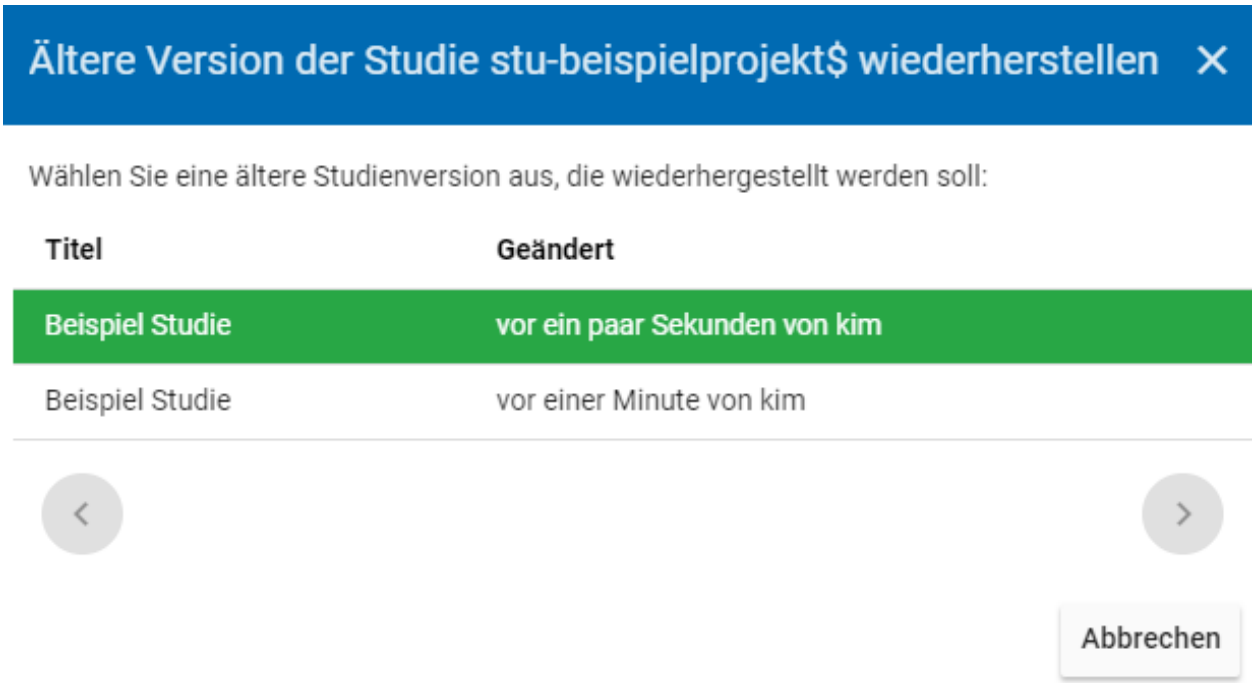


Abb. 2.21: Dialog zur Historisierung innerhalb einer Studie

2.4.2 Erhebungen (surveys)

Übersicht

Mit den Informationen über die Erhebung(en), die Sie innerhalb Ihrer Studie durchgeführt haben, wird im MDM folgende Übersichtsseite erstellt:

Eine neue Erhebung anlegen

Wenn Sie eine Studie angelegt haben (vgl. Kapitel 4.1), können Sie über den Reiter „Erhebungen“ eine neue Erhebung innerhalb Ihrer Studie erstellen. Hierzu finden Sie unten rechts auf der Seite – ebenso wie bei der Studie – einen orangefarbenen Plus-Button (vgl. Abb. 2.23). Wenn Sie mit dem Mauszeiger darüberfahren, erscheinen die beiden weißen Buttons, von denen Sie den Plus-Button anklicken, um die Eingabemaske zu öffnen. Bitte beachten Sie, dass Sie mehrere Erhebungen über die Eingabemaske in der richtigen Reihenfolge eingeben müssen, da die IDs beim Anlegen einer neuen Erhebung automatisch generiert werden und sich später nicht mehr verändern lassen.

Eingabemaske

Die Eingabemaske auf Erhebungsebene besteht aus den drei Abschnitten „Details“, „Weitere Informationen zum Rücklauf“ sowie „Materialien zu der Erhebung“. Im Folgenden wird der Abschnitt „Details“ – aufgrund der Länge in zwei Teilen – dargestellt:

Beim Anlegen einer Erhebung wird automatisch die ID auf Basis des Projektnamens generiert (s. rotes Kästchen, Abb. 2.24, hier als Beispiel der 21. Sozialerhebung). Neben den bereits aus der Studienebene bekannten Funktionen gibt

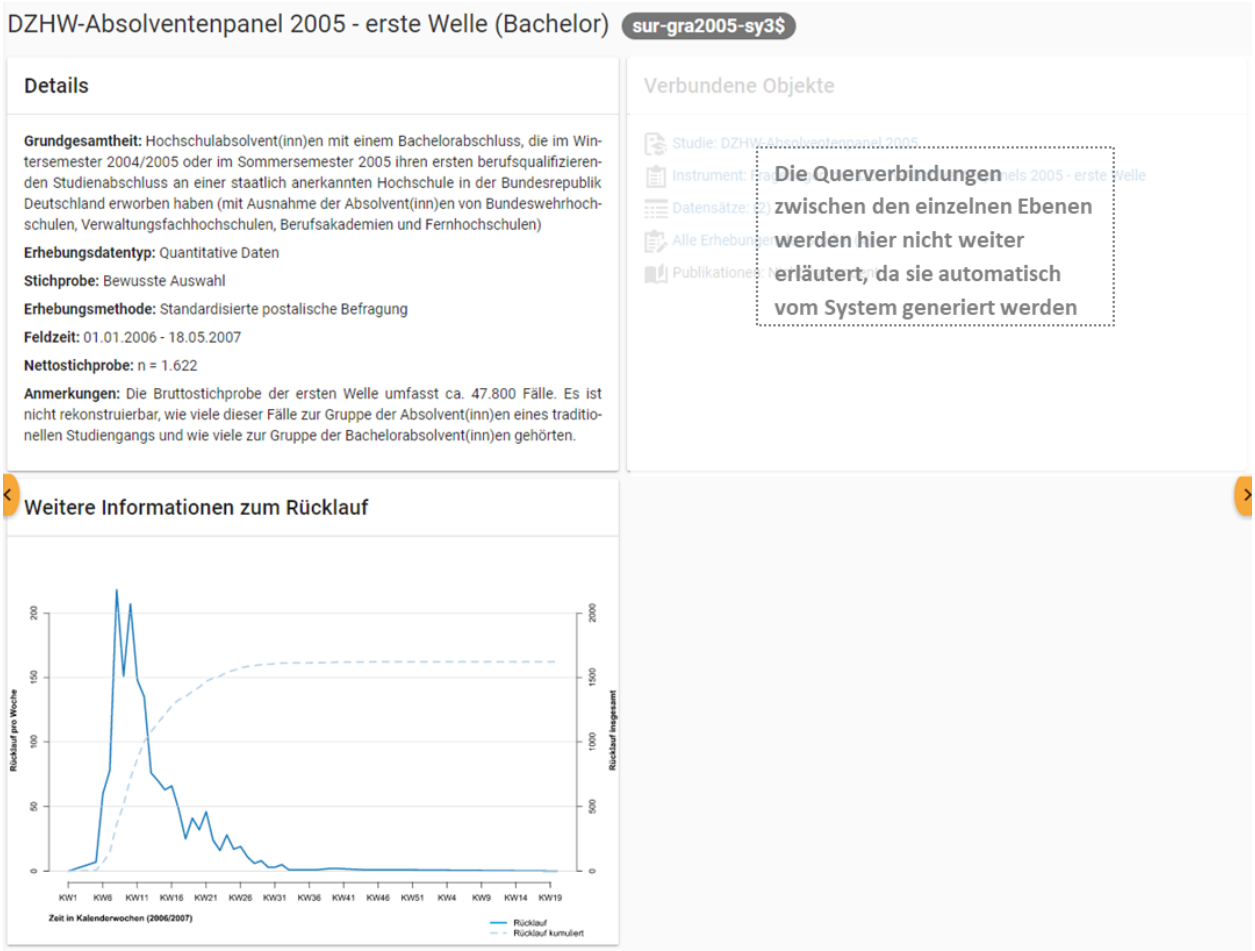


Abb. 2.22: Erhebungsübersicht im MDM am Beispiel der ersten Welle (Bachelor) im Absolventenpanel 2005

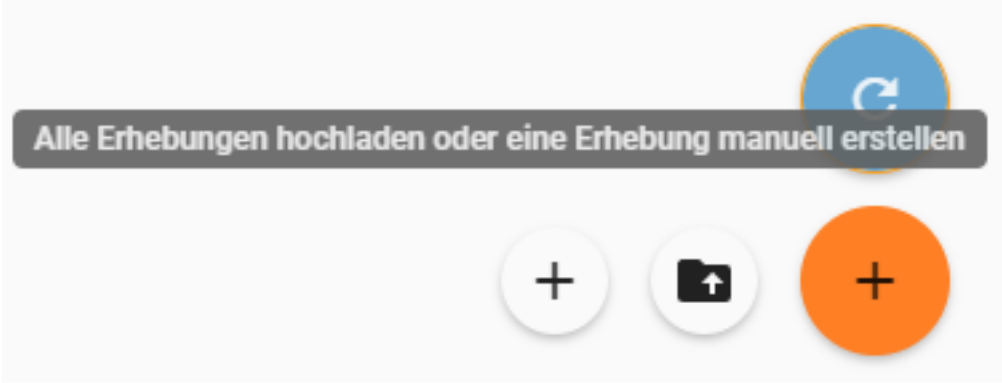


Abb. 2.23: Optionen für das Anlegen einer Erhebung

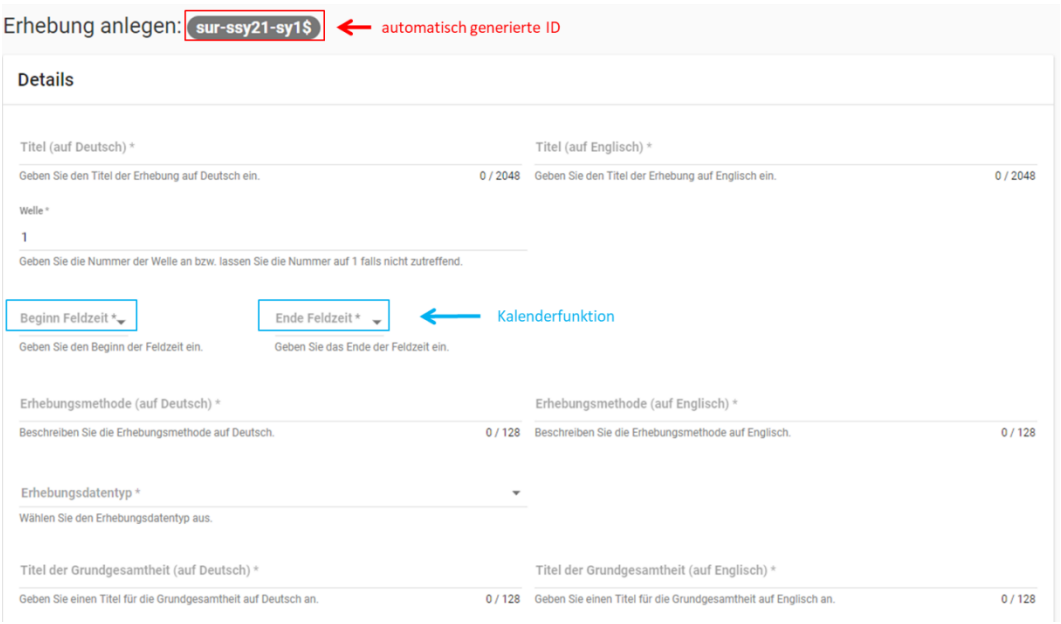


Abb. 2.24: Eingabemaske der Erhebungsebene, Abschnitt „Details“ Teil 1

es in dieser Eingabemaske zusätzlich eine Kalenderfunktion (s. blaue Kästchen, Abb. 2.24), welche die Feldzeit des Projekts erfasst und in Abb. 2.25 dargestellt ist:

Im zweiten Teil der Eingabemaske für die Erhebungsebene gibt es die Besonderheit, dass sich die Rücklaufquote automatisch ermitteln lässt (s. Abb. 2.26). Sie können den Rücklauf auch manuell eingeben. Hierbei ist zu jedoch beachten, dass sich bereits eingegebene Zahlen bei Brutto- und Netto-Stichprobe bei nicht automatisch anpassen.

Um den nächsten Abschnitt in der Eingabemaske („Weitere Informationen zum Rücklauf“[2]) bearbeiten zu können, müssen Sie die bisherigen Eingaben abspeichern. Dann können Sie deutschsprachige und/oder englischsprachige Grafiken zum Rücklauf entweder über den blauen Plus-Button oder per Drag & Drop hochladen und dann mit dem Save-Button speichern. Diese Grafiken dürfen im svg-, png- oder auch PDF-Format vorliegen. Über den Button mit dem Mülleimer-Symbol lassen sich hochgeladene Dateien wieder löschen (s. Abb. 2.27).

Im letzten Abschnitt der Eingabemaske können – wie auch bei der Studie – Materialien hinzugefügt werden (s. Abb. 2.28). Die Funktionsweise ist identisch zu der auf Studienebene.[3]

Editieren und historisieren

Falls Sie Ihre Informationen auf Erhebungsebene nicht in einem Vorgang eingeben und hochladen können oder möchten, ist es immer möglich, dass Sie Ihre bisherigen Eingaben abspeichern und zu einem späteren Zeitpunkt weiter bearbeiten. Hierfür wird Ihnen im Reiter „Erhebungen“ am rechten Rand ein Stift-Button angezeigt, über den Sie wieder in die Eingabemaske gelangen. Außerdem finden Sie dort auch einen Button mit Mülleimer-Symbol, mit dem Sie die Erhebung komplett löschen können (s. Abb. 2.29).

Es ist außerdem möglich, ältere Versionen der bereits gespeicherten Eingaben wiederherzustellen. Im Bearbeitungsmodus gibt es auch auf der Erhebungsebene einen Historisierungs-Button, den Sie rechts unten über dem Save-Button betätigen können (s. Abb. 2.30).

Bei einem Klick auf den Historisierungs-Button öffnet sich ein Dialog, der die verschiedenen Versionen der Erhebung anzeigt (s. Abb. 2.31). Zudem sind der Name des Nutzers, der die entsprechende Version der Studie gespeichert hat, sowie das Änderungsdatum sichtbar. Durch Klicken auf die Version wird diese wiederhergestellt, aber nicht automa-

[2] Rücklaufgrafiken sind nur im Dokumentationsstandard der Stufe 3 gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentationsstandards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“.

[3] Bitte beachten Sie, die dokumenteigenen Metadaten bei PDF-Dateien vorab zu löschen (vgl. Kapitel 3.3).

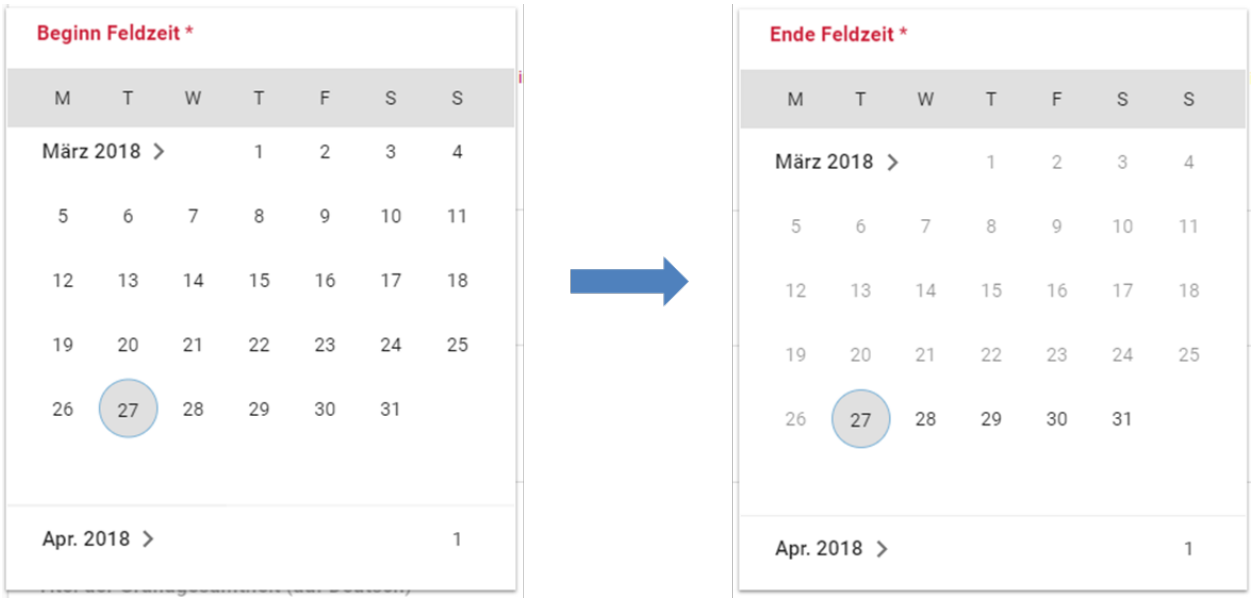


Abb. 2.25: Kalenderfunktion auf der Erhebungsebene

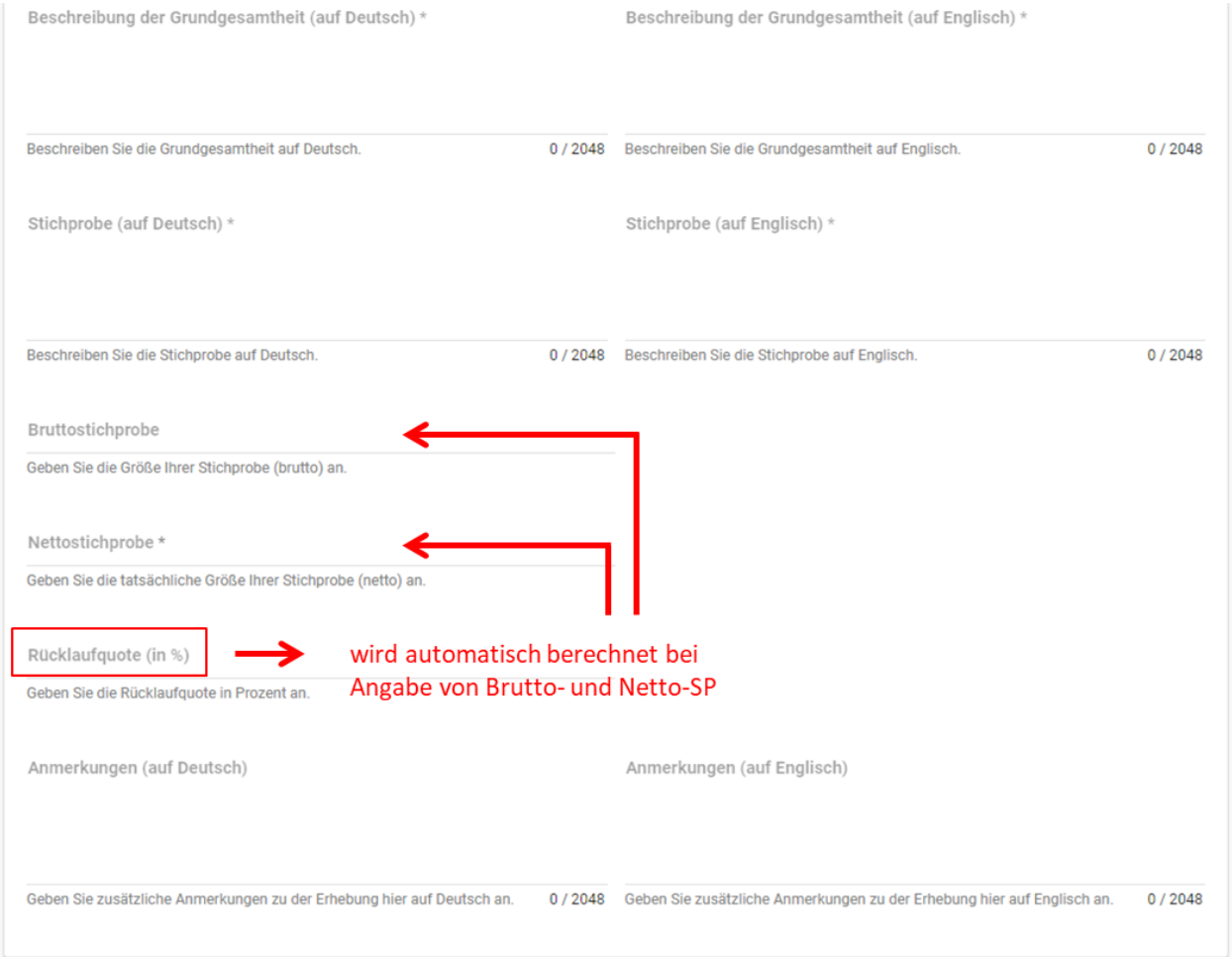


Abb. 2.26: Eingabemaske der Erhebungsebene, Abschnitt „Details“ Teil 2

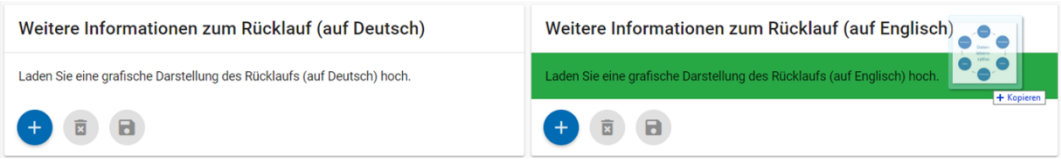


Abb. 2.27: Eingabemaske der Erhebungsebene, Abschnitt „Weitere Informationen zum Rücklauf“

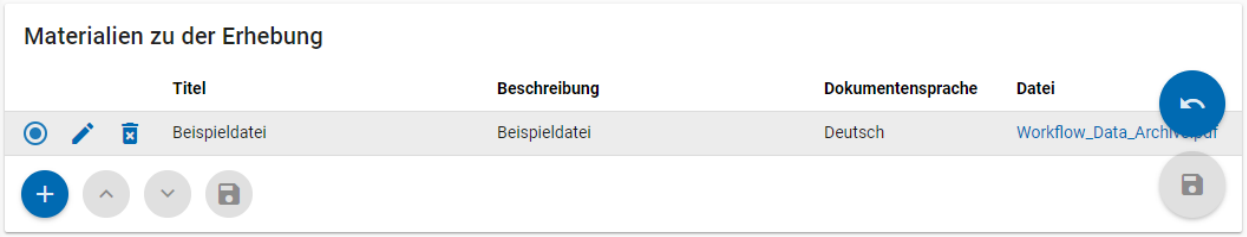


Abb. 2.28: Eingabemaske der Erhebungsebene, Abschnitt „Materialien zu der Erhebung“

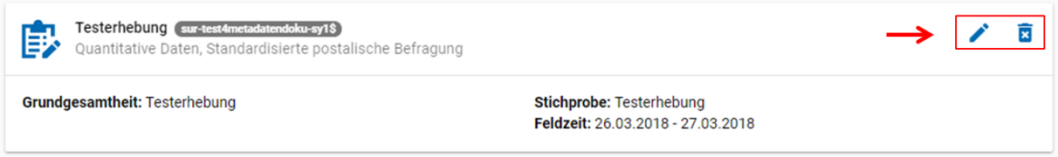


Abb. 2.29: Weitere Bearbeitung einer bereits abgespeicherten Erhebung

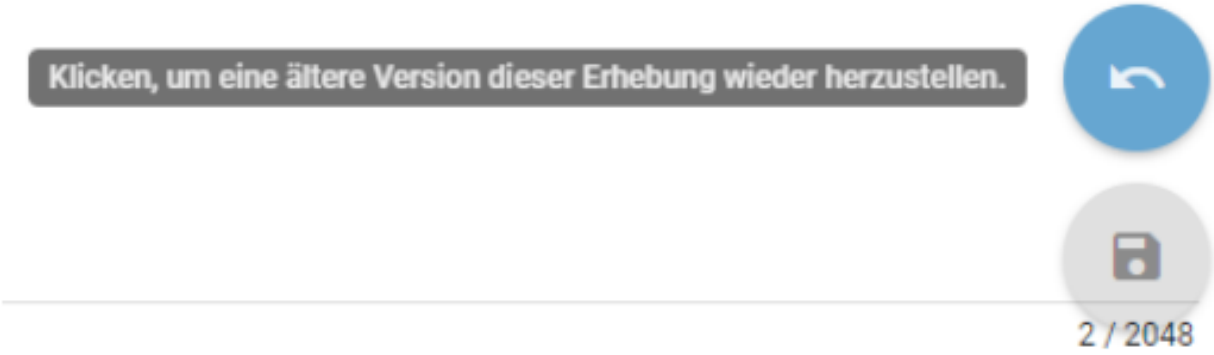


Abb. 2.30: Ältere Versionen einer Erhebung wiederherstellen

tisch als aktuelle Version gespeichert. Dies müsste über einen Klick auf den Save-Button erfolgen. Zu beachten ist, dass Materialien zur Erhebung nicht historisiert werden.

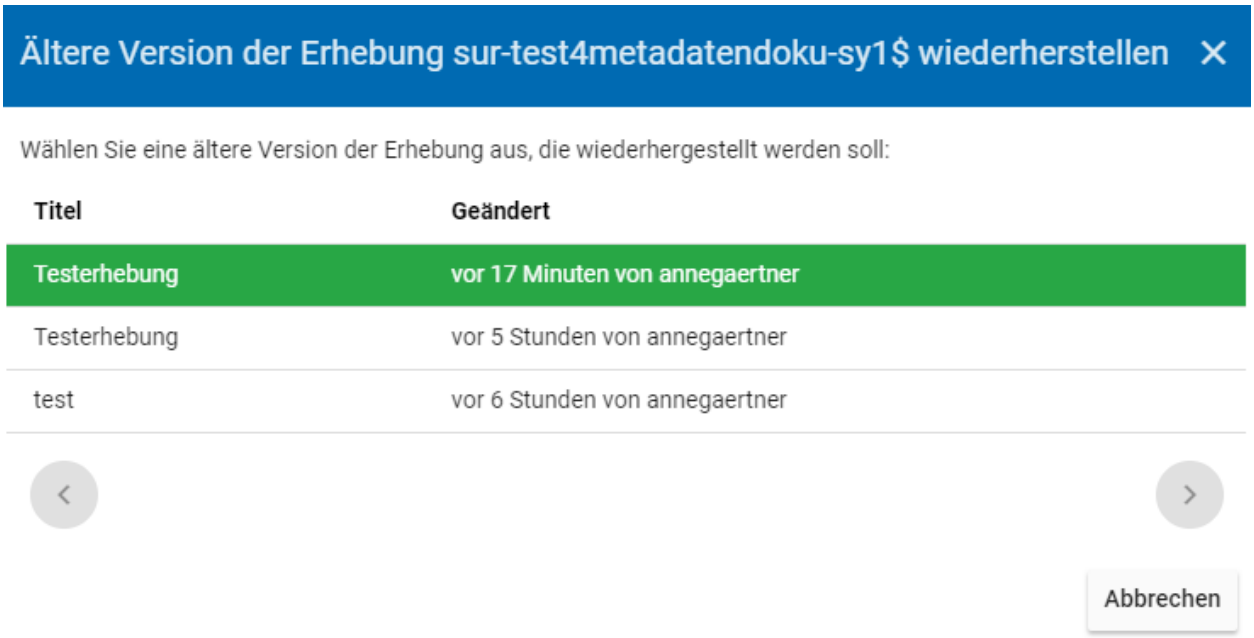


Abb. 2.31: Dialog zur Historisierung innerhalb einer Erhebung

2.4.3 Erhebungsinstrumente (instruments)

Übersicht

Wenn Sie Informationen über Ihre Erhebungsinstrumente aufnehmen, wird folgende Übersicht im MDM erstellt:

Eingabemaske

Erhebungsinstrumente lassen sich per Eingabemaske erfassen und editieren. Dafür darf die Studie aktuell nicht released sein. Um ein Erhebungsinstrument mittels Eingabemaske anzulegen muss man sich im Datenaufbereitungsprojekt im Instrumentereiter befinden. Anschließend wird der Plusbutton gedrückt und es öffnet sich der Dialog um ein neues Instrument anzulegen.

Die Eingabemaske besteht aus den Pflichtfeldern Beschreibung, Titel, Typ und Erhebung, sowie den nicht verpflichtenden Feldern Untertitel und Anmerkungen.

Des weiteren können weitere Materialien zum Instrument hochgeladen werden. Um weitere Materialien hochzuladen muss zunächst das Instrument abgespeichert sein. Im Anschluss muss der Plusbutton gedrückt werden, woraufhin sich ein Dialog öffnet (s. Abb. 2.35), in welchem der Anhang hochgeladen werden kann und Metadaten zur Datei eingegeben werden können. Um die Datei hochzuladen wird auf den Büroklammer-Button gedrückt und es öffnet sich ein weiterer Dialog. Alle Felder dieses Dialogs sind verpflichtend. Anschließend lässt sich der Anhang mit dem Speichern-Button (Diskettensymbol unten rechts) speichern.

Excel-Tabelle

Sie können für jede Ihrer Erhebungen alle zugehörigen Instrumente inkl. Anhänge innerhalb einer Tabelle eintragen und hochladen. Dazu müssen Sie die Excel-Datei *instruments.xlsx* ausfüllen, welche aus zwei Tabellenblättern besteht: *instruments* und *attachments*. Nachstehend finden Sie die Erläuterungen zur Tabelle:

Tabelle 2: Ausfüllanweisungen für die Excel-Tabelle „instruments“



Abb. 2.32: Instrumentenübersicht im MDM am Beispiel des Fragebogens der ersten Welle im Absolventenpanel 2005

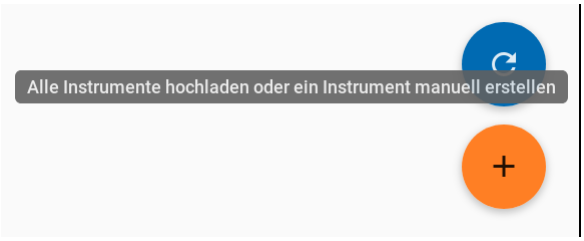


Abb. 2.33: Plusbutton

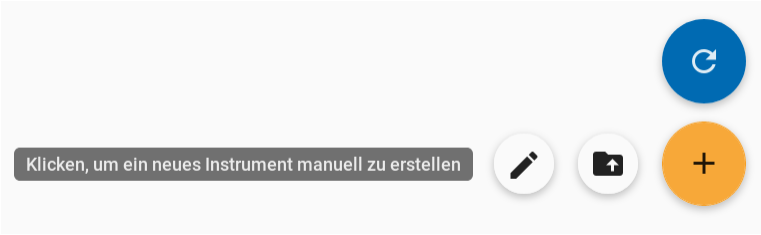


Abb. 2.34: Manuelles Anlegen des Instruments.

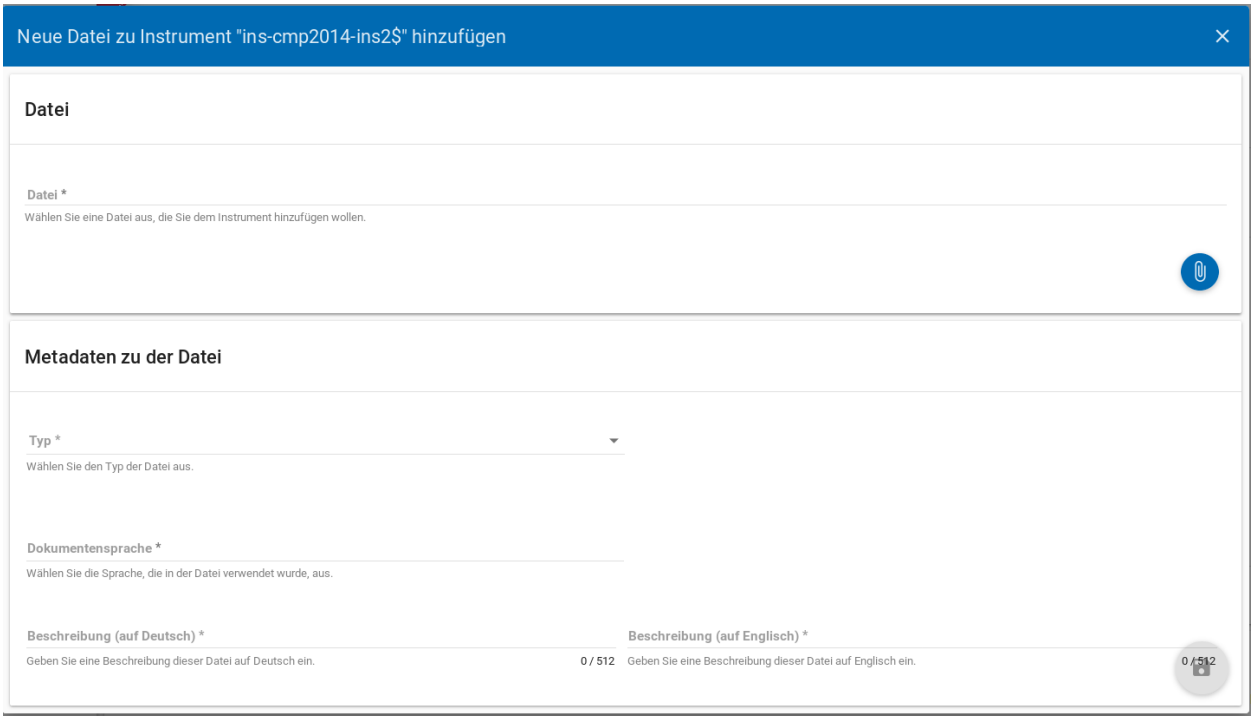


Abb. 2.35: Instrumente Anhang

Tabellenblatt 1: instruments		
Es können mehrere Instrumente eingetragen werden (= mehrere Zeilen möglich, ein Instrument pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
number	Ja	Nummer des Instrumentes (fortlaufend)
surveyNumbers	Ja	Nummer der zugehörigen Erhebungen
description.de/en	Ja	Beschreibung des Instrumentes
title.de/en	Ja	Titel des Instruments
subtitle.de/en	Nein	Untertitel des Instruments
type	Ja	Instrumenttyp: „PAPI“ (paper and pencil interview), „CAPI“ (computer-assisted personal interview), „CATI“ (computer-assisted telephone interview), „CAWI” (computer-assisted web interview)
annotations.de/en	Nein	Anmerkungen zum Instrument

Tabellenblatt 2: attachments		
Es können mehrere Anhänge eingetragen werden (= mehrere Zeilen möglich, ein Anhang pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
filename	Ja	Name der Datei im attachments-Ordner (z. B. „gra2005_W1_Questionnaire_de.pdf“)
type.de/en	Ja	de: „Fragebogen“, „Filterführungsdiagramm“, „Variablenfragebogen“, „Sonstige“; en: „Questionnaire“, „Question Flow“, „Variable Questionnaire“, „Other“
description.de/en	Ja	Beschreibung des Dokuments
language	Ja	Sprache des Anhangs *Bitte verwenden Sie eine Abkürzung nach ISO 639-1*: 26. (b) „de“, „en“
instrumentNumber	Ja	Nummer des dem Anhang zugehörigen Instruments

Zu den möglichen Anhängen zählen z. B. Fragebögen, Variablenfragebögen sowie Filterführungsdiagramme[4]. Diese müssen als PDF-Dateien vorliegen.[5] Außerdem können an dieser Stelle Codierlisten, welche als Excel-Tabelle vorliegen müssen, erfasst werden. Alle Anhänge werden im Ordner *attachments* abgelegt.

Für den Upload der Dateien ins MDM muss die Ordnerstruktur wie in Abb. 2.36 vorliegen.

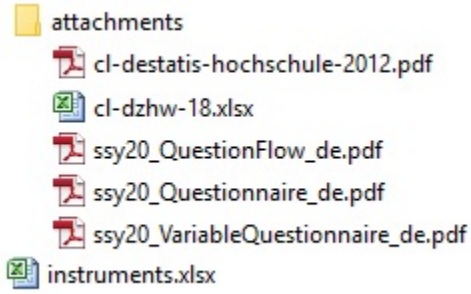


Abb. 2.36: Ordnerstruktur: Inhalt des Ordners instruments

Hochladen können Sie den Ordner im Reiter Instrumente entweder über den orangefarbenen Plus-Button (unten rechts) oder per Drag & Drop.

2.4.4 Fragen (questions)[6]

Übersicht

[4] Filterführungsdiagramme sind erst ab der 2. Dokumentationsstufe gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentationsstandards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“.

[5] Bitte beachten Sie, die dokumenteigenen Metadaten der PDF-Dateien vorab zu löschen (vgl. Kapitel 3.3).

[6] Metadaten auf Fragenebene sind erst ab der 2. Dokumentationsstufe gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentationsstandards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“.

Zu den einzelnen Fragen eines Instruments (sprich: Fragebogen) können Sie Informationen in das MDM übermitteln, in welchem dann für jede Frage folgende Übersichtsseite erstellt wird:

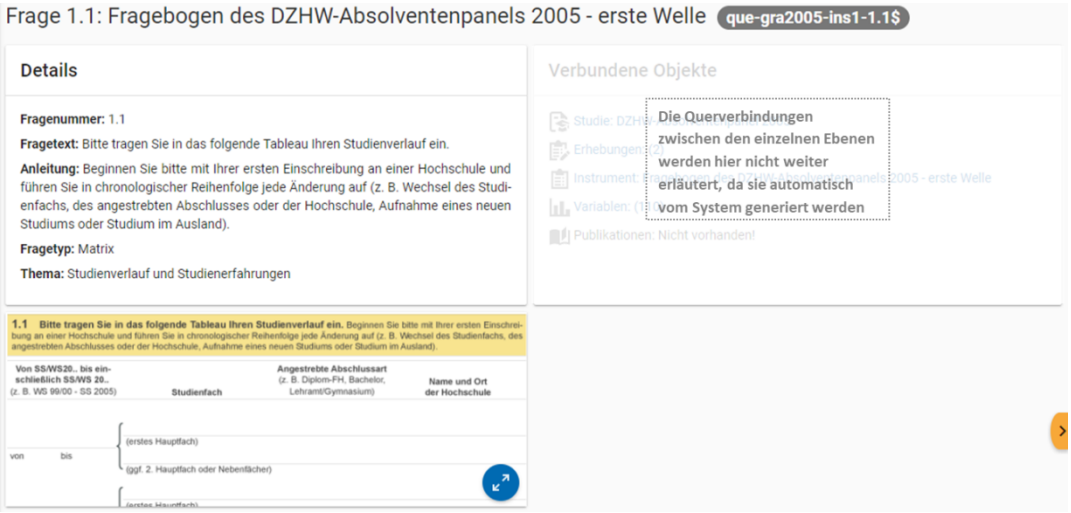


Abb. 2.37: Fragenübersicht im MDM am Beispiel der Frage 1.1 des Fragebogens der ersten Welle im Absolventenpanel 2005

Auf dieser Ebene werden Informationen über alle Fragen für jedes einzelne Erhebungsinstrument einer Studie abgeben. Der Einspeisungsprozess dieser Informationen hängt vom Typ des Erhebungsinstrumentes ab. Während Daten aus Onlinebefragungen, die mit ZOFAR, dem Datenerhebungssystem den DZHW, durchgeführt wurden, direkt aus dem System heraus extrahiert werden (siehe **Questions (ZOFAR)**), müssen Daten aus allen anderweitig durchgeführten Befragungen – sowohl andere Onlinebefragungen als auch PAPI-Befragungen – manuell erfasst werden (siehe **Questions (manuell)**). Im Folgenden werden beide Vorgehensweisen schrittweise beschrieben.

2.4.5 Questions (manuell)

Excel-Tabelle

Um Metadaten auf der Fragenebene in manueller Weise zu erfassen, müssen Sie die Excel-Datei *questions.xlsx* ausfüllen, welche die beiden Tabellenblätter *questions* und *images* beinhaltet. Sie können alle Fragen aus allen Erhebungsinstrumenten in einer einzigen Exceltabelle erfassen:

Tabelle 3: Ausfüllanweisungen für die Excel-Tabelle „questions“

Tabellenblatt 1: questions		
Es können mehrere Fragen eingetragen werden (= mehrere Zeilen möglich, eine Frage pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
indexInInstrument	Ja	Nummer der Frage im Fragebogen, nach der die Reihenfolge festgelegt wird (ganzzahlig)
questionNumber	Ja	Fragenummer, idealerweise selbsterklärend aus Instrument (z. B. 1.1)
instrumentNumber	Ja	Nummer des Instruments
questionsText.de/en	Ja	„Übergreifender" Fragetext, bei Itembatterien oder komplexen Fragen der einleitende Fragetext. Bei „einfachen" Fragetypen der komplette Fragetext.
instruction.de/en	Nein	wenn vorhanden, Anweisungstext der Frage
introduction.de/en	Nein	wenn vorhanden, Einleitungstext der Frage
type.de/en	Ja	de: „Einfachnennung", „Offen", „Mehrfachnennung", „Itembatterie" oder „Matrix" (eine Anleitung zur Einteilung der verschiedenen Fragetypen kann unter https://github.com/dzhw/metadatamanagement /files/1421895/Anleit ung_Vergabe_Fragetype n.docx gefunden werden) en: „Single Choice", „Open", „Multiple Choice", „Item Set" or „Grid".
topic.de/en	Nein	Themenblock, in dem die Frage im Instrument eingeordnet ist (idealerweise direkt aus Instrument entnehmbar)
successorNumbers	Nein	Fragenummern der nachfolgenden Frage(n) (Angabe in einer Zeile durch Komma getrennt)
technicalRepresentation.type	x*	Herkunft des Codeschnipsels (z. B. „ZOFAR-Question Markup Language")
technicalRepresentation.language	x*	Technische Sprache des Codeschnipsels (z. B. XML)
technicalRepresentation.source	x*	Codeschnipsel, um Frage technisch abbilden zu können (z. B. QML-Schnipsel)
additionalQuestionText.de/.en	Nein	Weitere Ausführungen der Frage, die nicht im Fragetext stehen, wie z. B. der Itemtext (bei Itembatterien) oder Antworttext (bei Mehrfachnennungen). Aktuell ist diese Information für den Nutzenden des MDM nicht sichtbar, sondern wird nur bei einer Volltextsuche berücksichtigt.
annotations.de/en	Nein	Anmerkungen zur Frage

x* = nur, wenn technicalRepresentation vorhanden (wird dann automatisch von ZOFAR geliefert)

Tabellenblatt 2: images		
Es können mehrere Bilder eingetragen werden (= mehrere Zeilen möglich, ein Bild pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
fileName	Ja	Dateiname des Bildes (z.B. „1.1_1.png“)
questionNumber	Ja	Dem Bild zugeordnete Fragenummer
instrumentNumber	Ja	Nummer des zum Bild gehörenden Instruments
language	Ja	Sprache des Bildes *Bitte verwenden Sie eine Abkürzung nach ISO 639-1*: z. B. „de“, „en“
indexInQuestion	Ja	Auf das wievielte Bild der Frage bezieht sich die Zeile? (Liegt pro Frage nur ein Bild vor, steht hier immer 1)

Mit dem zweiten Tabellenblatt *images* erfassen Sie Informationen zu den Fragebildern, welche Sie für jede Frage mit hochladen müssen. Zu jeder Frage muss mindestens ein Bild (es können auch mehrere sein) im png-Format vorhanden sein. Die Fragebilder können z. B. mit Ragtime extrahiert werden, sofern der Fragebogen auch mit Ragtime erstellt wurde. Ansonsten lassen sich die Fragebilder auch aus einer PDF-Datei erstellen.[7] Anleitung für beiden Varianten finden Sie unter https://github.com/dzhw/metadatamanagement-io/wiki/Bilderfassung-aus-RagTime und https://github.com/dzhw/metadatamanagement-io/wiki/Bilderfassung-aus-pdf.

Die fertig ausgefüllte Excel-Datei sowie die Bilder zu den Fragen speichern Sie dann in dem Ordner, den das FDZ für Sie vorbereitet hat. Das FDZ greift daraufhin auf die Dateien zu, verarbeitet sie weiter und lädt die Metadaten für die Fragenebene dann selbst ins MDM.

2.4.6 Questions (Zofar)

Bei Onlinebefragungen mit Zofar können die Metadaten für Fragen automatisch extrahiert werden (.jsons + .pngs).

Der Prozess befindet sich gerade im Aufbau…

2.4.7 Datensätze (dataSets)

Übersicht

Mit den Informationen über die Datensätze, welche Sie aus den Daten Ihrer Studie erstellt haben, wird für jeden dieser Datensätze folgende Übersicht im MDM angezeigt:

Eingabemaske

Datensätze lassen sich auch per Eingabemaske anlegen und editieren. Hierfür muss man auf den Reiter Datensätze klicken (Abb. 2.1), anschließend auf das Plussymbol (Abb. 2.39) in der unteren rechten Ecke klicken und dann auf das Stiftsymbol (Abb. 2.40) („Klicken um einen Datensatz manuell zu erstellen“).

Die mit * markierten Felder sind verpflichtend. Die verknüpften Erhebungen werden nach einem Klick in das Feld „Erhebungen“ automatisch vorgeschlagen und können per Klick ausgewählt werden. Im Anschluss werden die Subdatensätze per Eingabemaske auf der selben Seite eingegeben. Weitere Subdatensätze können per Klick auf das Plussymbol hinzugefügt werden. Nachdem gespeichert wurde, lassen sich weitere Materialien zum Datensatz hinzufügen.

Excel-Tabelle

[7] Bitte beachten Sie, die dokumenteigenen Metadaten der PDF-Dateien vorab zu löschen (vgl. Kapitel 3.3).



Abb. 2.38: Datensatzübersicht im MDM am Beispiel des Personendatensatzes (Bachelor) im Absolventenpanel 2005



Abb. 2.39: Neuen Datensatz hinzufügen.

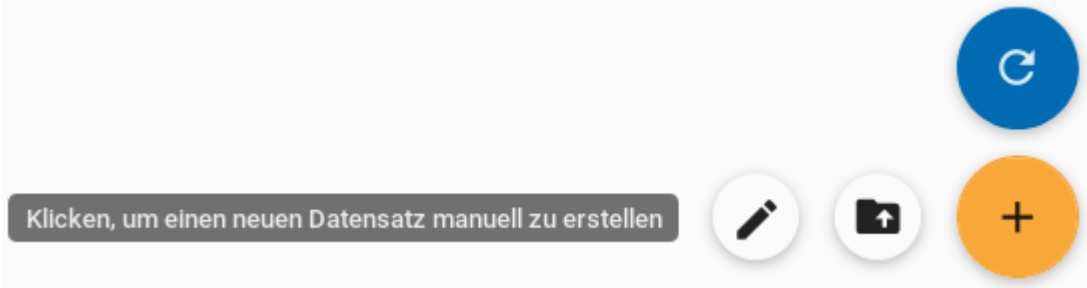


Abb. 2.40: Klicken um einen Datensatz manuell zu erstellen.

Auf der Datensatzebene werden alle Datensätze und Subdatensätze[8], die einer Studie zugeordnet sind, erfasst. Ihre Informationen zu den Datensätzen können Sie in die Excel-Tabelle *dataSets.xlsx*, welche aus drei Tabellenblättern (*dataSets*, *subDataSets* und *attachments*) besteht, eintragen.

Tabelle 4: Ausfüllanweisungen für die Excel-Tabelle „dataSets“

Tabellenblatt 1: dataSets		
Es können mehrere Datensätze eingetragen werden (= mehrere Zeilen möglich, ein Datensatz pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
number	Ja	Nummer des Datensatzes (laufende Nummer, eindeutig im Datenaufbereitungspro jekt)
description.de/en	Nein	Beschreibung des Datensatzes
type.de/en	Ja	de: „Personendatensatz“, „Episodendatensatz“; en: „Individual Data“, „Spell Data“
format.de/en	Nein	de: „breit“, „lang“; en: „wide“, „long“
surveyNumbers	Ja	Nummern der zum Datensatz gehörenden Erhebungen
annotations.de/en	Nein	Anmerkungen zum Datensatz

Tabellenblatt 2: subDataSets		
Es können mehrere Subdatensätze eingetragen werden (= mehrere Zeilen möglich, ein Subdatensatz pro Zeile)		
Spalten-über-schrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
name	Ja	Dateiname des „physikalischen“ Datensatzes ohne Dateikürzel (z. B. „gra2005_tS_p_c_1-0-0 “)
number-OfObserva-tions	Ja	Anzahl der Beobachtungseinheiten in einem Datensatz (z. B. Anzahl der Befragten)
accessWay	Ja	„download-cuf“, „download-suf“, „remote-desktop-suf“ oder „onsite-suf“
descripti-on.de/en	Ja	Beschreibung des Datensatzes
dataSet-Number	Ja	Nummer des zugehörigen Datensatzes
citation-Hint.de/en	Nein	Hinweise zur Zitation: Primärforscher_1, Primärforscher_2 & Primärforscher_3 (Jahr). Studientitel. Aufbereitet durch FDZMitarbeiter_1, FDZMitarbeiter_2 & FDZ-Mitarbeiter_3, doi: 10.21249/DZHW:a-Z0-9: 0-9.0-9.0-9, released Jahr. Hannover: FDZ-DZHW.

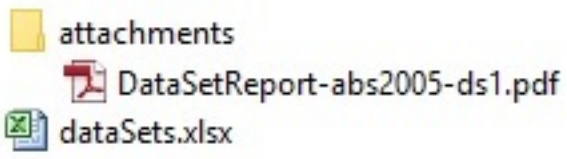
[8] Mit Subdatensätzen sind solche gemeint, die Sie nach einer Anonymisierung Ihrer Daten erhalten. Sie können mehrere Stufen der Anonymisierung verwenden, wobei jede Stufe einen eigenen Zugangsweg zu den anonymisierten Daten mit sich bringt. Für jeden Zugangsweg wird dann ein eigener Subdatensatz erstellt (vgl. hierzu „accessWay“ im Tabellenblatt „subDataSets“).

Tabellenblatt 3: attachments		
Es können mehrere Anhänge eingetragen werden (= mehrere Zeilen möglich, ein Anhang pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
fileName	Ja	Name der Datei im attachments-Ordner (z. B. „DataSetReport-abs200 5-ds1.pdf“)
title	Ja	Titel des Anhangs
description.de/en	Ja	Beschreibung des Anhangs
language	Ja	Sprache des Anhangs *Bitte verwenden Sie eine Abkürzung nach ISO 639-1*: z. B. „de“, „en“
dataSetNumber	Ja	Nummer des zugehörigen Datensatzes

Wenn Sie Materialien auf Ebene der Datensätze haben, können Sie diese auch hier wieder im Ordner *attachments* ablegen.[9]

Um den erfolgreichen Upload der Informationen ins MDM gewährleisten, ist auf Datensatzebene folgende Ordnerstruktur vorgegeben:

Ordnerstruktur: Inhalt des Ordners dataSets



Hochladen können Sie den Ordner im Reiter Datensätze entweder über den orangefarbenen Plus-Button (unten rechts) oder per Drag & Drop.

2.4.8 Variablen (variables)[10]

Übersicht

Anhand der Informationen, die Sie auf Ebene der Variablen abgeben, wird für jede Variable eine Übersichtsseite im MDM erstellt:

Die Erstellung der Variablenebene beinhaltet einerseits recht viel Aufwand, da für jeden Datensatz eine eigene Excel-Tabelle mit Informationen zu allen Variablen geliefert werden muss. Viele Informationen müssen manuell eingetragen werden, einige können – sofern die Befragung über Zofar stattgefunden hat – auch direkt aus Zofar extrahiert werden oder sogar aus der Excel-Tabelle der Frageebene importiert werden.

Die Variablenebene ist andererseits sehr wertvoll im Hinblick auf die Nachnutzbarkeit der Forschungsdaten. Wenn Metadaten auf dieser Ebene vorhanden sind, können die dazugehörigen Daten auch aus inhaltlicher Sicht umfassend durchsucht werden, sodass Analysepotential auch für sehr spezielle Fragestellungen direkt sichtbar wird.

Für die Darstellung der Metadatenaufnahme auf Variablenebene gilt es noch folgende Dinge zu beachten:

- Wenn Sie mehrere Datensätze liefern: Es darf kein Variablenname doppelt vorkommen.
- Missings müssen global definiert sein, d. h. sie müssen für alle Variablen eines Datensatzes gelten.

[9] Bitte beachten Sie, die dokumenteigenen Metadaten bei PDF-Dateien vorab zu löschen (vgl. Kapitel 3.3).

[10] Metadaten auf Variablenebene sind erst ab der 2. Dokumentationsstufe gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentationsstandards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“.

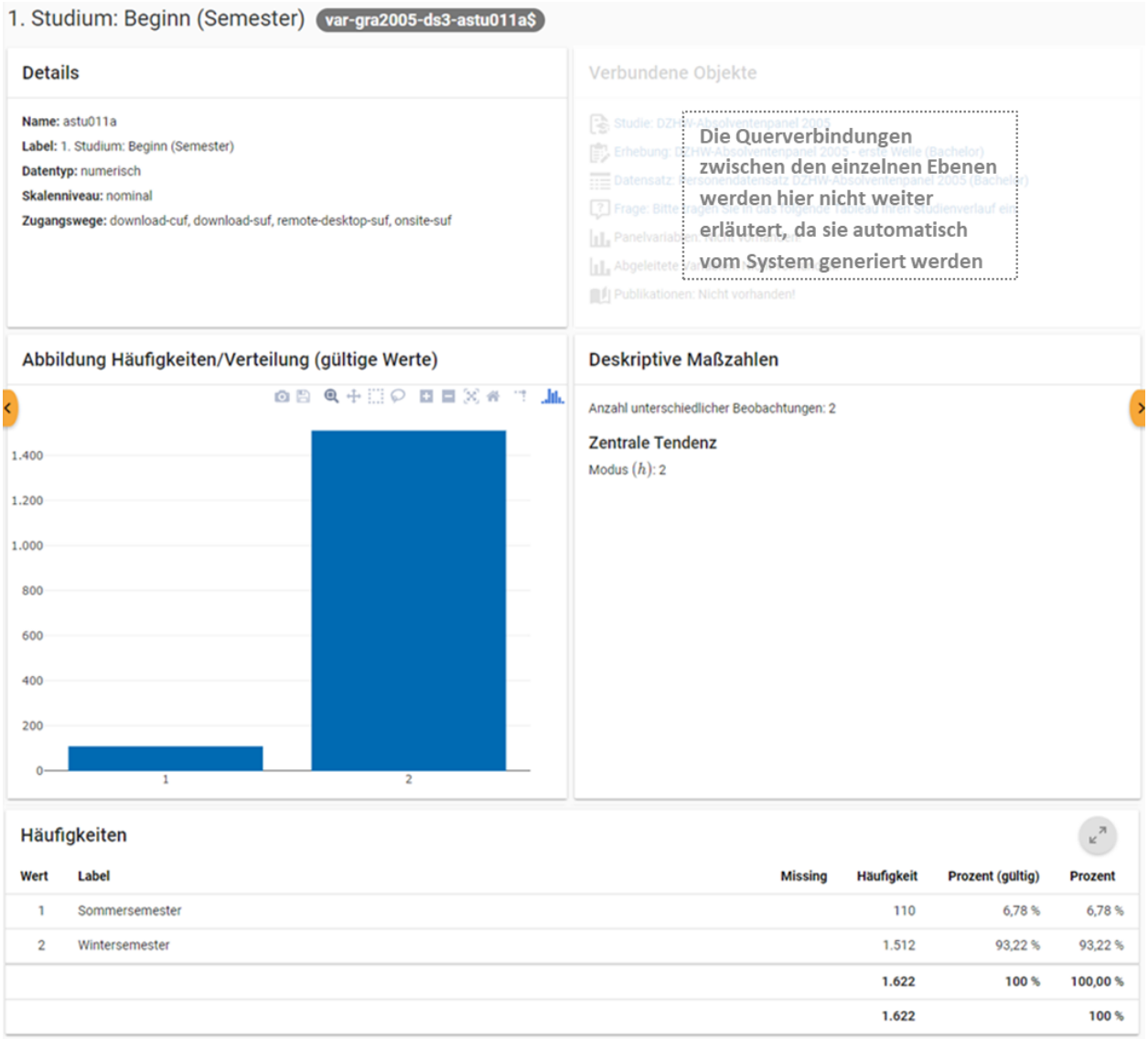


Abb. 2.41: Variablenübersicht im MDM am Beispiel der Variable „1. Studium: Beginn (Semester)“ im Absolventenpanel 2005, erste Welle (BA)

Excel-Tabelle

Ausfüllen müssen Sie je nach Anzahl der Datensätze mindestens eine Excel-Datei mit dem Namen *vimport_ds***Nr.*.xlsx*, wobei die **„Nr."** im Dateinamen der Nummer des dazugehörigen Datensatzes entsprechen muss, d. h. die Variablen des Datensatzes mit der Nummer 1 muss *vimport_ds1.xlsx* heißen usw. Die Datei enthält die beiden Tabellenblätter *variables* und *relatedQuestions*.

Tabelle 5: Ausfüllanweisungen für die Excel-Tabelle „vimport_ds*Nr*."

Tabellenblatt 1: variables		
Es können mehrere Variablen eingetragen werden (= mehrere Zeilen möglich, eine Variable pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
name	Ja	Variablenname
surveyNumbers	Ja*	Angabe aller der Variablen zugehörigen Erhebungsnummern (in einer Zelle durch Komma getrennt)
scaleLevel.de/.en	Ja	de: „nominal“, „ordinal“, „intervall“ oder „verhältnis“ en: „nominal“, „ordinal“, „intervall“ or „ratio“
panelIdentifier	Nein*	Identifier zur eindeutigen Zuordnung von Panelvariablen. Präfix muss aus der Projekt-ID + Nummer des Datensatzes bestehen (Beispiel: *gra2005-ds1*), der hintere Teil des Identifiers ist beliebig wählbar, muss aber eindeutig sein. Beispiel: Sind die Variablen *astu01a* und *bstu01a* aus dem 1. Datensatz des Projekts *gra2005* Panelvariablen, so könnte der Identifier *gra2005-ds1-stu01a* lauten.
annotations.de/en	Nein	Anmerkungen zur Variablen
accessWays	Ja*	Mögliche Zugangswege: Download-CUF, Download-SUF, Remote-Desktop-SUF, On-Site-SUF. Bei mehreren Zugangswegen sind den verschiedenen Zugangswegen entsprechend Spalten vorhanden, die mit „nicht verfügbar im … “ überschrieben sind. Für jede Variable muss dann ein „x“ gesetzt werden, wenn ´diese über den jeweiligen Zugangsweg nicht vorhanden ist.
filterDetails.description.de/.en	Nein	Verbalisierte Beschreibung des Variablenfilters
filterDetails.expression[11]	Ja, wenn Filter vorhanden	Regel, die in der angegebenen „Sprache“ (.expressionLanguage) beschreibt, welche Teilpopulation zu dieser Variable hin gefiltert wurde (auch verschachtelte Filterführung wird beachtet (PAPI))
filterDetails.expressionLanguage[12]	Ja, wenn Filter vorhanden	Sprache des Filterausdrucks: „Stata“
generationDetails.description.de/.en	Nein	Beschreibung, wie die Variable erzeugt wurde, wenn sie nicht direkt aus dem Fragebogen abgelesen werden kann (Beispiel, siehe Abschnitt „Generierungsdetails“)
generationDetails.rule	Ja, wenn Variable	Regel, die in der angegebenen „Sprache“ (.ruleExpressionLanguage) beschreibt, wie die Variable erzeugt wurde (Beispiel, siehe Abschnitt „Generierungsregel (Stata)“)

* Wenn eigene Konventionen verwendet werden, muss das Feld manuell ausgefüllt werden. Bei Verwendung von FDZ-eigenen Schemata kann dieses Feld auch leer gelassen werden.

Tabellenblatt 2: relatedQuestions		
Variablen, die mit mehreren Fragen verbunden sind, können mehrfach aufgeführt werden. Variablen, die keiner Frage (oder keinem Instrument) zugeordnet sind, müssen nicht eingetragen werden.		
Es können mehrere verbundene Fragen eingetragen werden (= mehrere Zeilen, eine verbundene Frage pro Zeile)		
Spaltenüberschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
name	Ja	Variablenname
relatedQuestionString s.de/.en	Nein	Text, der den Frageinhalt der Variable darstellt. Also Fragetext der dazugehörigen Frage plus evtl. weitere Ausführungen wie bspw. der Itemtext (bei Itembatterien) oder der Antworttext (bei Einfach- oder Mehrfachnennungen)
questionNumber	Ja	Nummer der zur Variablen zugehörigen Frage im Fragebogen
instrumentNumber	Ja	Nummer des zur Variablen zugehörigen Fragebogens

Dem Namen entsprechend wird aus den Informationen des zweiten Tabellenblatts die Verknüpfung zwischen einer Variablen und der dazugehörigen Frage aus dem Erhebungsinstrument erstellt. Für eine nachvollziehbare Dokumentation dieser Verbindung ist die Erstellung eines Variablenfragebogens sehr hilfreich. Aus diesem kann die Verknüpfung aus Variable und Frage problemlos abgelesen werden. Abb. 2.42 zeigt beispielhaft, dass den Variablen *astu08a* bis *astu08e* die Frage 1.8 zugeordnet ist.

Außer der/den Excel-Tabelle/n müssen Sie für jede Tabelle noch den zugehörigen Stata-Datensatz liefern, aus dem die Variablen stammen. Diese Dateien speichern Sie dann in dem Ordner, den das FDZ für Sie vorbereitet hat. Das FDZ greift daraufhin auf die Dateien zu, verarbeitet sie weiter und lädt die finalisierten Metadaten für die Variablenebene dann selbst ins MDM.

2.4.9 Publikationen (relatedPublications)

Überblick

Auf der Ebene der Publikationen werden wissenschaftliche Veröffentlichungen, welche auf Grundlage von Daten Ihres Projekts verfasst worden sind, erfasst. Die Informationen, die Sie im Hinblick auf Ihre Publikationen abgeben, werden im MDM für jede Veröffentlichung wie folgt dargestellt:

Wenn Sie Publikationen zu Ihren Daten abgeben möchten, senden Sie dem FDZ per Mail die PDF-Datei Ihrer Publikation sowie den dazugehörigen Zitationshinweis zu. Die weitere Bearbeitung wie z. B. die Registrierung für das Erhalten einer DOI und den Upload ins MDM übernimmt das FDZ.

2.5 Die Freigabe eines neuen Projekts

Wenn Sie alle Metadaten ausgefüllt bzw. ans FDZ gesendet haben, melden Sie sich beim FDZ mit dem Hinweis, dass Sie Ihre Daten nicht weiter editieren möchten. Das FDZ nimmt ihre Daten dann in die sogenannte Release-Pipeline auf. Die finale Freigabe erfolgt dann über einen dafür benannten Mitarbeiter des FDZ, den Release-Manager.

[11] Nur in der Dokumentationsstufe 3 gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentationsstandards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“.

[12] Nur in der Dokumentationsstufe 3 gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentationsstandards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“.

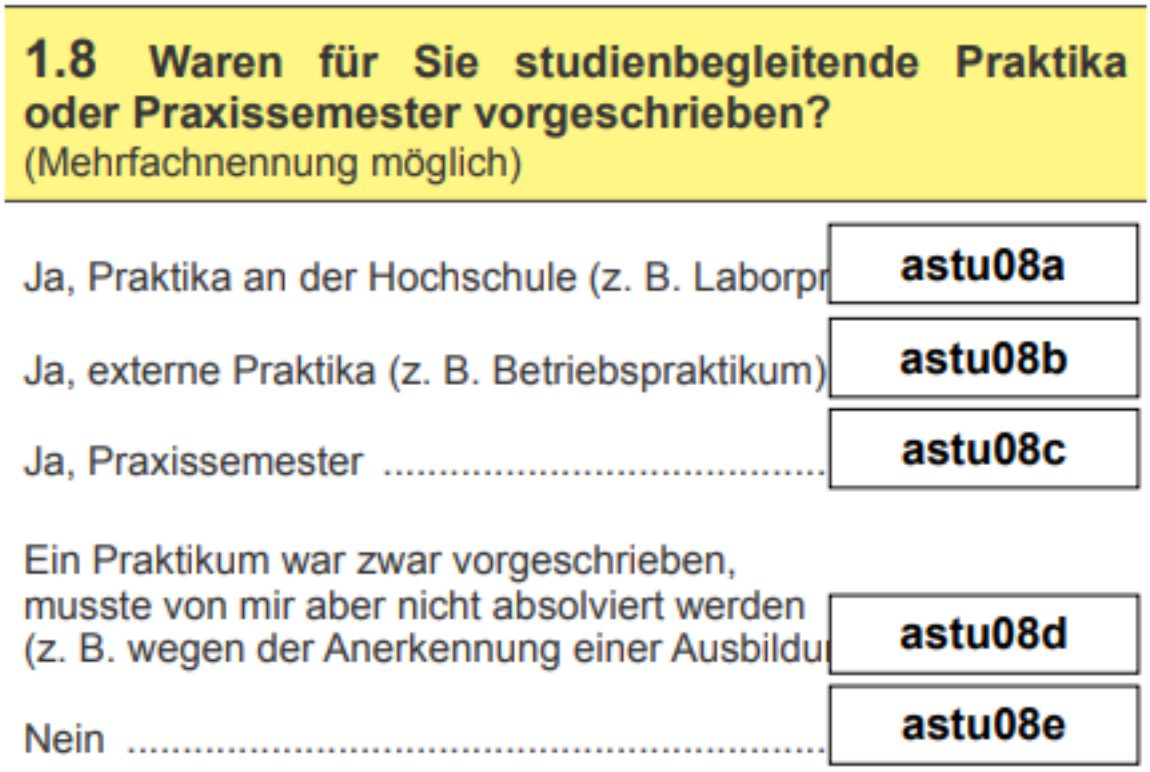


Abb. 2.42: Ausschnitt aus dem Variablenfragebogen des Absolventenpanels 2005, erste Welle, Frage 1.8

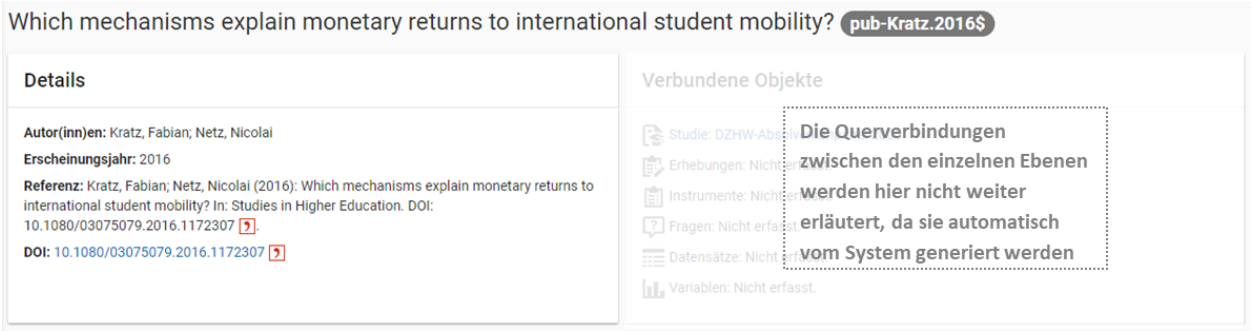


Abb. 2.43: Publikationsübersicht im MDM am Beispiel einer Veröffentlichung, welche im Rahmen des Absolventenpanels 2005 verfasst wurde

2.6 Anhang

2.6.1 Checkliste für Abgabe der Metadaten

Vor Abgabe bzw. dem Hochladen der Daten sind folgende Punkte zu überprüfen:

Ordnerstruktur und Dateinamen sind unverändert

Excel-Dateien sind richtig und vollständig ausgefüllt

- verpflichtende Felder sind ausgefüllt (vgl. Kapitel 4)
- es sind keine Fehlermeldungen vorhanden
- die Daten wurden auf Richtigkeit überprüft

Metadaten sind aus PDF-Dokumenten entfernt (vgl. Kapitel 3.3)

Nicht benötigte Zeilen entfernt (vgl. Kapitel 3.2)

- instruments.xlsx: löschen bis Zeile 20
- dataSets.xlsx: löschen bis Zeile 20
- questions.xlsx: löschen bis Zeile 2000
- variables.xlsx: löschen bis Zeile 2000

Dateien sind richtig abgegeben worden

In das Metadatensystem eingegeben

- Studie (study)
- Erhebungen (surveys)

In das Metadatensystem hochgeladen

- instruments.xlsx
- dataSets.xlsx

In der Ordnerstruktur abgelegt

- questions.xlsx
- variables.xlsx

KAPITEL 3

Javadoc

3.1 eu.dzhw.fdz.metadatamanagement.common.domain

Common domain objects which can be used in all other subdomains of this application.

3.1.1 AbstractRdcDomainObject

public abstract class **`AbstractRdcDomainObject`**
: Base class for all rdc domain objects. All domain objects inherit the fields from this base class.

Fields

createdBy

private String **`createdBy`**
: The name of the user which has created this object.

createdDate

private LocalDateTime **`createdDate`**
: The date and time (in UTC) when this domain object was created.

lastModifiedBy

private String **`lastModifiedBy`**
: The name of the user who last saved this object.

lastModifiedDate

private LocalDateTime **lastModifiedDate**
The date and time when this object was last saved.

version

private Long **version**
Number which is incremented on each save of this object.

3.1.2 Counter

public class **Counter**
Counter document which can be used to get an incremented sequence number per document id.

Fields

id

private String **id**
The id of the counter, e.g. „orders“.

seq

private long **seq**
The current sequence number.

3.1.3 I18nString

public class **I18nString**
Strings that can be represented in English and German.

Fields

de

private String **de**
The german version of this string.

en

private String **en**
The english version of this string.

3.1.4 Period

public class **`Period`**
Objects representing periods in time. All periods must have a start date and an end date and the start date must be before or equal to the end date.

Fields

end

private LocalDate **`end`**
The end date of the period. Mandatory and must not be before start date.

start

private LocalDate **`start`**
The start date of the period. Mandatory and must not be after end date.

3.1.5 Person

public class **`Person`**
A representation of a person.

Fields

firstName

private String **`firstName`**
The first name of the person. Must not be empty.

lastName

private String **`lastName`**
The last name of the person. Must not be empty.

middleName

private String **`middleName`**
The middle name of the person.

3.1.6 Resolution

public class **`Resolution`**
Representation of the resolution of images.

Fields

heightY

private Integer **heightY**
The height in pixel.

widthX

private Integer **widthX**
The width in pixel.

3.2 eu.dzhw.fdz.metadatamanagement.datasetmanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.datasetmanagement.domain.DataSet*s.

3.2.1 DataSet

public class **DataSet** extends *AbstractRdcDomainObject*
A dataset contains *Variable*s. It results from at least one *Survey*.

Fields

annotations

private *I18nString* **annotations**
Arbitrary additional text for the dataset. Must not contain more than 2048 characters.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which this dataset belongs. The dataAcquisitionProjectId must not be empty.

description

private *I18nString* **description**
A short description of the dataset. It must be specified in at least one language and it must not contain more than 2048 characters.

format

private *I18nString* **format**
The format of the dataset. Must be one of *Format*.

id

private String **id**
The id of the dataset which uniquely identifies the dataset in this application. The id must not be empty and must be of the form dat-{{dataAcquisitionProjectId}}-ds{{number}}$. The id must not contain more than 512 characters.

number

private Integer **number**
The number of the dataset. Must not be empty and must be unique within the *DataAcquisitionProject*.

studyId

private String **studyId**
The id of the *OrderedStudy* to which this dataset belongs. Must not be empty.

subDataSets

private List<*SubDataSet*> **subDataSets**
List of *SubDataSet*s (concrete accessible files) within this dataset. Must contain at least one element. There must not be more than one *SubDataSet* per *AccessWays*.

surveyIds

private List<String> **surveyIds**
List of ids of *Survey*s of this *DataAcquisitionProject*. The dataset contains results from these *Survey*s. Must contain at least one element.

surveyNumbers

private List<Integer> **surveyNumbers**
List of numbers of *Survey*s of this *DataAcquisitionProject*. The dataset contains results from these *Survey*s. Must contain at least one element.

type

private *I18nString* **type**
The type of the dataset. Must be one of *DataSetTypes* and must not be empty.

3.2.2 DataSetAttachmentMetadata

public class **DataSetAttachmentMetadata** extends *AbstractRdcDomainObject*
Metadata which will be stored with each attachment of a *DataSet*.

Fields

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which the *DataSet* of this attachment belongs. Must not be empty.

dataSetId

private String **dataSetId**
The id of the *DataSet* to which this attachment belongs. Must not be empty.

dataSetNumber

private Integer **dataSetNumber**
The number of the *DataSet* to which this attachment belongs. Must not be empty.

description

private *I18nString* **description**
A description for this attachment. It must be specified in at least one language and it must not contain more than 512 characters.

fileName

private String **fileName**
The filename of the attachment. Must not be empty and must contain only (german) alphanumeric characters and „_“ and „-“ and „.“.

id

private String **id**
The id of the attachment. Holds the complete path which can be used to download the file.

indexInDataSet

private Integer **indexInDataSet**
The index in the *DataSet* of this attachment. Used for sorting the attachments of this *DataSet*. Must not be empty.

language

private String **language**
The language of the attachments content. Must not be empty and must be specified as ISO 639 language code.

title

private String **title**
The title of the attachment in the language of the attachment. Must not be empty and must not contain more than 2048 characters.

3.2.3 DataSetTypes

public class **DataSetTypes**
All possible types of a *DataSet*.

Fields

ALL

public static final Set<*I18nString*> **ALL**

EPISODE_RECORD

public static final *I18nString* **EPISODE_RECORD**

PERSONAL_RECORD

public static final *I18nString* **PERSONAL_RECORD**

3.2.4 Format

public class **Format**
All possible formats of a *DataSet*.

Fields

ALL

public static final Set<*I18nString*> **ALL**

LONG

public static final *I18nString* **LONG**

WIDE

public static final *I18nString* **WIDE**

3.2.5 SubDataSet

public class **SubDataSet**

A subdataset is part of a *DataSet* and describes the concrete analyzable file which is accessible by a given access way.

Fields

accessWay

private String **accessWay**

The access way of this subdataset. Describes how the user will be able to work with the data set. Must not be empty and be one of *AccessWays* but not *AccessWays.NOT_ACCESSIBLE*.

citationHint

private *I18nString* **citationHint**

A hint telling how to cite this subdataset in publications. It must be specified in at least one language and it must not contain more than 2048 characters.

description

private *I18nString* **description**

A description for this subdataset. It must be specified in at least one language and it must not contain more than 512 characters.

name

private String **name**

The filename of the subdataset without extension. Must not be empty and must not contain more than 32 characters.

numberOfObservations

private Integer **numberOfObservations**

The number of rows (observations or episodes) which are present in this subdataset. Must not be empty.

3.3 eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain.Instrument*s.

3.3.1 Instrument

public class **Instrument** extends *AbstractRdcDomainObject*

An instrument (e.g. a questionnaire) which was used in at least one *Survey*.

Fields

annotations

private *I18nString* **annotations**
Arbitrary additional text for this instrument. Must not contain more than 2048 characters.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which this instrument belongs. The dataAcquisitionProjectId must not be empty.

description

private *I18nString* **description**
A short description of the instrument. It must be specified in at least one language and it must not contain more than 512 characters.

id

private String **id**
The id of the instrument which uniquely identifies the instrument in this application. The id must not be empty and must be of the form ins-{{dataAcquisitionProjectId}}-ins{{number}}$. The id must not contain more than 512 characters.

number

private Integer **number**
The number of the instrument. Must not be empty and must be unique within the *DataAcquisitionProject*.

studyId

private String **studyId**
The id of the *OrderedStudy* to which this instrument belongs. Must not be empty.

subtitle

private *I18nString* **subtitle**
An optional subtitle of the instrument. It must not contain more than 2048 characters.

surveyIds

private List<String> **surveyIds**
List of ids of *Survey*s of this *DataAcquisitionProject*. The instrument has been used in these *Survey*s. Must contain at least one element.

surveyNumbers

private List<Integer> **surveyNumbers**
List of numbers of *Survey*s of this *DataAcquisitionProject*. The instrument has been used in these *Survey*s. Must contain at least one element.

title

private *I18nString* **title**
The title of the instrument. It must be specified in at least one language and it must not contain more than 2048 characters.

type

private String **type**
The type of this instrument. Must be one of *InstrumentTypes* and must not be empty.

3.3.2 InstrumentAttachmentMetadata

public class **InstrumentAttachmentMetadata** extends *AbstractRdcDomainObject*
Metadata which will be stored with each attachment of a *Instrument*.

Fields

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which the *Instrument* of this attachment belongs. Must not be empty.

description

private *I18nString* **description**
A description for this attachment. It must be specified in at least one language and it must not contain more than 512 characters.

fileName

private String **fileName**
The filename of the attachment. Must not be empty and must contain only (german) alphanumeric characters and „_“ and „-“ and „.“.

id

private String **id**
The id of the attachment. Holds the complete path which can be used to download the file.

indexInInstrument

private Integer **indexInInstrument**
The index in the *Instrument* of this attachment. Used for sorting the attachments of this *Instrument*. Must not be empty.

instrumentId

private String **instrumentId**
The id of the *Instrument* to which this attachment belongs. Must not be empty.

instrumentNumber

private Integer **instrumentNumber**
The number of the *Instrument* to which this attachment belongs. Must not be empty.

language

private String **language**
The language of the attachments content. Must not be empty and must be specified as ISO 639 language code.

type

private *I18nString* **type**
The type of this attachment. Must not be empty and must be one of *InstrumentAttachmentTypes*.

3.3.3 InstrumentAttachmentTypes

public class **InstrumentAttachmentTypes**
All valid types of an instrument attachment.

Fields

ALL

public static final Set<*I18nString*> **ALL**

OTHER

public static final *I18nString* **OTHER**

QUESTIONNAIRE

public static final *I18nString* **QUESTIONNAIRE**

QUESTION_FLOW

public static final *I18nString* **QUESTION_FLOW**

VARIABLE_QUESTIONNAIRE

public static final *I18nString* **VARIABLE_QUESTIONNAIRE**

3.3.4 InstrumentTypes

public class **InstrumentTypes**
All valid types of an instrument.

Fields

ALL

public static final Set<String> **ALL**

CAPI

public static final String **CAPI**

CATI

public static final String **CATI**

CAWI

public static final String **CAWI**

PAPI

public static final String **PAPI**

3.4 eu.dzhw.fdz.metadatamanagement.ordermanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.ordermanagement.domain.Order*s.

3.4.1 Customer

public class **Customer**
Details of a customer who has ordered *Product*s.

Fields

email

private String **email**
Email address of the customer. Must be a valid email address and must not be empty.

name

private String **name**
Name of the customer as given in the shopping cart. Must not be empty.

3.4.2 Order

public class **Order** extends *AbstractRdcDomainObject*
Order (DTO) containing all relevant information of a `Customer` and her `Product`s.

Fields

client

private *OrderClient* **client**
The id of the client (one of @link `OrderClient`) who has last modified this order.

customer

private *Customer* **customer**
The `Customer` who has placed this order. Must not be null.

id

private String **id**
The id of an order. It is a number which is generated by a sequence (see `Counter`.

languageKey

private String **languageKey**
The key of the preferred language (either „de“ or „en“) of the `Customer`. Must not be empty.

products

private List<*Product*> **products**
List of data `Product`s the `Customer` want to order. Must not be empty.

state

private *OrderState* **state**
The current state of the order. One of *OrderState*.

3.4.3 OrderClient

public enum **OrderClient**
Enum holding possible clients with write access to the orders.

Enum Constants

DLP

public static final *OrderClient* **DLP**

MDM

public static final *OrderClient* **MDM**

3.4.4 OrderState

public enum **OrderState**
The states an *Order* can have.

Author René Reitmann

Enum Constants

CREATED

public static final *OrderState* **CREATED**

NOTIFIED

public static final *OrderState* **NOTIFIED**

3.4.5 OrderedStudy

public class **OrderedStudy**
Partial *eu.dzhw.fdz.metadatamanagement.studymanagement.domain.Study* which is part of a *Product*. It is a copy of the *eu.dzhw.fdz.metadatamanagement.studymanagement.domain.Study* attributes which is made when the *Customer* places the orders.

Fields

annotations

private *I18nString* **annotations**
The annotations of the *eu.dzhw.fdz.metadatamanagement.studymanagement.domain.Study*.

id

private String **id**
The id of the *eu.dzhw.fdz.metadatamanagement.studymanagement.domain.Study*. Must not be empty.

title

private *I18nString* **title**
The title of the *eu.dzhw.fdz.metadatamanagement.studymanagement.domain.Study*. Must not be empty neither in German nor in English.

3.4.6 Product

public class **Product**
Data Product which can be ordered by a customer.

Author René Reitmann

Fields

accessWay

private String **accessWay**
The access way to the *DataSet*s which the *Customer* wants to have.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* in which this product was generated. Must not be empty.

study

private *OrderedStudy* **study**
The (partial) *OrderedStudy* of this product. Must not be empty.

version

private String **version**
The version of the *DataSet*s which the *Customer* wants to have.

3.5 eu.dzhw.fdz.metadatamanagement.projectmanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.projectmanagement.domain.DataAcquisitionProject*s.

3.5.1 Configuration

public class **Configuration**
The project configuration describes which users are publishers or data providers for a project.

Fields

dataProviders

private List<String> **dataProviders**
User names having the role of a data provider for a project. Must contain at least one user name.

publishers

private List<String> **publishers**
User names having the role of a publisher for a project. Must contain at least one user name.

requirements

private *Requirements* **requirements**
Defines which object types are required before a project can be released.

3.5.2 DaraUpdateQueueItem

public class **DaraUpdateQueueItem** extends *AbstractRdcDomainObject*
Publishing metadata to dalra will be done asynchronously and repeated as long as the update queue item has not been processed successfully and has therefore been deleted.

Fields

id

private String **id**
The id of the update queue item. It is generated by the database.

projectId

private String **projectId**
The id of the *DataAcquisitionProject* which needs to be sent to dalra. Must not be empty and there must be at most one update queue item in the database for any project.

updateStartedAt

private LocalDateTime **updateStartedAt**
Timestamp at which the update has been started.

updateStartedBy

private String **updateStartedBy**
Id of the process who started the updated. Stored in order to avoid having multiple concurrent processes sending data to dalra.

3.5.3 DataAcquisitionProject

public class **DataAcquisitionProject** extends *AbstractRdcDomainObject*
The data acquisition project collects the metadata for the data products which are published by our RDC. One project can contain one *Study*, many *Survey*s, many *Instrument*s and *Question*s, and many *DataSet*s and *Variable*s. A project can be currently released (visible to public users) or not. When a publisher releases a project and its version is greater than or equal to 1.0.0 then the metadata is published to dalra.

Fields

configuration

private *Configuration* **configuration**
Contains the project configuration.

hasBeenReleasedBefore

private Boolean **hasBeenReleasedBefore**
Flag indicating whether this project has ever been released in its life. It is used to ensure that project cannot be deleted once they have been released.

id

private String **id**
The id of this project. Must not be empty and must only contain lower cased (english) letters and numbers. Must not contain more than 32 characters.

release

private *Release* **release**
A valid *Release* object. Null if the project is currently not released. The version of a *Release* must be a syntactically correct according to semver (major.minor.patch) and must not be decreased.

3.5.4 Release

public class **Release**
The release object contains the version and a timestamp of the current release.

Fields

date

private LocalDateTime **date**
The timestamp (in UTC) indicates when a publisher has released the *DataAcquisitionProject*. Must not be empty.

version

private String **version**
A valid semver version (major.minor.patch). Must not be empty and must not contain more than 32 characters. A version of a *DataAcquisitionProject* must not be decreased.

3.5.5 Requirements

public class **Requirements**
This configuration defines which object types have to be delivered before a project can be released.

Fields

isDataSetsRequired

private boolean **isDataSetsRequired**
Defines if data set data is required for a release.

isInstrumentsRequired

private boolean **isInstrumentsRequired**
Defines if instrument data is required for a release.

isQuestionsRequired

private boolean **isQuestionsRequired**
Defines if question data is required for a release.

isStudyRequired

private boolean **isStudyRequired**
Defines if study data is required for a release (this object type is mandatory and this setting is therefore always `true`.

isSurveysRequired

private boolean **isSurveysRequired**
Defines if survey data is required for a release.

isVariablesRequired

private boolean **isVariablesRequired**
Defines if variable data is required for a release.

3.6 eu.dzhw.fdz.metadatamanagement.questionmanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.questionmanagement.domain.Question*s.

3.6.1 ImageType

public enum **ImageType**
Enum representing supported types of question images.

Enum Constants

PNG

public static final *ImageType* **PNG**

3.6.2 Question

public class **Question** extends *AbstractRdcDomainObject*
A question is part of an *Instrument* which has been used in at least one *Survey*s. The responses to a question are stored in *Variable*s.

Fields

additionalQuestionText

private *I18nString* **additionalQuestionText**
Arbitrary additional question text which has been presented to the participant. Must not contain more than 1 MB characters.

annotations

private *I18nString* **annotations**
Arbitrary annotations to this question. Must not contain more than 2048 characters.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which this question belongs. The dataAcquisitionProjectId must not be empty.

id

private String **id**
The id of the question which uniquely identifies the question in this application. The id must not be empty and must be of the form que-{{dataAcquisitionProjectId}}-ins{{instrumentNumber}}-{{number}}$. The id must not contain more than 512 characters.

indexInInstrument

private Integer **indexInInstrument**
The index of the question in the *Instrument*. Used for sorting the questions.

instruction

private *I18nString* **instruction**
The instruction for the participant which tells how to give the answers to this question. Must not contain more than 1 MB characters.

instrumentId

private String **instrumentId**
The id of the *Instrument* to which this question belongs. Must not be empty.

instrumentNumber

private Integer **instrumentNumber**
The number of the *Instrument* to which this question belongs. Must not be empty.

introduction

private *I18nString* **introduction**
The introduction of this question which gives more context to the participant before asking the question. Must not contain more than 2048 characters.

number

private String **number**
The number of the question. Must not be empty and must be unique within the *Instrument*. Must contain only (german) alphanumeric characters and „_“,“-“ and „.“ and must not contain more than 32 characters.

questionText

private *I18nString* **questionText**
The question the *Survey*s participant was asked. It must be specified in at least one language and it must not contain more than 2048 characters.

studyId

private String **studyId**
The id of the *OrderedStudy* to which this question belongs. Must not be empty.

successorNumbers

private List<String> **successorNumbers**
List of numbers of the *Question*s which directly follow this question in the *Instrument*.

successors

private List<String> **successors**
List of ids of the *Question*s which directly follow this question in the *Instrument*.

technicalRepresentation

private *TechnicalRepresentation* **technicalRepresentation**
A *TechnicalRepresentation* of this question. This is optional and can be used to add the source code of the question which was used to generate it.

topic

private *I18nString* **topic**
The topic or section in the *Instrument* to which this question belongs. It must not contain more than 2048 characters.

type

private *I18nString* **type**
The type of the question. Must be one of QuestionTypes and must not be empty.

3.6.3 QuestionImageMetadata

public class **QuestionImageMetadata** extends *AbstractRdcDomainObject*
The metadata for one question images. One question image displays the question in one language with one given resolution.

Fields

containsAnnotations

private Boolean **containsAnnotations**
Flag indicating whether the image contains annotations which highlight parts that were only visible to specific participants. These annotations were not visible to the participants.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* of the *Question* to which this image belongs. Must not be empty.

fileName

private String **fileName**
The name of the images file. Must not be empty and must only contain (german) alphanumeric characters and „_“,“-“ and „.“.

imageType

private *ImageType* **imageType**
The type of this image. Must be one of *ImageType* and must not be empty.

indexInQuestion

private Integer **indexInQuestion**
The index in the *Question* of this image. Used for sorting the images of this *Question*. Must not be empty.

language

private String **language**
The language of the question text on this image. Must not be empty and must be a valid ISO 639 code.

questionId

private String **questionId**
The id of the *Question* to which this image belongs. Must not be empty.

resolution

private *Resolution* **resolution**
The resolution of the image.

3.6.4 QuestionTypes

public class **QuestionTypes**
All valid types of a *Question*.

Fields

ALL

public static final Set<*I18nString*> **ALL**

GRID

public static final *I18nString* **GRID**

ITEM_SET

public static final *I18nString* **ITEM_SET**

MULTIPLE_CHOICE

public static final *I18nString* **MULTIPLE_CHOICE**

OPEN

public static final *I18nString* **OPEN**

SINGLE_CHOICE

public static final *I18nString* **SINGLE_CHOICE**

3.6.5 TechnicalRepresentation

public class **TechnicalRepresentation**
The technical representation of a *Question* which was used to generate the question for instance in an online *Instrument*.

Fields

language

private String **language**
The technical language of the source of this representation. E.g. „qml“. Must not be empty and must not contain more than 32 characters.

source

private String **source**
The source code of the question. Must not be empty and must not contain more than 1 MB characters.

type

private String **type**
The type of the technical representation. E.g. „zofar“. Must not be empty and must not contain more than 32 characters.

3.7 eu.dzhw.fdz.metadatamanagement.relatedpublicationmanagement.domain

In the domain layer are all domain classes of the related publication.

author Daniel Katzberg

3.7.1 RelatedPublication

public class **RelatedPublication** extends *AbstractRdcDomainObject*
Domain Object for the Related Publications.

Author Daniel Katzberg

Fields

abstractSource

private *I18nString* **abstractSource**

authors

private String **authors**

dataSetIds

private List<String> **dataSetIds**

doi

private String **doi**

id

private String **id**

instrumentIds

private List<String> **instrumentIds**

language

private String **language**

publicationAbstract

private String **publicationAbstract**

questionIds

private List<String> **questionIds**

sourceLink

private String **sourceLink**

sourceReference

private String **sourceReference**

studyIds

private List<String> **studyIds**

studySerieses

private List<*I18nString*> **studySerieses**

surveyIds

private List<String> **surveyIds**

title

private String **title**

variableIds

private List<String> **variableIds**

year

private Integer **year**

3.8 eu.dzhw.fdz.metadatamanagement.studymanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.studymanagement.domain.Study*s.

3.8.1 DataAvailabilities

public class **DataAvailabilities**
The data's availability of a *Study* can be in one of these states.

Fields

ALL

public static final Set<*I18nString*> **ALL**

AVAILABLE

public static final *I18nString* **AVAILABLE**

IN_PREPARATION

public static final *I18nString* **IN_PREPARATION**

NOT_AVAILABLE

public static final *I18nString* **NOT_AVAILABLE**

3.8.2 Study

public class **Study** extends *AbstractRdcDomainObject* implements StudySubDocumentProjection
A study contains all metadata of a *DataAcquisitionProject*. It will get a DOI (Digital Object Identifier) when the *DataAcquisitionProject* is released.

Fields

annotations

private *I18nString* **annotations**
Arbitrary additional text for this instrument. Must not contain more than 2048 characters.

authors

private List<*Person*> **authors**
List of *Person*s which have performed this study. Must not be empty.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which this study belongs. The dataAcquisitionProjectId must not be empty.

dataAvailability

private *I18nString* **dataAvailability**
The current state of the data's availability. Must be one of *DataAvailabilities* and must not be empty.

description

private *I18nString* **description**
A description of the study. It must be specified in at least one language and it must not contain more than 2048 characters.

id

private String **id**
The id of the study which uniquely identifies the study in this application. The id must not be empty and must be of the form stu-{{dataAcquisitionProjectId}}$. The id must not contain more than 512 characters.

institution

private *I18nString* **institution**
The name of the institution which has performed this study. It must be specified in German and English and it must not contain more than 512 characters.

sponsor

private *I18nString* **sponsor**
The name of the sponsor who which has sponsored this study. It must be specified in German and English and it must not contain more than 512 characters.

studySeries

private *I18nString* **studySeries**
The name of the series of studies to which this study belongs.. If specified it must be specified in German and English. It must not contain more than 512 characters and must not contain „“.

surveyDesign

private *I18nString* **surveyDesign**
The survey design of this *Study*. Must be one of *SurveyDesigns* and must not be empty.

title

private *I18nString* **title**
The title of the study. It must be specified in German and English and it must not contain more than 2048 characters.

3.8.3 StudyAttachmentMetadata

public class **StudyAttachmentMetadata** extends *AbstractRdcDomainObject*
Metadata which will be stored with each attachment of a *Study*.

Fields

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which the *Study* of this attachment belongs. Must not be empty.

description

private *I18nString* **description**
A description for this attachment. It must be specified in at least one language and it must not contain more than 512 characters.

fileName

private String **fileName**
The filename of the attachment. Must not be empty and must contain only (german) alphanumeric characters and „_“ and „-“ and „.“.

id

private String **id**
The id of the attachment. Holds the complete path which can be used to download the file.

indexInStudy

private Integer **indexInStudy**
The index in the *Study* of this attachment. Used for sorting the attachments of this *Study*. Must not be empty.

language

private String **language**
The language of the attachments content. Must not be empty and must be specified as ISO 639 language code.

studyId

private String **studyId**
The id of the *Study* to which this attachment belongs. Must not be empty.

title

private String **title**
An optional title of this attachment in the attachments' language. It must not contain more than 2048 characters.

type

private *I18nString* **type**
The type of the attachment. Must be one of *StudyAttachmentTypes* and must not be empty.

3.8.4 StudyAttachmentTypes

public class **StudyAttachmentTypes**
All valid types of a *StudyAttachmentMetadata*.

Fields

ALL

public static final Set<*I18nString*> **ALL**

METHOD_REPORT

public static final *I18nString* **METHOD_REPORT**

OTHER

public static final *I18nString* **OTHER**

3.8.5 SurveyDataTypes

public class **SurveyDataTypes** extends *DataTypes*
List of types of data, which a *Study* can consist of. It will be computed from the *Survey*s of a *Study*.

Fields

ALL

public static final Set<*I18nString*> **ALL**

MIXED_METHODS

public static final *I18nString* **MIXED_METHODS**

3.8.6 SurveyDesigns

public class **SurveyDesigns**
List of currently supported survey designs.

Fields

ALL

public static final Set<*I18nString*> **ALL**

CROSS_SECTION

public static final *I18nString* **CROSS_SECTION**

PANEL

public static final *I18nString* **PANEL**

3.9 eu.dzhw.fdz.metadatamanagement.surveymanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.surveymanagement.domain.Survey*s.

3.9.1 DataTypes

public class **DataTypes**
Types of data, which a *Survey* can produce.

Fields

ALL

public static final Set<*I18nString*> **ALL**

QUALITATIVE_DATA

public static final *I18nString* **QUALITATIVE_DATA**

QUANTITATIVE_DATA

public static final *I18nString* **QUANTITATIVE_DATA**

3.9.2 Population

public class **Population**
Details of the population of a *Survey*.

Fields

description

private *I18nString* **description**
A description of the population. It must be specified in at least one language and it must not contain more than 2048 characters.

title

private *I18nString* **title**
A short title for the population. It must be specified in at least one language and it must not contain more than 512 characters.

3.9.3 Survey

public class **Survey** extends *AbstractRdcDomainObject*
A survey is conducted to examine a population on the basis of a sample. The resulting *DataSet*s can be used to make statements about the population.

Fields

annotations

private *I18nString* **annotations**
Arbitrary additional text for this survey. Must not contain more than 2048 characters.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which this survey belongs. The dataAcquisitionProjectId must not be empty.

dataType

private *I18nString* **dataType**
The type of data which the survey produced. Must be one of *DataTypes* and must not be empty.

fieldPeriod

private *Period* **fieldPeriod**
The period during which the survey has been conducted or is expected to be conducted. Must not be empty.

grossSampleSize

private Integer **grossSampleSize**
The gross sample size represents the number of participants which have been invited to take part in the *Survey*. Must not be negative.

id

private String **id**
The id of the survey which uniquely identifies the survey in this application. The id must not be empty and must be of the form sur-{{dataAcquisitionProjectId}}-sy{{number}}$. The id must not contain more than 512 characters.

number

private Integer **number**
The number of the instrument. Must not be empty and must be unique within the *DataAcquisitionProject*.

population

private *Population* **population**
Details about the *Population*. Must not be empty.

responseRate

private Double **responseRate**
The response rate is the quotient of the gross sample size and the sample size. Must be between 0 and 100.

sample

private *I18nString* **sample**
The sampling method is the procedure for selecting sample members from a population. It must be specified in at least one language and it must not contain more than 2048 characters.

sampleSize

private Integer **sampleSize**
The sample size is the number of participant which took part in the survey. Must not be empty and must not be negative.

studyId

private String **studyId**
The id of the *Study* to which this survey belongs. Must not be empty.

surveyMethod

private *I18nString* **surveyMethod**
The survey method briefly describes how the data were collected. It must be specified in German and English and it must not contain more than 2048 characters.

title

private *I18nString* **title**
The title of the instrument. It must be specified in German and English and it must not contain more than 2048 characters.

wave

private Integer **wave**
Number of the wave which this *Survey* represents. Will be ignored if the *Study* is not organized in waves. Must not be empty and must be greater than or equal to 1.

3.9.4 SurveyAttachmentMetadata

public class **SurveyAttachmentMetadata** extends *AbstractRdcDomainObject*
Metadata which will be stored with each attachment of a *Survey*.

Fields

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which the *Survey* of this attachment belongs. Must not be empty.

description

private *I18nString* **description**
A description for this attachment. It must be specified in at least one language and it must not contain more than 512 characters.

fileName

private String **fileName**
The filename of the attachment. Must not be empty and must contain only (german) alphanumeric characters and „_“ and „-“ and „.“.

id

private String **id**
The id of the attachment. Holds the complete path which can be used to download the file.

indexInSurvey

private Integer **indexInSurvey**
The index in the *Survey* of this attachment. Used for sorting the attachments of this *Survey*. Must not be empty.

language

private String **language**
The language of the attachments content. Must not be empty and must be specified as ISO 639 language code.

surveyId

private String **surveyId**
The id of the *Survey* to which this attachment belongs. Must not be empty.

surveyNumber

private Integer **surveyNumber**
The number of the *Survey* to which this attachment belongs. Must not be empty.

title

private String **title**
A title of this attachment in the attachments‘ language. It must not contain more than 2048 characters.

3.9.5 SurveyResponseRateImageMetadata

public class **SurveyResponseRateImageMetadata** extends *AbstractRdcDomainObject*
Metadata which will be stored with each response rate image of a *Survey*.

Fields

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which the *Survey* of this response rate image belongs. Must not be empty.

fileName

private String **fileName**
The filename of the image. Must not be empty and must contain only (german) alphanumeric characters and „_“ and „-“ and „.“.

id

private String **id**
The id of the response rate image. Holds the complete path which can be used to download the file.

language

private String **language**
The language used in the response rate image. Must be either „de“ or „en“.

surveyId

private String **surveyId**
The id of the *Survey* to which this response rate image belongs. Must not be empty.

surveyNumber

private Integer **surveyNumber**
The number of the *Survey* to which this response rate image belongs. Must not be empty.

3.10 eu.dzhw.fdz.metadatamanagement.variablemanagement.domain

Domain objects describing *eu.dzhw.fdz.metadatamanagement.variablemanagement.domain.Variable*s.

3.10.1 AccessWays

public class **AccessWays**
An access way of a *Variable* or a *DataSet* indicates how the data user will be able to work with the data.

Fields

ALL

public static final Set<String> **ALL**

DOWNLOAD_CUF

public static final String **DOWNLOAD_CUF**

DOWNLOAD_SUF

public static final String **DOWNLOAD_SUF**

NOT_ACCESSIBLE

public static final String **NOT_ACCESSIBLE**

ONSITE_SUF

public static final String **ONSITE_SUF**

REMOTE_DESKTOP

public static final String **REMOTE_DESKTOP**

3.10.2 DataTypes

public class **DataTypes**
The technical type which the *ValidResponse*s have.

Fields

ALL

public static final Set<*I18nString*> **ALL**

DATE

public static final *I18nString* **DATE**

NUMERIC

public static final *I18nString* **NUMERIC**

STRING

public static final *I18nString* **STRING**

3.10.3 Distribution

public class **Distribution**

A distribution contains the descriptives of a *Variable* meaning its *ValidResponse*s, *Missing*s and *Statistics*.

Fields

maxNumberOfDecimalPlaces

private Integer **maxNumberOfDecimalPlaces**

Integer used for rounding the values of this *Variable* when displaying it. It is computed during the import of the *Variable* by finding the maximum number of decimal places in the list of *ValidResponse*s.

missings

private List<*Missing*> **missings**

List of *Missing*s of this *Variable*. Must not contain more than 7000 entries and the code of the *Missing*s must be unique.

statistics

private *Statistics* **statistics**

Descriptive metrics of this *Variable*.

totalAbsoluteFrequency

private Integer **totalAbsoluteFrequency**

The total absolute number of *ValidResponse*s and *Missing*s. Must not be empty.

totalValidAbsoluteFrequency

private Integer **totalValidAbsoluteFrequency**

The total absolute number of *ValidResponse*s. Must not be empty.

totalValidRelativeFrequency

private Double **totalValidRelativeFrequency**

The quotient from totalValidAbsoluteFrequency and totalAbsoluteFrequency. Must not be empty.

validResponses

private List<*ValidResponse*> **validResponses**
List of *ValidResponse*s of this variable. Must not contain more than 7000 entries and the value of the *ValidResponse*s must be unique.

3.10.4 FilterDetails

public class **FilterDetails**
Filter details of a *Variable* describe the condition which must have evaluated to true before a participant was asked a *Question* resulting in this *Variable*. All participants for which the conditions evaluates to false will have a *Missing* in this *Variable*.

Fields

description

private *I18nString* **description**
A description of this filter condition. Must not contain more than 2048 characters

expression

private String **expression**
A technical expression describing the condition which must have evaluated to true. The expression is given in the expressionLanguage. Must not be empty and must not contain more than 2048 characters.

expressionLanguage

private String **expressionLanguage**
The name of the language in which the expression was given. Must not be empty and must be one of *FilterExpressionLanguages*.

3.10.5 FilterExpressionLanguages

public class **FilterExpressionLanguages**
All supported expression languages for *FilterDetails*.

Fields

ALL

public static final Set<String> **ALL**

SPEL

public static final String **SPEL**

STATA

public static final String **STATA**

3.10.6 GenerationDetails

public class **GenerationDetails**
Generation details describe how a *Variable* was generated from one or more input *Variable*s.

Fields

description

private *I18nString* **description**
A description of this generation rule. Must not contain more than 2048 characters

rule

private String **rule**
The computation rule in the ruleExpressionLanguage which was used to generate this *Variable*. Must not contain more than 1 MB characters.

ruleExpressionLanguage

private String **ruleExpressionLanguage**
The language which was used to describe this rule. Must be one of *RuleExpressionLanguages*.

3.10.7 Missing

public class **Missing**
A missing or missing value is a value in a *Variable* which represents a reason why no observation (*ValidResponse*) has been stored. It also contains its frequency.

Fields

absoluteFrequency

private Integer **absoluteFrequency**
The absolute number of occurrences of this missing. Must not be empty.

code

private String **code**
A (unique in this *Variable*) code for this missing. Must not be empty.

label

private *I18nString* **label**
A label describing this missing. Must not contain more than 512 characters.

relativeFrequency

private Double **relativeFrequency**
The quotient from absoluteFrequency and *Distribution*.totalAbsoluteFrequency. Must not be empty.

3.10.8 RelatedQuestion

public class **RelatedQuestion**
A related question is a *Question* which has been asked to generate the values of a *Variable*. It contains the ids of the *Instrument* and the *Question* as well as all Strings of the *Question* which are related to this *Variable*.

Fields

instrumentId

private String **instrumentId**
The id of the *Instrument* of this *Question*. Must not be empty.

instrumentNumber

private String **instrumentNumber**
The number of the *Instrument* of this *Question*. Must not be empty.

questionId

private String **questionId**
The id of the corresponding *Question*. Must not be empty.

questionNumber

private String **questionNumber**
The number of the corresponding *Question*. Must not be empty.

relatedQuestionStrings

private *I18nString* **relatedQuestionStrings**
All Strings (concatenated) of this *Question* which „belong“ to this *Variable*. These Strings typically overlap with String from other *Variable*s of the same *Question*.

3.10.9 RuleExpressionLanguages

public class **RuleExpressionLanguages**
All supported expression languages for *GenerationDetails*.

Fields

ALL

public static final Set<String> **ALL**

R

public static final String **R**

STATA

public static final String **STATA**

3.10.10 ScaleLevels

public class **ScaleLevels**
The scale level (or level of measurement) classifies the nature of information within the values assigned to a *Variable* (*ValidResponse*s). It determines which mathematical operations can be performed with the values.

Fields

ALL

public static final Set<*I18nString*> **ALL**

INTERVAL

public static final *I18nString* **INTERVAL**

NOMINAL

public static final *I18nString* **NOMINAL**

ORDINAL

public static final *I18nString* **ORDINAL**

RATIO

public static final *I18nString* **RATIO**

3.10.11 Statistics

public class **Statistics**
Descriptive metrics of this *Variable*.

Fields

deviance

private Double **deviance**
See Deviance (Wikipedia).

firstQuartile

private String **firstQuartile**
Splits off the lowest 25% of the values (*ValidResponse*s) of this *Variable* from the highest 75%. Must not contain more than 32 characters.

highWhisker

private Double **highWhisker**
The highest value still within 1.5 IQR of the third quartile.

kurtosis

private Double **kurtosis**
See Kurtosis (Wikipedia).

lowWhisker

private Double **lowWhisker**
The lowest value still within 1.5 IQR of the first quartile.

maximum

private String **maximum**
The maximum of the values (*ValidResponse*s) of this *Variable*. Must not contain more than 32 characters.

meanDeviation

private Double **meanDeviation**
See Mean Absolute Deviation (Wikipedia).

meanValue

private Double **meanValue**
The arithmetic mean of the values (*ValidResponse*s) of this *Variable*.

median

private String **median**
The median is the value separating the higher half from the lower half of the values (*ValidResponse*s) of this *Variable*. Must not contain more than 32 characters.

minimum

private String **minimum**
The minimum of the values (*ValidResponse*s) of this *Variable*. Must not contain more than 32 characters.

mode

private String **mode**
The mode is the value (*ValidResponse*) that appears most often.

skewness

private Double **skewness**
See Skewness (Wikipedia).

standardDeviation

private Double **standardDeviation**
Measure that is used to quantify the amount of variation of the values (*ValidResponse*s) of this *Variable*.

thirdQuartile

private String **thirdQuartile**
Splits off the highest 25% of the values (*ValidResponse*s) of this *Variable* from the lowest 75%. Must not contain more than 32 characters.

3.10.12 StorageTypes

public class **StorageTypes**
All supported storage types of *Variable*s.

Fields

ALL

public static final Set<String> **ALL**

ANY

public static final String **ANY**

BUILTIN

public static final String **BUILTIN**

BYTECODE

public static final String **BYTECODE**

CHAR

public static final String **CHAR**

CHARACTER

public static final String **CHARACTER**

CLOSURE

public static final String **CLOSURE**

COMPLEX

public static final String **COMPLEX**

DOTDOTDOT

public static final String **DOTDOTDOT**

DOUBLE

public static final String **DOUBLE**

ENVIRONMENT

public static final String **ENVIRONMENT**

EXPRESSION

public static final String **EXPRESSION**

EXTERNALPTR

public static final String **EXTERNALPTR**

INTEGER

public static final String **INTEGER**

LANGUAGE

public static final String **LANGUAGE**

LIST

public static final String **LIST**

LOGICAL

public static final String **LOGICAL**

NULL

public static final String **NULL**

PAIRLIST

public static final String **PAIRLIST**

PROMISE

public static final String **PROMISE**

RAW

public static final String **RAW**

S4

public static final String **S4**

SPECIAL

public static final String **SPECIAL**

SYMBOL

public static final String **SYMBOL**

WEAKREF

public static final String **WEAKREF**

3.10.13 ValidResponse

public class **ValidResponse**
A valid response represents one observation of a *Variable* and its frequency.

Fields

absoluteFrequency

private Integer **absoluteFrequency**
The absolute number of occurrences of this observation. Must not be empty.

label

private *I18nString* **label**
An optional label for the value of this observation.

relativeFrequency

private Double **relativeFrequency**
The quotient from absoluteFrequency and *Distribution*.totalAbsoluteFrequency. Must not be empty.

validRelativeFrequency

private Double **validRelativeFrequency**
The quotient from absoluteFrequency and *Distribution*.totalValidAbsoluteFrequency. Must not be empty.

value

private String **value**
The value which has been observed (e.g. was responded by the participant). Must not be empty and must not contain more than 256 characters.

3.10.14 Variable

public class **Variable** extends *AbstractRdcDomainObject*
A variable contains the results from at least one *Survey*. These results can be the responses from participants of an online survey, hence a variable can result from *RelatedQuestion*s. A variable is part of exactly one *DataSet*.

Fields

accessWays

private List<String> **accessWays**
The access way of this variable. Depends on the sensitivity of the data and describes how the data user will be able to work with the data. Must not be empty and be one of *AccessWays*.

annotations

private *I18nString* **annotations**
Arbitrary additional text for this variable. Must not contain more than 2048 characters.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
The id of the *DataAcquisitionProject* to which this variable belongs. The dataAcquisitionProjectId must not be empty.

dataSetId

private String **dataSetId**
The id of the *DataSet* to which this variable belongs. Must not be empty.

dataSetNumber

private Integer **dataSetNumber**
The number of the *DataSet* to which this variable belongs. Must not be empty.

dataType

private *I18nString* **dataType**
The technical type which the *ValidResponse*s have. Must be one of *DataTypes* and must not be empty.

derivedVariablesIdentifier

private String **derivedVariablesIdentifier**

Identifier used to group variables within this *DataSet* which have been derived from each other. For instance one variable might be an aggregated version of the other. Must be of the form {{dataAcquisitionProjectId}}-ds{{dataSetNumber}}-{{string}}$. Must not contain more than 512 characters and must contain only (german) alphanumeric characters and „_“ and „-„.

distribution

private *Distribution* **distribution**

The *Distribution* contains the descriptives of this variable meaning *ValidResponse*s, *Missing*s and *Statistics*.

doNotDisplayThousandsSeparator

private Boolean **doNotDisplayThousandsSeparator**

Flag indicating whether the *ValidResponse*s should be displayed with a thousands separator or not. For instance years (1970) are numeric but should not be displayed with a thousands separator. Default value is false indicating that the *ValidResponse*s are displayed with thousands separator.

filterDetails

private *FilterDetails* **filterDetails**

FilterDetails of a variable describe the condition which must have evaluated to true before a participant was asked a *Question* resulting in this variable.

generationDetails

private *GenerationDetails* **generationDetails**

GenerationDetails describe how this variable was generated from one or more input variables.

id

private String **id**

The id of the variable which uniquely identifies the variable in this application. The id must not be empty and must be of the form var-{{dataAcquisitionProjectId}}-ds{{dataSetNumber}}-{{name}}$. The id must not contain more than 512 characters.

indexInDataSet

private Integer **indexInDataSet**

The index in the *DataSet* of this variable. Used for sorting the variables of this *DataSet* and for displaying successors and predecessors of this variable. Must not be empty and the successor of this variable must have indexInDataSet incremented by one.

label

private *I18nString* **label**
The label of the variable should describe its content. It must be specified in at least one language and it must not contain more than 512 characters.

name

private String **name**
The name of the variable as it is used in the *DataSet*. It must not be empty and must be unique in the *DataSet*. It must contain only alphanumeric (english) characters and „_“. The first character must not be a number. It must not contain more than 32 characters.

panelIdentifier

private String **panelIdentifier**
Identifier used to group variables within this *DataSet* which measure the same across multiple waves. Must be of the form {{dataAcquisitionProjectId}}-ds{{dataSetNumber}}-{{string}}$. Must not contain more than 512 characters and must contain only (german) alphanumeric characters and „_“ and „-„.

relatedQuestions

private List<*RelatedQuestion*> **relatedQuestions**
List of *RelatedQuestion*s which have been asked to generate the values of this variable.

relatedVariables

private List<String> **relatedVariables**
List of ids of variables which are „related“ to this variable. The type of relation is arbitrary.

scaleLevel

private *I18nString* **scaleLevel**
The scale level (or level of measurement) classifies the nature of information within the values assigned to this variable (*ValidResponse*s). It determines which mathematical operations can be performed with the values. It must be one of *ScaleLevels* and must not be empty. If the data type of this variable is *DataTypes.DATE* then the ScaleLevel must be *ScaleLevels.ORDINAL*.

storageType

private String **storageType**
Associated with each data type is a storage type. For instance numerics can be stored as integer or double. Must be one of *StorageTypes* and must not be empty.

studyId

private String **studyId**
Id of the *Study* to which this variable belongs.

surveyIds

private List<String> **surveyIds**
List of ids of *Survey*s which have been conducted to create this variable. Must not be empty.

surveyNumbers

private List<Integer> **surveyNumbers**
List of numbers of *Survey*s which have been conducted to create this variable. Must not be empty.

KAPITEL 4

Indizes und Tabellen

- genindex

Stichwortverzeichnis

A

absoluteFrequency (Java field), 73, 80
AbstractRdcDomainObject (Java class), 35
abstractSource (Java field), 58
accessWay (Java field), 42, 49
AccessWays (Java class), 69
accessWays (Java field), 81
additionalQuestionText (Java field), 53
ALL (Java field), 41, 45, 46, 57, 60, 63, 64, 70, 72, 75, 78
annotations (Java field), 38, 43, 49, 53, 60, 65, 81
ANY (Java field), 78
authors (Java field), 58, 61
AVAILABLE (Java field), 60

B

BUILTIN (Java field), 78
BYTECODE (Java field), 78

C

CAPI (Java field), 46
CATI (Java field), 46
CAWI (Java field), 46
CHAR (Java field), 78
CHARACTER (Java field), 78
citationHint (Java field), 42
client (Java field), 47
CLOSURE (Java field), 78
code (Java field), 73
COMPLEX (Java field), 78
Configuration (Java class), 50
configuration (Java field), 51
containsAnnotations (Java field), 56
Counter (Java class), 36
CREATED (Java field), 48
createdBy (Java field), 35
createdDate (Java field), 35
CROSS_SECTION (Java field), 64
Customer (Java class), 46
customer (Java field), 47

D

DaraUpdateQueueItem (Java class), 50
data provider, 3, 5
DataAcquisitionProject (Java class), 51
dataAcquisitionProjectId (Java field), 38, 40, 43, 44, 49, 54, 56, 61, 62, 65, 67, 69, 81
DataAvailabilities (Java class), 60
dataAvailability (Java field), 61
dataProviders (Java field), 50
DataSet (Java class), 38
DataSetAttachmentMetadata (Java class), 39
dataSetId (Java field), 40, 81
dataSetIds (Java field), 58
dataSetNumber (Java field), 40, 81
DataSetTypes (Java class), 41
dataType (Java field), 66, 81
DataTypes (Java class), 64, 70
DATE (Java field), 70
date (Java field), 52
Datenaufnahme, 3, 5
de (Java field), 36
derivedVariablesIdentifier (Java field), 82
description (Java field), 38, 40, 42–44, 61, 62, 65, 67, 72, 73
deviance (Java field), 76
Distribution (Java class), 71
distribution (Java field), 82
DLP (Java field), 48
doi (Java field), 58
doNotDisplayThousandsSeparator (Java field), 82
DOTDOTDOT (Java field), 78
DOUBLE (Java field), 78
DOWNLOAD_CUF (Java field), 70
DOWNLOAD_SUF (Java field), 70

E

email (Java field), 47
en (Java field), 36
end (Java field), 37

ENVIRONMENT (Java field), 79
EPISODE_RECORD (Java field), 41
eu.dzhw.fdz.metadatamanagement.common.domain (package), 35
eu.dzhw.fdz.metadatamanagement.datasetmanagement.domain (package), 38
eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain (package), 42
eu.dzhw.fdz.metadatamanagement.ordermanagement.domain (package), 46
eu.dzhw.fdz.metadatamanagement.projectmanagement.domain (package), 50
eu.dzhw.fdz.metadatamanagement.questionmanagement.domain (package), 53
eu.dzhw.fdz.metadatamanagement.relatedpublicationmanagement.domain (package), 58
eu.dzhw.fdz.metadatamanagement.studymanagement.domain (package), 60
eu.dzhw.fdz.metadatamanagement.surveymanagement.domain (package), 64
eu.dzhw.fdz.metadatamanagement.variablemanagement.domain (package), 69
EXPRESSION (Java field), 79
expression (Java field), 72
expressionLanguage (Java field), 72
EXTERNALPTR (Java field), 79

F

fieldPeriod (Java field), 66
fileName (Java field), 40, 44, 56, 62, 68, 69
FilterDetails (Java class), 72
filterDetails (Java field), 82
FilterExpressionLanguages (Java class), 72
firstName (Java field), 37
firstQuartile (Java field), 76
Format (Java class), 41
format (Java field), 38

G

GenerationDetails (Java class), 73
generationDetails (Java field), 82
GRID (Java field), 57
grossSampleSize (Java field), 66

H

hasBeenReleasedBefore (Java field), 51
heightY (Java field), 38
highWhisker (Java field), 76

I

I18nString (Java class), 36
id (Java field), 36, 39, 40, 43, 44, 47, 49–51, 54, 58, 61, 62, 66, 68, 69, 82
ImageType (Java enum), 53
imageType (Java field), 56
IN_PREPARATION (Java field), 60
indexInDataSet (Java field), 40, 82
indexInInstrument (Java field), 45, 54
indexInQuestion (Java field), 56
indexInStudy (Java field), 62
indexInSurvey (Java field), 68
institution (Java field), 61
instruction (Java field), 54
Instrument (Java class), 42
InstrumentAttachmentMetadata (Java class), 44
InstrumentAttachmentTypes (Java class), 45
instrumentId (Java field), 45, 54, 74
instrumentIds (Java field), 59
instrumentNumber (Java field), 45, 54, 74
InstrumentTypes (Java class), 46
INTEGER (Java field), 79
INTERVAL (Java field), 75
introduction (Java field), 54
isDataSetsRequired (Java field), 52
isInstrumentsRequired (Java field), 52
isQuestionsRequired (Java field), 52
isStudyRequired (Java field), 52
isSurveysRequired (Java field), 53
isVariablesRequired (Java field), 53
ITEM_SET (Java field), 57

K

kurtosis (Java field), 76

L

label (Java field), 74, 80, 83
LANGUAGE (Java field), 79
language (Java field), 40, 45, 56, 57, 59, 63, 68, 69
languageKey (Java field), 47
lastModifiedBy (Java field), 35
lastModifiedDate (Java field), 36
lastName (Java field), 37
LIST (Java field), 79
LOGICAL (Java field), 79
LONG (Java field), 41
lowWhisker (Java field), 76

M

maximum (Java field), 76
maxNumberOfDecimalPlaces (Java field), 71
MDM (Java field), 48
meanDeviation (Java field), 77
meanValue (Java field), 77
median (Java field), 77
METHOD_REPORT (Java field), 63
middleName (Java field), 37
minimum (Java field), 77

Missing (Java class), 73
missings (Java field), 71
MIXED_METHODS (Java field), 64
mode (Java field), 77
MULTIPLE_CHOICE (Java field), 57

N

name (Java field), 42, 47, 83
NOMINAL (Java field), 75
NOT_ACCESSIBLE (Java field), 70
NOT_AVAILABLE (Java field), 60
NOTIFIED (Java field), 48
NULL (Java field), 79
number (Java field), 39, 43, 54, 66
numberOfObservations (Java field), 42
NUMERIC (Java field), 70

O

ONSITE_SUF (Java field), 70
OPEN (Java field), 57
Order (Java class), 47
OrderClient (Java enum), 48
OrderedStudy (Java class), 48
OrderState (Java enum), 48
ORDINAL (Java field), 75
OTHER (Java field), 45, 63

P

PAIRLIST (Java field), 79
PANEL (Java field), 64
panelIdentifier (Java field), 83
PAPI (Java field), 46
Period (Java class), 37
Person (Java class), 37
PERSONAL_RECORD (Java field), 41
PNG (Java field), 53
Population (Java class), 65
population (Java field), 66
Product (Java class), 49
products (Java field), 47
projectId (Java field), 50
Projekt anlegen, 5
PROMISE (Java field), 79
publicationAbstract (Java field), 59
publishers (Java field), 50

Q

QUALITATIVE_DATA (Java field), 65
QUANTITATIVE_DATA (Java field), 65
Question (Java class), 53
QUESTION_FLOW (Java field), 46
questionId (Java field), 56, 74
questionIds (Java field), 59
QuestionImageMetadata (Java class), 55
QUESTIONNAIRE (Java field), 45
questionNumber (Java field), 74
questionText (Java field), 55
QuestionTypes (Java class), 57

R

R (Java field), 75
RATIO (Java field), 76
RAW (Java field), 79
RelatedPublication (Java class), 58
RelatedQuestion (Java class), 74
relatedQuestions (Java field), 83
relatedQuestionStrings (Java field), 74
relatedVariables (Java field), 83
relativeFrequency (Java field), 74, 80
Release (Java class), 52
release (Java field), 51
REMOTE_DESKTOP (Java field), 70
Requirements (Java class), 52
requirements (Java field), 50
Resolution (Java class), 37
resolution (Java field), 56
responseRate (Java field), 66
rule (Java field), 73
ruleExpressionLanguage (Java field), 73
RuleExpressionLanguages (Java class), 75

S

S4 (Java field), 80
sample (Java field), 66
sampleSize (Java field), 67
scaleLevel (Java field), 83
ScaleLevels (Java class), 75
seq (Java field), 36
SINGLE_CHOICE (Java field), 57
skewness (Java field), 77
source (Java field), 58
sourceLink (Java field), 59
sourceReference (Java field), 59
SPECIAL (Java field), 80
SPEL (Java field), 72
sponsor (Java field), 61
standardDeviation (Java field), 77
start (Java field), 37
STATA (Java field), 73, 75
state (Java field), 48
Statistics (Java class), 76
statistics (Java field), 71
storageType (Java field), 83
StorageTypes (Java class), 77
STRING (Java field), 71
Study (Java class), 60
study (Java field), 49

StudyAttachmentMetadata (Java class), 62
StudyAttachmentTypes (Java class), 63
studyId (Java field), 39, 43, 55, 63, 67, 84
studyIds (Java field), 59
studySeries (Java field), 61
studySerieses (Java field), 59
SubDataSet (Java class), 42
subDataSets (Java field), 39
subtitle (Java field), 43
successorNumbers (Java field), 55
successors (Java field), 55
Survey (Java class), 65
SurveyAttachmentMetadata (Java class), 67
SurveyDataTypes (Java class), 63
surveyDesign (Java field), 62
SurveyDesigns (Java class), 64
surveyId (Java field), 68, 69
surveyIds (Java field), 39, 43, 59, 84
surveyMethod (Java field), 67
surveyNumber (Java field), 68, 69
surveyNumbers (Java field), 39, 44, 84
SurveyResponseRateImageMetadata (Java class), 68
SYMBOL (Java field), 80

T

TechnicalRepresentation (Java class), 57
technicalRepresentation (Java field), 55
thirdQuartile (Java field), 77
title (Java field), 41, 44, 49, 59, 62, 63, 65, 67, 68
topic (Java field), 55
totalAbsoluteFrequency (Java field), 71
totalValidAbsoluteFrequency (Java field), 71
totalValidRelativeFrequency (Java field), 71
type (Java field), 39, 44, 45, 55, 58, 63

U

updateStartedAt (Java field), 51
updateStartedBy (Java field), 51

V

validRelativeFrequency (Java field), 80
ValidResponse (Java class), 80
validResponses (Java field), 72
value (Java field), 81
Variable (Java class), 81
VARIABLE_QUESTIONNAIRE (Java field), 46
variableIds (Java field), 59
version (Java field), 36, 49, 52

W

wave (Java field), 67
WEAKREF (Java field), 80
WIDE (Java field), 41
widthX (Java field), 38

Y

year (Java field), 60