MetaDataManagement Documentation

Release 1.0.106

Robert Birkelbach, Anne Gärtner, René Reitmann

23.11.2020

Inhaltsverzeichnis:

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

2 Metadatenabgabe (DatengeberInnen) **3**
2.1 Allgemeines . . . 3
2.2 Vorbereitende Schritte . . . 4
2.3 Notwendige Schnittstellen und Dateien . . . 10
2.4 Die Abgabe von Metadaten für die einzelnen Ebenen . . . 10
2.5 Projekte freigeben . . . 33

3 FDZ-MitarbeiterIn (Publisher, Developer) **35**
3.1 ID-Vergabe . . . 35
3.2 Neues Projekt anlegen . . . 36
3.3 Bilderfassung aus PDF Dateien . . . 38
3.4 Bilderfassung aus RagTime . . . 46
3.5 Excel Makros verwenden . . . 51
3.6 Editing Macros Excel . . . 54
3.7 Anpassungen LateX Template DSR . . . 56
3.8 Datensatzreport (Lektorat) . . . 57
3.9 Fragen (questions) . . . 60
3.10 Variablen (variables) . . . 61
3.11 Erstellung der Variable-JSON Dateien . . . 64
3.12 Variables (Zofar) . . . 66
3.13 Related Publications . . . 67
3.14 Datensatzreport erzeugen . . . 67
3.15 Prüfung der Jsons nach Umstellung der Generierungsskripte . . . 67
3.16 Skalenniveau prüfen . . . 68
3.17 Skalenniveau Diese Informationen dienen dazu die verschiedenen . . . 68
3.18 Testen von MDM-Issues . . . 69
3.19 Datentypen . . . 70
3.20 Die korrekte Anordnung der Dateien im Ordner . . . 71
3.21 Übersicht über verschiedene R-Helfer-Skripte . . . 71
3.22 Verantwortlichkeiten . . . 72
3.23 Projektrelease . . . 72

3.24 Domänenmodell 72
3.25 Javadoc 72

4 Indizes und Tabellen **149**

Stichwortverzeichnis **151**

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

StudentIn, ForscherIn

1.2 Datengeber (Data Provider)

DatengeberIn, also PrimärforscherIn, der/die Daten am FDZ abgibt, siehe *Metadatenabgabe (DatengeberInnen)*.

1.3 Publisher

DZHW-FDZ MitarbeiterIn

1.4 Admin

Weist User-Rollen zu.

KAPITEL 2

Metadatenabgabe (DatengeberInnen)

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 Hochschul- und Wissenschaftsforschung zur Verfügung. Ein fester Bestandteil dieser Arbeit ist zunächst der Prozess der Datenaufnahme, welcher im FDZ des DZHW durch das Metadatenmanagementsystem (MDM) unterstützt wird. Im Metadatenmanagementsystem werden Metadaten über die von Ihnen bereitgestellten Daten erfasst. Dies erleichtert es SekundärforscherInnen für sie passende Datenpakete auszuwählen. Für die strukturierte Aufnahme der Metadaten sind sieben unterschiedliche Ebenen im MDM vorgesehen: *Datenpaket*, *Erhebungen*, *Erhebungsinstrumente*, *Fragen*, *Datensätze*, *Variablen*, *Konzepte* und *Publikationen*.

Hierbei wird erfasst, welchem Datenpaket sowie welcher 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.

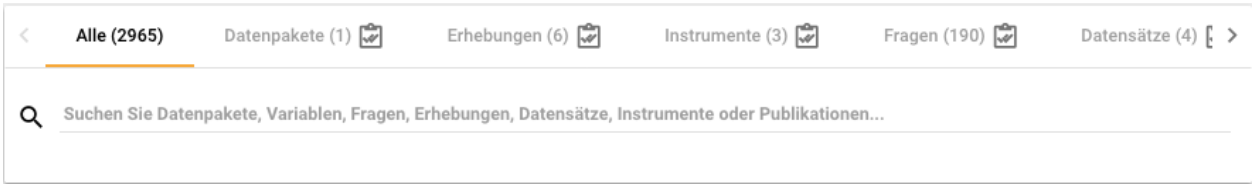


Abb. 2.1: Darstellung der verschiedenen Ebenen im MDM; Pfeil rechts klicken um die weiteren Ebenen anzuzeigen

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 größtenteils eigenständig in das MDM hochladen.

Die Abgabe der Metadaten ist innerhalb der einzelnen Ebenen unterschiedlich komplex, sodass die Daten für jede der acht Ebenen separat erfasst 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 eine erfolgreiche Erfassung der einzelnen Metadaten gewährleistet werden kann.

Aktuell können einige Metadaten über eine Eingabemaske im MDM direkt eingetragen werden. Für die übrigen Ebenen müssen JSON-Dateien hochgeladen werden, bzw eine Exceldatei ans FDZ übergeben werden bzw. hochgeladen werden. In dieser Dokumentation wird lediglich auf den Standardfall eingegangen, bei dem die Metadaten per Eingabemaske eingetragen werden. Zusätzlich ist auf manchen Ebenen das Erstellen von weiteren Anhängen (vgl. *Anhänge*) vorgesehen. Die Excel-Tabellen und eventuelle Anhänge müssen für jede Ebene dann entweder 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
Datenpaket	Eingabemaske	Anhänge (PDF)	Eingabemaske ausfüllen
Erhebungen	Eingabemaske	Anhänge (PDF)	Eingabemaske ausfüllen
Erhebungs- instrumente	Eingabemaske	Anhänge (PDF, Excel)	Eingabemaske ausfüllen
Fragen	Excel-Tabelle (oder JSON- Dateien von Zofar)	Fragebilder (.png)	Im Vorlage-Ordner speichern
Datensätze	Eingabemaske	Anhänge (PDF)	Eingabemaske ausfüllen
Variablen	mind. 1 Excel-Tabelle	mind. 1 Datensatz (Stata)	Im Vorlage-Ordner speichern
Publikationen	•	•	Informationen dem FDZ schicken
Konzepte	•	•	Informationen dem FDZ schicken

2.2 Vorbereitende Schritte

2.2.1 Vergabe der DAP-ID

Die Data Acquisition Project-ID (DAP-ID) ist das Kürzel des Datenpakets. Bitte sprechen Sie mit dem Publisher einen Vorschlag ab, welcher dann mit dem Release-Manager (aktuell Robert Birkelbach, stellv. Anne Weber) rückgesprochen wird, sodass es zu keinen Inkonsistenzen kommt.

2.2.2 Registrierung

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

Im Anschluss erhalten Sie eine Bestätigungsmail. In dieser müssen Sie auf den Aktivierungslink klicken, welcher nach drei Tagen automatisch abläuft. Im Anschluss werden Sie von uns dem Projekt in der Rolle Datengeber/in hinzugefügt.

Abb. 2.2: Registrierung im MDM Schritt 1

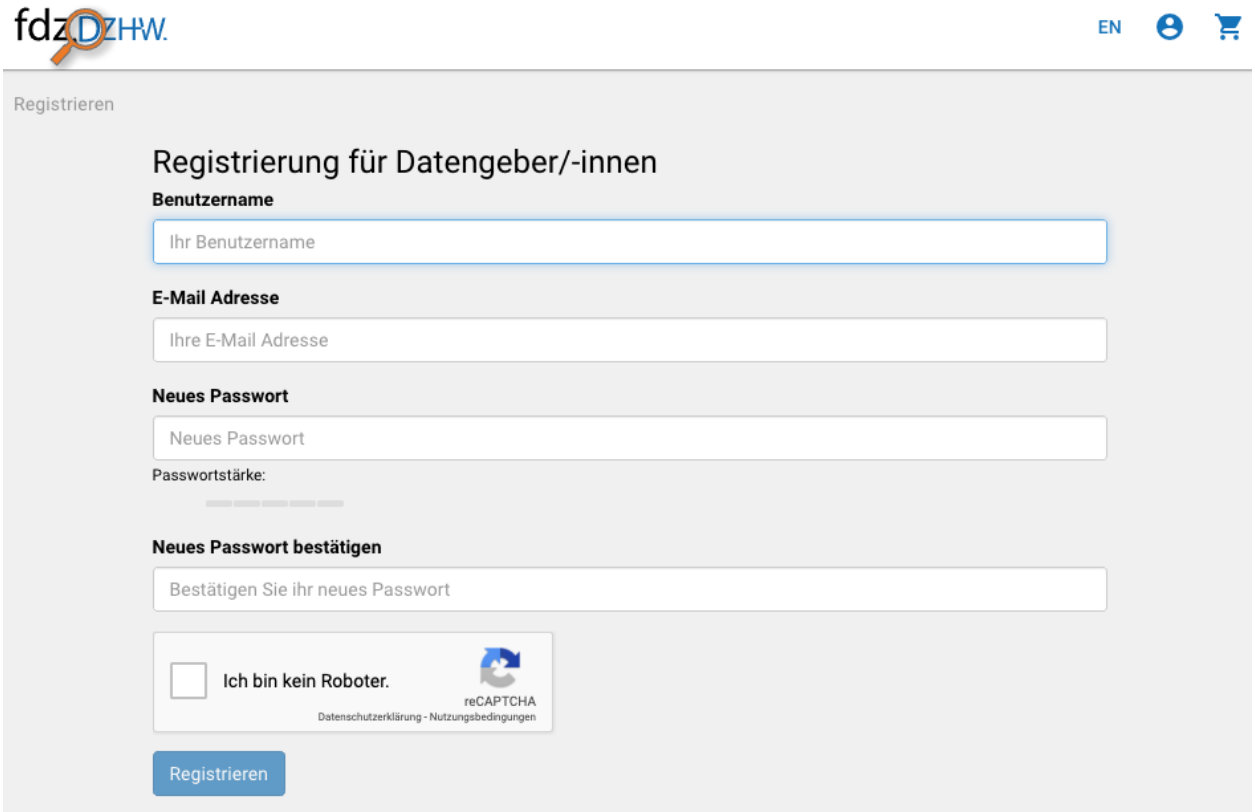


Abb. 2.3: Registrierung im MDM Schritt 2

Sie erhalten jeweils eine E-Mail wenn Sie Ihrem Datenaufbereitungsprojekt hinzugefügt wurden und wenn das Projekt Ihnen von einer FDZ-Mitarbeiterin zur Bearbeitung freigegeben wurde. Falls Sie nicht kurz darauf dem Projekt zugewiesen worden sind, sprechen Sie bitte noch einmal den/die FDZ-MitarbeiterIn an. Der/die FDZ-MitarbeiterIn muss einem der Admins Bescheid geben, zu welchem Projekt Sie hinzugefügt werden sollen.

2.2.3 Allgemeiner Hinweis zur Formatierung

An einigen Stellen gibt es die Möglichkeit Texte mittels Markdown zu formatieren. Hier finden Sie eine Anleitung. Eingabefelder welche mit einem M↓-Symbol gekennzeichnet sind, können mit Markdown formatiert werden. Es kann sein, dass die konkreten Überschriften-Ebenen nicht der angegebenen Überschriften-Ebene ensprechen. So kann es sein, dass eine H1-Überschrift auf Ebene H2 „rutscht“. Das liegt daran, dass die Überschriften-Ebenen auf der gesamten Detailseite vom System konsistent gemacht werden. Die niedrigste Überschriften-Ebene ist H6.

Abb. 2.4: Beispiel eines Eingabefeldes mit Markdown-Unterstützung.

2.2.4 Verwaltung des Projektes im Projekt-Cockpit

Das Projekt-Cockpit dient der Zusammenarbeit zwischen FDZ-MitarbeiterInnen und den DatengeberInnen (also Ihnen). In der Navigationsleiste links, welche ggfs. aufgeklappt werden muss, finden Sie den Zugang zum Projekt-Cockpit (vgl. Abb. 2.5). Wenn Sie ins Projekt-Cockpit gehen, sehen Sie unter dem Punkt Einstellungen (vgl. Abb. 2.6), welche Publisher (FDZ-MitarbeiterInnen) und DatengeberInnen dem Projekt zugewiesen sind und welche Metadaten erwartet werden. Es werden lediglich die erwarteten Metadatenebenen im Projektcockpit angezeigt.

Abb. 2.5: Projekt-Cockpit Button.

Das Status-Menü (siehe Abb. 2.7) hat einerseits Funktionen zum Projektmanagement und andererseits Funktionen um Metadaten anzulegen:

Es wird angezeigt, ob das Projekt freigegeben ist, also die eingegebenen Metadaten für alle öffentlichen Nutzer des Systems sichtbar sind. Daneben wird angezeigt, ob das Projekt gerade bei den Publishern liegt, oder bei den DatengeberInnen zur Bearbeitung liegt.

Zuerst liegt das Projekt zur Bearbeitung bei den Publishern und Sie als Datengeber/in werden per Mail benachrichtigt, wenn es Ihnen zugewiesen wird. Um das Projekt wieder den Publishern zuzuweisen, klicken Sie den „Papierflieger“-Button (siehe Abb. **??**) über dem „Zugewiesen an Datengeber“ bzw „Assigned to Publishers“ steht.

Die Vorraussetzung, dass Sie das Projekt zurückgeben können ist, dass Sie die erwarteten Metadaten eingegeben haben mittels des „Neu“ bzw. „Hochladen“ Buttons und als „fertig“ markiert haben (siehe Abb. **??**).

Wenn Sie auf den Neu-Button klicken, gelangen Sie zur Eingabemaske der jeweiligen Ebene und mit Klick auf den Hochladen Button erscheint ein File-Explorer Fenster. Details zur Abgabe der Metadaten der einzelnen Ebenen wird

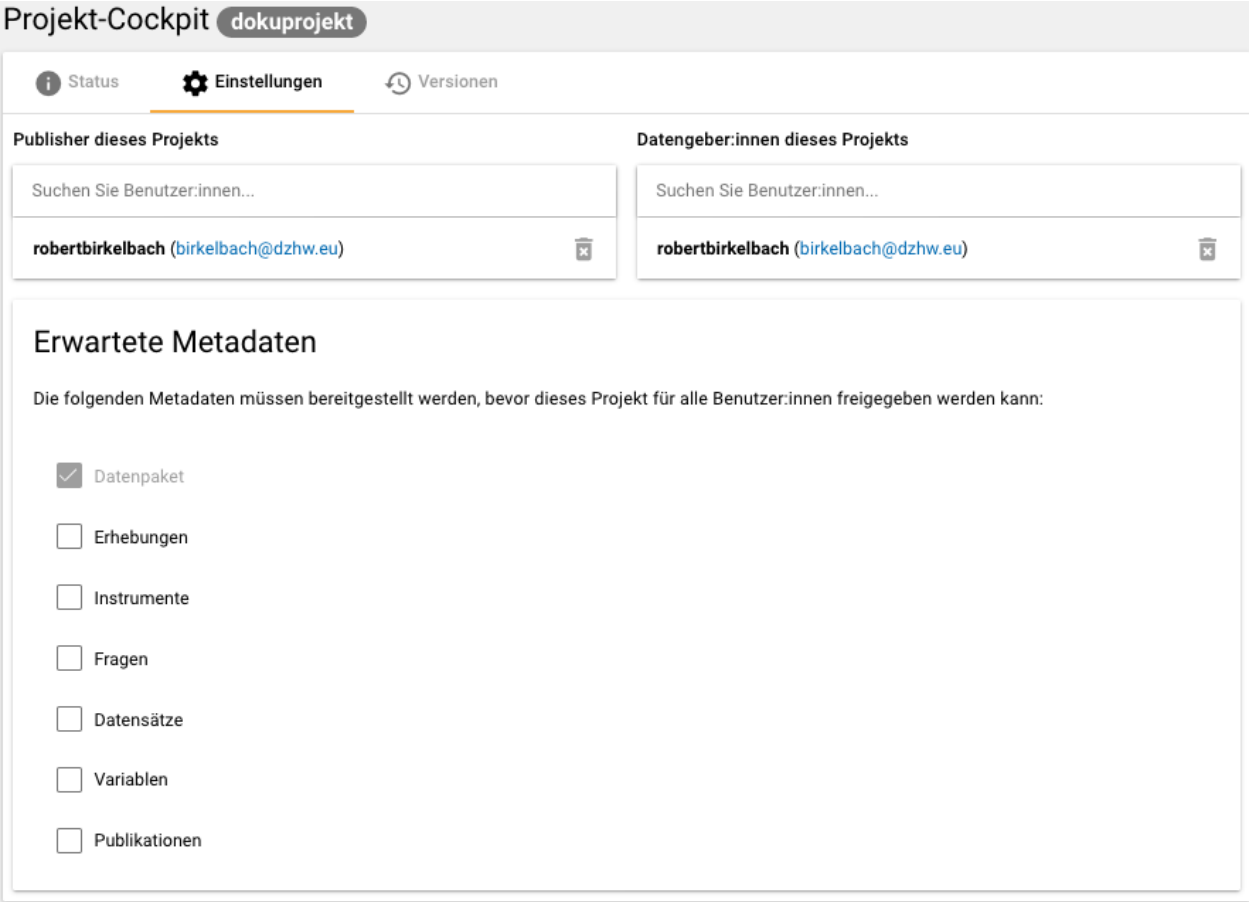


Abb. 2.6: Projekt-Cockpit Einstellungen.

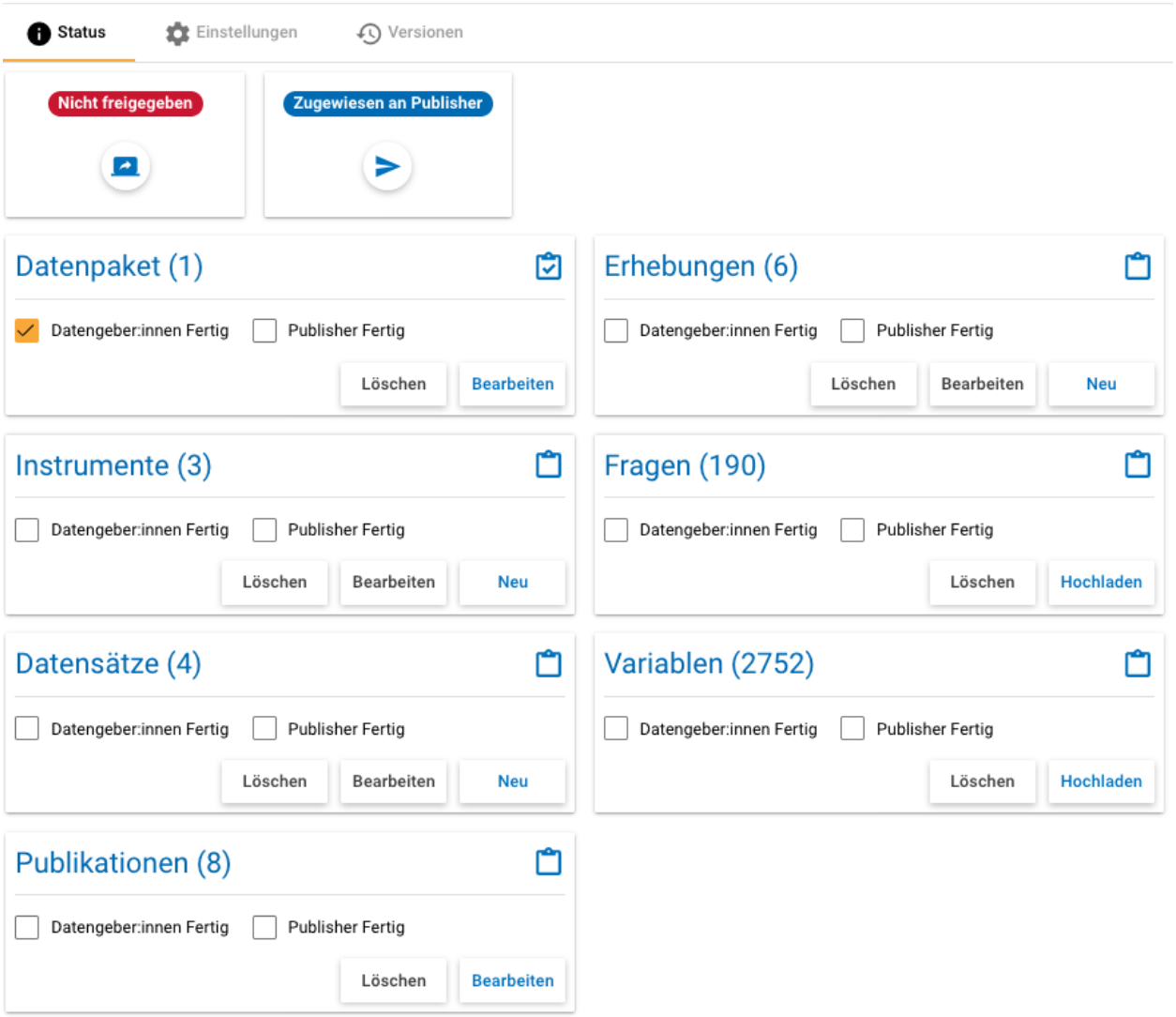
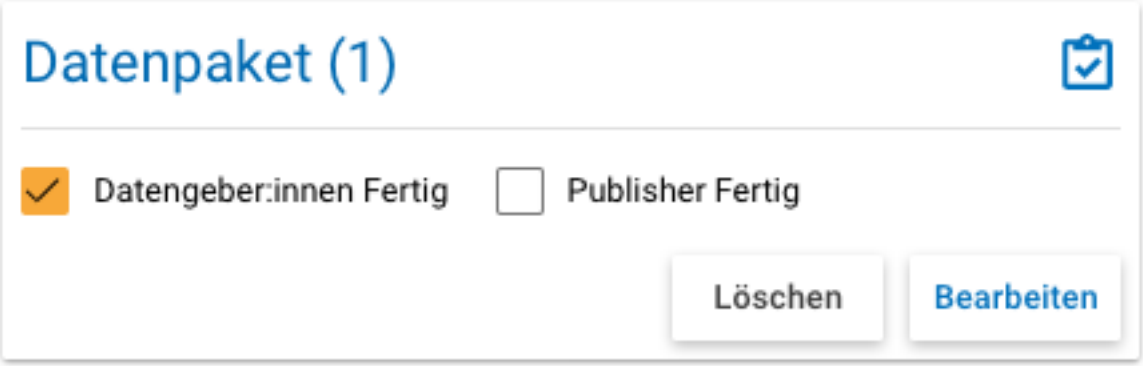


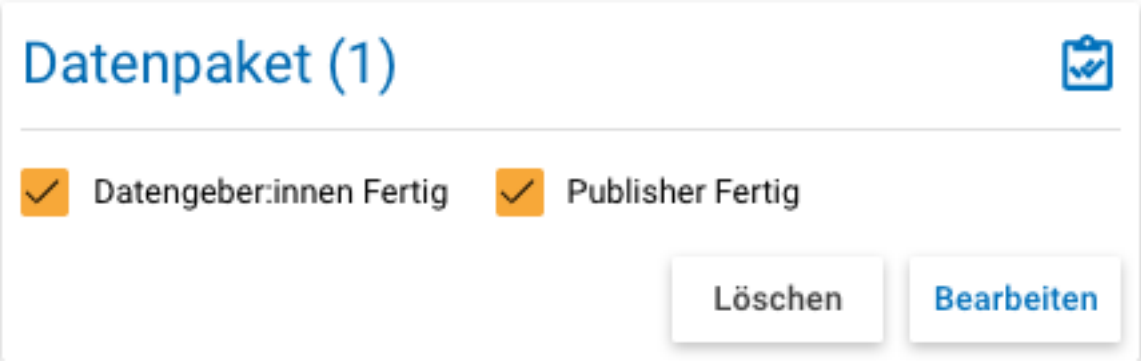
Abb. 2.7: Projekt-Cockpit Status.

im jeweiligen Kapitel erklärt. Wenn Sie fertig mit der Eingabe der Metadaten einer Ebene sind, setzen Sie bitte das entsprechende Häkchen. Es wird automatisch abgespeichert.



Sie sehen, dass sich der Status von keinem Häkchen hin zu einem Häkchen ändert, nachdem Sie „fertig“ angeklickt haben. Nachdem der Papierflieger-Button geklickt wurde, erscheint der „Nachricht an Publisher“ Dialog (siehe Abb. **??**).

Falls die Publisher denken, dass noch irgend etwas vergessen wurde oder anders eingegeben werden sollte, weisen die Publisher Ihnen das Projekt zurück zu und Sie werden per Email darüber benachrichtigt. Sollte der Publisher denken, dass die Eingabe der Metadaten auf der jeweiligen Ebene fertig sind, markiert er/sie die Ebene auch als „fertig“, was durch einen zweiten Haken signalisiert wird (siehe Abb. **??**). Sind alle erwarteten Ebenen mit zwei Häkchen markiert, können die Publisher das Projekt für alle öffentlichen Nutzern des Systems freigeben.



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 *Die Abgabe von Metadaten für die einzelnen Ebenen* detailliert beschrieben werden.

2.3 Notwendige Schnittstellen und Dateien

Entweder werden Metadaten über Eingabemasken oder per Dateiupload angelegt. Die Eingabemasken ermöglichen eine komfortable Abgabe der Metadaten direkt auf der Website und in den meisten Fällen werden Sie die Metadaten lediglich per Eingabemaske anlegen und editieren.

2.3.1 Eingabemasken

Für die Ebenenen Datenpakete, Erhebungen, Datensätze und Instrumente steht 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. Im Rahmen der relevanten Ebenen Datenpakete, Erhebungen, Datensätze und Instrumente wird die Handhabung der jeweiligen Eingabemasken dann konkret gezeigt (s. *Erhebungen (surveys) Erhebungsinstrumente (instruments)*).

Für die Ebenen Fragen, Variablen und Publikationen müssen Dateien hochgeladen werden – wenden Sie sich hierfür bitte ans FDZ. Konzepte werden vom FDZ angelegt.

2.3.2 Anhänge

Anhänge werden über die Eingabemasken im MDM hochgeladen. Zu den Anhängen zählen z.B. der Daten- und Methodenbericht auf der Datenpaketebene sowie Fragebögen oder Codierlisten auf Instrumentenebene. Diese Dokumente müssen als Dateien im PDF- oder Excel-Format vorliegen (Details dazu finden Sie in den jeweils relevanten Kapiteln zu den einzelnen Ebenen) 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 der PDF-Datei gelöscht werden. Eine Anleitung finden Sie hier: https://helpx.adobe.com/acrobat/using/pdf-properties-metadata.html

2.4 Die Abgabe von Metadaten für die einzelnen Ebenen

2.4.1 Datenpaket (data package; ehemals Studie/study)

Übersicht

Anhand der Informationen, die Sie bzgl. Ihres Datenpakets im MDM erfassen, wird dort später eine Übersichtsseite erstellt, die im Folgenden am Beispiel des Absolventenpanels 2005 dargestellt wird:

Eine neues Datenpaket anlegen

Nachdem ein neues Projekt erstellt wurde, können Sie nun innerhalb des Projektes ein Datenpaket über das Project-Cockpit anlegen.

Datenpaket anlegen über das Projekt-Cockpit

Im Projekt-Cockpit sehen Sie im Status-Bereich die Felder zu den einzelnen Metadatenebenen. Wenn Sie unter dem Punkt Datenpaket auf den „Neu“-Button klicken, gelangen Sie zur Eingabemaske. Machen Sie beim Punkt *Eingabemaske* weiter.

Datenpaket anlegen über die Suche

Dazu finden Sie im Reiter „Datenpakete“ unten rechts auf der Seite einen orangefarbenen Plus-Button (vgl. Abb. 2.10).

Mit einem Klick auf den Plus-Button öffnet sich die Eingabemaske, in der Sie Ihre Informationen zum Datenpaket ablegen können.

Datensuche > Datenpaket gra2005

DZHW-Absolventenpanel 2005

Schlagwörter: Hochschulforschung

Details

Studienreihe:	DZHW-Absolventenstudien
Erhebende Institution(en):	Deutsches Zentrum für Hochschul- und Wissenschaftsforschung (DZHW)
Gefördert von:	Bundesministerium für Bildung und Forschung (BMBF)
Projektmitarbeiter:innen:	Briedis, Kolja; Euler, Thorsten; Grotheer, Michael; Isleib, Sören; Minks, Karl-Heinz; Netz, Nicolai; Schaeper, Hildegard; Trennt, Fabian; Trommer, Maximilian
Erhebungsdesign:	Panel
Datenkuratierung:	Baillet, Florence; Franken, Andreas; Weber, Anne

Datenpaketbeschreibung

Das DZHW-Absolventenpanel 2005 ist Teil der DZHW-Absolventenstudienreihe, in der anhand von standardisierten Befragungen Informationen zu Studium, Berufseintritt, Berufsverlauf und Weiterqualifizierung von Hochschulabsolvent(inn)en erfasst werden. Das erste Absolventenpanel wurde 1989 durchgeführt, seitdem wird jeder vierte Absolvent(inn)enjahrgang (Kohorte) untersucht. Für jede Absolvent(inn)enkohorte werden mehrere Befragungswellen durchgeführt, wobei jede Welle in unterschiedlichem zeitlichen Abstand zum Studienabschluss stattfindet. Beim Absolventenpanel 2005 handelt es sich um die fünfte Absolvent(inn)enkohorte der Studienreihe. Im Unterschied zu vorangegangenen Absolvent(inn)enkohorten ist die Studienphase der Kohorte 2005 durch den Hochschulwandel im Rahmen des Bologna-Prozesses geprägt. In diesem Zuge wurden nicht nur Absolvent(inn)en traditioneller Studiengänge, sondern erstmalig auch eine nennenswerte Anzahl an Bachelorabsolvent(inn)en befragt, wobei lediglich Absolvent(inn)en aus Fächern berücksichtigt wurden, für die bereits eine größere Absolvent(inn)enzahl vorlag. Zudem ist die berufliche Einstiegs- und Konsolidierungsphase der Kohorte 2005 durch den Beginn der Wirtschafts- und Finanzkrise im Jahr 2008 gekennzeichnet.

Materialien zu diesem Datenpaket

Datenpaket

Instrumente

Datensätze

Verbundene Objekte - Version 2.0.0

Erhebungen (6) | Instrumente | Fragen | Datensätze | Variablen | Publikationen | Konzepte

Suchen (Erhebungen) Suchen

Sortiert nach Relevanz Einträge pro Seite 10 1 - 6 von 6

DZHW-Absolventenpanel 2005 - erste Welle (Absolvent(inn)en traditioneller Studiengänge)

Quantitative Daten, Standardisierte postalische Befragung

Grundgesamtheit: Hochschulabsolvent(inn)en mit einem traditionellen

Stichprobenverfahren: Kombination aus Wahrscheinlichkeits- und Nicht-

Abb. 2.8: Datenpaketübersicht im MDM am Beispiel des Absolventenpanels 2005

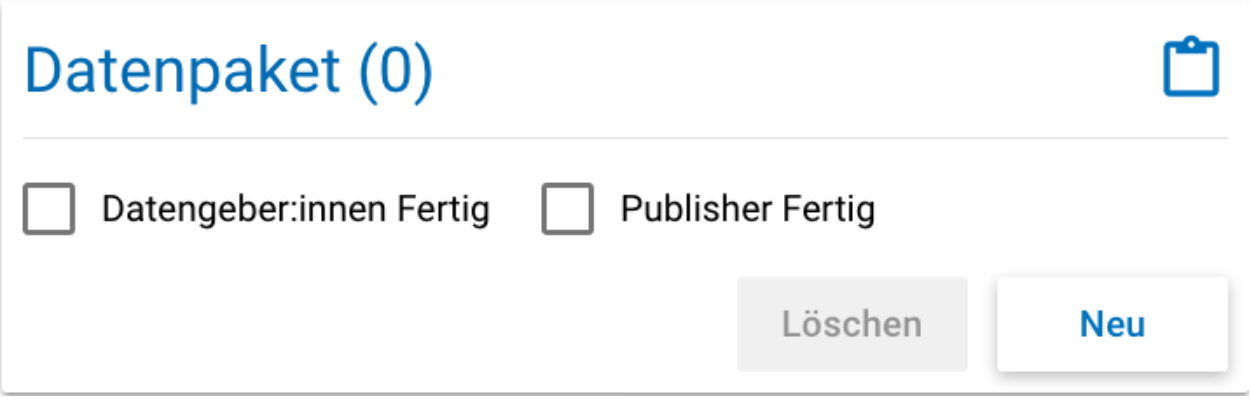


Abb. 2.9: Datenpaket per Cockpit anlegen

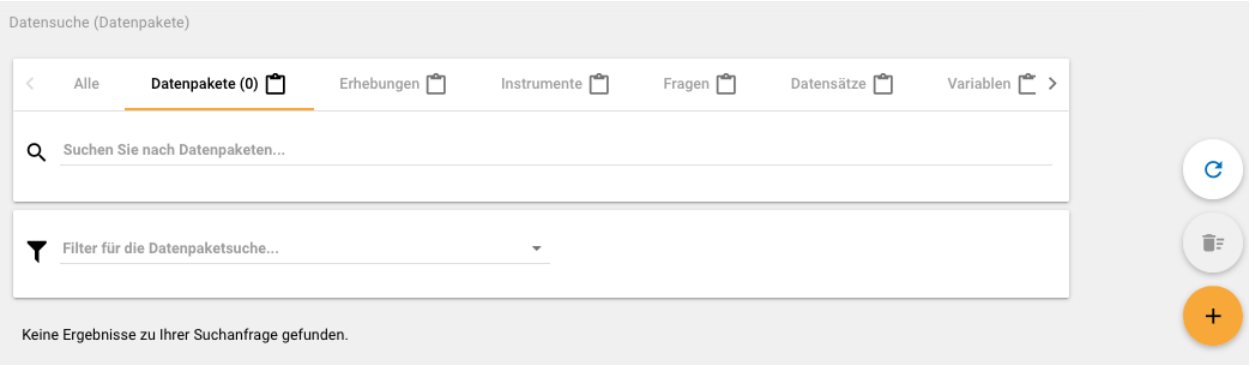


Abb. 2.10: Datenpaket per Suche anlegen

Eingabemaske

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

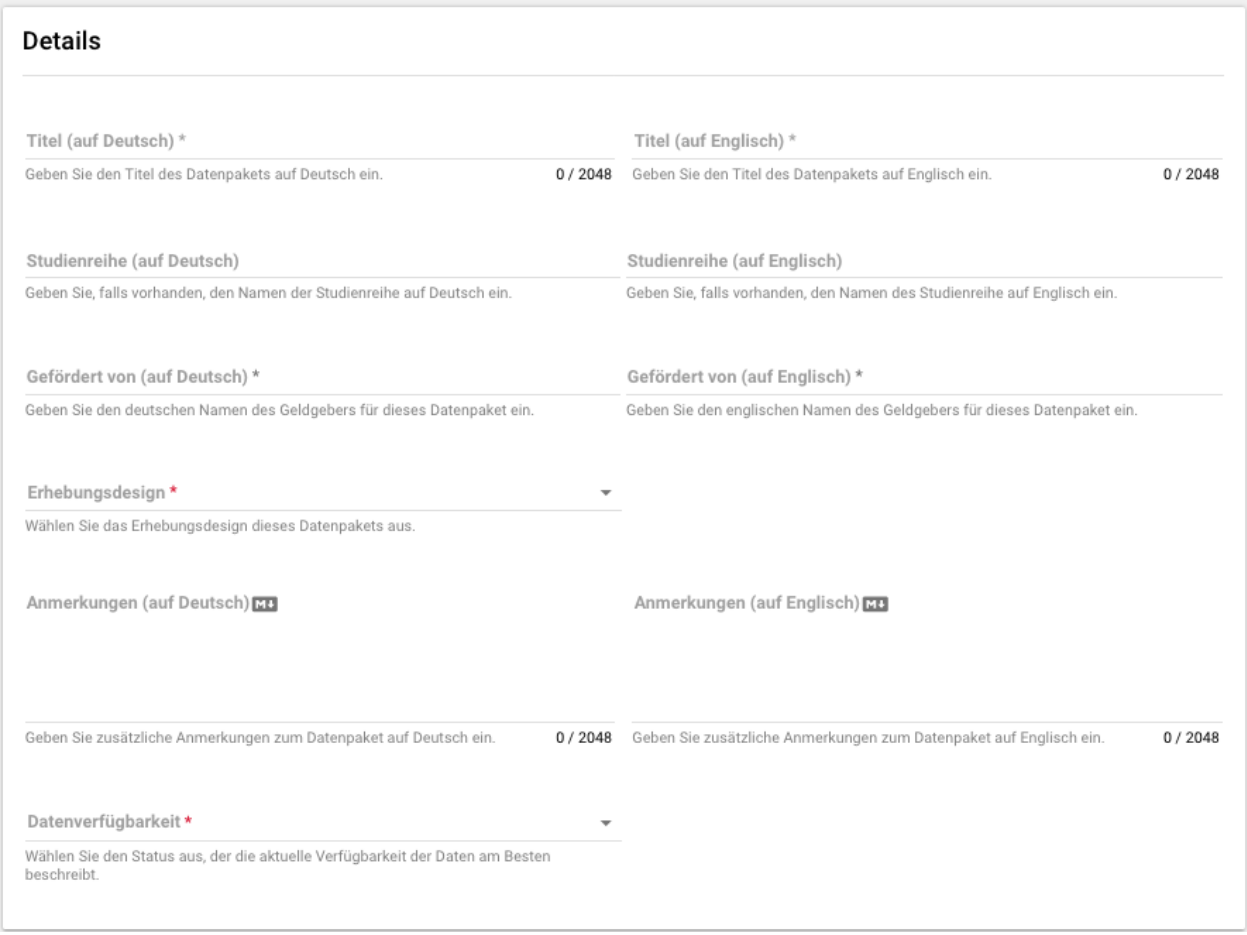


Abb. 2.11: Eingabemaske auf Datenpaketebene, Abschnitt „Details“ am Beispiel der 21. Sozialerhebung

Nach dem Öffnen der Eingabemaske erscheint ganz oben die aus ihrem Projektnamen automatisch generierte ID für die Datenpaketseite (s. rotes Kästchen in Abb. 2.11). 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. 2.11). Außerdem finden Sie teilweise Drop-Down-Menüs vor, in denen Sie aus vorgegebenen Alternativen auswählen können (s. grünes Kästchen in Abb. 2.11).

Im Feld Erhebende Institution(en) (s. Abb. **??**) geben Sie an welche Institution die Konzeption und Durchführung der Erhebung des Datenpakets durchgeführt hat. Es werden Institutionen vorgeschlagen, die bereits andere Datenpakete durchgeführt haben. Sollte kein Vorschlag passen, geben Sie bitte den Institutionsnamen auf Deutsch und Englisch ein.

Im zweiten Abschnitt der Eingabemaske müssen Sie eine Beschreibung Ihres Datenpaket sowohl auf Deutsch als auch auf Englisch eingeben. Als 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. 2.13). Wenn mindestens zwei Personen eingetragen sind, erscheinen die Pfeil-Buttons als aktiv (Farbwechsel von grau zu blau). Dann können Sie die Reihenfolge der Per-

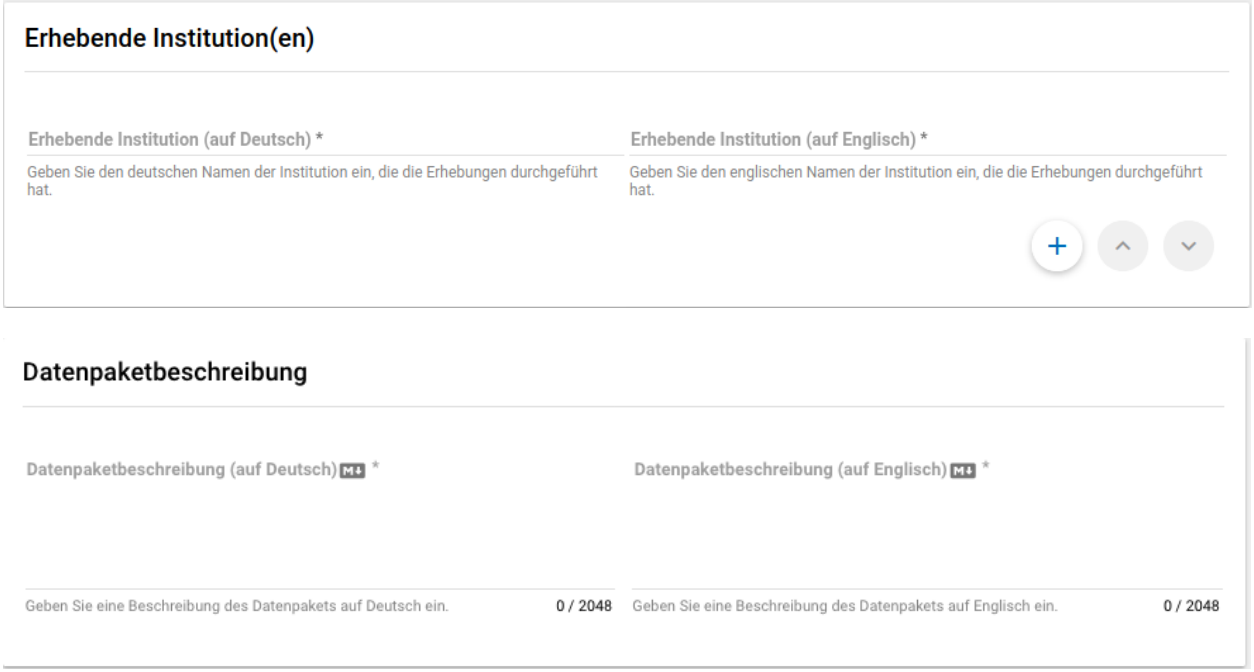


Abb. 2.12: Eingabemaske auf Datenpaketebene, Abschnitt „Datenpaketbeschreibung“ am Beispiel der 21. Sozialerhebung

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 zum Datenpaket“) nicht bearbeiten können.

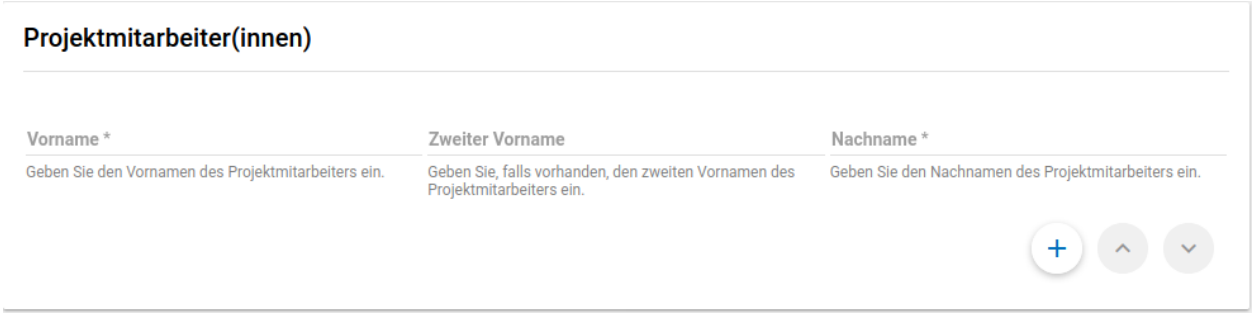


Abb. 2.13: Eingabemaske auf Datenpaketebene, Abschnitt „Projektmitarbeiter(innen)“

Im Feld Tags (Schlagwörter) zum Datenpaket sind kurze Schlagwörter anzugeben, die dabei helfen, schnell einen Überblick über die wichtigsten Themen des Datenpakets zu erhalten und ihr Datenpaket schnell auffindbar zu machen. Außerdem erleichtert es forschenden ähnliche Datenpakete, die das selbe Schlagwort verwendet haben, zu finden.

Im vierten und letzten Abschnitt der Eingabemaske können Sie Materialien zum Datenpaket ablegen. Dazu klicken Sie auf den blauen Plus-Button (s. Abb. 2.17), 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 deutsch oder englischsprachige *Datenpaketübersicht/data package overview*.[1] Die Sprache der Materialien muss nach ISO 639-1 angegeben werden. Bei den Metadaten der Materia-

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

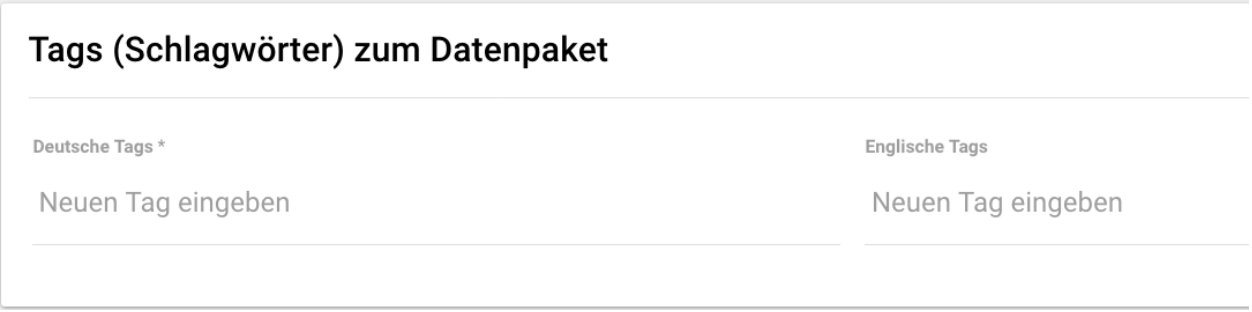


Abb. 2.14: Eingabemaske Schlagwörter/ Tags

lien ist darauf zu achten, dass diese korrekt eingegeben worden sind. 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.

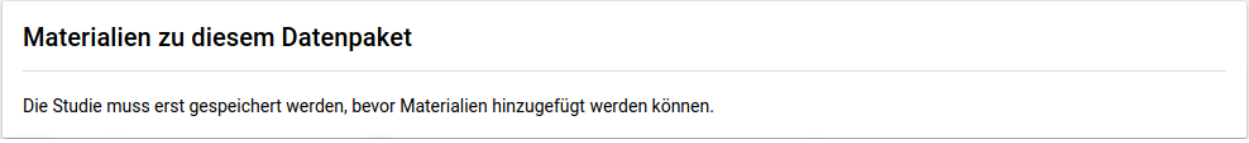


Abb. 2.15: Materialienabschnitt ist noch ausgegraut

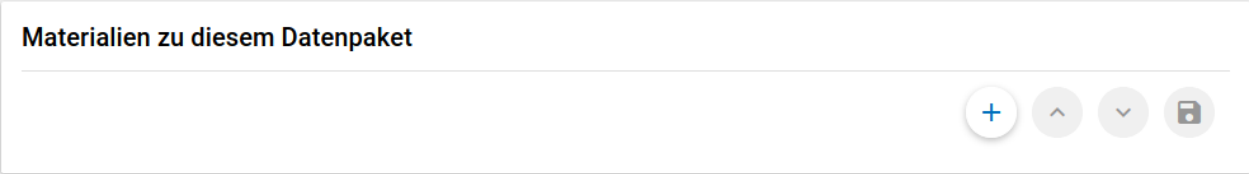


Abb. 2.16: Materialien können hinzugefügt werden

@TODO describe attachments

Editieren und historisieren

Falls Sie Ihre Informationen auf Datenpaketebene 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 „Datenpakete“ am rechten Rand neben Ihrers Datenpakets ein Stift-Button angezeigt, über den Sie wieder in die Eingabemaske gelangen (s. Abb. 2.18).

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. 2.19).

Bei einem Klick auf den Historisierungs-Button öffnet sich ein Dialog, der die verschiedenen Versionen des Datenpakets anzeigt (s. Abb. 2.20). Zudem sind der Name des Nutzers, der die entsprechende Version des Datenpakets 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 zum Datenpaket (Dateianhänge) nicht historisiert werden, sondern lediglich deren ins MDM eingegebene Metadaten.

Abb. 2.17: Eingabemaske zu den Materialien des Datenpakets

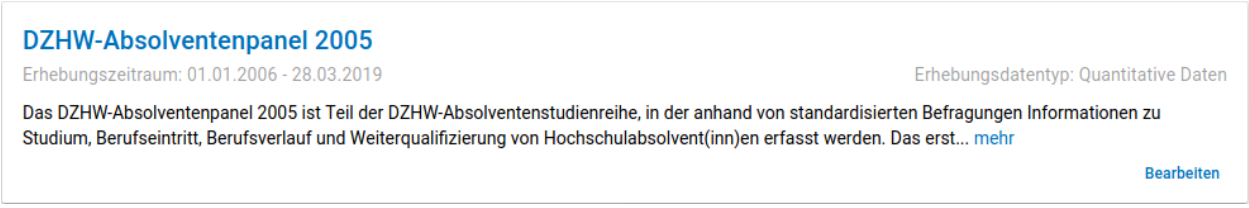


Abb. 2.18: Weitere Bearbeitung eines bereits abgespeicherten Datenpakets

Abb. 2.19: Ältere Versionen eine Datenpakets wiederherstellen

Abb. 2.20: Dialog zur Historisierung innerhalb eines Datenpakets

2.4.2 Erhebungen (surveys)

Übersicht

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

Eine neue Erhebung anlegen

Wenn Sie ein Datenpaket angelegt haben (vgl. Kapitel *Datenpaket (data package; ehemals Studie/study)*), können Sie über den Reiter „Erhebungen“ eine neue Erhebung innerhalb Ihres Datenpakets erstellen. Hierzu finden Sie unten rechts auf der Seite – ebenso wie bei Datenpaketen – einen orangefarbenen Plus-Button. 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. 2.22, hier als Beispiel der 21. Sozialerhebung). Neben den bereits aus der Datenpaketebene bekannten Funktionen gibt es in dieser Eingabemaske zusätzlich eine Kalenderfunktion (s. blaue Kästchen, Abb. 2.22), welche die Feldzeit des Projekts erfasst und in Abb. 2.23 dargestellt ist:

Die Rücklaufquote wird automatisch ermittelt. 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,

[2] Rücklaufgrafiken sind nur im Dokumentationsstandard der Stufe 3 gefordert. Die Erläuterungen zu den drei verschiedenen Dokumentations-

Datensuche > Datenpaket gra2005 > Erhebung 1

DZHW-Absolventenpanel 2005 - erste Welle (Absolvent(inn)en traditioneller Studiengänge)

Details

Grundgesamtheit:	Hochschulabsolvent(inn)en mit einem traditionellen Studienabschluss, die im Wintersemester 2004/2005 oder im Sommersemester 2005 ihren ersten berufsqualifizierenden Studienabschluss an einer staatlich anerkannten Hochschule in der Bundesrepublik Deutschland erworben haben (mit Ausnahme der Absolvent(inn)en von Bundeswehrhochschulen, Verwaltungsfachhochschulen, Berufsakademien und Fernhochschulen)
Erhebungseinheit:	Hochschulabsolventen
Erhebungsdatentyp:	Quantitative Daten
Stichprobenverfahren:	Kombination aus Wahrscheinlichkeits- und Nicht-Wahrscheinlichkeitsauswahl
Erhebungsmethode:	Standardisierte postalische Befragung
Feldzeit:	01.01.2006 - 18.05.2007
Nettostichprobe:	n = 10.165
Anmerkungen:	Die Bruttostichprobe der ersten Welle umfasst ca. 47.800 Fälle. Es ist nicht rekonstruierbar, wie viele dieser Fälle zur Gruppe der Absolvent(inn)en eines traditionellen Studiengangs und wie viele zur Gruppe der Bachelorabsolvent(inn)en gehörten.
Untersuchungsgebiet:	Deutschland

Weitere Informationen zum Rücklauf

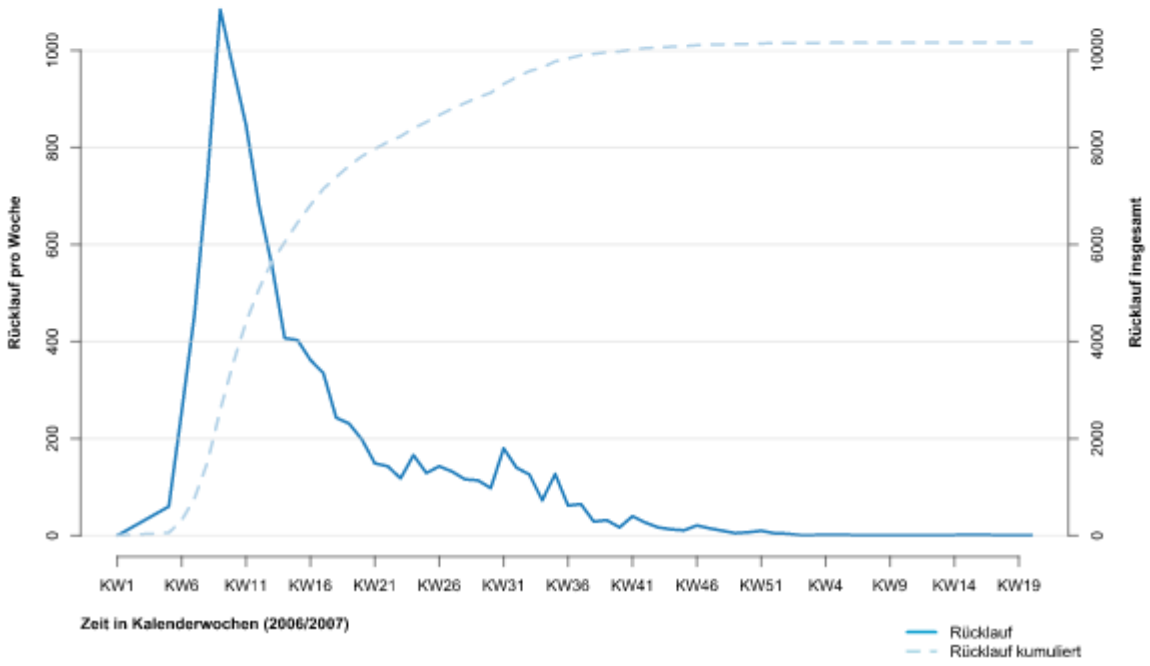


Abb. 2.21: Erhebungsübersicht im MDM am Beispiel der ersten Welle (traditioneller Studiengänge) im Absolventenpanel 2005

Details

Titel (auf Deutsch) *
Geben Sie den Titel der Erhebung auf Deutsch ein. 0 / 2048

Titel (auf Englisch) *
Geben Sie den Titel der Erhebung auf Englisch ein. 0 / 2048

Welle *
1
Geben Sie die Nummer der Welle an bzw. lassen Sie die Nummer auf 1 falls nicht zutreffend.

Beginn Feldzeit *
Geben Sie den Beginn der Feldzeit ein.

Ende Feldzeit *
Geben Sie das Ende der Feldzeit ein.

Erhebungsmethode (auf Deutsch) *
Beschreiben Sie die Erhebungsmethode auf Deutsch.

Erhebungsmethode (auf Englisch) *
Beschreiben Sie die Erhebungsmethode auf Englisch.

Erhebungsdatentyp *
Wählen Sie den Erhebungsdatentyp aus.

Beschreibung der Grundgesamtheit (auf Deutsch) MD *
Beschreiben Sie die Grundgesamtheit auf Deutsch. 0 / 2048

Beschreibung der Grundgesamtheit (auf Englisch) MD *
Beschreiben Sie die Grundgesamtheit auf Englisch. 0 / 2048

Stichprobenverfahren *

Bruttostichprobe
Geben Sie die Größe Ihrer Stichprobe (brutto) an.

Nettostichprobe *
Geben Sie die tatsächliche Größe Ihrer Stichprobe (netto) an.

Rücklaufquote (in %)
Geben Sie die Rücklaufquote in Prozent an.

Wählen Sie eine Erhebungseinheit aus *
Wählen Sie eine Erhebungseinheit aus

Anmerkungen (auf Deutsch) MD
Geben Sie zusätzliche Anmerkungen zu der Erhebung hier auf Deutsch an. 0 / 2048

Anmerkungen (auf Englisch) MD
Geben Sie zusätzliche Anmerkungen zu der Erhebung hier auf Englisch an. 0 / 2048

Abb. 2.22: Eingabemaske der Erhebungsebene, Abschnitt „Details“ Teil 1

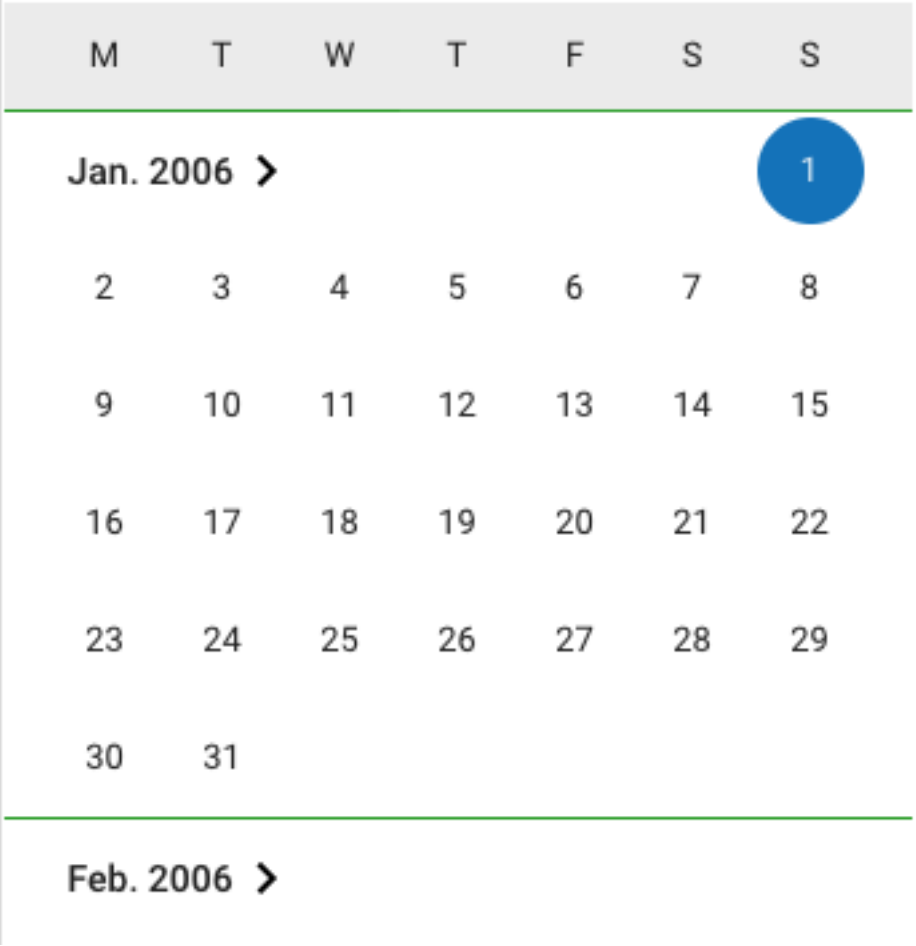


Abb. 2.23: Kalenderfunktion auf der Erhebungsebene

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-Format vorliegen. Über den Button mit dem Mülleimer-Symbol lassen sich hochgeladene Dateien wieder löschen (s. Abb. 2.24).

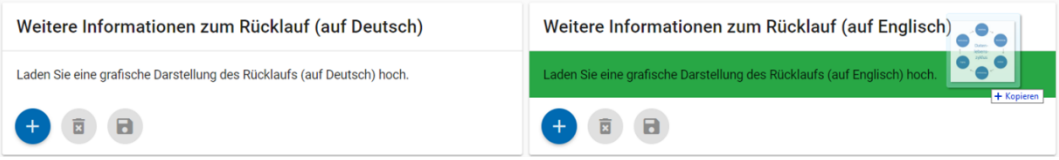


Abb. 2.24: Eingabemaske der Erhebungsebene, Abschnitt „Weitere Informationen zum Rücklauf“

Im letzten Abschnitt der Eingabemaske können – wie auch beim Datenpaket – Materialien hinzugefügt werden (s. Abb. 2.25). Die Funktionsweise ist identisch zu der auf Datenpaketebene.[3]

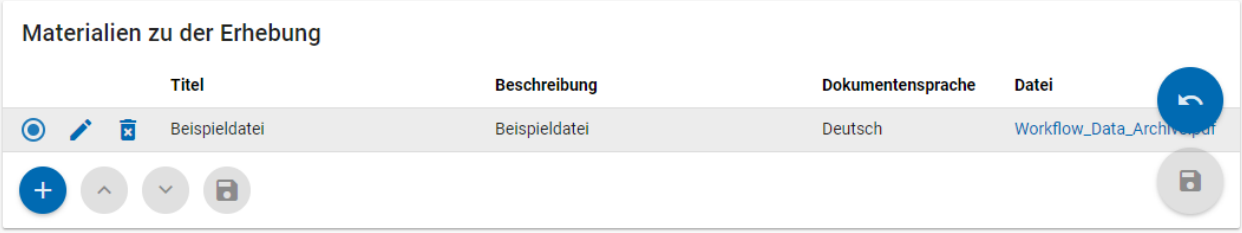


Abb. 2.25: Eingabemaske der Erhebungsebene, Abschnitt „Materialien zu der Erhebung“

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. 2.26).

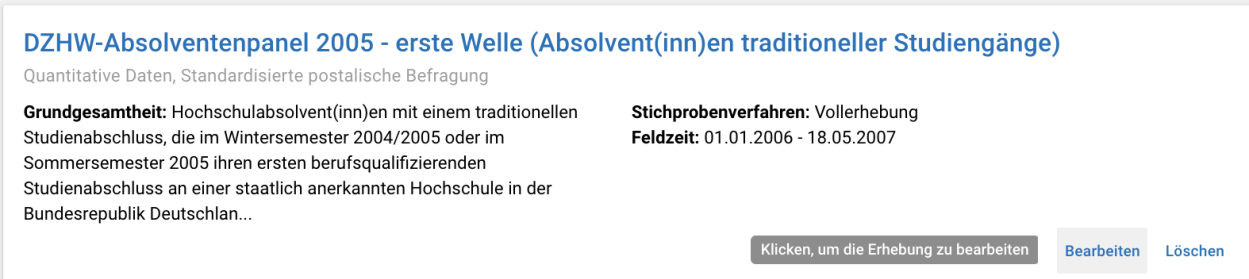


Abb. 2.26: Weitere Bearbeitung einer bereits abgespeicherten Erhebung

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. 2.27).

Bei einem Klick auf den Historisierungs-Button öffnet sich ein Dialog, der die verschiedenen Versionen der Erhebung anzeigt (s. Abb. 2.28). Zudem sind der Name des Nutzers, der die entsprechende Version des Datenpakets gespeichert

standards finden Sie in den Dokumenten „Anforderungen an Daten und Dokumentation im FDZ des DZHW“. Sie erhalten es auf Anfrage.

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

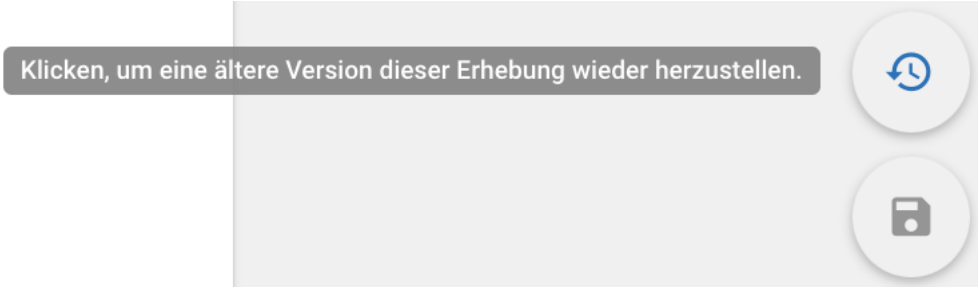


Abb. 2.27: Ältere Versionen einer Erhebung wiederherstellen

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 Erhebung nicht historisiert werden.

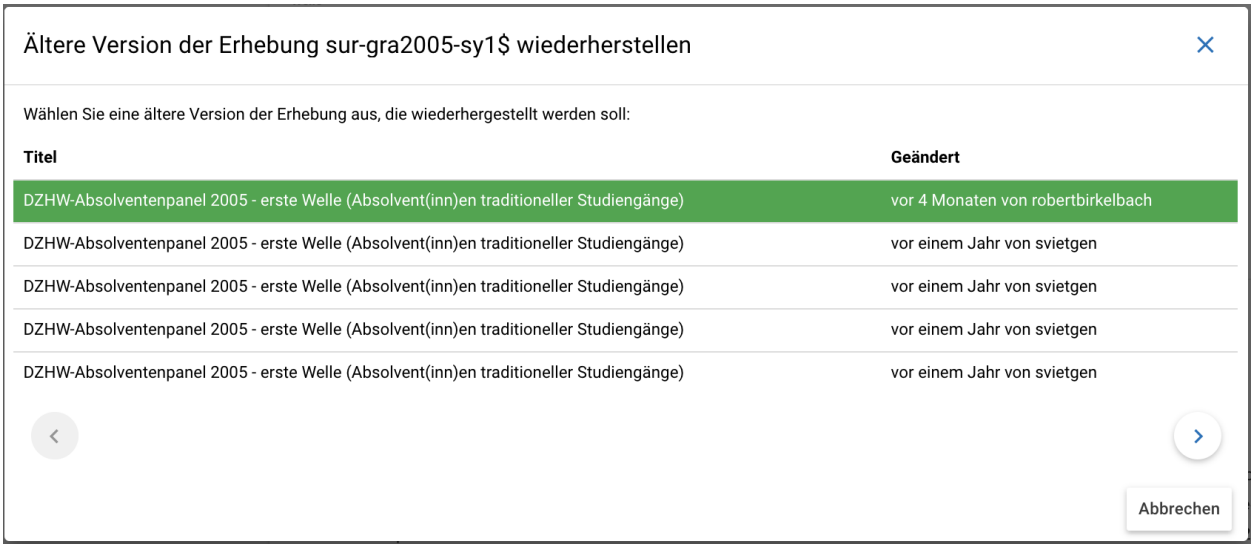


Abb. 2.28: Dialog zur Historisierung innerhalb einer Erhebung

Prüfschritte

Der Titel der Erhebung wird bei Veröffentlichung (mit Versionsnummer von mindestens 1.0.0) bei dalra vor einige Attribute (z.B. Referenzzeitraum) gehängt. Der Titel der Erhebung muss daher eindeutig sein und im Falle von längsschnittlich erhobenen Datenpaketen die Welle enthalten.

2.4.3 Erhebungsinstrumente (instruments)

Als Instrument wird das Erhebungsinstrument bezeichnet (z.B. Fragebogen).

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 das Datenpaket aktuell nicht released sein. Um ein Erhebungsinstrument mittels Eingabemaske anzulegen, wird im Projektcockpit unter Instrumen-

Abb. 2.29: Instrumentenübersicht im MDM am Beispiel des Fragebogens der ersten Welle im Absolventenpanel 2005

te auf den Neu-Button oder geht über die Suche in die Instrumentenebene und klickt auf den Plus-Button. Es öffnet sich bei beiden Herangehensweisen die Eingabemaske um ein neues Instrument anzulegen.

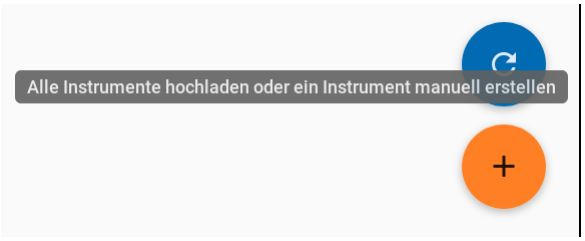


Abb. 2.30: Plusbutton

Die Eingabemaske besteht aus den Pflichtfeldern Beschreibung, Titel, Typ und Erhebung, sowie den nicht verpflichtenden Feldern Untertitel, Anmerkungen und Konzepte (s. Abb. 2.31). Konzepte müssen zuerst, wie gleichnamigen Kapitel erklärt, angelegt werden und können danach über die Eingabemaske verlinkt werden.

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. 2.32), 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.

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.

Sollte es Erhebungsinstrumente geben, welche in einer anderen Sprache als deutsch oder englisch existieren, werden diese nur als Attachment und nicht auf Variablenebene bereitgestellt.

[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“. Sie erhalten das Dokument auf Anfrage.

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

Instrument bearbeiten: ins-gra2005-ins1$

Details

Beschreibung (auf Deutsch) *
Fragebogen des DZHW-Absolventenpanels 2005 - erste Welle
Geben Sie eine kurze Beschreibung für das Instrument auf Deutsch ein. 56 / 512

Beschreibung (auf Englisch) *
Questionnaire of the DZHW Graduate Panel 2005 (first wave)
Geben Sie eine kurze Beschreibung für das Instrument auf Englisch ein. 58 / 512

Titel (auf Deutsch) *
Zwischen Hochschule und Arbeitsmarkt
Geben Sie den Titel des Instruments auf Deutsch ein. 36 / 2048

Titel (auf Englisch) *
Between university and labour market
Geben Sie den Titel des Instruments auf Englisch ein. 36 / 2048

Untertitel (auf Deutsch)
Geben Sie den Untertitel des Instruments auf Deutsch ein. 0 / 2048

Untertitel (auf Englisch)
Geben Sie den Untertitel des Instruments auf Englisch ein. 0 / 2048

Typ *
PAPI
Wählen Sie den Typ des Instruments aus.

Erhebungen *
sur-gra2005-sy1$ sur-gra2005-sy3$
Erhebungen suchen...
Wählen Sie die Erhebungen aus, in denen dieses Instrument verwendet wurde.

Anmerkungen (auf Deutsch) MD
Geben Sie zusätzliche Anmerkungen zu dem Instrument hier auf Deutsch an. 0 / 2048

Anmerkungen (auf Englisch) MD
Geben Sie zusätzliche Anmerkungen zu dem Instrument hier auf Englisch an. 0 / 2048

Konzepte
Konzepte suchen...
Wählen Sie die Konzepte aus, die mit diesem Instrument gemessen wurden.

Materialien zum Instrument

Typ	Fragebogen	Fragebogen	Filterführungsdiagramm
Beschreibung	eingesetztes Erhebungsinstrument	englische Übersetzung des deutschen Originalfragebogens	grafische Darstellung des dur
Dokumentensprache	Deutsch	Englisch	Deutsch
Datei	gra2005_W1_Questionnaire_de.pdf	gra2005_W1_Questionnaire_en.pdf	gra2005_W1_QuestionFlow_c

Abb. 2.31: Eingabemaske Instrument

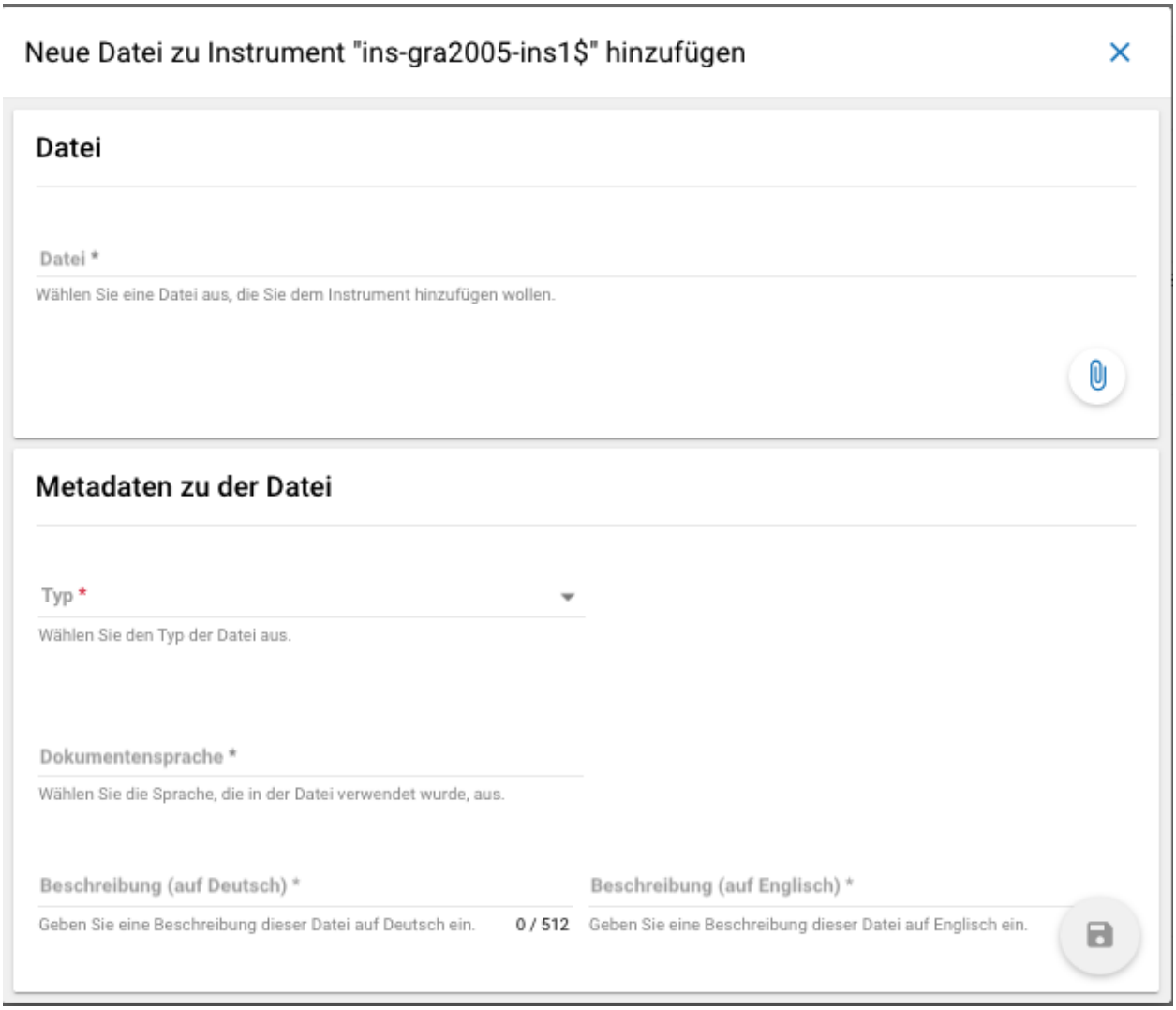


Abb. 2.32: Instrumente Anhang

2.4.4 Datensätze (dataSets)

Übersicht Für die Dokumentation der Datensätze werden die „Master“(AIP)-Datensätze (siehe Zwiebelmodell) genutzt. Diese Datensätze sind die größte mögliche Vereinheitlichung eines Datensatzes, also keine Teilpopulation oder Teilmenge von Variablen eines Datensatz. Datensätze die sich als Teilmenge eines „Master“-Datensatzes abbilden lassen werden über die Subdatensätzen (SubDataSets) dokumentiert. 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.

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

Personendatensatz DZHW-Absolventenpanel 2005 (Bachelor)

Details

Typ:	Personendatensatz
Format:	breit

Verfügbare Subdatensätze

Zugangsweg	Beschreibung	Name	Variablen	Analysierbare Variablen	Fälle	Zitation
download-suf	Kann nach Abschluss eines Datennutzungsvertrags heruntergeladen werden	gra2005_ba_p_d_2-0-0	1.777	1.584	1.622	
remote-desktop-suf	Kann über Remote Desktop genutzt werden	gra2005_ba_p_r_2-0-0	1.777	1.665	1.622	
onsite-suf	Ist am Gastwissenschaftlerarbeitsplatz im DZHW in Hannover verfügbar	gra2005_ba_p_o_2-0-0	1.777	1.746	1.622	

Materialien zu dem Datensatz

Titel	Beschreibung	Dokumentensprache	Datei
Variablendokumentation: DZHW-Absolventenpanel 2005 (Datensatz 3)	Datensatzreport des dritten Datensatzes des DZHW-Absolventenpanels 2005	Deutsch	dsreport-gra2005-ds3.pdf

Verbundene Objekte - Version 2.0.0

Abb. 2.33: Datensatzübersicht im MDM am Beispiel des Personendatensatzes (Bachelor) im Absolventenpanel 2005

Eingabemaske

Datensätze lassen sich per Eingabemaske anlegen und editieren. Hierfür muss man entweder über das Projektcockpit gehen, oder in der Suche auf den Reiter Datensätze klicken (Abb. 2.1), anschließend auf das Plussymbol (Abb. 2.34) in der unteren rechten Ecke klicken. Anschließend öffnet sich die Eingabemaske (siehe Abb. 2.35).

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.

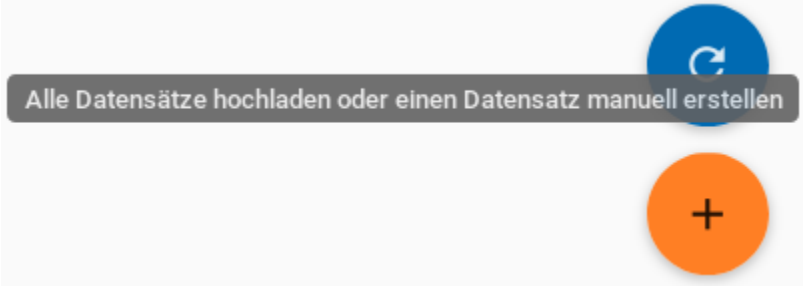


Abb. 2.34: Neuen Datensatz hinzufügen.

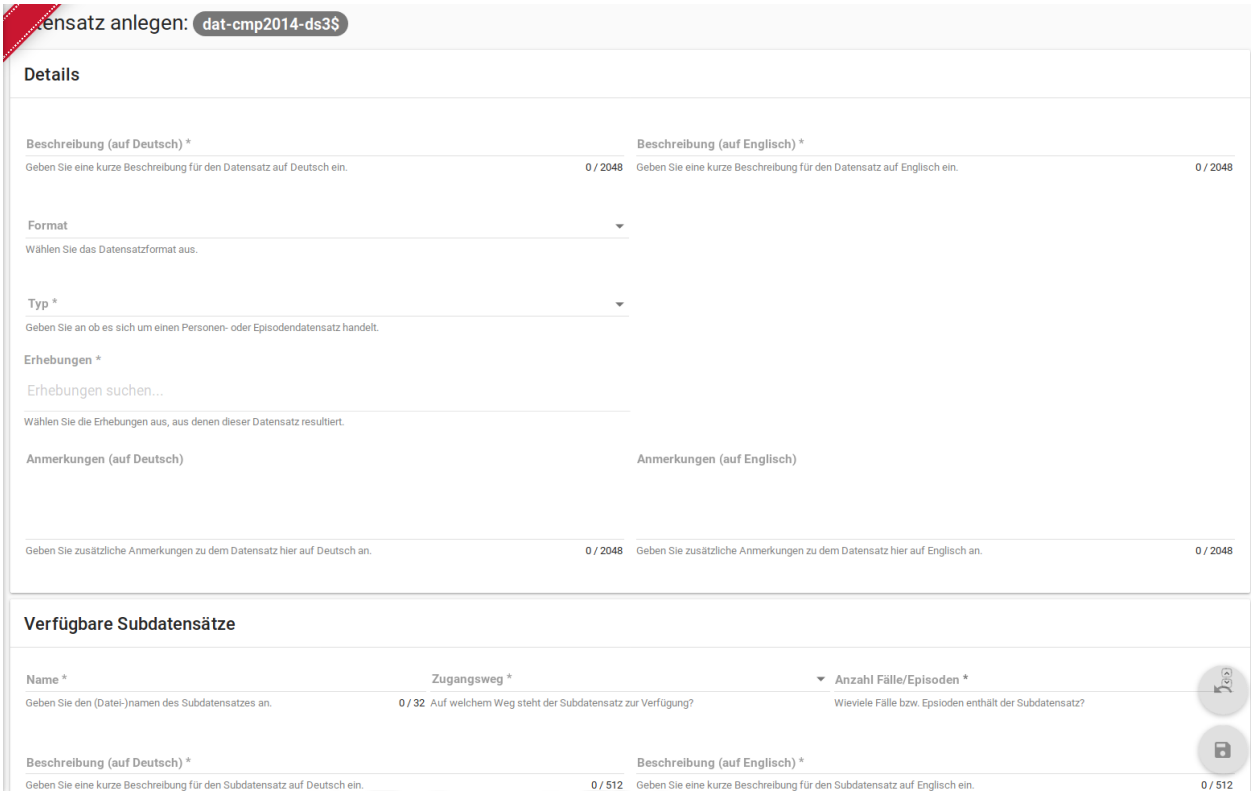


Abb. 2.35: Eingabemaske der Datensatzebene.

Wenn Sie zusätzliche Materialien (z.B. Variablen-Dokumentation) auf Ebene der Datensätze haben, können Sie diese hinzufügen. Hierfür muss zunächst der Datensatz angelegt sein. Anschließend wird in der unteren linken Ecke auf den blauen Plus-Button geklickt.[6]

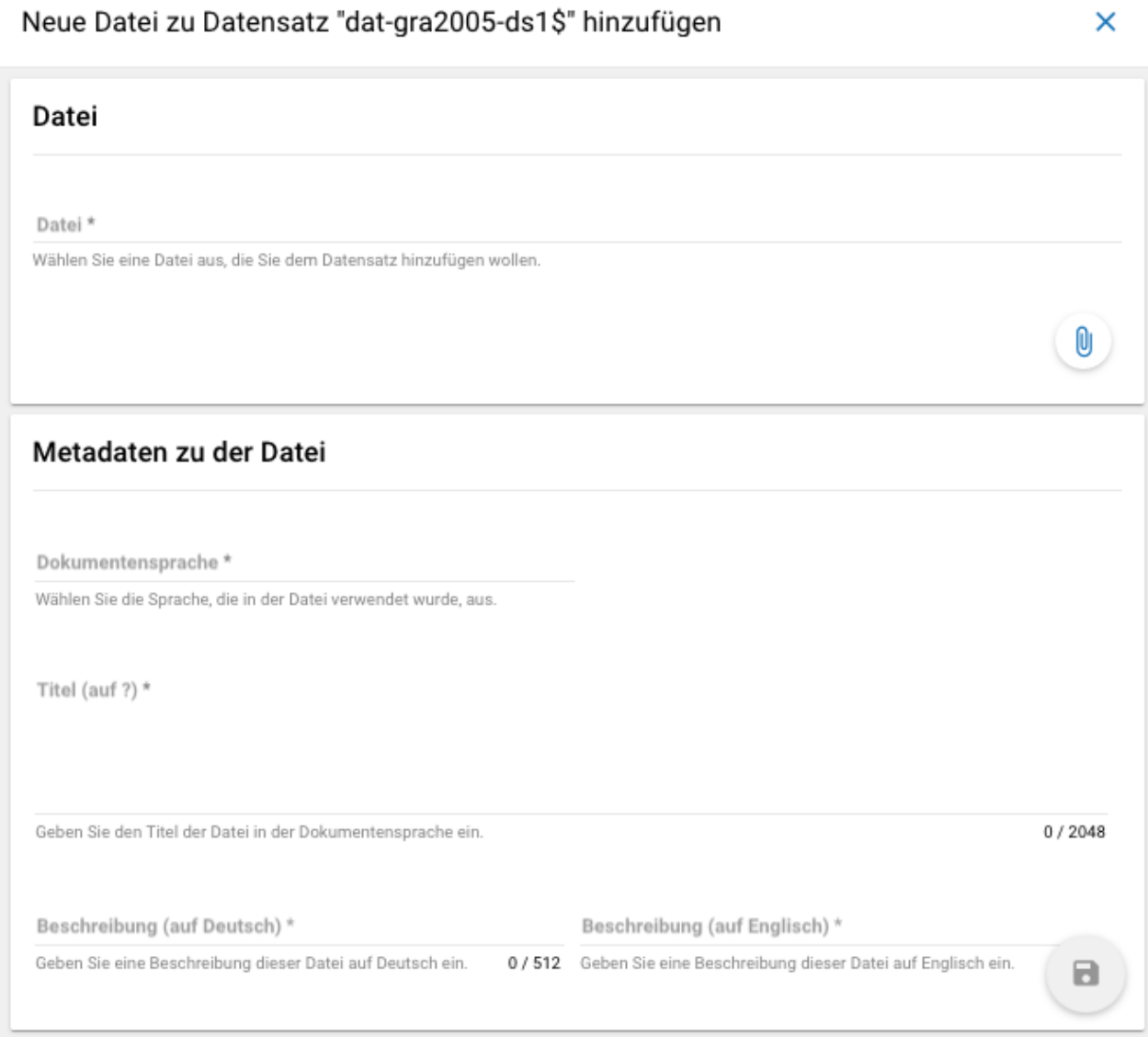


Abb. 2.36: Datensatz Anhänge

Es öffnet sich ein Fenster (siehe Abb. 2.36) in dem Sie eine Datei hochladen können und Metadaten zur Datei angeben müssen.

2.4.5 Fragen

Sollten Sie mit uns die Eingabe von Frage-Metadaten vereinbart haben, sprechen Sie uns bitte darauf an. Wir erläutern Ihnen dann den Prozess. Metadaten für Fragen müssen als JSON und png Dateien vorliegen. Sollte eine Umfrage mit Zofar durchgeführt worden sein, bitten wir Sie uns die Metadaten zukommen zu lassen. Die Fragen-Metadaten werden

[6] Bitte beachten Sie, die dokumenteigenen Metadaten bei PDF-Dateien vorab zu löschen (vgl. *Anhänge*).

von uns hochgeladen. Hier befindet sich die technische Dokumentation zur Erstellung der Fragemetadaten. Um Fragen ins MDM zu laden benutzen Sie den Upload-Button im Projektcockpit.

2.4.6 Variablen

Sollten Sie die Bereitstellung von Variablenmetadaten mit uns vereinbart haben, sprechen Sie uns bitte an. Wir erläutern Ihnen dann die notwendigen Schritte. Hier befindet sich die technische Dokumentation zur Erstellung der Variablenmetadaten. Um Variablen ins MDM zu laden benutzen Sie den Upload-Button im Projektcockpit.

2.4.7 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:

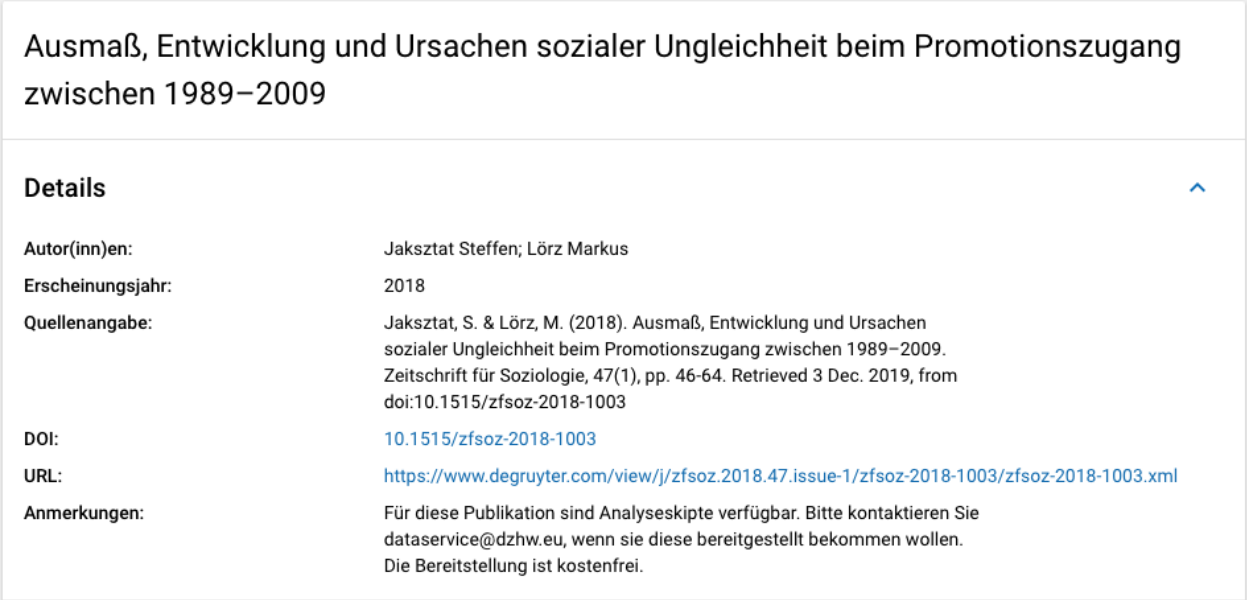


Abb. 2.37: Publikationsübersicht im MDM am Beispiel einer Veröffentlichung, welche im Rahmen des Absolventenpanels 2005 verfasst wurde

Wenn Sie Publikationen zu Ihren Daten abgeben möchten, senden Sie dem FDZ per Mail die PDF-Datei Ihrer Publikation sowie den dazugehörigen Zitationshinweis zu.

2.4.8 Konzepte

Konzepte können ausschließlich von Publishern angelegt werden. Der Begriff „Konzept“ kann auf mehreren Ebenen angewendet werden. Im Kontext des MDM sind konkrete Konzept-Instrumente gemeint. Im folgendes wird es am Beispiel des Konzeptes Persönlichkeit erklärt: Es gibt mehrere Modelle, die das Konzept „Persönlichkeit“ erfassen können: z.B. Big5 und DISG. Ins MDM tragen Sie bitte konkrete Messinstrumente ein, also z.B. eine bestimmte Big5 Kurzskala. Da so eingetragene Konzepte mit mehreren Datenpaketen (auf verschiedenen Ebenen) verknüpft werden können, kann der/die EndnutzerIn so Datenpakete heraussuchen, die ein bestimmtes Konzept auf gleiche Art und Weise gemessen haben.

Konzept anlegen:

Details

FDZ-ID *

Geben Sie die ID dieses Konzeptes in unserem FDZ an. 0 / 507

Titel (auf Deutsch) *

Geben Sie den Titel des Konzepts auf Deutsch ein. 0 / 512

Titel (auf Englisch) *

Geben Sie den Titel des Konzepts auf Englisch ein. 0 / 512

Zitationshinweis M↓ *

Geben Sie an, wie dieses Konzept zitiert werden soll. 0 / 2048

DOI

Bitte geben Sie die DOI für dieses Konzept an (falls verfügbar). 0 / 512

Abb. 2.38: Konzeptdetails

Zunächst müssen Sie eine Konzept-ID festlegen. Diese folgt der Form Abkürzung.Jahreszahl, wobei sich die Jahreszahl auf das Publikationsdatum des Zitationshinweises bezieht. Die ID, Titel und Zitationshinweis sind verpflichtend auszufüllen, während die DOI lediglich angegeben werden muss, wenn eine DOI registriert wurde.

Abb. 2.39: Konzeptbeschreibung

Eine Beschreibung des Konzepts ist verpflichtend auf Deutsch und Englisch.

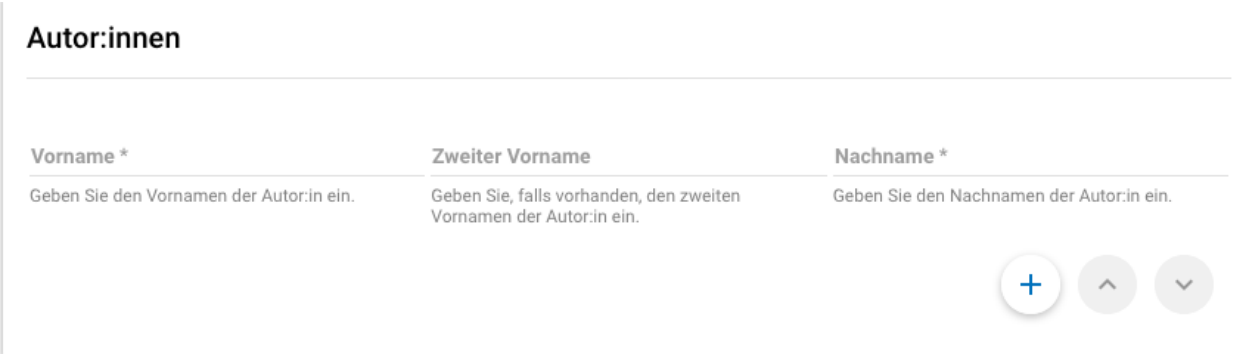


Abb. 2.40: Konzept-Autor:innen

Bitte geben Sie außerdem alle Autor:innen des Konzeptes an.

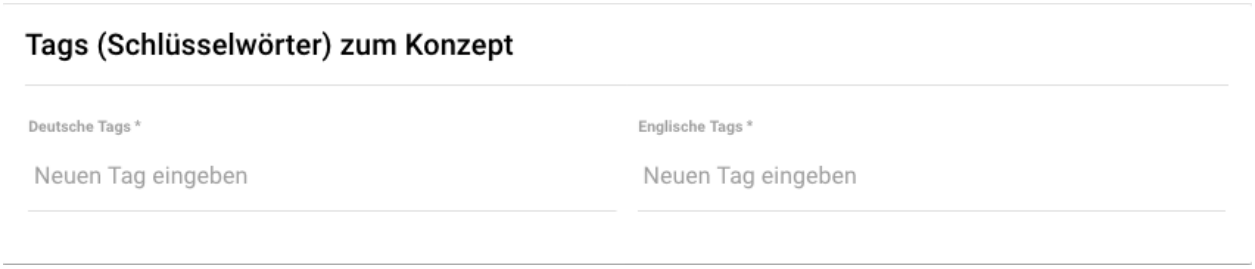


Abb. 2.41: Konzept-Tags

Sie können außerdem Tags, also Schlüsselwörter zum Konzept angeben. Diese sind nicht verpflichtend, erleichtern es aber Datennutzer:innen sehr, für sie relevante Datenpakete zu finden.

Idealerweise hat ein Konzept eine Lizenz, sodass rechtlich geregelt ist, unter welchen Umständen das Konzept weitergegeben, verwendet oder modifiziert werden darf. Dieses wird auf Englisch eingetragen (ggfs. muss es übersetzt werden). Außerdem kann ein Link zur Lizenz angegeben werden.

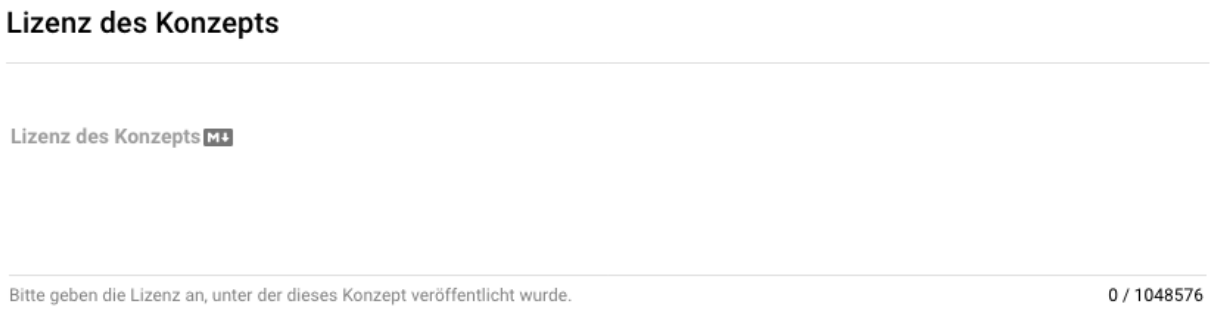


Abb. 2.42: Konzept-Lizenz

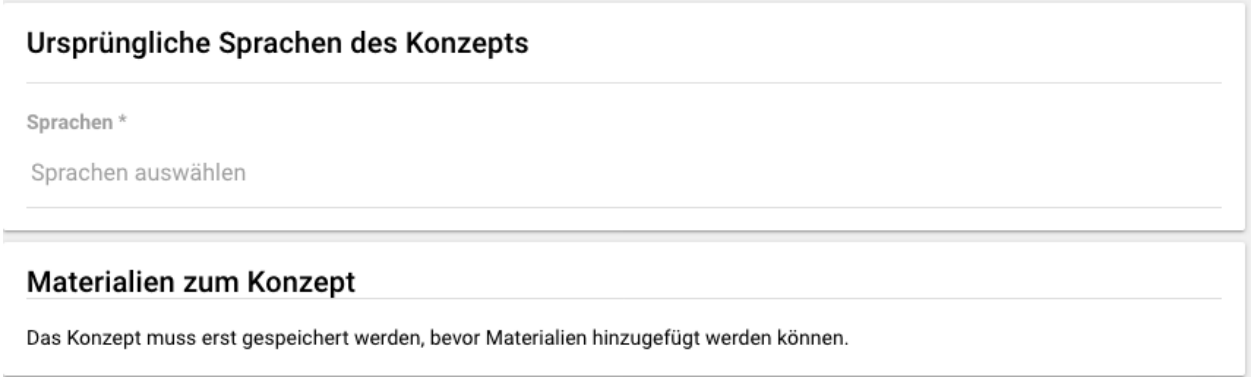


Abb. 2.43: Konzept-Sprache und -Materialien

Die ursprüngliche Sprache(n) des Konzepts müssen Sie auch angeben. Materialien zum Konzept, also Anhänge, können erst nachdem das Konzept gespeichert wurde angehängt werden.

Fragen und Instrumente können mit Konzepten verbunden werden. Die Verknüpfung von Fragen und Konzepten geschieht im Handcrafted-to-MDM-Schritt der Frage-Metadatenerstellung.

2.5 Projekte freigeben

Wenn Sie alle Metadaten ausgefüllt bzw. ans FDZ gesendet haben, markieren Sie im Project-Cockpit die Metadaten als fertig. 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. Mit der Freigabe sind Ihre Metadaten und damit auch Ihre Datenprodukte für alle öffentlichen Nutzer des Systems verfügbar.

2.5.1 Checkliste für Abgabe der Metadaten

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

Ordnerstruktur und Dateinamen sind unverändert

Falls nötig: Excel-Dateien sind richtig und vollständig ausgefüllt

- verpflichtende Felder sind ausgefüllt
- es sind keine Fehlermeldungen vorhanden
- die Daten wurden auf Richtigkeit überprüft

Metadaten sind im PDF gelöscht (vgl. *Anhänge*)

Nicht benötigte Zeilen entfernt (falls Fragen oder Variablen abgegeben werden)

- questions.xlsx: löschen bis Zeile 2000
- variables.xlsx: löschen bis Zeile 2000

Dateien sind richtig abgegeben worden

In das Metadatensystem eingegeben

- Datenpaket (data package)
- Erhebungen (surveys)

In der Ordnerstruktur abgelegt

- questions.xlsx
- variables.xlsx

KAPITEL 3

FDZ-MitarbeiterIn (Publisher, Developer)

3.1 ID-Vergabe

Manuell vergebene ids (DAP-ids) müssen in folgender Tabelle festgehalten werden.

Das Datenaufbereitungs-Team kann sich das Kürzel selbst überlegen. Es orientiert sich in der Regel am englischsprachigen Titel des Datenpakets: drei Buchstaben + Jahr der Erhebung. Die ID muss mit dem Release Manager rückgesprochen werden und es sollten alle anderen FDZ-Aufbereitungsteams kurz informiert werden. Hintergrund ist, dass Kürzel für geplante Datenpakete reserviert sein könnten, beispielsweise SLC.

3.1.1 Logik

Metadaten	Id-Generierung
DataAcquesitionProject (DAP-id)	wird manuell vergeben, siehe Tabelle oben Übersicht über alle Projekte des DZ-HW
Data Package	„stu-“ + DAP-id + „$“
Survey	„sur-“ + DAP-id + „-“ + „sy“ + survey.number + „$“
DataSet	„dat-“ + DAP-id + „-“ + „ds“ + dataSet.number + „$“
Variable	„var-“ + DAP-id + „-“ + „ds“ + variable.dataSetNumber + „-“ + variable name + „$“
Instrument	„ins-“ + DAP-id + „-“ + „ins“ + number + „$“
Question	„que-“ + DAP-id + „-ins“ + instrumentNumber + „-“ + number + „$“
relatedPublication	„pub-“ + citaviId + „$“

3.1.2 Beispiele am Absolventenpanel 2005

Metadaten	Id
DataAcquesitionProject	gra2005
Data Package	stu-gra2005$
Survey	sur-gra2005-sy1$
DataSet	dat-gra2005-ds1$
Variable	var-gra2005-ds1-stu01$
Instrument	ins-gra2005-ins1$
Question	que-gra2005-ins1-1.1$
BibliographicalReference	pub-Meier.2010$

3.2 Neues Projekt anlegen

Im Folgenden wird zunächst der Prozess zur Erstellung eines Projektes und dann die Vergabelogik der DataAcquisitionProject-ID erklärt.

3.2.1 Prozess

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

Abb. 3.1: Anmeldung im MDM

Nach erfolgreicher Anmeldung erscheint in der Sidebar das folgende Feld:

Über den Plus-Button können *Publisher* ein neues Projekt anlegen (vgl. Abb. 3.3). Als Projektname müssen diese 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).

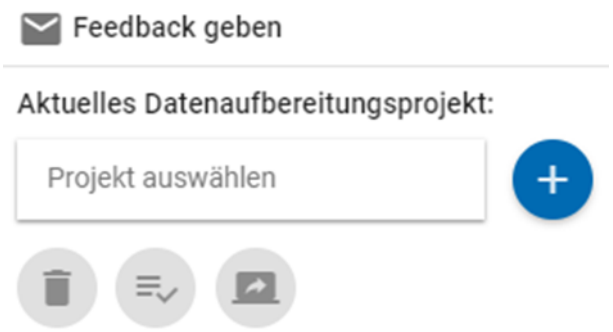


Abb. 3.2: Bereich für Verwaltung von Projekten im MDM

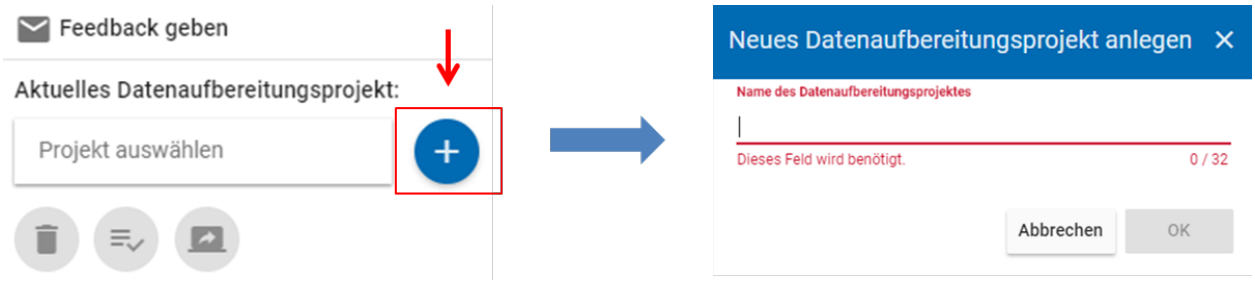


Abb. 3.3: Ein neues Projekt im MDM anlegen

3.2.2 DataAcquisitionProject-ID

- **drei Kleinbuchstaben**: Um die internationale Nutzbarkeit der Daten zu erleichtern, ergeben sich die drei Kleinbuchstaben aus der englischsprachigen Abkürzung des Projektes. Es kann Ausnahmen geben, wenn z.B. bestimmte Projekte ein besonders griffiges Kürzel haben wie z.B. bei Libertas 2016 - lib2016.
- **(zwei bis) vier Ziffern**: Die Ziffern sind in der Regel die vier Ziffern des Jahres, das die Zugehörigkeit zur Stichprobe definiert, z. B. der Abschluss des Studiums (z. B. im (Prüfungs-)Jahr 2005 oder der Erwerb der Hochschulzugangsberechtigung im Jahr 2008). Davon kann in begründeten Ausnahmefällen abgewichen werden.
- Beispielsweise ist die Sozialerhebung deutlich stärker unter der jeweiligen Nummer des Datenpakets als unter dem Jahr, das die Zugehörigkeit zur Stichprobe definiert, bekannt, so dass beispielsweise für die 19. Sozialerhebung aus dem Jahr 2009 die Ziffern 19 (und nicht 2009) vergeben werden.
- Bei Befragungen, die sich auf mehrere Jahreszahlen beziehen, kann eine andere eindeutige Jahreszahl verwendet werden. Beispielweise gehören in der KomPaed-Befragung die zuletzt aktiven Panelteilnehmer mehrerer Kohorten zur Stichprobe. Hier werden die vier Ziffern des Erhebungsjahres (2014) genutzt.

Für jedes weitere Objekt wird später ebenfalls eine ID generiert, die die DAP-id enthält und sich nach der folgenden Logik zusammensetzt:

Metadaten	Id-Generierung
Datenpaket	„stu-“ + DAP-id + „$“
Survey	„sur-“ + DAP-id + „-“ + „sy“ + survey.number + „$“
DataSet	„dat-“ + DAP-id + „-“ + „ds“ + dataSet.number + „$“
Variable	„var-“ + DAP-id + „-“ + „ds“ + variable.dataSetNumber + „-“ + variable name + „$“
Instrument	„ins-“ + DAP-id + „-“ + „ins“ + number + „$“
Question	„que-“ + DAP-id + „-ins“ + instrumentNumber + „-“ + number + „$“
relatedPublication	„pub-“ + citaviId + „$“

Das Projekt ist dadurch im MDM angelegt, aber noch nicht automatisch freigegeben. Solange das Projekt noch nicht freigegeben wurde, können Nutzer der Gruppe *publisher* es jederzeit über den Mülleimer-Button ganz links (vgl. Abb. 3.3) wieder löschen.

3.3 Bilderfassung aus PDF Dateien

3.3.1 Voraussetzungen:

Fragebögen, die als eine pdf Datei vorliegen sollen gemäß der nächsten Schritte bearbeitet werden.

Liegt zu dem Fragebogen, den es von Ihnen zu bearbeiten gilt, eine äquivalente Excel Datei vor, so wäre es ratsam diese als Grundlage für die weitere Arbeit zu verwenden. Hierbei ist es notwendig, dass jeder aufgeführten Frage in der Excel Datei eine finale Bilddatei gegenüber steht.

Für die Bilderfassung aus einer pdf Datei kann sowohl der Adobe Acrobat XI Pro als auch der Adobe Acrobat DC genutzt werden. In dieser Anleitung soll es aber im Detail um das Arbeiten mit dem Adobe Acrobat XI Pro gehen, während die Vorgehensweise mit dem Adobe Acrobat DC nur sehr knapp dargestellt wird.

3.3.2 Adobe Acrobat DC (knapp):

- Schritt 1: Öffnen der Datei in Adobe Acrobat DC
- Schritt 2: Klicken Sie mittels der linken Maustaste auf „Werkzeuge“. Unter dieser Schlagzeile öffnet sich eine weitere Leiste. Wählen Sie nun „pdf Datei bearbeiten“ und anschießend „Seite beschneiden“ aus. Nun sind Sie in der Lage einzelne Fragen auszuschneiden.
- Schritt 3.1: Nachdem Sie alle Fragen mittels der in Schritt 2 genannten Tools ausgeschnitten haben, können Sie diese nun Extrahieren und speichern. Dies erfolgt mit dem Werkzeug „Seiten verwalten“. Wählen Sie auch dies durch einen Linksklick mittels der Maus aus. Hier sind zwei Wege denkbar um Fortzufahren.
- Schritt 3.2.1 Wählen Sie die jeweiligen Seite, die es zu extrahieren gilt aus und ziehen Sie in einen von Ihnen gewählten Ordner (durch stetiges Festhalten der linken Maustaste).

  oder

  - 3.2.2 Wählen Sie in der rechten Leiste (Werkzeuge) „exportieren“, bestimmen Sie die Seite, die exportiert werden soll und speichern Sie es in einem von Ihnen gewählten Ordner.

  Achten Sie darauf, dass der jeweilige Dateiname der Fragenummer entspricht! (zur Not auch nochmal mit Excel Tabelle abgleichen, falls vorhanden)

3.3.3 Adobe Acrobat XI Pro

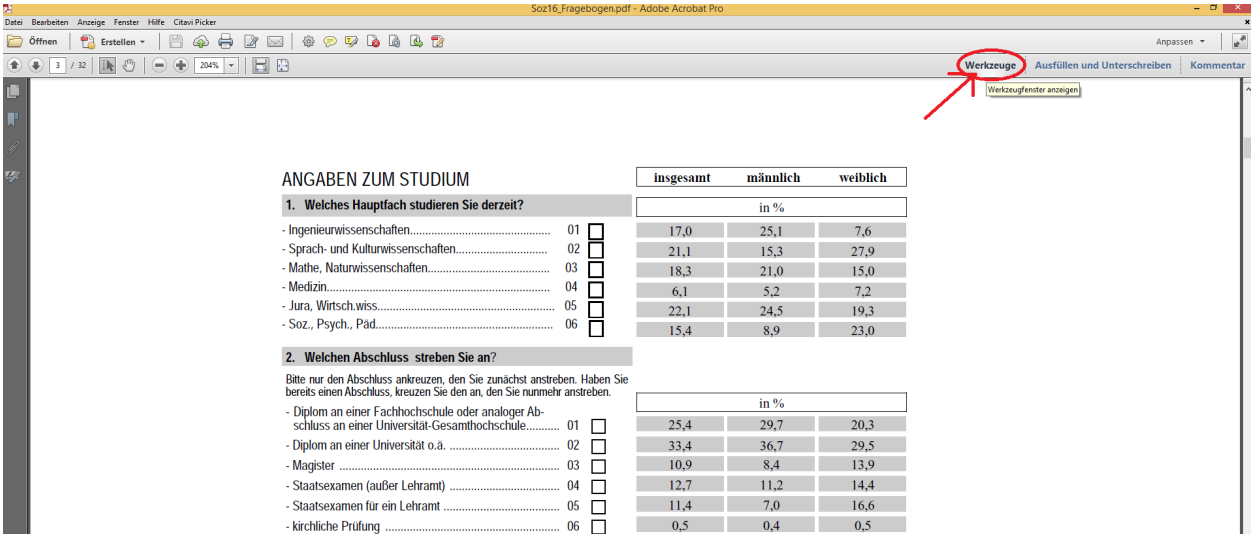
- Schritt 1: Öffnen Sie die Datei mit dem Adobe Acrobat XI Pro.
- Schritt 2.1.1: Es erscheint eine obere Leiste in der rechts der Begriff „Werkzeuge“ zu finden ist. Wählen Sie diesen Befehl durch einen Linksklick der Maus aus (siehe Abb. 1.1).

Abb. 1.1

Nun sollte sich ein weiterer Spaltenblock öffnen (siehe Abb. 1.2).

Abb. 1.2

An dieser Stelle klicken Sie auf „Seiten“. Auch hier eröffnet sich ein weiterer Spaltenblock mit ergänzenden Werkzeugen. Unter der Rubrik „Seiten manipulieren“ finden Sie die Instrumente „Zuschneiden“ sowie „Extrahieren“. Diese sind für unser weiteres Vorgehen wesentlich (siehe Abb. 1.3)


Soz16_Fragebogen.pdf - Adobe Acrobat Pro
Datei Bearbeiten Anzeige Fenster Hilfe Citavi Picker
Öffnen
Erstellen
Anpassen
Werkzeuge
Ausfüllen und Unterschreiben
Kommentar
Werkzeugfenster anzeigen
ANGABEN ZUM STUDIUM
insgesamt männlich weiblich
1. Welches Hauptfach studieren Sie derzeit?
in %
- Ingenieurwissenschaften 01 17,0 25,1 7,6
- Sprach- und Kulturwissenschaften 02 21,1 15,3 27,9
- Mathe, Naturwissenschaften 03 18,3 21,0 15,0
- Medizin 04 6,1 5,2 7,2
- Jura, Wirtsch.wiss. 05 22,1 24,5 19,3
- Soz., Psych., Päd. 06 15,4 8,9 23,0
2. Welchen Abschluss streben Sie an?
Bitte nur den Abschluss ankreuzen, den Sie zunächst anstreben. Haben Sie bereits einen Abschluss, kreuzen Sie den an, den Sie nunmehr anstreben.
in %
- Diplom an einer Fachhochschule oder analoger Abschluss an einer Universität-Gesamthochschule 01 25,4 29,7 20,3
- Diplom an einer Universität o.ä. 02 33,4 36,7 29,5
- Magister 03 10,9 8,4 13,9
- Staatsexamen (außer Lehramt) 04 12,7 11,2 14,4
- Staatsexamen für ein Lehramt 05 11,4 7,0 16,6
- kirchliche Prüfung 06 0,5 0,4 0,5

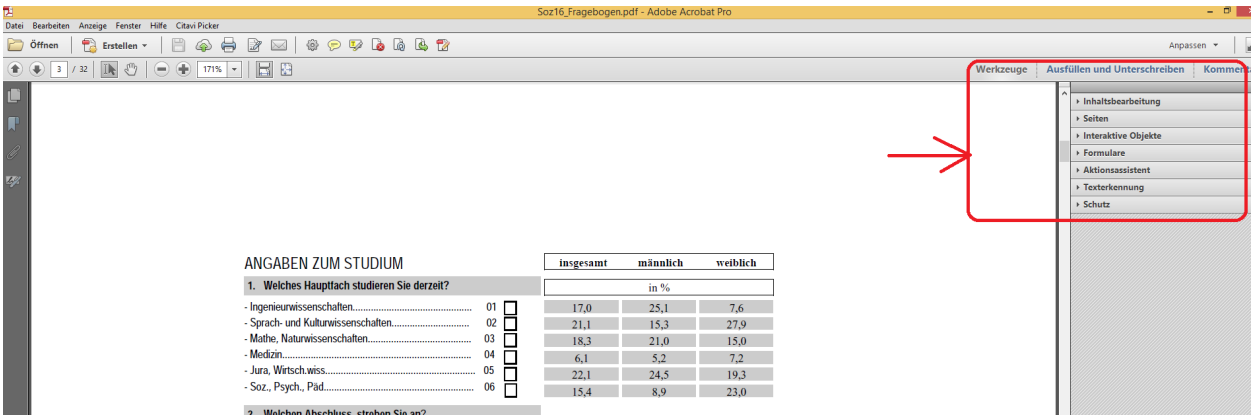
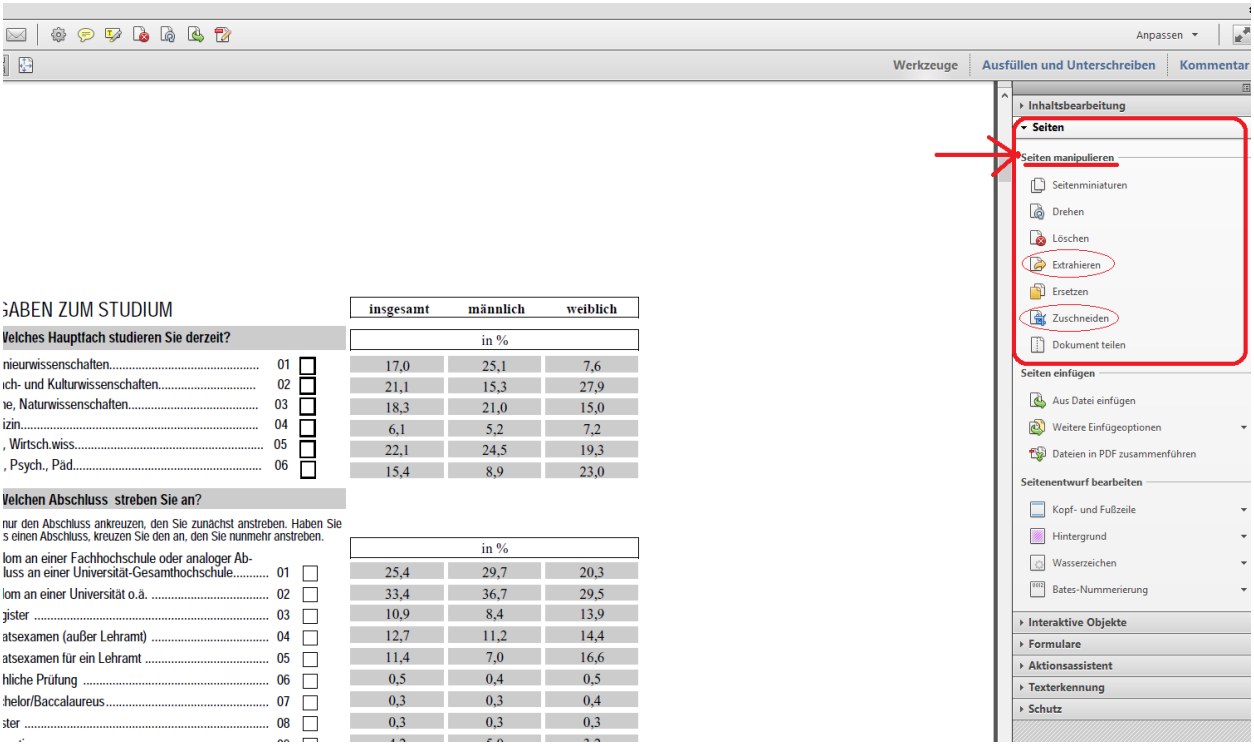
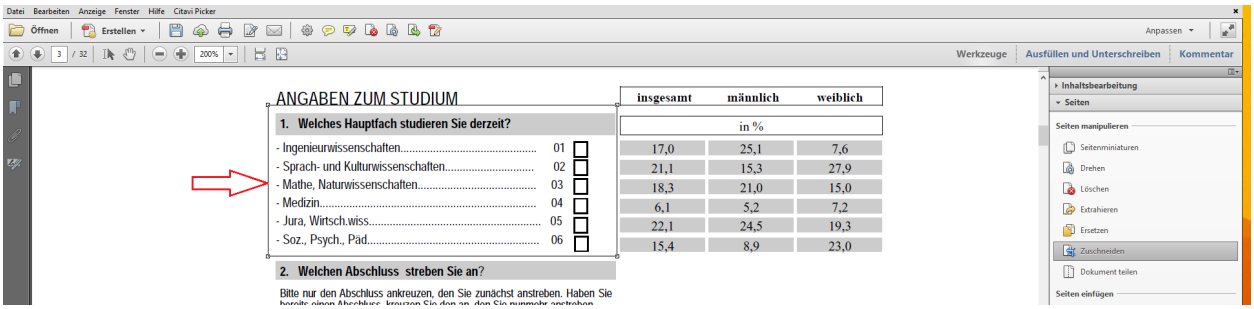

Soz16_Fragebogen.pdf - Adobe Acrobat Pro
Datei Bearbeiten Anzeige Fenster Hilfe Citavi Picker
Öffnen
Erstellen
Anpassen
Werkzeuge
Ausfüllen und Unterschreiben
Inhaltsbearbeitung
Seiten
Interaktive Objekte
Formulare
Aktionsassistent
Texterkennung
Schutz
ANGABEN ZUM STUDIUM
insgesamt männlich weiblich
1. Welches Hauptfach studieren Sie derzeit?
in %
- Ingenieurwissenschaften 01 17,0 25,1 7,6
- Sprach- und Kulturwissenschaften 02 21,1 15,3 27,9
- Mathe, Naturwissenschaften 03 18,3 21,0 15,0
- Medizin 04 6,1 5,2 7,2
- Jura, Wirtsch.wiss. 05 22,1 24,5 19,3
- Soz., Psych., Päd. 06 15,4 8,9 23,0

Abb. 1.3



Um nun mit dem Zu- und Ausschneiden der einzelnen Fragen zu beginnen, wählen Sie den Befehl „Zuschneiden“ aus. Klicken sie durch Betätigen der linken Maustaste links oberhalb der Frage, die es auszuschneiden gilt und markieren Sie durch stetiges Festhalten der linken Maustaste alle relevanten Elemente der Frage. Dabei erscheint ein schwarz umrandetes Viereck. In diesem müssen alle Bestandteile der Frage enthalten sein!(siehe Abb. 1.4)

Abb. 1.4



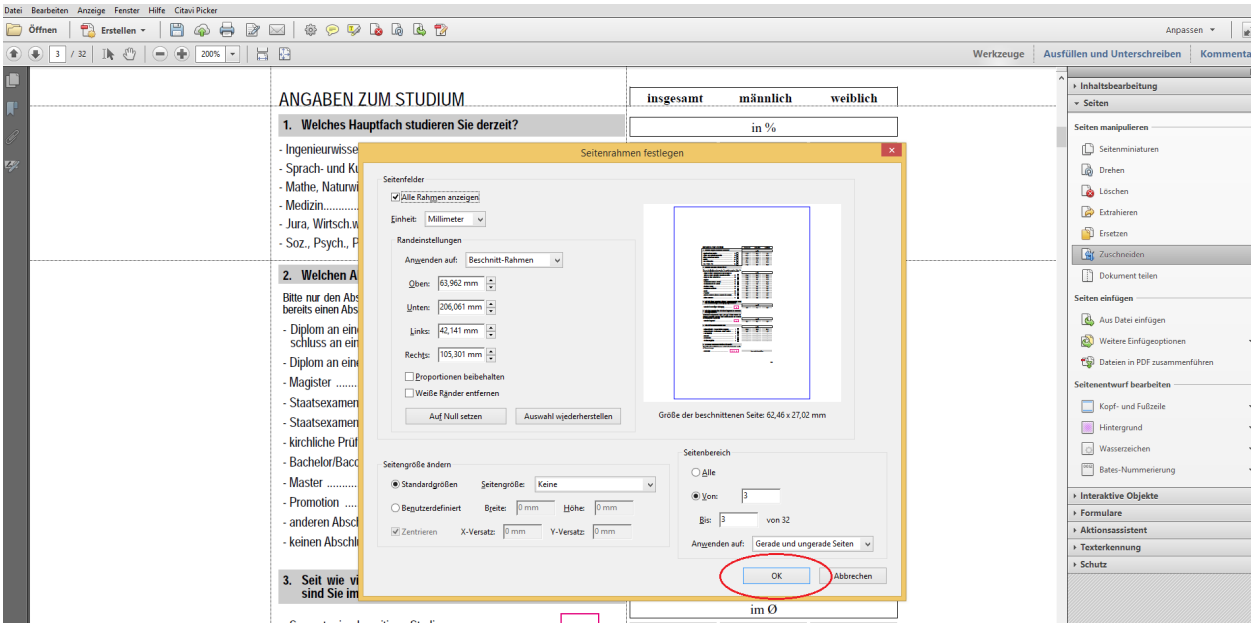
Anschließend betätigen Sie die Enter Taste ihrer Tastatur. Dabei eröffnet sich eine Maske (siehe Abb. 1.5) welche Sie mit „OK“ bestätigen.

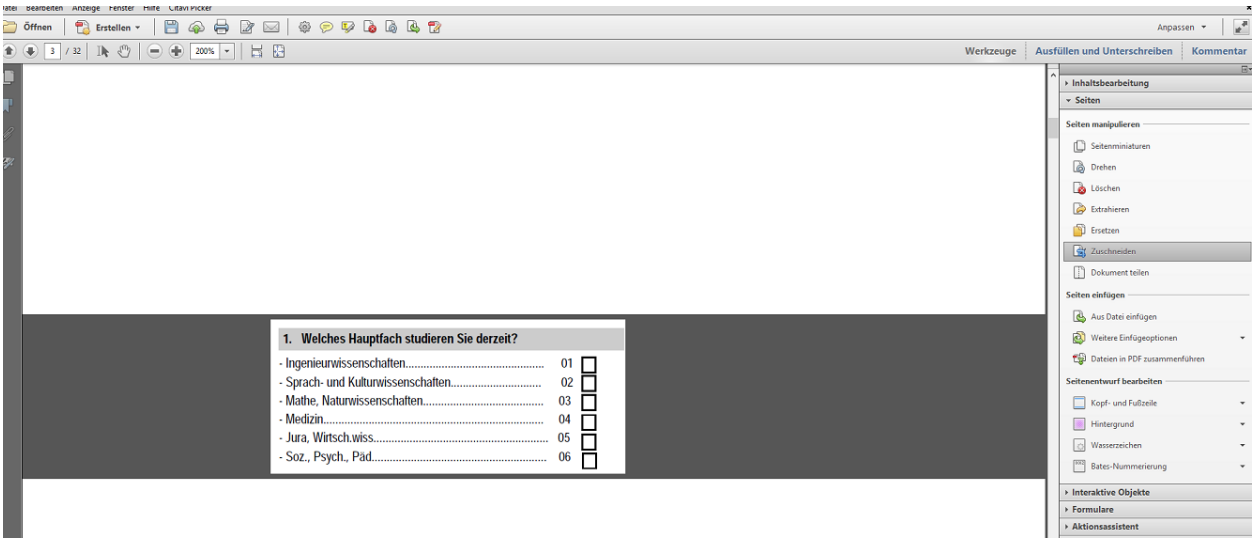
Abb. 1.5

Nachdem Sie die Maske durch das Bestätigen des „OK“ Buttons geschlossen haben, sollte auf ihrem Bildschirm nun eine ähnliche Darstellung wie in Abb. 1.6 erscheinen.

Abb. 1.6

Hinweis: Dies ist der einfachste Fall der in Schritt 2.1 beschrieben wird. Es kann aber durchaus vorkommen, dass eine Frage sich über zwei Seiten erstreckt, was das Ausschneiden der Frage erschwert. Hierbei ist erst einmal das


Datei Bearbeiten Anzeige Fenster Hilfe Citavi Picker
Öffnen
Erstellen
Anpassen
Werkzeuge
Ausfüllen und Unterschreiben
Kommentar
ANGABEN ZUM STUDIUM
insgesamt
männlich
weiblich
in %
1. Welches Hauptfach studieren Sie derzeit?
Seitenrahmen festlegen
Seitenfelder
Alle Rahmen anzeigen
Einheit: Millimeter
Randeinstellungen
Anwenden auf: Beschnitt-Rahmen
Oben: 63,962 mm
Unten: 206,061 mm
Links: 42,141 mm
Rechts: 105,301 mm
Proportionen beibehalten
Weiße Ränder entfernen
Auf Null setzen
Auswahl wiederherstellen
Größe der beschnittenen Seite: 62,46 x 27,02 mm
Seitengröße ändern
Standardgrößen
Seitengröße: Keine
Benutzerdefiniert
Breite: 0 mm
Höhe: 0 mm
Zentrieren
X-Versatz: 0 mm
Y-Versatz: 0 mm
Seitenbereich
Alle
Von: 3
Bis: 3
von 32
Anwenden auf: Gerade und ungerade Seiten
OK
Abbrechen
Inhaltsbearbeitung
Seiten
Seiten manipulieren
Seitenminiaturen
Drehen
Löschen
Extrahieren
Ersetzen
Zuschneiden
Dokument teilen
Seiten einfügen
Aus Datei einfügen
Weitere Einfügeoptionen
Dateien in PDF zusammenführen
Seitenentwurf bearbeiten
Kopf- und Fußzeile
Hintergrund
Wasserzeichen
Bates-Nummerierung
Interaktive Objekte
Formulare
Aktionsassistent
Texterkennung
Schutz

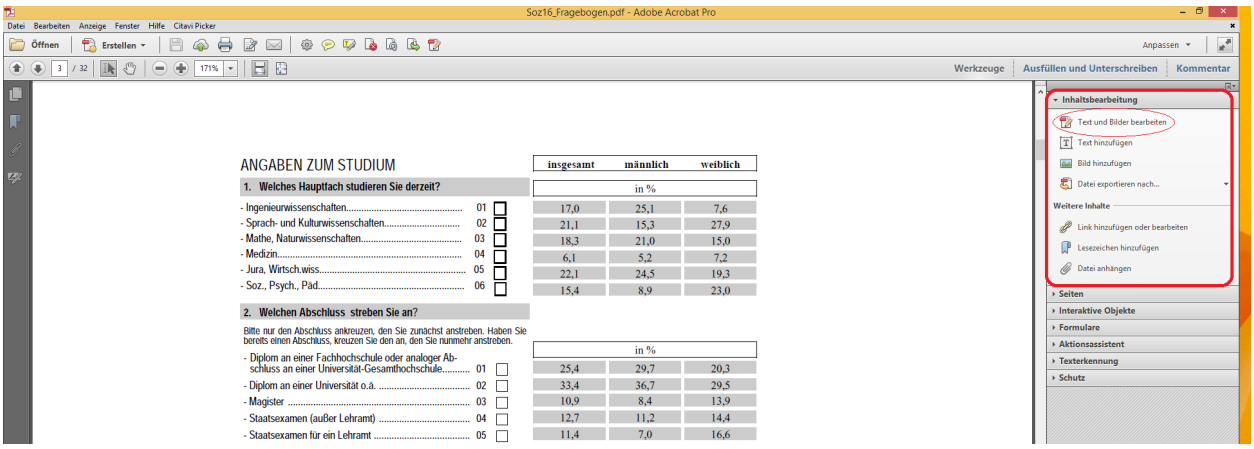

Öffnen
Erstellen
Anpassen
Werkzeuge
Ausfüllen und Unterschreiben
Kommentar
1. Welches Hauptfach studieren Sie derzeit?
- Ingenieurwissenschaften 01
- Sprach- und Kulturwissenschaften 02
- Mathe, Naturwissenschaften 03
- Medizin 04
- Jura, Wirtsch.wiss. 05
- Soz., Psych., Päd. 06
Inhaltsbearbeitung
Seiten
Seiten manipulieren
Seitenminiaturen
Drehen
Löschen
Extrahieren
Ersetzen
Zuschneiden
Dokument teilen
Seiten einfügen
Aus Datei einfügen
Weitere Einfügeoptionen
Dateien in PDF zusammenführen
Seitenentwurf bearbeiten
Kopf- und Fußzeile
Hintergrund
Wasserzeichen
Bates-Nummerierung
Interaktive Objekte
Formulare
Aktionsassistent

Zusammenführen der Frage auf einer gemeinsamen Seite notwendig, bevor man mit dem Zuschneiden beginnen kann. Wie dies zum Beispiel aussehen kann ist in Abb. 1.7 zu sehen.

Abb. 1.7 .. figure:: ./_static/pdf_extraction_1_7.png

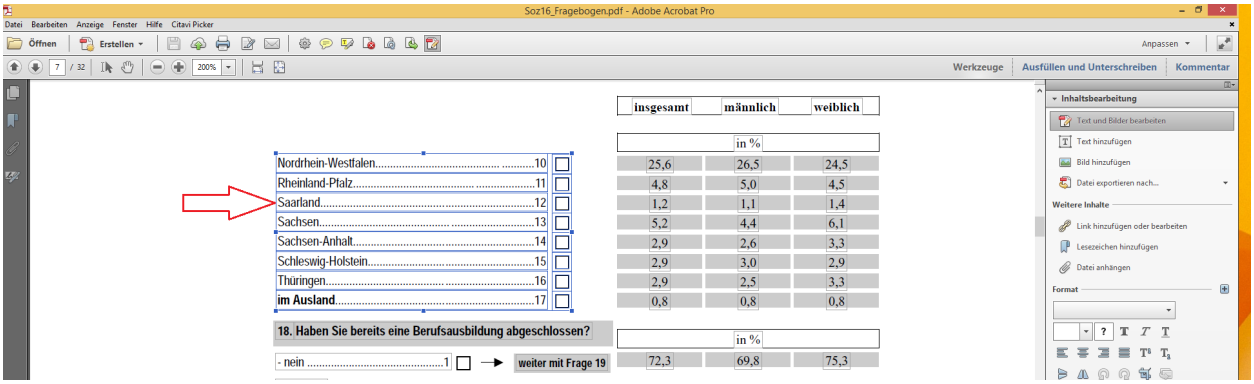
- Schritt 2.1.2 Liegt diese Ausgangssituation vor, so ist es notwendig Werkzeuge unter dem Begriff „Inhaltsbearbeitung“ zu verwenden. Dies ist ebenfalls in der rechten Leiste unter Werkzeuge zu finden (Siehe Abb. 1.8)

Abb. 1.8



In diesem Zusammenhang ist lediglich der Befehl „Text und Bilder bearbeiten“ relevant. Äquivalent zum Vorgang des Zuschneidens, klicken wir auch hier per Linksklick neben die auszuschneidenden Elemente und ziehen durch Halten der linken Maustaste ein Viereck um diese. Alle ausgewählten Elemente müssen nun blau umrandet sein (siehe Abb. 1.9).

Abb. 1.9



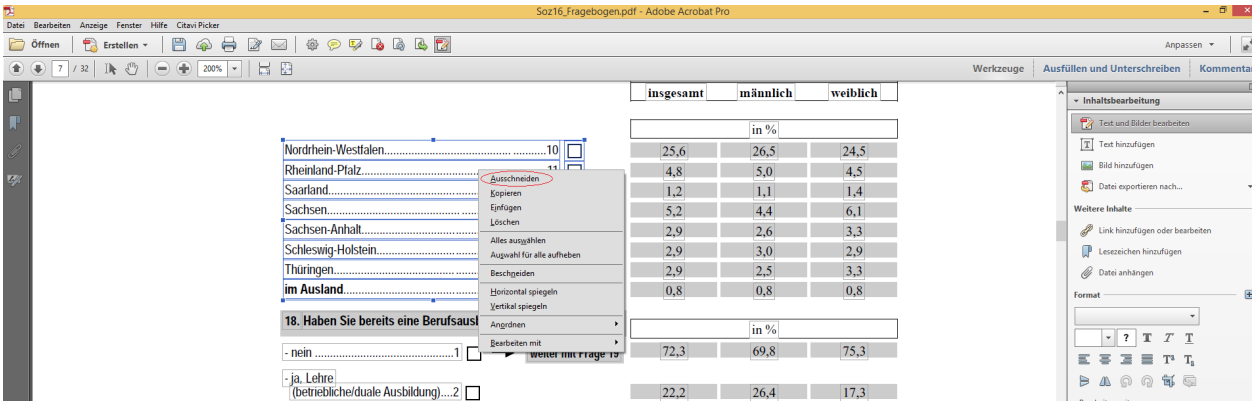
Als nächstes klickt man mittels der rechten Maustaste in das Viereck und wählt schließlich „ausschneiden“ aus (siehe Abb. 1.10).

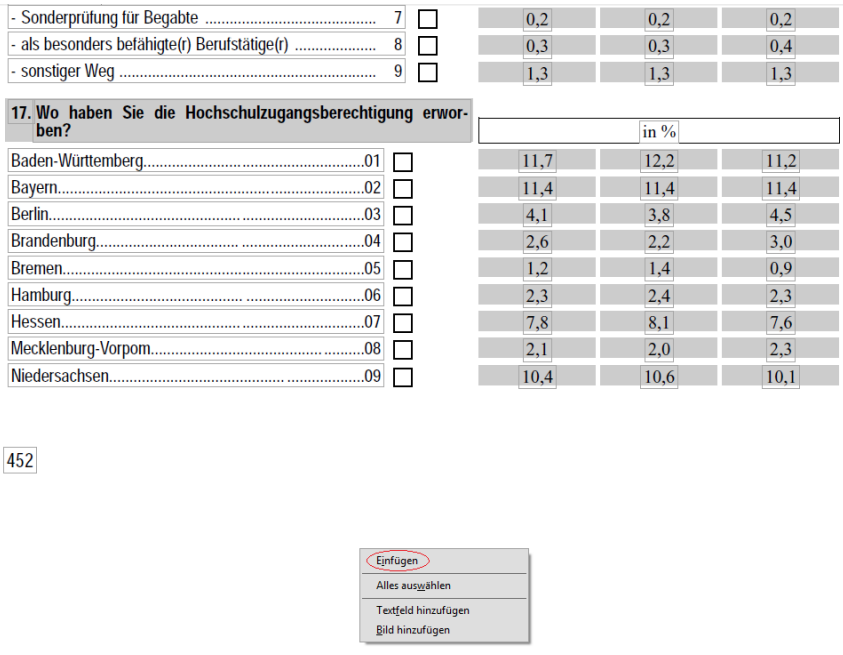
Abb. 1.10

Um die Frage nun auf einer Seite zusammenzuführen, klicken Sie nun mit der rechten Maustaste auf die Seite, auf der die soeben ausgeschnittenen Elemente hinzugefügt werden sollen. Dabei eröffnet sich ein Spaltenblock in dem Sie bitte „einfügen“ auswählen (siehe Abb. 1.11).

Abb. 1.11

Schließlich können Sie die Frageelemente so verschieben, dass die Frage vollständig auf einer gemeinsamen Seite vorliegt. Verschieben Sie die Frageelemente, indem Sie den Mauszeiger auf den äußersten blauen Rand bewegen, wodurch dann ein Kreuz aufzeigen müsste, welches an jedem Ende mit Pfeilen versehen ist. Klicken sie mittels





- Sonderprüfung für Begabte	7	0,2	0,2	0,2
- als besonders befähigte(r) Berufstätige(r)	8	0,3	0,3	0,4
- sonstiger Weg	9	1,3	1,3	1,3
17. Wo haben Sie die Hochschulzugangsberechtigung erworben?		in %		
Baden-Württemberg	01	11,7	12,2	11,2
Bayern	02	11,4	11,4	11,4
Berlin	03	4,1	3,8	4,5
Brandenburg	04	2,6	2,2	3,0
Bremen	05	1,2	1,4	0,9
Hamburg	06	2,3	2,4	2,3
Hessen	07	7,8	8,1	7,6
Mecklenburg-Vorpom	08	2,1	2,0	2,3
Niedersachsen	09	10,4	10,6	10,1

452

der linken Maustaste darauf und verschieben Sie die Elemente unter stetigem Halten der linken Maustaste an den gewünschten Ort. Es ist möglich, dass vorher allerdings noch weitere Elemente auf der Seite zu entfernen sind, da es sonst zu Überschneidungen und damit zu Unleserlichkeiten kommen könnte (siehe Abb. 1.12/ 1.13)

Abb. 1.12

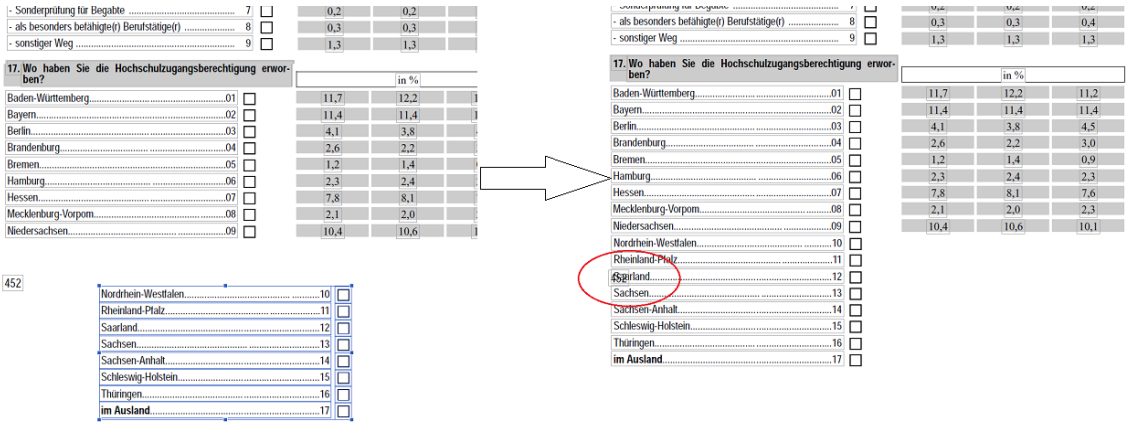
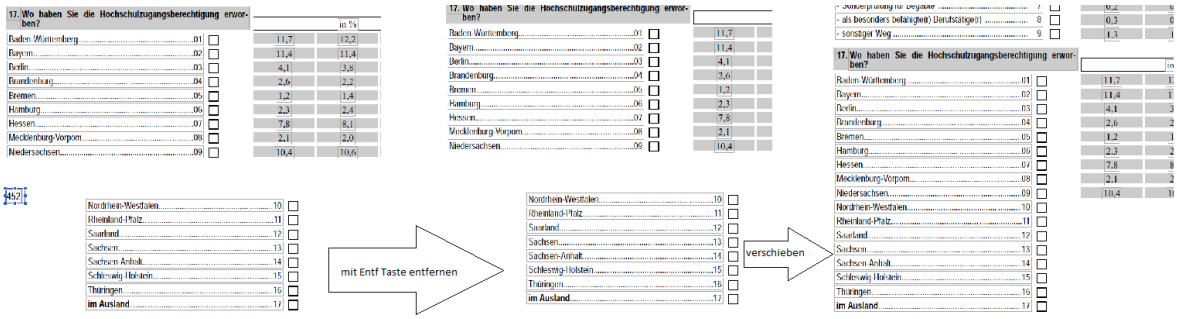


Abb. 1.13



In Abbildung 1.12 und 1.13 sehen Sie eine beispielhafte Vorgehensweise. In Abbildung 1.13 wurde die Zahl 452 entfernt, indem es ebenfalls mittels des Instruments „Texte und Bilder bearbeiten“ markiert wurde und durch anschließendes Drücken der „Entf“ Taste der Tastatur gelöscht worden ist.

Im Anschluss daran können Sie wie in Schritt 2.1.1 dargestllet mit dem Zuschneiden der Frage fortfahren.

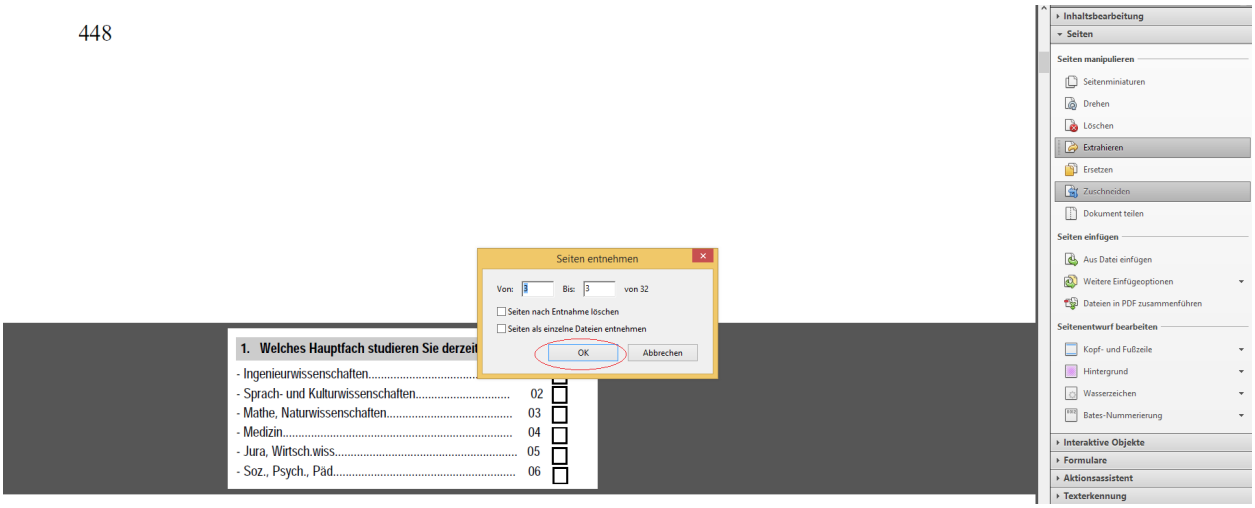
- Schritt 2.2: Nun können wir die ausgeschnittene Frage transportieren. Dafür wählen Sie das Instrument „Extrahieren“ aus. Wie bereits in Abb. 1.3 dargestellt, befindet sich dieses Werkzeug ebenfalls unter der Kategorie „Seiten manipulieren“. Dabei öffnet sich erneut eine Maske, in der Sie die zu entnehmende Seite auswählen sollen. Hierbei ist meist, die zuvor zugeschnittene Seite vorausgewählt. Überprüfen Sie es gegebenenfalls noch einmal und Bestätigen Sie dann durch Klicken auf den „OK“ Button. Dies ist in Abb. 1.14 dargestellt.

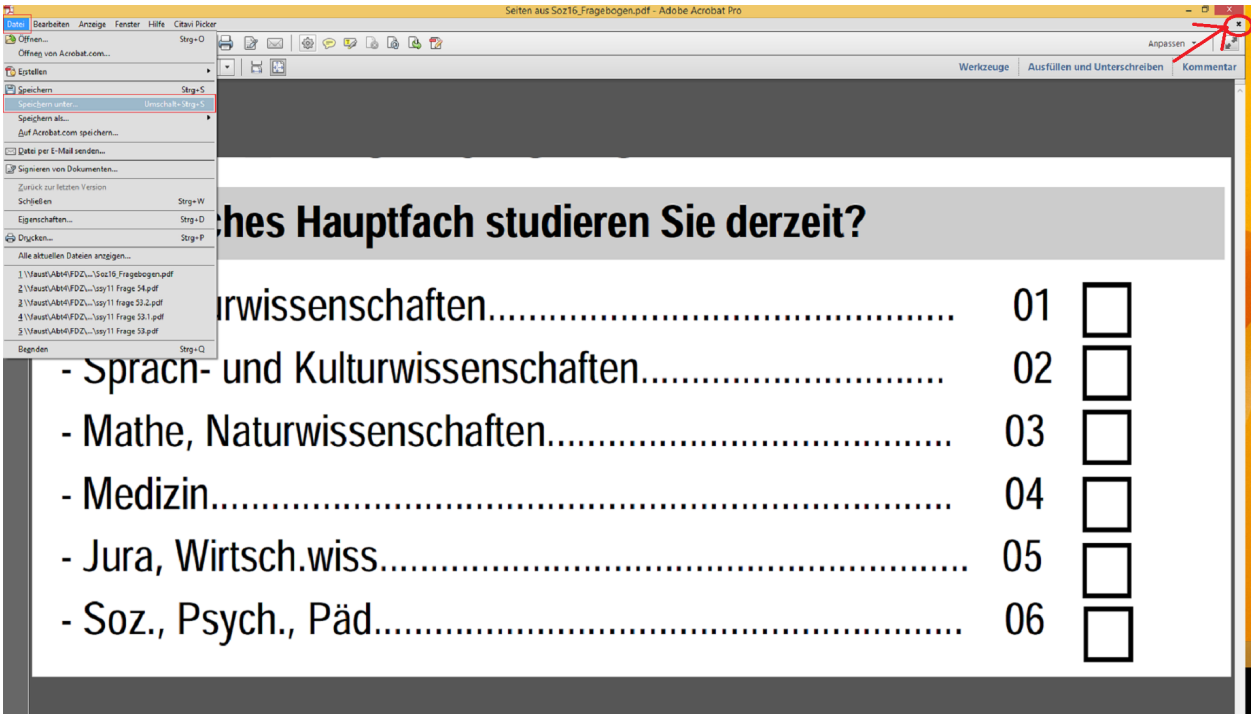
Abb. 1.14

Anschließend sollte auf Ihrem Bildschirm ein ähnliches Bild erscheinen, wie in Abb. 1.15 abgebildet.

Abb. 1.15

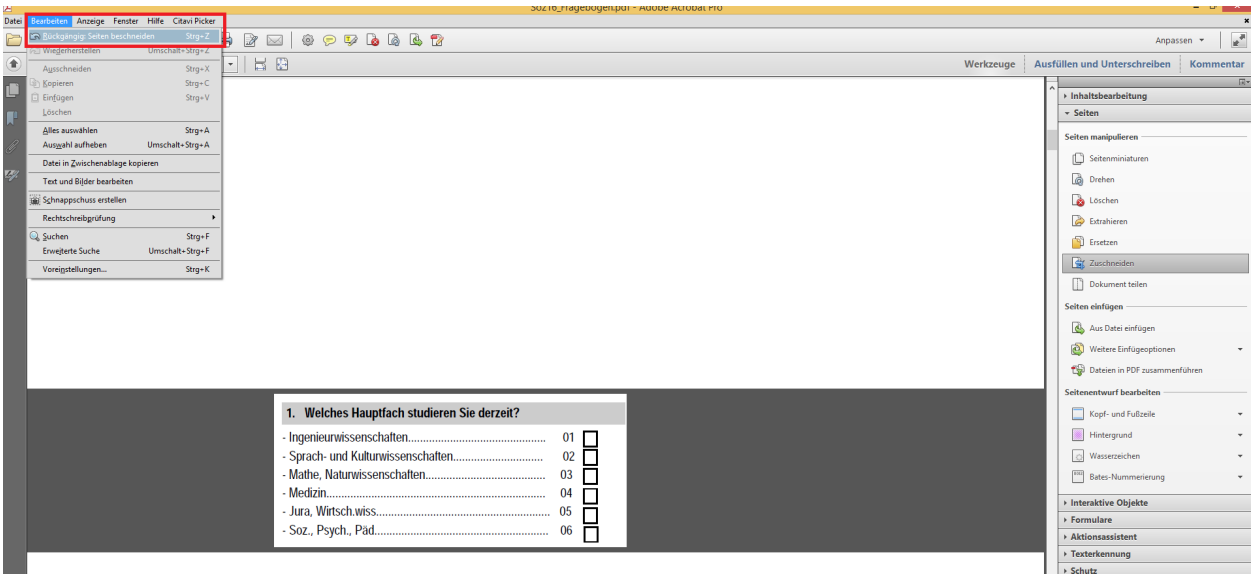
Um das Transportieren nun vollständig abzuschließen klicken Sie oben rechts auf Datei und anschließend auf „Speichern unter“. Wichtig ist hierbei, dass der ausgewählte Dateiname, unter dem die neue Datei gespeichert werden soll der Fragenummer entspricht. (eventuell mit der Excel Tabelle abgleichen)


448
Seiten entnehmen
Von: 3 Bis: 3 von 32
Seiten nach Entnahme löschen
Seiten als einzelne Dateien entnehmen
OK
Abbrechen
1. Welches Hauptfach studieren Sie derzeit
- Ingenieurwissenschaften
- Sprach- und Kulturwissenschaften 02
- Mathe, Naturwissenschaften 03
- Medizin 04
- Jura, Wirtsch.wiss 05
- Soz., Psych., Päd 06
Inhaltsbearbeitung
Seiten
Seiten manipulieren
Seitenminiaturen
Drehen
Löschen
Extrahieren
Ersetzen
Zuschneiden
Dokument teilen
Seiten einfügen
Aus Datei einfügen
Weitere Einfügeoptionen
Dateien in PDF zusammenführen
Seitenentwurf bearbeiten
Kopf- und Fußzeile
Hintergrund
Wasserzeichen
Bates-Nummerierung
Interaktive Objekte
Formulare
Aktionsassistent
Texterkennung


Seiten aus Soz16_Fragebogen.pdf - Adobe Acrobat Pro
Datei Bearbeiten Anzeige Fenster Hilfe Citavi Picker
Öffnen... Strg+O
Öffnen von Acrobat.com...
Erstellen
Speichern Strg+S
Speichern unter... Umschalt+Strg+S
Speichern als...
Auf Acrobat.com speichern...
Datei per E-Mail senden...
Signieren von Dokumenten...
Zurück zur letzten Version
Schließen Strg+W
Eigenschaften... Strg+D
Drucken... Strg+P
Alle aktuellen Dateien anzeigen...
Beenden Strg+Q
Anpassen
Werkzeuge
Ausfüllen und Unterschreiben
Kommentar
hes Hauptfach studieren Sie derzeit?
rwissenschaften 01
- Sprach- und Kulturwissenschaften 02
- Mathe, Naturwissenschaften 03
- Medizin 04
- Jura, Wirtsch.wiss 05
- Soz., Psych., Päd 06

- Schritt 2.3: Um nun fortzufahren und weitere Fragen als pdf Datei zu extrahieren, klicken Sie auf das kleine Kreuz (wie in Abb. 1.15 gekennzeichnet). Dabei wird das Fenster geschlossen. Damit die ursprüngliche Datei wieder als solche vorliegt, wie es vor dem Beschneiden der Seiten der Fall war, wählen Sie nun in der oberen Leiste erst „Bearbeiten“ aus und anschließend „Rückgängig: Seiten beschneiden“ (siehe Abb. 1.16). Dieser Schritt ist vor allem empfehlenswert, wenn sich mehrere Fragen auf einer Seite befinden. Hierdurch wird das stetige Wiederöffnen der Datei vermieden und es wirkt sich zeitsparend aus.

Abb. 1.16



- Schritt 3: Der letzte Schritt beinhaltet das Konvertieren der extrahierten Fragen vom pdf Format ins png Format. Hierfür können Sie den Online Converter „pdf2png“ nutzen. https://online2pdf.com/de/pdf-zu-png-konvertieren

ACHTUNG: Hier ist nur ein begrenztes Volumen an Konvertierungen pro Tag von 150MB möglich.

3.4 Bilderfassung aus RagTime

3.4.1 Vorraussetzungen

Fragebögen, die als eine Ragtime Datei vorliegen sollen gemäß der nächsten Schritte bearbeitet werden.

Liegt zu dem Fragebogen, den es von Ihnen zu bearbeiten gilt, eine äquivalente Excel Datei vor, so wäre es ratsam diese als Grundlage für die weitere Arbeit zu verwenden. Hierbei ist es notwendig, dass jeder aufgeführten Frage in der Excel Datei eine finale Bilddatei gegenüber steht.

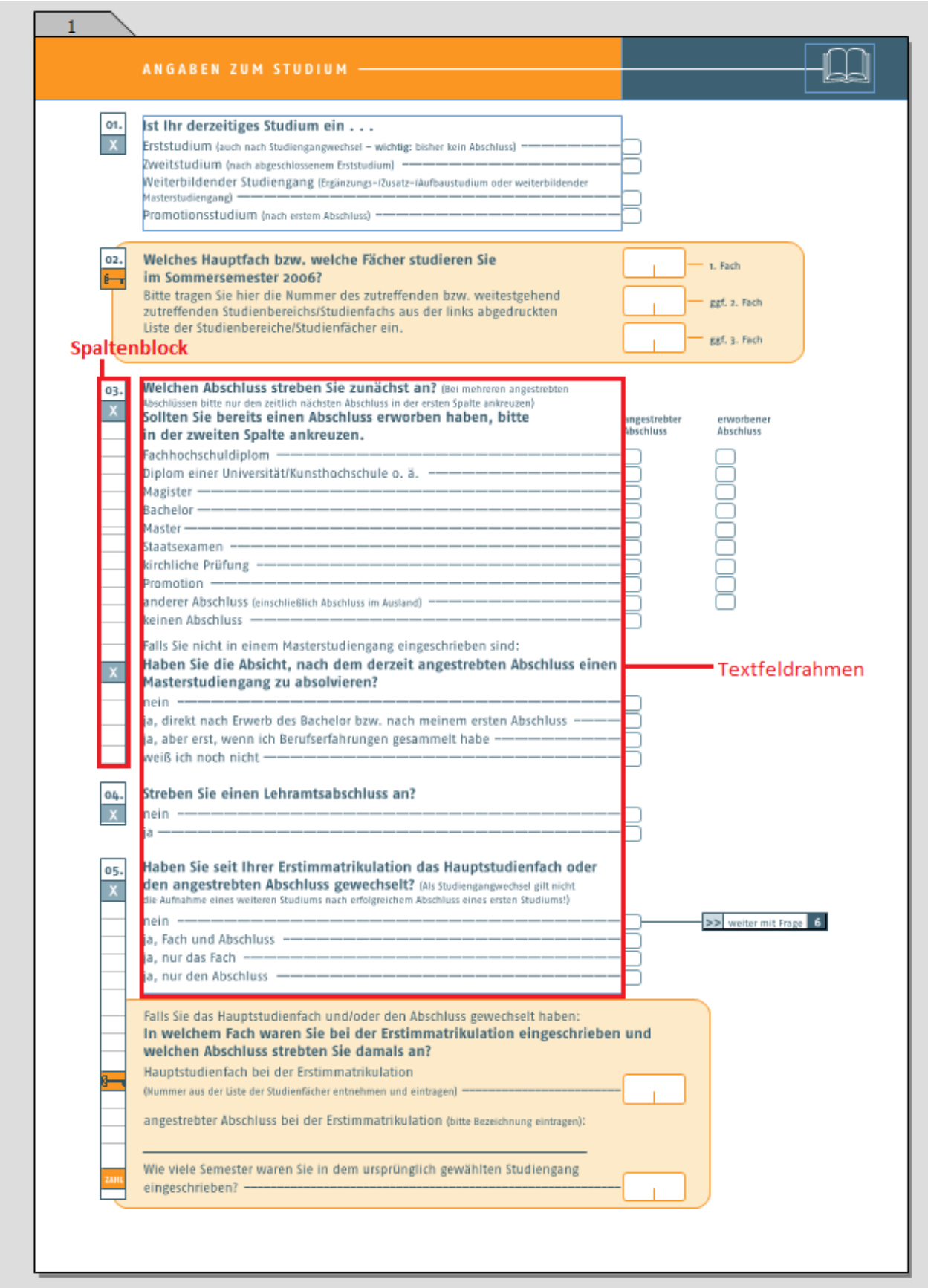
Schritt 1:

Zuerst öffnen Sie die zu bearbeitende Datei mit Ragtime.

Schritt 2:

Schritt 3.0: Markieren von Elementen in Ragtime

Wählen Sie nun die Elemente aus, die es als Bilddatei zu exportieren gilt. Markieren Sie die ausgewählten Inhalte, indem Sie den Mauszeiger vom Beginn des zu markierenden Feldes unter Festhalten der linken Maustaste bis zum Ende des gewünschten Feldes bewegen (ziehen). Während dieses Vorgangs bildet sich ein schwarzes Rechteck, in welchem


1
ANGABEN ZUM STUDIUM
01.
X
Ist Ihr derzeitiges Studium ein . . .
Erststudium (auch nach Studiengangwechsel – wichtig: bisher kein Abschluss)
Zweitstudium (nach abgeschlossenem Erststudium)
Weiterbildender Studiengang (Ergänzungs-/Zusatz-/Aufbaustudium oder weiterbildender Masterstudiengang)
Promotionsstudium (nach erstem Abschluss)
02.
Welches Hauptfach bzw. welche Fächer studieren Sie im Sommersemester 2006?
Bitte tragen Sie hier die Nummer des zutreffenden bzw. weitestgehend zutreffenden Studienbereichs/Studienfachs aus der links abgedruckten Liste der Studienbereiche/Studienfächer ein.
1. Fach
ggf. 2. Fach
ggf. 3. Fach
Spaltenblock
03.
X
Welchen Abschluss streben Sie zunächst an? (Bei mehreren angestrebten Abschlüssen bitte nur den zeitlich nächsten Abschluss in der ersten Spalte ankreuzen)
Sollten Sie bereits einen Abschluss erworben haben, bitte in der zweiten Spalte ankreuzen.
angestrebter Abschluss
erworbener Abschluss
Fachhochschuldiplom
Diplom einer Universität/Kunsthochschule o. ä.
Magister
Bachelor
Master
Staatsexamen
kirchliche Prüfung
Promotion
anderer Abschluss (einschließlich Abschluss im Ausland)
keinen Abschluss
Falls Sie nicht in einem Masterstudiengang eingeschrieben sind:
X
Haben Sie die Absicht, nach dem derzeit angestrebten Abschluss einen Masterstudiengang zu absolvieren?
Textfeldrahmen
nein
ja, direkt nach Erwerb des Bachelor bzw. nach meinem ersten Abschluss
ja, aber erst, wenn ich Berufserfahrungen gesammelt habe
weiß ich noch nicht
04.
X
Streben Sie einen Lehramtsabschluss an?
nein
ja
05.
X
Haben Sie seit Ihrer Erstimmatrikulation das Hauptstudienfach oder den angestrebten Abschluss gewechselt? (Als Studiengangwechsel gilt nicht die Aufnahme eines weiteren Studiums nach erfolgreichem Abschluss eines ersten Studiums!)
nein
>> weiter mit Frage 6
ja, Fach und Abschluss
ja, nur das Fach
ja, nur den Abschluss
Falls Sie das Hauptstudienfach und/oder den Abschluss gewechselt haben:
In welchem Fach waren Sie bei der Erstimmatrikulation eingeschrieben und welchen Abschluss strebten Sie damals an?
Hauptstudienfach bei der Erstimmatrikulation
(Nummer aus der Liste der Studienfächer entnehmen und eintragen)
angestrebter Abschluss bei der Erstimmatrikulation (bitte Bezeichnung eintragen):
ZAHL
Wie viele Semester waren Sie in dem ursprünglich gewählten Studiengang eingeschrieben?

alle zu markierenden Elemente vollständig enthalten sein müssen; Elemente außerhalb dieses Vierecks werden nicht markiert.

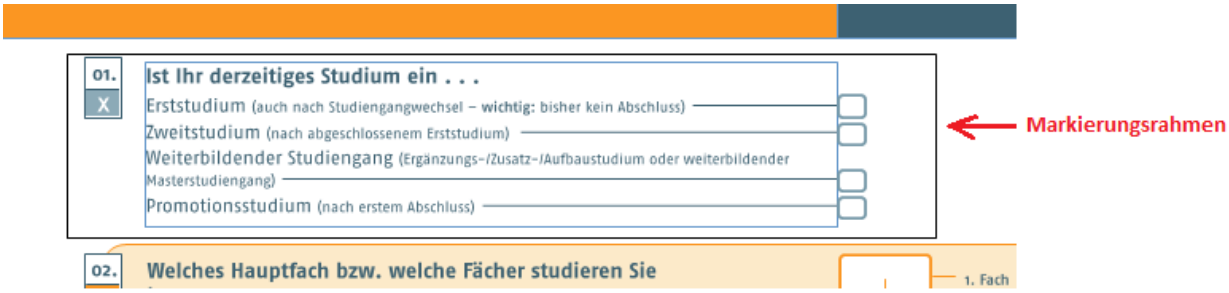


Abb. 3.4: Elemente markieren.

- Falls der Text nicht markiert wurde, da der Textfeldrahmen noch weitere Fragen beinhaltet fahren Sie fort mit Schritt 3.1 ff.
- Ist dies nicht der Fall; weiter mit Schritt 4

Schritt 3.1: Text in einem Textfeldelement bearbeiten

Um gewünschte Textelemente aus einem zu großen Textfeldrahmen auszuschneiden, muss eine Platzierung des Mauszeigers innerhalb des Textfeldes erfolgen. Durch einen darauffolgenden Rechtsklick öffnet sich ein Menü, in welchem Sie die Auswahlmöglichkeit „Komponente öffnen“ erwählen. Dadurch öffnet sich ein neues Fenster, in dem die Bearbeitung des gesamten Textes des ausgewählten Textfeldrahmens möglich ist. Nun können Sie den Text bearbeiten: Löschen Sie alle irrelevanten Textpassagen und schließen Sie das Fenster.

ACHTUNG: Beachten Sie, dass Sie das kleine, graue X drücken, um das Komponentenfenster zu schließen.

Abb. 3.5: Komponentenfenster schließen.

Schritt 3.2: Anpassen des Textfeldrahmens

Durch das Löschen irrelevanter Elemente kann sich der Text innerhalb des Textfeldrahmens verschieben. Deswegen muss der Textfeldrahmen in diesem Schritt angepasst werden: Durch Anklicken des Textfeldrahmens ist dies möglich. Bewegen Sie hierfür den Mauszeiger auf den mittleren schwarzen Markierungspunkt des unteren Randes des Textfeldrahmens. Dabei müsste sich der Mauszeiger zu einem Kreuz verändern. Durch Linksklick und anschließendes Festhalten können Sie den Rahmen sowohl nach unten als auch nach oben anpassen. Ziehen Sie den Rahmen bis zum Ende der letzten Textpassage des Textelementes hoch. Dadurch ist gewährleistet, dass der Text ebenfalls bei dem Exportieren markiert werden kann.

Schritt 3.3: Verschieben der zusätzlichen Elemente

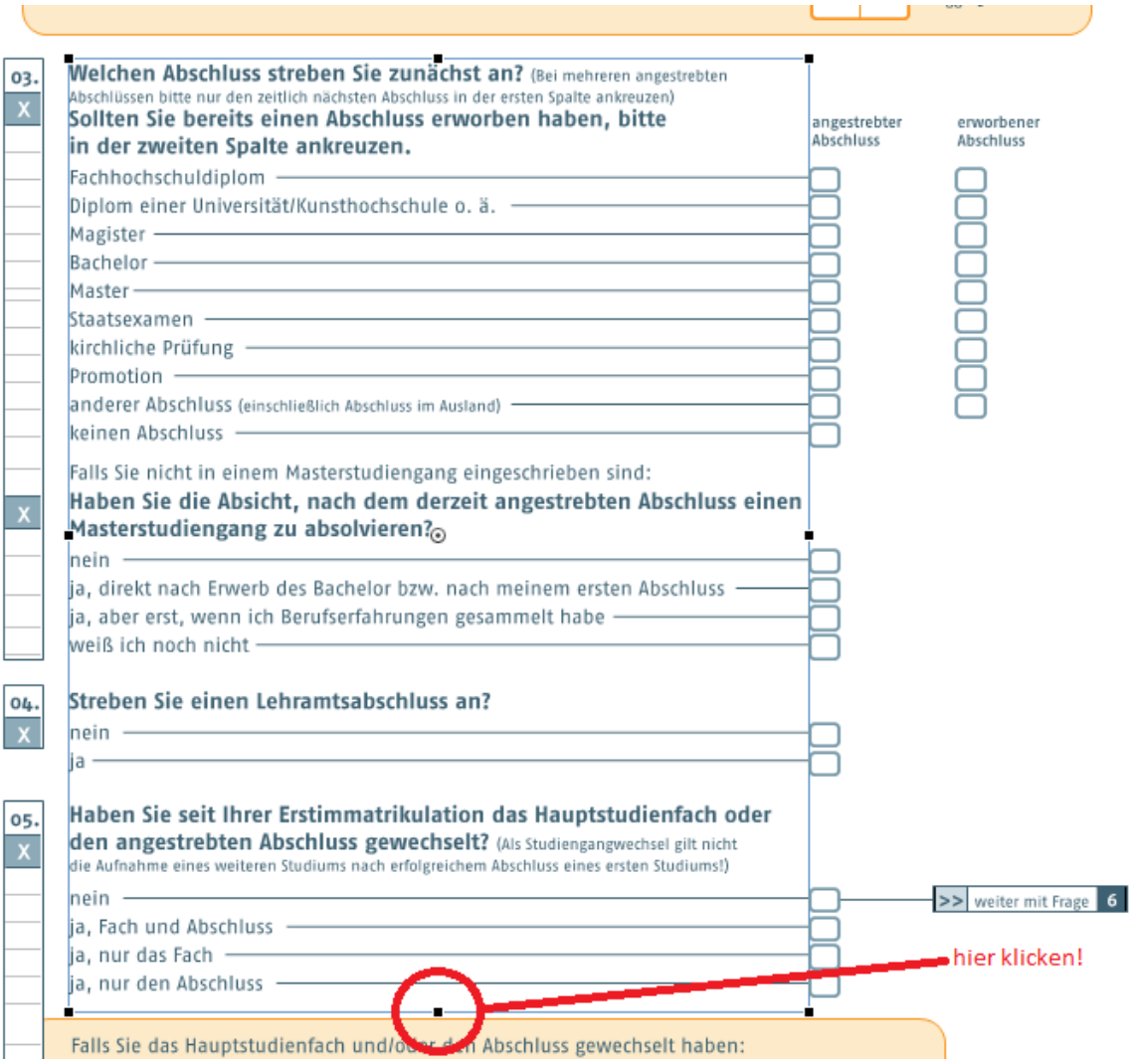


Abb. 3.6: Textfeldrahmen anpassen

Durch das mögliche Verschieben des Textes, müssen die dazugehörigen Elemente wie z.B. Antwortkästchen einer Frage etc. ebenfalls angepasst werden. Solche Elemente können wie in Schritt 3.0 beschrieben verschoben werden.

Schritt 3.4: Anpassen des Spaltenblocks

Es kann zudem der Fall eintreten, dass der Spaltenblock angepasst werden muss. Dazu wählt man diesen durch einen Klick aus. Dabei öffnet sich auf der linken Seite des Fensters eine graue, durchnummerierte Leiste (1). In dieser ist es möglich Elemente wie z.B. grau unterlegte Kreuze zu entfernen. Durch das Markieren einer Nummer wird das dazugehörige Kästchen des Spaltenblocks blau unterlegt (2) und man kann es durch Drücken der Entf - Taste löschen (3). Wie bei dem Textfeldrahmen, ist es bei dem Spaltenblock möglich die Ränder durch die schwarzen Punkte zu verschieben.

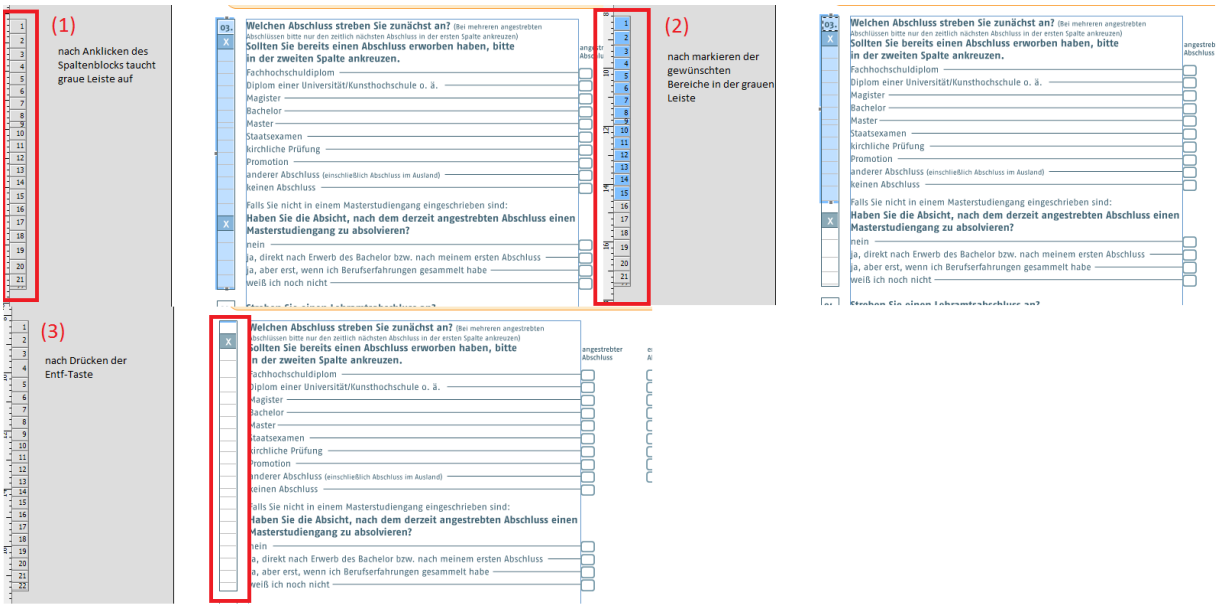


Abb. 3.7: Anpassen des Spaltenblocks

Schritt 3.5: Fertigstellung

Liegen nun alle gewünschten Elemente in der richtigen Anordnung, werden alle markiert und setzt mit Schritt 4 fort.

Schritt 4: Exportieren

Sind alle Elemente markiert (WICHTIG: überprüfen Sie, ob wirklich jedes Element von Markierungspunkten umrandet ist, ansonsten werden nicht markierte Elemente nicht exportiert), kann durch Betätigen der rechten Maustaste der markierte Bereich exportiert werden. Hierbei ist es wichtig darauf zu achten, dass der Mauszeiger auf einem der schwarzen Markierungspunkte liegt, da ansonsten die Markierung aufgehoben wird. Im weiteren Verlauf öffnet sich ein Menü, in welchem die Option „Exportieren. . . “ aufzufinden ist. Wählen Sie diese Option aus um die ausgewählten Elemente gesondert von der ursprünglichen Datei zu speichern.

Schritt 5: Speichern

Beim Speichern müssen Sie folgende Details beachten:

- Falls eine Excel- Datei vorhanden ist, achten Sie auf übereinstimmenden Dateinamen und der dazugehörigen Fragennummer (die in der Excel-Datei aufgeführt ist).
- Stellen sie sicher, dass der Dateityp „PostScript-Illustraition (EPSF) (*.eps)“ ausgewählt ist.
- Unter dem Punkt „Exportieren“ muss zudem die Option „Auswahl“ gewählt sein.

Schritt 6: Konvertieren von .eps zu .png

Die abgespeicherten .eps - Dateien müssen in einem weiteren Schritt in .png - Dateien konvertiert werden. Dies erfolgt am Besten mit dem ReaConverter7Pro. (dieser Konverter ist kostenpflichtig, daher handelt es sich hierbei um eine Testversion. Aus diesem Grund können hierbei lediglich fünf Dateien gleichzeitig konvertiert werden.)

- Um die Dateien zu konvertieren, zieht man die eps Dateien in das offene Feld indem man Dateien markiert, die rechte Maustaste festhält und Dateien mittels Bewegens des Mauszeigers schließlich rüberzieht.
- Wenn Sie ab einem bestimmten Punkt alle Objekte markieren wollen, dann klicken Sie mit der linken Maustaste zunächst auf die erste Datei / Ordner und halten Sie die Shift-Taste gedrückt, während Sie auf das letzte Objekt Ihrer Auswahl mit der linken Maustaste klicken.

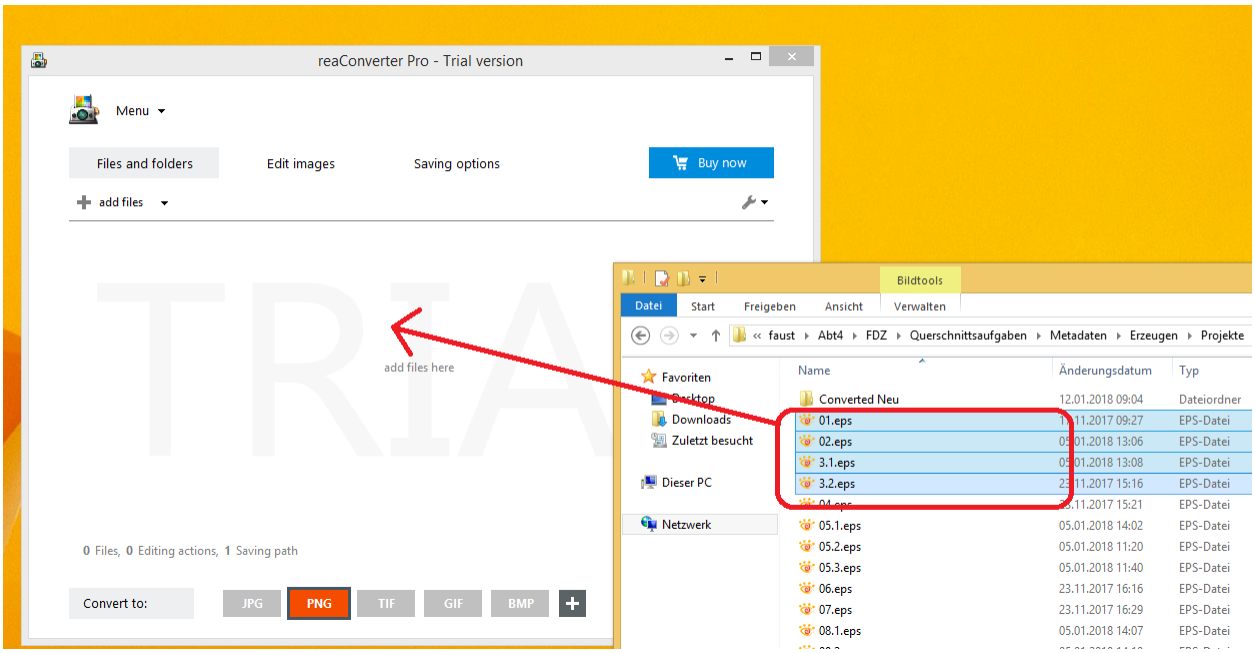


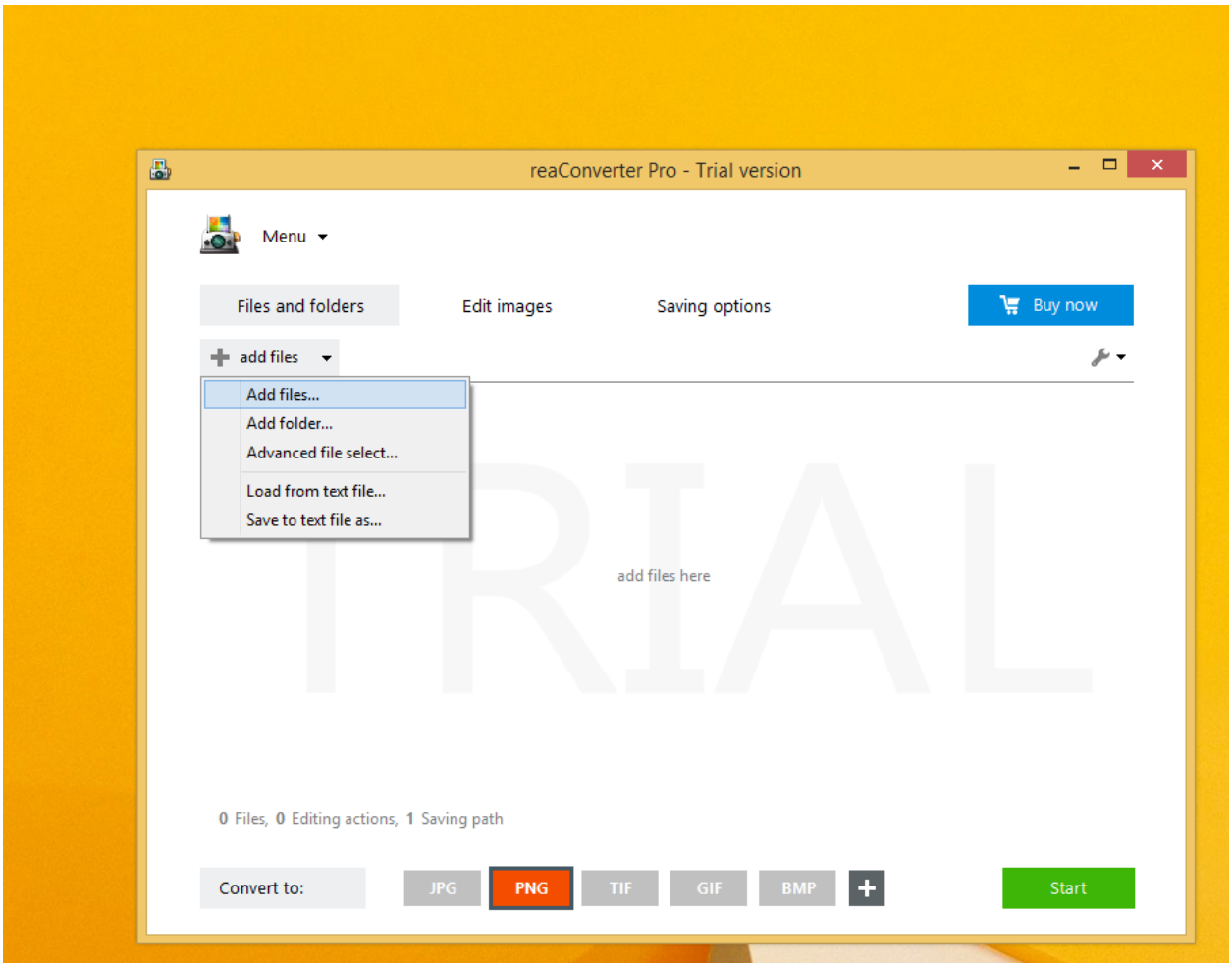
Abb. 3.8: Anpassen des Spaltenblocks

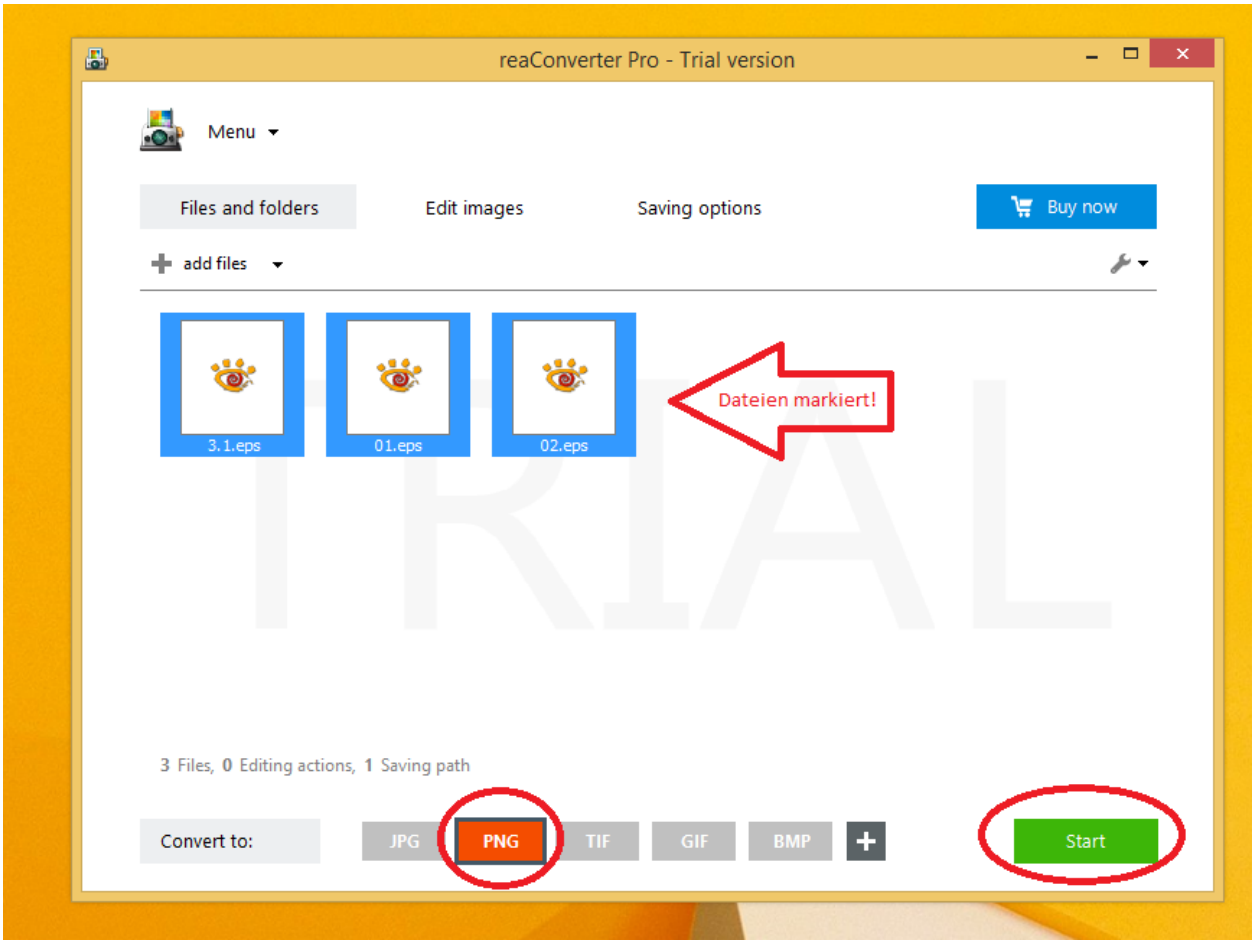
- Oder man klickt auf „add files“. Der Computer kann dann auf Dateien durchsucht werden und ausgewählte Dateien können hinzugefügt werden.
- Als nächstes wählt man unter Convert to: das gewünschte Format „PNG“ aus. Das ausgewählte Format sollte nun orange aufleuchten. Als letztes markiert man die zu konvertierenden Dateien und klickt unten rechts auf den grünen Button „Start“.

3.5 Excel Makros verwenden

Um Makros in Excel zu nutzen, wird der VBA Editor benötigt. Dieser wird mit Alt+F11 geöffnet. Über `Einfügen` > `Modul` wird ein neues Modul angelegt, in das die fertigen Skripte einfach reinkopiert werden können.

In der Excel Tabelle können die Makros über `Ansicht` > `Makros` > `Makros anzeigen` > `Ausführen` gestartet werden.


reaConverter Pro - Trial version
Menu
Files and folders
Edit images
Saving options
Buy now
add files
Add files...
Add folder...
Advanced file select...
Load from text file...
Save to text file as...
add files here
0 Files, 0 Editing actions, 1 Saving path
Convert to:
JPG
PNG
TIF
GIF
BMP
Start


reaConverter Pro - Trial version
Menu
Files and folders
Edit images
Saving options
Buy now
add files
3.1.eps
01.eps
02.eps
Dateien markiert!
TRIAL
3 Files, 0 Editing actions, 1 Saving path
Convert to:
JPG
PNG
TIF
GIF
BMP
Start

Name	Was kann es?	Wo wird es genutzt?
panelIdentifier	Generiert den PanelIdentifier durch den Vergleich von Variablennamen. Dabei dürfen Variablennamen verschiedene Versionierungen und/oder Zugangswege haben. Die Eingabe des Projektnamens, der Datensatznummer, der Spalte in die der panelIdentifier eingefügt werden soll und die Auswahl der Variablen erfolgt über eine Inputbox Beispiel: der panelIdentifier zu adem01_g1v1r und bdem01_g1v3r lautet z.B. abs2005-ds1-dem01_g1	panelIdentifier in variables.xlsx
addPrefix	Fügt einer oder mehreren Variablen in einer Zelle und durch Komma getrennt ein Präfix hinzu (z.B. abs2005). Der Bereich in der die Variablen stehen und das Projektkürzel werden über eine InputBox eingegeben Beispiel: bski01d_v1,bski01a_v1 bski01u_v1 wird zu abs2005-bski01d_v1,ab s2005-bski01a_v1,abs2 005-bski01u_v1	Panelvariablen in manMetadaten.xlsx

3.6 Editing Macros Excel

Die folgenden Makros können unterstützend nach nach dem manuellen Ausfüllen der Exceltabelle vimport_ds**Nr**.xlsx genutzt werden. Vor dem Ausführen der Makros sollte sichergestellt werden, dass die Tabellenblätter der Exceltabelle richtig benannt wurden (variables und relatedQuestions) und die für das jeweilige Makro notwendigen Spalten vorhanden sind.

3.6.1 Wie werden Makros ausgeführt?

Um Makros in Excel zu nutzen, wird der VBA Editor benötigt. Dieser wird mit Alt+F11 geöffnet. Über `Einfügen > Modul` wird ein neues Modul angelegt, in das die fertigen Skripte einfach reinkopiert werden können. In der Excel Tabelle können die Makros über `Ansicht > Makros > Makros anzeigen > Ausführen` ausgeführt werden.

3.6.2 Makro-Übersicht

panelIdentifier

Wo wird es genutzt? panelIdentifier in vimport.xlsx

- generiert den panelIdentifier durch den Vergleich vom Variablenstamm (ohne-Präfix)
- dabei dürfen Variablennamen verschiedene Versionierungen und/oder Zugangswege-haben
- das FDZ Variablenschema muss verwendet worden sein (Variablenname z.B.-astu01_g1v1r mit den möglichen Zugangswegen c,d,o,r,a)
- im Tabellenblatt variables müssen die beiden Spaltennamen name und-panelIdentifier vorhanden sein
- der Projektnamen und die Datensatznummer werden über eine Inputbox eingegeben
- Beispiel: der panelIdentifier zu adem01_g1v1r und bdem01_g1v3r lautet z.B.-abs2005-ds1-dem01_g1

derivedVariablesIdentifier

Wo wird es genutzt? derivedVariablesIdentifier in vimport.xlsx

- generiert den derivedVariablesIdentifier durch den Vergleich vom Variablenstamm (mit Präfix)
- das FDZ Variablenschema muss verwendet worden sein (Variablenname z.B. astu01_g1v1r mit den möglichen Zugangswegen c,d,o,r,a)
- im Tabellenblatt variables müssen die beiden Spaltennamen name und derivedVariablesIdentifier vorhanden sein
- der Projektname und die Datensatznummer werden über eine Inputbox eingegeben
- Beispiel: der derivedVariablesIdentifier zu adem01 und adem01_g1r des Projektes gra2005-ds1 lautet z.B. gra2005-ds1-adem01

accessWaysInOneColumn

accessWaysInOneColumn-Skript

Wo wird es genutzt? accessWays in vimport.xlsx - wenn die vier Spalten nicht verfügbar im Download-CUF, nicht verfügbar im Download-SUF, nicht verfügbar im Remote-Desktop-SUF und nicht verfügbar im On-Site-SUF ausgefüllt wurden

- wurden in der Excel Tabelle die vier Spalten nicht verfügbar im Download-CUF, nicht verfügbar im Download-SUF, nicht verfügbar im Remote-Desktop-SUF und nicht verfügbar im On-Site-SUF durch ankreuzen mit „x“ ausgefüllt, können daraus die Zugangswege in einer Spalte generiert werden
- dafür müssen die Spaltennamen nicht verfügbar im Download-CUF, nicht verfügbar im Download-SUF, nicht verfügbar im Remote-Desktop-SUF, nicht verfügbar im On-Site-SUF, accessWays und name im Tabellenblatt variables vorhanden sein
- Beispiel: wurden alle vier Spalten einer Variablen mit „x“ ausgefüllt, wird der zugehörige Zugangsweg „not-accessible“ ermittelt

matchSurveyNumbers

matchSurveyNumbers

Wo wird es genutzt? surveyNumbers in vimport.xlsx - wenn das Präfix im Variablennamen in Abhängigkeit von der surveyNumber vergeben wurde

- wurden das Präfix des Variablennamens in Abhängigkeit von der surveyNumber vergeben, kann die surveyNumber automatisch generiert werden
- es muss ein zusätzlichen Tabellenblatt mit dem Namen optionalEntries eingefügt werden
- dieses Tabellenblatt hat 2 Spalten mit den Namen prefix (hier steht z.B. a, b oder c) und surveyNumber (hier wird die zum Präfix zugehörige Nummer der Erhebung eingetragen)
- das Tabellenblatt variables muss die beiden Spalten name und surveyNumbers enthalten
- Beispiel: alle Variablen mit dem Präfix „a“ (z.B. astu01, adem05) bekommen die surveyNumber 1, alle Variablen mit dem Präfix „b“ (z.B. bstu02, bdem03) bekommen die surveyNumber 2, alle Variablen mit abweichendem (bzw. ohne) Präfix (z.B. pid, wave, wgt-Variablen) bekommen die surveyNumbers 1,2

addRelatedQuestionsForGeneratedVariables

addRelatedQuestionsForGeneratedVariables

Wo wird es genutzt? vimport.xlsx > relatedQuestions - das Makro ergänzt im Tabellenblatt relatedQuestions generierte Variablen von Fragebogenvariabeln. Das Makro funktioniert nur, wenn die generierten, wenn die generierte Variablen denselben Stamm haben, wie ihre Ausgangsvariablen (z.B. astu01_g1 von astu01), d.h. wenn das FDZ-Variablennamenschema (mindestens Silber) verwendet wurde!

- Es müssen alle Variablen aus dem Datensatz im Tabellenblatt variables eingetragen sein.
- Es müssen alle Fragebogenvariablen (d. h. „Originalvariablen“) inkl. questionNumber und instrumentNumber im Tabellenblatt relatedQuestions eingetragen sein.
- Das Makro gleicht die Variablenliste vom Tabellenblatt relatedQuestions mit der aus dem Tabellenblatt variables ab und ergänzt die generierten Variablen der eingetragenen Fragebongenvariablen im Tabellenblatt relatedQuestions.
- Die generierten Variablen werden inklusive questionNumber und instrumentNumber am Ende des Tabellenblatts eingefügt
- zur Kontrolle werden die neuen Zellen blau gefüllt
- relatedQuestionString.de/.en muss nicht ausgefüllt werden. name im Tabellenblatt variables und name, questionNumber und instrumentNumber im Tabellenblatt relatedQuestions müssen gefüllt sein- name, questionNumber, instrumentNumber im Tabellenblatt relatedQuestions

3.7 Anpassungen LateX Template DSR

Diese Seite enthält eine Übersicht über die im LateX-Template zur Erzeugung der Variablendetailseiten des DSR notwendigen Änderungen.

3.7.1 nominal skalierte Variablen

- Maßzahlen ergänzen
- Struktur Häufigkeitstabelle anpassen nach Vorlage MDM (Reihenfolge Spalten, Reihenfolge Zeilen)

3.7.2 ordinal skalierte Variablen

- Maßzahlen anpassen (wie MDM)
- Boxplot ergänzen
- Struktur Häufigkeitstabelle anpassen nach Vorlage MDM (Reihenfolge Spalten, Reihenfolge Zeilen)

3.7.3 intervall skalierte Variablen

- Maßzahlen ergänzen
- Boxplot ergänzen
- Häufigkeitstabelle ergänzen

3.7.4 verhältnis skalierte Variablen

- Maßzahlen anpassen (wie MDM)
- Struktur Häufigkeitstabelle ändern (wie MDM)
- sicherstellen, dass alle verhältnis-Variablen ein Histogramm besitzen (Bsp. aocc226h)

3.7.5 Skalenniveau-übergreifend

- sicherstellen, dass die Häufigkeitstabelle bei Variablen ohne value-labels wie gewünscht dargestellt wird (Bsp. astu061b)
- sicherstellen, dass Variablen mit String-values wie gewünscht dargestellt werden (Bsp. bfec162h_g3o)
- sicherstellen, dass Variablen mit mehr als einer zugehörigen Frage wie gewünscht dargestellt werden

3.8 Datensatzreport (Lektorat)

Diese Seite ist aktuell nicht relevant, da im Moment der redaktionelle Teil des Datensatzreports nicht verwendet wird. Eventuell wird es in Zukunft wieder relevant und wird daher hier dokumentiert.

Diese Seite dokumentiert das geplante Aussehen des Datensatzreport und die dafür notwendigen Änderungen bei Inhalt, Struktur und Layout sowie den Stand der vorgenommenen Anpassungen.

3.8.1 1 geplanter Aufbau des DSR

Übersicht

Im folgenden wird der Aufbau des DSR skizziert. Aktuell wird auf den redaktionellen Teil und den Anhang verzichtet.

Titelseite

```
1.1 Titelblatt
1.2 Zweite Seite
1.3 Inhaltsverzeichnis
1.4 Verzeichnis der Variablenseiten (+ Verlinkung auf entsprechende Seite)
```

Redaktioneller Teil

```
2.1 Einleitung
2.2 Informationen zum Datensatz / Datensatzstruktur
2.3 Variablenbenennung, Vergabe von Labels
2.4 Codierung fehlender Werte
2.5 Lesehilfe / Legende zu Variablendetailseiten
```

Variablendetailseiten

Anhang

```
4.1 Tabelle Übersicht über Panelvariablen
4.2 Tabelle Übersicht über vercodete ursprünglich offen erfragte Variablen
4.3 Tabelle Übersicht über anonymisierte Variablen
4.4 Tabelle Übersicht über generierte Variablen (andere Gründen als Anonymisierung)
```

bisherige Festlegungen

- Redaktioneller Teil und Anhang werden jeweils eigene LaTeX-Dateien, die in die main.tex eingebunden werden
- Variablennamen im Anhang werden auf entsprechende Variablenseite im DSR verlinkt

noch offene Aspekte

- Existenz des Anhangs ist abhängig von vorhandenen Personal- und Zeitressourcen; Inhalt des Anhangs muss noch endgültig festgelegt werden
- enthält der Anhang irgendeine Übersicht/Text etc. zu Gewichten?
- soll es ein Verzeichnis der eingesetzten Codierlisten geben (mit oder ohne Links auf die Listen)?
- wird Titelei eigene LaTeX-Datei oder Bestandteil der main.tex

aktuelle ToDos

- Inhalt Titelblatt und zweite Seite festlegen
- Redaktionellen Teil schreiben (2.1 bis 2.4)
- Inhalt und Darstellung Lesehilfe (2.5) festlegen
- Abbildungs- und Tabellenverzeichnis rausnehmen (klären mit LaTeX-Firma, ob wir grundlegender Typ des Dokumentes ändern, kein *Buch* mit linker/rechter Seite etc.)

3.8.2 2 Variablendetailseiten

bisherige Festlegungen

Skalenniveau/Block	nominal	ordinal	intervall	verhältnis
Variablendetails	ja	ja	ja	ja
Fragedetails	ja	ja	ja	ja
Maßzahlen	wie MDM	wie MDM	wie MDM	wie MDM
Boxplot	nein	ja	ja	ja
Häufigkeitstabelle	wie MDM	wie MDM	wie MDM	wie MDM
Histogramm	nein	nein	nein	ja

- *Eingangsfilter* und *Generierungsregel* werden nur angezeigt, wenn sie eine noch zu definierende String-Länge nicht überschreiten. Bei längeren Strings in diesen Atrributen wird aufs MDM verlinkt (teilweise Darstellung im DSR oder dort nur Standardtext?)

noch offene Aspekte

- Block Variablendetails: Sichtbarkeitsbedingung für Attribute *Beschreibung*, *Panelvariablen* und *Eingangsfilter*
- Block Variablendetails: Schreibweise der Skalenniveaus im DSR festlegen (Groß-oder Kleinschreibung von *intervall* und *verhältnis* oder *intervall skaliert*)
- Block Fragendetails: Sichtbarkeitsbedingung für Attribute *Einleitung der Frage* und *Ausfüllanweisung*
- Block Maßzahlen: entscheiden, ob *Devianz* bleibt oder raus soll
- lange *Eingangsfilter* und *Generierungsregel*: festlegen, was im DSR angezeigt wird (s.o.)

aktuelle ToDos

- [[Anpassungen LaTeX-Template | Anpassungen LateX-Template DSR]], um Anforderungen (vgl. Tabelle) zu erfüllen
- Sonderfall-Variablen identifizieren, bei denen abweichend von den Vorgaben der Tabelle bestimmte Elemente (z.B. Boxplot bei aocc12) nicht angezeigt werden soll. Sonderfallklassen bilden und im LaTeX-Template behandeln.
- Block Fragendetails: festlegen, welcher Text erscheinen soll, wenn es keine zugehörige Frage gibt
- LaTeX-Template: Fallunterscheidungen für Attribute *Eingangsfilter* und *Generierungsregel* bezüglich String-Länge ergänzen
- LaTeX-Template: Tausendertrennzeichen einbauen
- LaTeX-Template: Darstellung des Histogramms anpassen, so dass die Balken verbunden sind
- Daten: ungewünschte Zeilenumbrüche entfernen

3.8.3 3 Layout-Anpassungen

aktuelle ToDos

- festlegen, an welchen Stellen das Layout angepasst werden soll und rausfinden, ob das noch im Rahmen des externen LaTeX-Auftrags möglich ist

3.8.4 4 Verschiedenes

bisherige Festlegungen

- der Datensatzreport wird zunächst nicht übersetzt. Falls die neuen Kolleginnen (Übersetzerinnen) in den nächsten Wochen noch Ressourcen frei haben, wird dieser Aspekt neu diskutiert.

noch offene Aspekte

- wie verfahren wir mit Episodendatensätzen? Datensatzreport macht wenig Sinn, evtl. stattdessen nur ein PDF mit einer Beschreibung, was ein Episodendatensatz ist, wie die Struktur des konkreten Datensatzes aussieht (Anzahl Fälle, Anzahl Variablen etc.) und wie man die Episodendaten an den Personendatensatz anspielt, praktisch eine Miniversion des redaktionellen Teils des normalen Datensatzreports

3.9 Fragen (questions)

Übersicht

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:

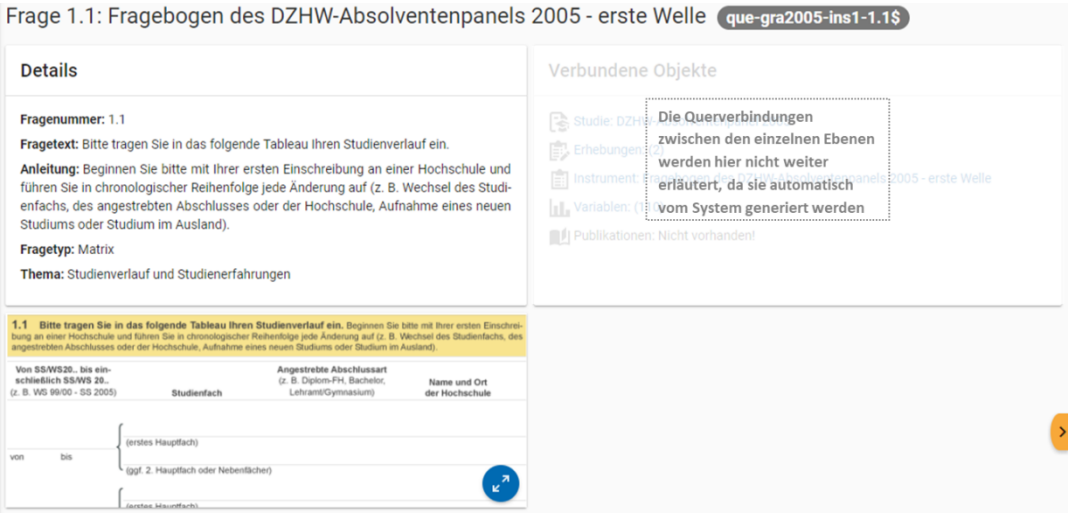


Abb. 3.9: Fragenübersicht im MDM am Beispiel der Frage 1.1 des Fragebogens der ersten Welle im Absolventenpanel 2005

Auf dieser Ebene werden Informationen über alle Fragen für jedes einzelne Erhebungsinstrument eines Datenpakets 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 bzw. handcrafted)**). Im Folgenden werden beide Vorgehensweisen schrittweise beschrieben.

3.9.1 Fragestruktur

Fragen sind gekennzeichnet durch einen einleitenden/übergreifenden Fragetext, sowie eine „natürliche“ sichtbare Abgrenzung gegenüber anderer Fragen und eine meist „erkennbare“ Nummerierung. Es wird zwischen fünf Fragetypen differenziert:

- Single Choice: Auf die Frage kann nur mit einer Antwortmöglichkeit geantwortet werden (z.B. Einfachauswahl aus mehren Antwortmöglichkeiten oder Angabe eines numerischen Wertes).
- Mehrfachnennung: Für die Frage gibt es eine Auswahl an Antwortmöglichkeiten bei denen eine oder mehre ausgewählt werden können.
- Itembatterie: Besitzt überleitenden Fragetext, welche jeweils weitere Items mit den gleichen Antwortmöglichkeiten besitzen.
- Matrix: Ist ein komplexer Fragetyp in dem viele Unterfragen geschachtelt werden können und die nicht durch die anderen Fragetypen abgedeckt werden (z.B. Tableaufragen des Absolventenpanels).
- Undocumented: Die „Restkategorie“, sollte die Frage nicht mit einem der oben genannten Fragetypen abzubilden sein.

Die technische Dokumentation zum erstellen der Frage-Metadaten finden Sie hier: https://dzhw.github.io/questionMetadataPreparation/index.html

Die Anleitung zur Zusammenstellung der Metadaten (inkl. Beschreibung der Attribute) finden Sie hier: https://dzhw.github.io/questionMetadataPreparation/articles/question_metadata_preparation_introduction.html

3.10 Variablen (variables)[1]

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 (das Onlinebefragungstool des DZHW) 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 das 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.

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*.“

[1] 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“.

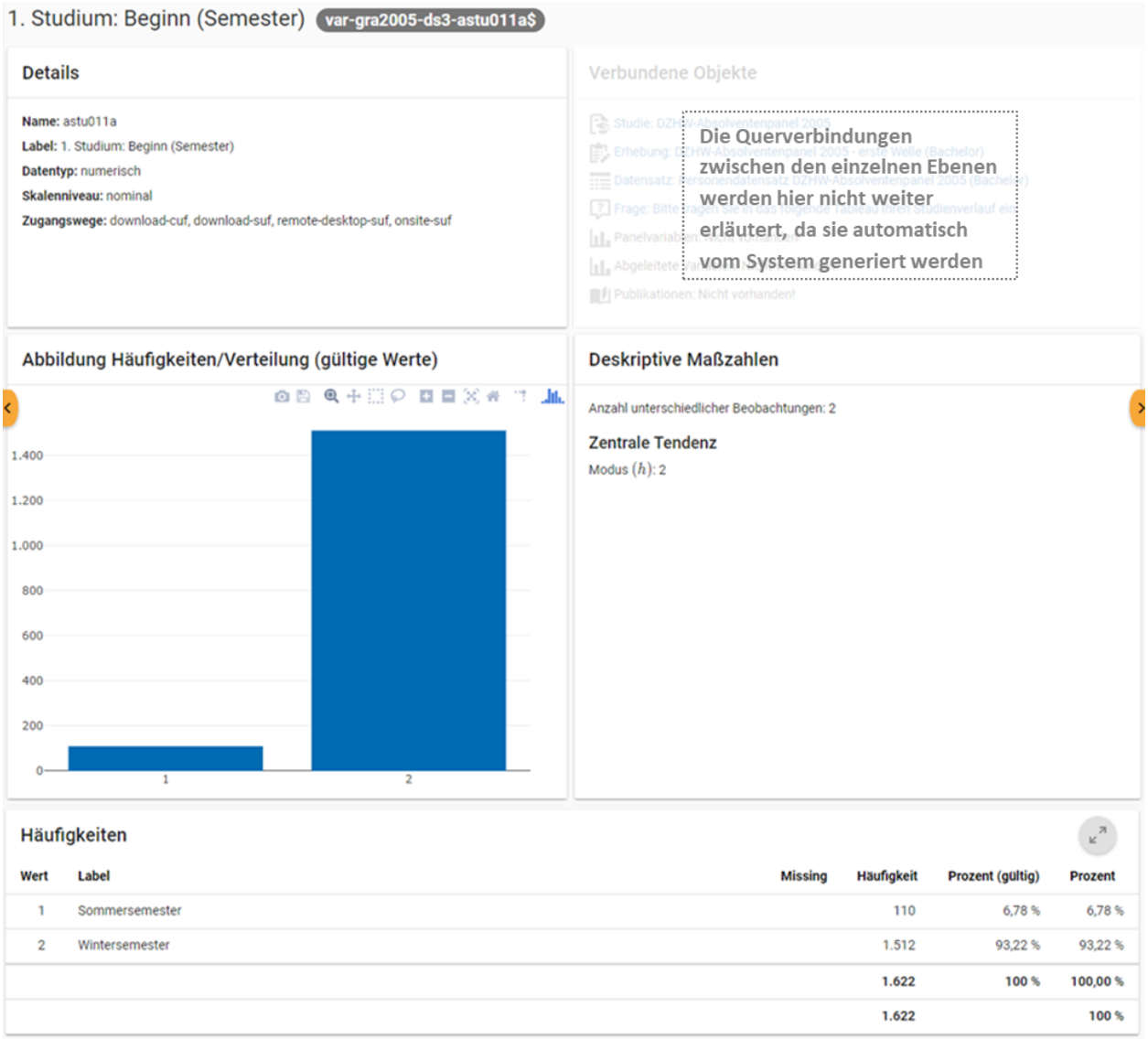


Abb. 3.10: Variablenübersicht im MDM am Beispiel der Variable „1. Studium: Beginn (Semester)“ im Absolventenpanel 2005, erste Welle (BA)

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[2]	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[2]	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 generiert	Regel, die in der angegebenen „Sprache“ (.ruleExpressionLangu age) beschreibt, wie die Variable erzeugt wurde (Beispiel, siehe Abschnitt „Generierungsregel (Stata)“)
ge-	Ja,	Sprache der Erzeugungsregel: „Stata“ oder „R“

* 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)		
Spalten-überschrift	**Muss ich das ausfüllen?**	**Was muss ich eintragen?**
name	Ja	Variablenname
relatedQuestionStrings.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) **wichtig**: da diese auch in den Datensatzreport als „Fragetext“ ausgelesen werden, muss unbedingt auf eine korrekt Formatierung (Leerzeichen, Zeilenumnrüche etc.) geachtet werden.
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. 3.11 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.

3.11 Erstellung der Variable-JSON Dateien

Die Erstellung der Variablen JSONs erfolgt komplett im geschützten Bereich. Benötigt werden pro Datensatz ein zugehöriger Stata-Datensatz und eine Exceltabelle. Die Exceltabelle (vimport_dsNR.xlsx) enthält die beiden Tabellenblätter variables und relatedQuestions. Pflichtspalten und zugehörige Ausfüllanweisungen werden im folgenden Abschnitt beschrieben.

Es ist erlaubt die Exceltabellen um weitere optionale Spalten zu erweitern, z.B. Varname_alt, Var_Erh, Var_Thema, Var_Nr, Var_Indiz, Var_g, Var_h, Var_x, Var_p, Var_v, Var_Zugang, Varlabel_alt, Varlabel_neu, On-Site, Remote-Desktop, Download-SUF, Download-CUF, AIP, SIP, delete, ...

Momentan liegen die Import Dateien der Projekte, sowie die Skripte zur Erzeugung der JSONs im geschützten Bereich unter Q:Variablenexport. Der Aufbau der Ordnerstruktur ist wie folgt:

```
|--Variablenexport
   |--Projekte
      |--gra2005
         |--variablesToJsons.bat
```

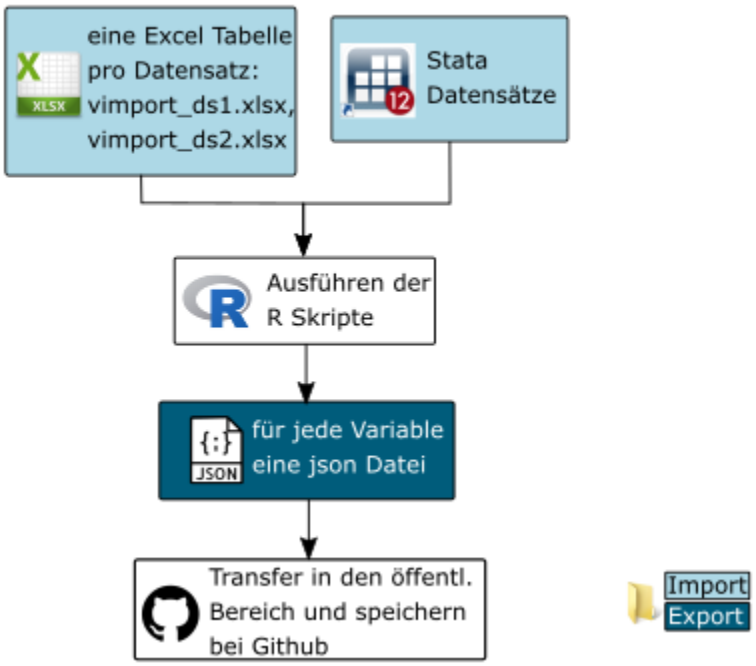
(Fortsetzung auf der nächsten Seite)

[2] 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“.

1.8 Waren für Sie studienbegleitende Praktika oder Praxissemester vorgeschrieben?
(Mehrfachnennung möglich)

Ja, Praktika an der Hochschule (z. B. Laborpr	astu08a
Ja, externe Praktika (z. B. Betriebspraktikum)	astu08b
Ja, Praxissemester	astu08c
Ein Praktikum war zwar vorgeschrieben, musste von mir aber nicht absolviert werden (z. B. wegen der Anerkennung einer Ausbildu	astu08d
Nein	astu08e

Abb. 3.11: Ausschnitt aus dem Variablenfragebogen des Absolventenpanels 2005, erste Welle, Frage 1.8



(Fortsetzung der vorherigen Seite)

```
        |--output
           |--ds1
           |--ds2
        |--data-raw
           |--stata
              |--ds1.dta
              |--ds2.dta
           |--excel
              |--vimport_ds1.xlsx
              |--vimport_ds2.xlsx
              |--conditions.xlsx
|--variable-generation_productive
   |--variablesToJsons.bat.tmpl
```

Um json Dateien für ein neues Projekt zu generieren, muss zunächst ein Ordner für das neue Projekt angelegt werden und die oben gezeigt Ordnerstruktur aufgebaut werden. Im Ordner stata befinden sich die jeweiligen Stata Datensätze (ds1, ds2, ds3, . . .) und im Ordner excel die zugehörigen Exceltabellen mit den beiden Tabellenblättern variables und relatedQuestions (vimport_ds1.xlsx, vimport_ds2.xlsx, vimport_ds1.xlsx, . . .), sowie die Datei mit den missing conditions (conditions.xlsx). Zum Generieren der json Dateien das R-Skript variablesToJsons.bat.tmpl in den Projektordner kopieren, das .tmpl entfernen, die Datei anpassen und danach ausführen. Das aktuelle Template liegt hier: https://github.com/dzhw/variableMetadataPreparation/blob/development/bin/variablesToJson.bat.tmpl

Es ist möglich die Missing Bedingungen für numerische und string Variablen in der datei conditions.xlsx anzupassen. Außerdem können in der batch-Datei Variablennamen angegeben werden, die im MDM keine Verteilung bekommen sollen. Dies sind z.B. id Variablen. Variablen mit accessway „not-accessible“ müssen hier nicht eingetragen werden.

Missing Conditions

In der Exceltabelle conditions.xlsx können für numerische und string Variablen Missingbedingungen angegeben werden. Die Exceltabelle enthält die beiden Tabellenblättern missingConditionNumeric und missingConditionString. Es ist möglich für numerische und string Variablen jeweils mehrere Bedingungen anzugeben. Die Bedingungen werden mit ODER verknüpft. Das heißt, wenn eine der Bedingungen für einen Wert zutrifft, wird dieser Wert als Missing gewertet. Die verfügbaren Operatoren können in der Exceltabelle über ein Drop-Down Menü ausgewählt werden und sind im Tabellenblatt list of valid operators dokumentiert.

Ein Fehler der auftreten kann ist, dass im Stata-Datensatz nicht die richtige Sprache gewählt wurde. Ist das der Fall können nicht die richtigen Wertelabel zugeordnet werden.

Nach erfolgreichem Durchlauf

Nach erfolgreicher Generierung der Variablen-Metadaten werden der Ordner data-raw und die Datei variableToJson.bat im Projektordner in den Unterordner Metadatenexport kopiert. In Zukunft wird es eine Option im Programm variableMetadataGeneration geben, dass dieses den Ordner und die variableToJson.bat kopiert.

Transfer in den öffentlichen Bereich Die Json-Dateien und Bilder müssen noch in den öffentlichen Bereich transferiert werden. Da es nicht möglich ist, Ordner zu transferieren, werden die Ordner gezippt (7-Zip), transferiert und im öffentlichen Bereich wieder entpackt. Die Ordner mit den Variablen-Metadaten werden anschließend auf Variablenebene ins MDM per Drag and Drop oder über den Plusbutton rechts unten hochgeladen.

3.12 Variables (Zofar)

Bei Onlinebefragungen mit ZOFAR können fragenbezogene Metadaten auf Variablenebene automatisch extrahiert werden. Eine .csv Tabelle die den Variablennamen, die Instrumentnummer, die Fragenummer und den relatedQuestionString (Fragetext + zugehöriger Variablentext) enthält, wird geliefert.

Der Prozess befindet sich im Aufbau. . .

3.13 Related Publications

Arbeiten mit der Citavi-Datenbank

Für dieses Objekt wird eine Citavi-Datenabank angelegt. Diese liegt unter: *\faust-Abt4FDZQuerschnittsaufgabenMetadatenErzeugenLiteraturexportrelatedPublication*.

Die Citavi-Einträge lassen sich exportieren, indem man einen Eintrag in der Literaturübersicht markiert (linke Seite) und Str+Alt+t drückt. Die Tabellenansicht öffnet sich und durch klicken auf Spalten (oben links) kann ausgewählt werden, welche Spalten exportiert werden sollen. Aus der Tabellenansicht kann die die Datei jetzt nach Excel als relatedPublications.xls exportiert werden (Datei > nach Microsoft Excel exportieren). Einige Spaltennamen müssen evtl. später noch manuell umbenannt werden (z.B. BibTeXKey wird zu id). Die Excel-Tabelle wird *hier<https://github.com/dzhw/metadatamanagement-io/tree/master/references/relatedPublications>* gepflegt.

Eigenschaft	Ausfüllanweisung	muss ausgefüllt werden?
id	von Citavi erzeugter BibTex-Key	ja
sourceReference	Quellangabe der Publikation (default)	ja
publicationAbstract	Zusammenfassung	nein
doi	doi der Publikation	nein
sourceLink	valide URL	nein
title	Titel	ja
authors	Autoren (Nachname1, Vorname1; Nachname2, Vorname2)	ja
year	Jahr der Veröffentlichung (muss kleiner oder gleich dem aktuellen Jahr sein)	ja
abstractSource.de/.en	??	nein
dataSetIds		nein
instrumentIds		nein
surveyIds		nein
variableIds		nein
questionIds		nein
studySerieses.de	mindestens eine dataPackageId oder mindestens 1 studySeries	Falls vorhanden ja
language	Sprache der Publikation (2-Buchstaben Code nach ISO 639-1	ja
annotations.de	Anmerkungen zur Publikation auf Deutsch	nein
annotations.en	Anmerkungen zur Publikation auf Englisch	nein

3.14 Datensatzreport erzeugen

Wenn ein Datensatz und die zugehörigen Variablen im MDM vorliegen, kann mit Hilfe des MDMs ein Datensatzreport erstellt werden. Hierzu wird im MDM auf der Datensatz-Seite auf den PDF-Button geklickt. Nach einiger Zeit (je nach Anzahl an Variablen länger als eine Minute) erhält der Publisher eine Mail, dass der Datensatzreport erzeugt wurde. Der Datensatzreport wird automatisch im MDM eingefügt. Es ist die jeweils aktuelle Version des Datenpakets als Version anzugeben.

3.15 Prüfung der Jsons nach Umstellung der Generierungsskripte

Es ist mögliche, zwei jsons online miteinander zu vergleiche, z.B. hier . Dazu einfach die Texte aus den zu vergleichenden jsons kopieren und in die jeweiligen Felder einfügen.

3.15.1 Struktur der Jsons

Die vorgegebene Struktur der json Dateien ist hier zu finden:

- Link to Variable Import File
- Link to Question Import File

3.15.2 Variablenprüfung

Beim Prüfen ist besonders drauf zu achten, Variablen mit:

- verschiedenen Skalenniveaus (besonders **intervall** und **verhältnis**, da viele Maßzahlen)
- verschiedenen Datentypen
- einer oder mehreren zugeordneten Fragen

auszuwählen.

3.15.3 Allgemeine Prüfung

- ist ein Attribut leer muss der Wert null sein (alternativ kann das Attribut auch aus dem json herausgenommen werden)
  - richtig: „panelIdentifier“: null
  - falsch: „panelIdentifier“: „“
  - falsch: „validResponses“: []
- der Datentyp muss richtig angegeben werden (wie im Beispiel Import File) -> die häufigsten Datentypen sind:

Datentyp	Beispiel
string	„storageType“: „integer“
I18nString	„scaleLevel“: {„en“: „interval“, „de“: „intervall“}
integer	„indexInDataSet“: 642
double	„standardDeviation“: 2.7881
list	„accessWays“: [„download-cuf“, „download-suf“, „remote-desktop-suf“, „onsite-suf“]
boolean	„containsAnnotations“: false

3.16 Skalenniveau prüfen

3.17 Skalenniveau Diese Informationen dienen dazu die verschiedenen

Skalentypen voneinander unterscheiden zu können und Variablen selbständig einordnen zu können. Das Forschungsdatenzentrum (**FDZ**) hat sich dazu entschieden vier Skalentypen zu unterscheiden _Nominal-, _*Ordinal*-, _\`Intervall <>__ und _\`Verhältnisskala <>__. Siehe hierzu Wikipedia-Eintrag

3.17.1 Skalentypen **_\ **\ `Nominalskala <>__` - Nominale Messung besteht

in der Erstellung einer einfachen Klasseneinteilung, die _\ `jedes <>__` Objekt genau _\ `einer <>__` Klasse zuordnet: Weder darf ein bestimmtes Objekt überhaupt nicht zugeordnet werden, noch darf ein Objekt mehreren Klassen zugeordnet werden.

Beispiele (Variable) | :—:—| | (=/) | Geschlecht, Universitätsnamen, Studienfächer

**_\ **\ `Ordinalskala <>__` - Sie besitzt eine Rangordnung der Objekte in Bezug auf die interessierende Dimension. Die entsprechend zugeordneten Zahlen müssen diese Rangordnung wiedergeben.

Beispiele (Variable) :— | (=/ ; </>) | Schulnoten, höchster Bildungsabschluss, Zufriedenheitensskalen

**_\ **\ `Intervallskala <>__` - ist ein Skalenniveau in der Statistik. Sie zählt zum metrischen Messniveau, da sich die Ausprägungen dieses Skalenniveaus quantitativ mittels Zahlen darstellen lassen. Insbesondere bedeutet das auch, dass Rangunterschiede und Abstand zwischen Werten gemessen werden können; das heißt, quantitative Merkmale gehen in ihren Anforderungen über ordinale oder gar nominale Eigenschaften hinaus.

Beispiele (Variable) | :—:—| | =/ ; </> ; ; + | Temperatur (Celcius, Fahrenheit), Zeitskala (Datum)

**_\ **\ `Verhältnisskala <>__` - ist das höchste Skalenniveau in der Statistik. Bei ihr handelt es sich um eine metrische Skala, im Unterschied zur Intervallskala existiert jedoch ein absoluter Nullpunk

| logische / mathematische Operationen | Beispiele (Variable) | Stata-Beispiel Syntax| | :—:—| | =/ ; </> ; +/ ; ÷ | Alter, Einkommen

Übersicht

| | Verschiedenartigkeit | natürl. Reihenfolge | Interpretierbarkeit der Verhältnisse der Differenzen | natürl. Nullpunkt | natürl. Einheit |

:—:—:—| ja | nein | nein | nein | nein | ja | ja | nein | nein | nein | ja | ja | ja | nein | nein | ja | ja | ja | ja | ja |

FAQ (ausfüllen): - Was ist mit „Sonstige“-Kategorien - Was mit

fehlenden Werten machen? - Prinzipiell immer konservativ (niedriges Niveau) Skalenniveau vergeben

Typische Vergaben -

[1]: R. Schnell, P. Hill, and E. Esser. Oldenbourg, München u.a., 6., völlig überarb. und erw. Aufl. edition, (1999); S.134ff. [2]: Stata commands sind kursiv. Vor einem kursiven Ausdruck muss ein Zeilenumbruch sein; Bsp.: _\ `sysuse <>__` auto (Zeilenumbruch) _\ `fre <>__` make

3.18 Testen von MDM-Issues

Issues des metadatamanagement-Repos werden im Testsystem getestet: https://test.metadata.fdz.dzhw.eu Für einige Issues ist eine Anmeldung notwendig.

Wie genau beim Testen vorzugehen ist, hängt stark vom jeweiligen Issue ab.

Generell sind die folgenden Punkte zu beachten:

- ist alles so umgesetzt, wie erwartet?
- verschiedene Browser benutzen (Firefox, Chrome, Internet Explorer 11 und - wenn möglich - Edge)
- sowohl deutsche als auch englische Seiten testen

- Issues, die den Upload betreffen, können nur über Chrome und Firefox getestet werden, da der Upload über IE nicht funktioniert
- bei Issues, die den Upload betreffen, auch prüfen, ob das Fehlerprotokoll nachvollziehbar ist
- nicht nur prüfen, ob etwas funktioniert, sondern auch, ob es nicht funktioniert, wenn es nicht funktionieren soll
- um den Upload mit gültigen und nicht-gültigen Daten zu prüfen: Daten im io-Testprojekt manipulieren
- überprüfen, ob Funktionen auch auf verkleinertem Bildschirm funktionieren (responsives Design)
- überprüfen, ob Funktionen auch bei Verwendung eines Handys funktionieren (auch hier versch. Browser testen)
- prüfen, ob die wiki-Dokumentation im metadatamanagement-Repo richtig angelegt wurde

Wenn **keine Fehler gefunden** wurden: Label „status: testing“ entfernen, unassignen und Issue schließen. Wenn **Fehler gefunden** wurden: Fehler im Issue möglichst genau dokumentieren (gerne auch Screenshots), label „status: testing“ entfernen und „status: development“ hinzufügen, sich selbst unassignen und den Developer assignen.

3.19 Datentypen

Der jeweilige Datentyp für eine Eigenschaft kann hier nachgeschaut werden.

Datentyp	Beschreibung	Excel Export	Json Export
string	Text	ein Objekt in einer Zelle	z.B.: `"value" : 1`
I18nString	de: string en: string	2 Spalten, z.B.: `annotations.de` `annotations.en`	als Liste, z.B: „questionText“: { „de“: „Wie alt sind Sie?“, „en“: „How old are you?“ }
Period	start: LocalDateTime end: LocalDateTime	2 Spalten, z.B.: `fieldPeriod.start` `fieldPeriod.end`	
LocalDateTime	yyyy-mm-dd, z.B.: 2011-12-23		
Double	Gleitkommazahlen		
Integer	Ganze Zahlen		
List	Eigenschaft mit mehreren Objekten	Objekte werden in einer Zelle durch Komma getrennt, z.B.: download-cuf, download-suf, remote-desktop-suf, onsite-suf	als Array, z.B.: „successorNumbers“: [„1“,“2“]
ImageType	PNG		

Dabei gibt es für strings und l18nStrings eine Obergrenze für die maximale Länge (Zeichenanzahl):

- SMALL: 32 characters
- MEDIUM: 128 characters
- LARGE: 2048 characters (2KB)
- X-LARGE: 1 MB

3.19.1 Vokabular

Es ist zu beachten, dass für einige Eigenschaften nur spezielles Vokabular zugelassen ist! Dieses kann entweder auf der io-wiki Seite der jeweiligen Exportfunktion (DataPackage, Survey, DataSet, Variable, Instrument, Question) oder hier nachgeschaut werden.

Beispiel:

```
- accesWay: "download-cuf", "download-suf", "remote-desktop-suf", "onsite-suf", "not-
↪accessible"
- scaleLevel.de = "nominal", "ordinal", "intervall" oder "verhältnis"
- scaleLevel.en = "nominal", "ordinal", "interval" oder "ratio"
```

3.20 Die korrekte Anordnung der Dateien im Ordner

Für einen erfolgreichen Upload der Metadaten attachments 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. Der Unterordner, welche alle Anhänge enthält, heißt unabhängig von der Ebene immer „attachments“.

3.21 Übersicht über verschiedene R-Helfer-Skripte

Name	Was kann es?	Wo wird es genutzt?	Was muss manuell gemacht werden?
csv2json	erstellt aus der `ProjID.csv` Tabelle und den `FrageNr.pdf` für jede Frage eine json Datei die json Dateien werden im Ordner `\questions\json\insNr` gespeichert	Fragenexport im öffentlichen Bereich	die Tabelle `ProjID.csv` muss ferig aufbereitet sein und im Ordner `\questions` liegen zu jeder Frage muss eine `FrageNr.pdf` Datei im Ordner `questions\Bilder\insNr\pdf` vorliegen Alternativ kann auch ocr für Bilder genutzt werden, dies muss jedoch im Skript auskommentiert werden Angabe des Projektnames im Skript Ausführen (Strg + r). Bilder sind nicht mehr verpflichtend.
responseRateSVG	Erstellt für alle csv Tabellen im Ordner \csv jeweils ein deutsches und ein englisches Diagramm die Diagramme werden als responserate_de.svg und responserate_en.svg im Ordner \images gespeichert	Responserate Diagramme für den Surveyexport	für eine Survey muss eine csv Tabelle erstellt werden die Angaben `einghauf, einghaufcum, datumw` müssen in der csv Tabelle stehen Ausführen (Strg + r)

Unter https://github.com/dzhw/variableMetadataExtractor findet man die Skripte zur Generierung der Variable-JSONs.

Die Struktur der Dateipfade muss immer eingehalten werden (nach dem Beispiel test2017), sonst funktioniert es nicht!

3.21.1 Response Rate Diagram

Das Skript [responserateSVG](http://github.com/dzhw/metadatamanagement-io/blob/master/generation/example/survey/Responserate/responserateSVG.R) erzeugt Responserate Diagramme als SVG in deutsch und englisch für die verschiedenen Surveys. Benötigt wird dafür je eine csv Tabelle pro Survey, in der folgende Spalten enthalten sind: - einghauf = Rücklauf pro Woche - einghaufcum = Rücklauf kummuliert - datumw = Wochendatum im Format Jahr_w_Kalenderwoche (z.B. 2006w52)

3.22 Verantwortlichkeiten

Objekt	Teilobjekt	Verantwortlicher
[[Data package]]	alles	aufbereitendes Projektteam
[[Survey]]	alles	aufbereitendes Projektteam
[[DataSet]]	alles	aufbereitendes Projektteam
[[Variable]]	vimport.xlsx	aufbereitendes Projektteam
	variable.xlsx	aufbereitendes Projektteam
	variable.json	Metadatenverantworlic her
[[Instrument]]	alles	aufbereitendes Projektteam
[[Question]]	qimport.xlsx	aufbereitendes Projektteam
	question.json	Metadatenverantworlic her
	question.bilder	Metadatenverantworlic her
[[Related Publication]]	alles	Publikationsverantwor tlicher (Karsten/Bene)

3.23 Projektrelease

User mit der Rolle Publisher können Projekte bei denen alle erwarteten Metadaten als fertig markiert wurden releasen. Dazu muss in der Navbar (Menü links) auf den Release Button geklickt werden. Bei Release wird eine Postvalidierung durchgeführt, näheres dazu findet sich *hier<https://github.com/dzhw/metadatamanagement/wiki/Domain-Model#dataacquisitionproject-post-validation>*. Ab Versionsnummer 1.0.0 wird das Projekt an dalra weitergegeben und erhält eine doi. Wenn der Release Button ein weiteres Mal geklickt wird, wird die Freigabe zurückgezogen und Metadaten können weiter editiert werden. Bei erneutem Klick kann das Projekt dann neu released werden. Gegebenenfalls wird eine neue Versionsnummer vergeben (Versionierungskonzept folgt).

3.24 Domänenmodell

Im *Domänenmodell* werden alle Domänenobjekte, ihre Relationen zueinander und, ob diese verpflichtend auszufüllen sind, dokumentiert. .. _Domänenmodell: https://github.com/dzhw/metadatamanagement/wiki/Domain-Model

3.25 Javadoc

3.25.1 eu.dzhw.fdz.metadatamanagement.common.domain

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

AbstractRdcDomainObject

public abstract class **`AbstractRdcDomainObject`** implements Serializable
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.

serialVersionUID

private static final long **`serialVersionUID`**

version

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

AbstractShadowableRdcDomainObject

public abstract class **`AbstractShadowableRdcDomainObject`** extends *AbstractRdcDomainObject*
Base class for all rdc domain objects which can exist as multiple versions (shadows).

Fields

hidden

private boolean **hidden**
: True if and only if the shadow copy must not be available for the public user. Only shadow copies which have a successor may be hidden.

serialVersionUID

private static final long **serialVersionUID**

shadow

private boolean **shadow**
: Determines whether this document is a shadow copy.

successorId

private String **successorId**
: The document id which is the successor to this shadow copy.

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.

Country

public class **Country** implements Serializable
: Represents a country with it's 2-letter country code and it's display name in german and english.

Fields

code

private String **code**

de

private String **de**

en

private String **en**

serialVersionUID

private static final long **serialVersionUID**

I18nString

public class **I18nString** implements Serializable
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.

serialVersionUID

private static final long **serialVersionUID**

ImmutableI18nString

public class **ImmutableI18nString** extends *I18nString*
Immutable (constant) version of `I18nString`s.

Fields

serialVersionUID

private static final long **serialVersionUID**

Language

public class **Language** implements Comparable<*Language*>

Wrapper for a language code and it's respective display name.

Fields

displayName

private final String **displayName**

Display name.

languageCode

private final String **languageCode**

Language code.

Period

public class **Period** implements Serializable

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.

serialVersionUID

private static final long **serialVersionUID**

start

private LocalDate **start**

The start date of the period. Mandatory and must not be after end date.

Person

public class **`Person`** implements Serializable
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.

serialVersionUID

private static final long **`serialVersionUID`**

Resolution

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

Fields

heightY

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

serialVersionUID

private static final long **`serialVersionUID`**

widthX

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

ShadowCopyCreateNotAllowedException

public class **`ShadowCopyCreateNotAllowedException`** extends IllegalArgumentException
Exception that should be thrown if client tries to create a shadowed domain object.

Fields

serialVersionUID

private static final long **`serialVersionUID`**

ShadowCopyDeleteNotAllowedException

public class **`ShadowCopyDeleteNotAllowedException`** extends IllegalArgumentException
Exception thrown if client tries to delete a shadowed domain object.

Fields

serialVersionUID

private static final long **`serialVersionUID`**

ShadowCopySaveNotAllowedException

public class **`ShadowCopySaveNotAllowedException`** extends IllegalArgumentException
Exception that should be thrown if a client tries to update a shadow version of a domain object.

Fields

serialVersionUID

private static final long **`serialVersionUID`**

Task

public class **`Task`** extends *AbstractRdcDomainObject*
Task entity holding the current state of a long running task.

Author tgehrke

Fields

errorList

private ErrorListDto **`errorList`**
The list of errors which occurred during execution of the task.

id

private String **id**
The id or task number of the task.

location

private String **location**
The location URI of the result of the task.

serialVersionUID

private static final long **serialVersionUID**

state

private *TaskState* **state**
The current state of the task.

type

private *TaskType* **type**
The type of the task.

Task.TaskState

public enum **TaskState**
State of tasks.

Author tgehrke

Enum Constants

DONE

public static final *Task*.*TaskState* **DONE**

FAILURE

public static final *Task*.*TaskState* **FAILURE**

RUNNING

public static final *Task*.*TaskState* **RUNNING**

Task.TaskType

public enum **TaskType**
type of tasks.

Author tgehrke

Enum Constants

DATA_SET_REPORT

public static final *Task.TaskType* **DATA_SET_REPORT**

TaskErrorNotification

public class **TaskErrorNotification**
DTO holding all information required for sending notifications to users in case an error occurred during task execution.

Author René Reitmann

Fields

domainObjectId

private String **domainObjectId**
An id of a domainObject. May be empty.

errorMessage

private String **errorMessage**
An error message indicating the reason of the error. Must not be empty.

onBehalfOf

private String **onBehalfOf**
The name of the user for which the task has been executed. May be empty.

taskType

private *TaskType* **taskType**
The type of the task which has been executed. Must not be empty.

3.25.2 eu.dzhw.fdz.metadatamanagement.conceptmanagement.domain

Concept

public class **Concept** extends *AbstractRdcDomainObject* implements ConceptSubDocumentProjection

A concept is something which cannot be observed directly but there is a model which helps observing the concept. E.g.: The concept „Personality“ can be observed with the help of the five-factor model (Big5).

Fields

authors

private List<*Person*> **authors**

List of `Person`s which have defined this concept. Must not be empty.

citationHint

private String **citationHint**

Hint on how to cite this concept. Markdown is supported. Must not be empty and must not contain more than 2048 characters.

description

private *I18nString* **description**

A description of the concept. Markdown is supported. It must be specified in German and English and it must not contain more than 2048 characters.

doi

private String **doi**

The doi of the paper defining the concept. Must not contain more than 512 characters.

id

private String **id**

The id of the concept which uniquely identifies the concept in this application. Must not be empty and must not contain more than 512 characters. Must start with „con-“ and end with „$“ and must not contain any whitespace.

license

private String **license**

The license of this concept. Markdown is supported. Must not contain more than 1 MB characters.

originalLanguages

private Set<String> **originalLanguages**

The original languages of the definition of the concept as ISO 639 code. Must not be empty.

serialVersionUID

private static final long **serialVersionUID**

tags

private *Tags* **tags**
Keywords for the concept. Must not be empty.

title

private *I18nString* **title**
The title of the concept. It must be specified in German and English and it must not contain more than 512 characters.

ConceptAttachmentMetadata

public class **ConceptAttachmentMetadata** extends *AbstractRdcDomainObject*
Metadata which will be stored with each attachment of a `Concept`.

Fields

conceptId

private String **conceptId**
The id of the `Concept` 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.

indexInConcept

private Integer **indexInConcept**

The index in the `Concept` of this attachment. Used for sorting the attachments of this `Concept`. 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.

serialVersionUID

private static final long **serialVersionUID**

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 `ConceptAttachmentTypes` and must not be empty.

ConceptAttachmentTypes

public class **ConceptAttachmentTypes**

All valid types of a `ConceptAttachmentMetadata`.

Fields

ALL

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

DOCUMENTATION

public static final *I18nString* **DOCUMENTATION**

INSTRUMENT

public static final *I18nString* **INSTRUMENT**

OTHER

public static final *I18nString* **OTHER**

ConceptInUseException

public class **ConceptInUseException** extends RuntimeException

Thrown if a delete attempt was made while the Concept is referenced by an instance of `eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain.Instrument` or `eu.dzhw.fdz.metadatamanagement.questionmanagement.domain.Question`.

Fields

instrumentIds

private final Set<String> **instrumentIds**

questionIds

private final Set<String> **questionIds**

serialVersionUID

private static final long **serialVersionUID**

Tags

public class **Tags** implements Serializable

Contains tags associated with a concept.

Fields

de

private Set<String> **de**

German tags. At least one tag must be provided.

en

private Set<String> **en**

English tags. At least one tag must be provided.

serialVersionUID

private static final long **serialVersionUID**

3.25.3 eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain

Domain objects describing `eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain.DataPackage`s.

DataPackage

public class **`DataPackage`** extends *AbstractShadowableRdcDomainObject* implements DataPackageSubDocumentProjection

A data package 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 dataPackage. Markdown is supported. Must not contain more than 2048 characters.

dataAcquisitionProjectId

private String **`dataAcquisitionProjectId`**

The id of the `DataAcquisitionProject` to which this dataPackage belongs. The dataAcquisitionProjectId must not be empty.

dataCurators

private List<*Person*> **`dataCurators`**

List of `Person`s which have curated this data package. Must not be empty.

description

private *I18nString* **`description`**

A description of the dataPackage. Markdown is supported. It must be specified in German and English and it must not contain more than 2048 characters.

id

private String **`id`**

The id of the dataPackage which uniquely identifies the dataPackage in this application.

institutions

private List<*I18nString*> **`institutions`**

The names of the institutions which have performed this dataPackage. It must be specified in German and English and it must not contain more than 512 characters.

masterId

private String **masterId**

The master id of the dataPackage. The master id must not be empty, must be of the form `stu-{{dataAcquisitionProjectId}}$` and the master id must not contain more than 512 characters.

projectContributors

private List<*Person*> **projectContributors**

List of `Person`s which have performed this dataPackage. Must not be empty.

serialVersionUID

private static final long **serialVersionUID**

sponsor

private *I18nString* **sponsor**

The name of the sponsor who which has sponsored this dataPackage. 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 dataPackages to which this dataPackage 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 `DataPackage`. Must be one of `SurveyDesigns` and must not be empty.

tags

private *Tags* **tags**

Keywords for the dataPackage.

title

private *I18nString* **title**

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

DataPackageAttachmentMetadata

public class **DataPackageAttachmentMetadata** extends *AbstractShadowableRdcDomainObject*

Metadata which will be stored with each attachment of a `DataPackage`.

Fields

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**

The id of the `DataAcquisitionProject` to which the `DataPackage` of this attachment belongs. Must not be empty.

dataPackageId

private String **dataPackageId**

The id of the `DataPackage` 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.

indexInDataPackage

private Integer **indexInDataPackage**

The index in the `DataPackage` of this attachment. Used for sorting the attachments of this `DataPackage`. 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.

masterId

private String **masterId**
The master id of the dataPackage attachment.

serialVersionUID

private static final long **serialVersionUID**

title

private String **title**
An optional title of this attachment in the attachments' language. Must not be empty and it must not contain more than 2048 characters.

type

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

DataPackageAttachmentTypes

public class **DataPackageAttachmentTypes**
All valid types of a `DataPackageAttachmentMetadata`.

Fields

ALL

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

METHOD_REPORT

public static final *I18nString* **METHOD_REPORT**

OTHER

public static final *I18nString* **OTHER**

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**

Tags

public class **Tags** implements Serializable
Contains tags associated with a dataPackage.

Fields

de

private Set<String> **de**
German tags. At least one tag must be provided.

en

private Set<String> **en**
English tags (optional).

serialVersionUID

private static final long **serialVersionUID**

TimeMethods

public class **TimeMethods**
Describes the time dimension of the data collection. Used by DARA as time dimension and harvested by the VFDB.

Fields

CROSSSECTION

public static final String **CROSSSECTION**

CROSSSECTIONADHOCFOLLOWUP

public static final String **CROSSSECTIONADHOCFOLLOWUP**

LONGITUDINAL

public static final String **LONGITUDINAL**

LONGITUDINAL_COHORTEVENTBASED

public static final String **LONGITUDINAL_COHORTEVENTBASED**

LONGITUDINAL_PANEL

public static final String **LONGITUDINAL_PANEL**

LONGITUDINAL_PANEL_CONTINOUS

public static final String **LONGITUDINAL_PANEL_CONTINOUS**

LONGITUDINAL_PANEL_INTERVAL

public static final String **LONGITUDINAL_PANEL_INTERVAL**

LONGITUDINAL_TRENDREPEATEDCROSSSECTION

public static final String **LONGITUDINAL_TRENDREPEATEDCROSSSECTION**

Other

public static final String **Other**

TIMESERIES

public static final String **TIMESERIES**

TIMESERIES_CONTINOUS

public static final String **TIMESERIES_CONTINOUS**

TIMESERIES_DISCRETE

public static final String **TIMESERIES_DISCRETE**

3.25.4 eu.dzhw.fdz.metadatamanagement.datasetmanagement.domain

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

DataFormat

public enum **DataFormat**
File format for `SubDataSet`.

Enum Constants

R

public static final *DataFormat* **R**

SPSS

public static final *DataFormat* **SPSS**

Stata

public static final *DataFormat* **Stata**

Word

public static final *DataFormat* **Word**

DataSet

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

Fields

annotations

private *I18nString* **annotations**
Arbitrary additional text for the dataset. Markdown is supported. 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.

dataPackageId

private String **dataPackageId**
The id of the `OrderedDataPackage` to which this dataset belongs. 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.

languages

private List<String> **languages**
The languages used in the dataset. Must be specified as ISO 639 language codes.

masterId

private String **masterId**
The master id of this dataset. It must not contain more than 512 characters, must not be empty and must be of the form `dat-{{dataAcquisitionProjectId}}-ds{{number}}$`.

number

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

serialVersionUID

private static final long **serialVersionUID**

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.

DataSetAttachmentMetadata

public class **DataSetAttachmentMetadata** extends *AbstractShadowableRdcDomainObject*

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.

masterId

private String **masterId**

serialVersionUID

private static final long **serialVersionUID**

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.

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**

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**

SubDataSet

public class **SubDataSet** implements Serializable

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`.

dataFormats

private Set<*DataFormat*> **dataFormats**

Set of available file formats of the `SubDataSet`.

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.

serialVersionUID

private static final long **serialVersionUID**

3.25.5 eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain

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

CollectionModes

public class **CollectionModes**

The procedure, technique, or mode of inquiry used to attain the data. Used by DARA as collection mode type and harvested by the VFDB.

Fields

AGGREGATION

public static final String **AGGREGATION**

COMPILATIONSYNTHESIS

public static final String **COMPILATIONSYNTHESIS**

CONTENTCODING

public static final String **CONTENTCODING**

EXPERIMENT

public static final String **EXPERIMENT**

EXPERIMENT_FIELDINTERVENTION

public static final String **EXPERIMENT_FIELDINTERVENTION**

EXPERIMENT_LABORATORY

public static final String **EXPERIMENT_LABORATORY**

EXPERIMENT_WEBBASED

public static final String **EXPERIMENT_WEBBASED**

FOCUSGROUP

public static final String **FOCUSGROUP**

FOCUSGROUP_FACETOFACE

public static final String **FOCUSGROUP_FACETOFACE**

FOCUSGROUP_ONLINE

public static final String **FOCUSGROUP_ONLINE**

FOCUSGROUP_TELEPHONE

public static final String **FOCUSGROUP_TELEPHONE**

INTERVIEW

public static final String **INTERVIEW**

INTERVIEW_EMAIL

public static final String **INTERVIEW_EMAIL**

INTERVIEW_FACETOFACE

public static final String **INTERVIEW_FACETOFACE**

INTERVIEW_FACETOFACE_CAPICAMI

public static final String **INTERVIEW_FACETOFACE_CAPICAMI**

INTERVIEW_FACETOFACE_PAPI

public static final String **INTERVIEW_FACETOFACE_PAPI**

INTERVIEW_TELEPHONE

public static final String **INTERVIEW_TELEPHONE**

INTERVIEW_TELEPHONE_CATI

public static final String **INTERVIEW_TELEPHONE_CATI**

INTERVIEW_WEBBASED

public static final String **INTERVIEW_WEBBASED**

MEASUREMENTSANDTESTS

public static final String **MEASUREMENTSANDTESTS**

MEASUREMENTSANDTESTS_EDUCATIONAL

public static final String **MEASUREMENTSANDTESTS_EDUCATIONAL**

MEASUREMENTSANDTESTS_PHYSICAL

public static final String **MEASUREMENTSANDTESTS_PHYSICAL**

MEASUREMENTSANDTESTS_PSYCHOLOGICAL

public static final String **MEASUREMENTSANDTESTS_PSYCHOLOGICAL**

OBSERVATION

public static final String **OBSERVATION**

OBSERVATION_COMPUTERBASED

public static final String **OBSERVATION_COMPUTERBASED**

OBSERVATION_FIELD

public static final String **OBSERVATION_FIELD**

OBSERVATION_FIELD_NONPARTICIPANT

public static final String **OBSERVATION_FIELD_NONPARTICIPANT**

OBSERVATION_FIELD_PARTICIPANT

public static final String **OBSERVATION_FIELD_PARTICIPANT**

OBSERVATION_LABARATORY

public static final String **OBSERVATION_LABARATORY**

OBSERVATION_LABARATORY_NONPARTICIPANT

public static final String **OBSERVATION_LABARATORY_NONPARTICIPANT**

OBSERVATION_LABARATORY_PARTICIPANT

public static final String **OBSERVATION_LABARATORY_PARTICIPANT**

OTHER

public static final String **OTHER**

RECORDING

public static final String **RECORDING**

SELFADMINISTEREDQUESTIONNAIRE

public static final String **SELFADMINISTEREDQUESTIONNAIRE**

SELFADMINISTEREDQUESTIONNAIRE_COMPUTERASSISTED

public static final String **SELFADMINISTEREDQUESTIONNAIRE_COMPUTERASSISTED**

SELFADMINISTEREDQUESTIONNAIRE_EMAIL

public static final String **SELFADMINISTEREDQUESTIONNAIRE_EMAIL**

SELFADMINISTEREDQUESTIONNAIRE_PAPER

public static final String **SELFADMINISTEREDQUESTIONNAIRE_PAPER**

SELFADMINISTEREDQUESTIONNAIRE_SMSORMMS

public static final String **SELFADMINISTEREDQUESTIONNAIRE_SMSORMMS**

SELFADMINISTEREDQUESTIONNAIRE_WEBBASED

public static final String **SELFADMINISTEREDQUESTIONNAIRE_WEBBASED**

SELFADMINISTEREDWRITINGSANDDIARIES

public static final String **SELFADMINISTEREDWRITINGSANDDIARIES**

SELFADMINISTEREDWRITINGSANDDIARIES_EMAIL

public static final String **SELFADMINISTEREDWRITINGSANDDIARIES_EMAIL**

SELFADMINISTEREDWRITINGSANDDIARIES_PAPER

public static final String **SELFADMINISTEREDWRITINGSANDDIARIES_PAPER**

SELFADMINISTEREDWRITINGSANDDIARIES_WEBBASED

public static final String **SELFADMINISTEREDWRITINGSANDDIARIES_WEBBASED**

SIMULATION

public static final String **SIMULATION**

SUMMARY

public static final String **SUMMARY**

TRANSCRIPTION

public static final String **TRANSCRIPTION**

Instrument

public class **Instrument** extends *AbstractShadowableRdcDomainObject*

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. Markdown is supported. Must not contain more than 2048 characters.

conceptIds

private List<String> **conceptIds**
List of ids of `Concept`s to which are covered by this instrument.

dataAcquisitionProjectId

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

dataPackageId

private String **dataPackageId**
The id of the `OrderedDataPackage` to which this instrument belongs. 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.

masterId

private String **masterId**
The instrument's master id. It must not be empty, must be of the form `ins-{{dataAcquisitionProjectId}}-ins{{number}}$` and 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`.

originalLanguages

private List<String> **originalLanguages**

The languages of the instrument during the data collection. Must be specified as ISO 639 language codes.

serialVersionUID

private static final long **serialVersionUID**

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.

InstrumentAttachmentMetadata

public class **InstrumentAttachmentMetadata** extends *AbstractShadowableRdcDomainObject*

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.

masterId

private String **masterId**
The master id of the instrument attachment metadata.

serialVersionUID

private static final long **serialVersionUID**

type

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

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**

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**

INTERVIEW

public static final String **INTERVIEW**

PAPI

public static final String **PAPI**

3.25.6 eu.dzhw.fdz.metadatamanagement.ordermanagement.domain

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

Order

public class **Order** extends *AbstractRdcDomainObject*

Order (DTO) containing all relevant information for ordered `Product`s.

Fields

client

private *OrderClient* **client**

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

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.

serialVersionUID

private static final long **serialVersionUID**

state

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

OrderAlreadyCompletedException

public class **OrderAlreadyCompletedException** extends IllegalArgumentException
Orders with `OrderState.ORDERED` must not be updated. This exception should be thrown whenever an update attempt is made on such orders.

Fields

serialVersionUID

private static final long **serialVersionUID**

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**

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**

ORDERED

public static final *OrderState* **ORDERED**

OrderedDataPackage

public class **OrderedDataPackage** implements Serializable

Partial `eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain.DataPackage` which is part of a `Product`. It is a copy of the `eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain.DataPackage` attributes which is made when the customer places the orders.

Fields

annotations

private *I18nString* **annotations**

The annotations of the `eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain.DataPackage`.

id

private String **id**

The id of the `eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain.DataPackage`. Must not be empty.

serialVersionUID

private static final long **serialVersionUID**

studySeries

private *I18nString* **studySeries**
The name of the series of dataPackages to which this dataPackage belongs. May be null.

surveyDataTypes

private List<*I18nString*> **surveyDataTypes**
List of `DataTypes`. Must not be empty.

title

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

Product

public class **Product** implements Serializable
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.

dataFormats

private Set<*DataFormat*> **dataFormats**
The available data formats of the dataPackage. It must not be empty.

serialVersionUID

private static final long **serialVersionUID**

study

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

version

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

3.25.7 eu.dzhw.fdz.metadatamanagement.projectmanagement.domain

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

AssigneeGroup

public enum **AssigneeGroup**
Possible assignee groups.

Enum Constants

DATA_PROVIDER

public static final *AssigneeGroup* **DATA_PROVIDER**

PUBLISHER

public static final *AssigneeGroup* **PUBLISHER**

Configuration

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

Fields

dataPackagesState

private *ProjectState* **dataPackagesState**
The state of the dataPackage.

dataProviders

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

dataSetsState

private *ProjectState* **dataSetsState**
The state of data sets.

instrumentsState

private *ProjectState* **instrumentsState**
The state of instruments.

publicationsState

private *ProjectState* **publicationsState**
The state of related publications.

publishers

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

questionsState

private *ProjectState* **questionsState**
The state of questions.

requirements

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

serialVersionUID

private static final long **serialVersionUID**

surveysState

private *ProjectState* **surveysState**
The State of surveys.

variablesState

private *ProjectState* **variablesState**
The state of variables.

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.

serialVersionUID

private static final long **serialVersionUID**

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.

DataAcquisitionProject

public class **DataAcquisitionProject** extends *AbstractShadowableRdcDomainObject* implements Serializable
The data acquisition project collects the metadata for the data products which are published by our RDC. One project can contain one `DataPackage`, 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

assigneeGroup

private *AssigneeGroup* **assigneeGroup**
Determines which assignee group is able to edit data on the project.

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

lastAssigneeGroupMessage

private String **lastAssigneeGroupMessage**
The last message provided by an assignee group user before `DataAcquisitionProject.assigneeGroup` value changed.

masterId

private String **masterId**
The master id of this project. Must not be empty, must only contain lower cased (english) letters and numbers and 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.

serialVersionUID

private static final long **serialVersionUID**

FreeResourceTypes

public class **`FreeResourceTypes`**
Resource Types as they are harvested from DARA by the VFDB.

Fields

MIXED_DATA

public static final *I18nString* **`MIXED_DATA`**

QUALITATIVE_DATA

public static final *I18nString* **`QUALITATIVE_DATA`**

SURVEY_DATA

public static final *I18nString* **`SURVEY_DATA`**

ProjectState

public class **`ProjectState`** implements Serializable
State of a data acquisition project. Used for all metadata

Author tgehrke

Fields

isDataProviderReady

private boolean **`isDataProviderReady`**
indicates if the data providers marked it's metadata as ready.

isPublisherReady

private boolean **`isPublisherReady`**
indicates if the publisher marked the metadata as ready.

serialVersionUID

private static final long **`serialVersionUID`**

Release

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

Fields

firstDate

private LocalDateTime **firstDate**

The timestamp (in UTC) indicates when a publisher has released the `DataAcquisitionProject` with the current version for the first time. Will be generated by the server and will not be empty.

lastDate

private LocalDateTime **lastDate**

The timestamp (in UTC) indicates when a publisher has released the `DataAcquisitionProject` with the current version the last time. Must not be empty.

pinToStartPage

private Boolean **pinToStartPage**

Boolean which indicates whether the data package shall be displayed on the startpage or not.

serialVersionUID

private static final long **serialVersionUID**

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.

Requirements

public class **Requirements** implements Serializable

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

Fields

isDataPackagesRequired

private boolean **isDataPackagesRequired**

Defines if dataPackage data is required for a release (this object type is mandatory and this setting is therefore always `true`.

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.

isPublicationsRequired

private boolean **isPublicationsRequired**
Defines if publication data is required for a release.

isQuestionsRequired

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

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.

serialVersionUID

private static final long **serialVersionUID**

ShadowCopyQueueItem

public class **ShadowCopyQueueItem** extends *AbstractRdcDomainObject*
Represents a queued shadow copy task of a project.

Fields

action

private *Action* **action**
The action which will be performed for the shadow copies.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**
Project id for which a shadow copy should be created or hidden or unhidden.

id

private String **id**
Queue item id.

release

private *Release* **release**
The release object of the project which has been released.

serialVersionUID

private static final long **serialVersionUID**

updateStartedAt

private LocalDateTime **updateStartedAt**
Start time of the copy process.

ShadowCopyQueueItem.Action

public enum **Action**
The action which will be performed for the shadows.

Enum Constants

CREATE

public static final *ShadowCopyQueueItem.Action* **CREATE**

HIDE

public static final *ShadowCopyQueueItem.Action* **HIDE**

UNHIDE

public static final *ShadowCopyQueueItem.Action* **UNHIDE**

ShadowCopyReleaseToDaraNotAllowed

public class **ShadowCopyReleaseToDaraNotAllowed** extends IllegalArgumentException
Thrown if client attempts to release a shadowed project to dara.

Fields

serialVersionUID

private static final long **`serialVersionUID`**

ShadowHidingNotAllowedException

public class **`ShadowHidingNotAllowedException`** extends Exception
Exception which is thrown when the user tries to hide a master or the most recent shadow copy.

Author René Reitmann

Fields

serialVersionUID

private static final long **`serialVersionUID`**

ShadowUnhidingNotAllowedException

public class **`ShadowUnhidingNotAllowedException`** extends Exception
Exception which is thrown when the user tries to unhide shadows which are already unhidden.

Author René Reitmann

Fields

serialVersionUID

private static final long **`serialVersionUID`**

3.25.8 eu.dzhw.fdz.metadatamanagement.questionmanagement.domain

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

ImageType

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

Enum Constants

PNG

public static final *ImageType* **`PNG`**

Question

public class **Question** extends *AbstractShadowableRdcDomainObject*

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. Markdown is supported. Must not contain more than 2048 characters.

conceptIds

private List<String> **conceptIds**

List of ids of `Concept`s to which this question belongs.

dataAcquisitionProjectId

private String **dataAcquisitionProjectId**

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

dataPackageId

private String **dataPackageId**

The id of the `OrderedDataPackage` to which this question belongs. Must not be empty.

id

private String **id**

The id of the question which uniquely identifies the question in this application.

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.

masterId

private String **masterId**

The master id of the question. It must not be empty, must be of the form `que-{{dataAcquisitionProjectId}}-ins{{instrumentNumber}}-{{number}}$` and must not contain more than 512 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.

serialVersionUID

private static final long **serialVersionUID**

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.

QuestionImageMetadata

public class **QuestionImageMetadata** extends *AbstractShadowableRdcDomainObject*

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 „.“.

id

private String **id**

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.

masterId

private String **masterId**

The master id of question image metadata.

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.

serialVersionUID

private static final long **serialVersionUID**

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**

UNDOCUMENTED

public static final *I18nString* **UNDOCUMENTED**

TechnicalRepresentation

public class **TechnicalRepresentation** implements Serializable
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.

serialVersionUID

private static final long **serialVersionUID**

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.

3.25.9 eu.dzhw.fdz.metadatamanagement.relatedpublicationmanagement.domain

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

author Daniel Katzberg

RelatedPublication

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

Author Daniel Katzberg

Fields

abstractSource

private *I18nString* **abstractSource**

annotations

private *I18nString* **annotations**
Arbitrary additional text for this publication. Markdown is supported. Must not contain more than 2048 characters.

authors

private String **authors**

dataPackageIds

private List<String> **dataPackageIds**

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**

serialVersionUID

private static final long **serialVersionUID**

sourceLink

private String **sourceLink**

sourceReference

private String **sourceReference**

studySerieses

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

surveyIds

private List<String> **surveyIds**

title

private String **title**

variableIds

private List<String> **variableIds**

year

private Integer **year**

3.25.10 eu.dzhw.fdz.metadatamanagement.surveymanagement.domain

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

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**

GeographicCoverage

public class **GeographicCoverage** implements Serializable

Contains data regarding the location where survey data was collected.

Fields

country

private String **country**

ISO 3166-1 alpha-2 country code.

description

private *I18nString* **description**

Free text description for additional information regarding the location. Markdown is supported.

serialVersionUID

private static final long **serialVersionUID**

Population

public class **Population** implements Serializable

Details of the population of a `Survey`.

Fields

description

private *I18nString* **description**

A description of the population. Markdown is supported. It must be specified in all languages and it must not contain more than 2048 characters.

geographicCoverages

private List<*GeographicCoverage*> **geographicCoverages**

A list of geographic coverages. Must contain at least one entry.

serialVersionUID

private static final long **serialVersionUID**

unit

private *I18nString* **unit**

Unit type. Mandatory field which only allows values specified by VFDB.

See also: GNERD: Survey Unit Educational Research (Version 1.0)

Survey

public class **Survey** extends *AbstractShadowableRdcDomainObject*

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. Markdown is supported. 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.

dataPackageId

private String **dataPackageId**

The id of the `DataPackage` to which this survey belongs. 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.

masterId

private String **masterId**

The master id of the survey. It must not be empty, must be of the form `sur-{{dataAcquisitionProjectId}}-sy{{number}}$` and 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 match the controlled vocabulary specified by VFDB.

See also: ‘ Catalog: GNERD: Sampling Procedure Educational Research (Version 1.0) <https://mdr.iqb.hu-berlin.de/#/catalog/1d791cc7-6d8d-dd35-b1ef-0eec9c31bbb5“>‘_

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.

serialVersionUID

private static final long **serialVersionUID**

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 `DataPackage` is not organized in waves. Must not be empty and must be greater than or equal to 1.

SurveyAttachmentMetadata

public class **SurveyAttachmentMetadata** extends *AbstractShadowableRdcDomainObject*

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.

masterId

private String **masterId**

serialVersionUID

private static final long **serialVersionUID**

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. Must not be empty and it must not contain more than 2048 characters.

SurveyResponseRateImageMetadata

public class **SurveyResponseRateImageMetadata** extends *AbstractShadowableRdcDomainObject*

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“.

masterId

private String **masterId**

The master id of the survey response rate image metadata.

serialVersionUID

private static final long **serialVersionUID**

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.

SurveySampleTypeProvider

public class **SurveySampleTypeProvider**

Sample types for survey. This list is based on official VFDB vocabulary.

See also: Catalog: GNERD: Sampling Procedure Educational Research (Version 1.0)

Fields

CONTROLLED_SAMPLE_VOCABULARY

private static final Set<*I18nString*> **CONTROLLED_SAMPLE_VOCABULARY**

UnitValueProvider

public class **UnitValueProvider**
Provides valid unit values for Survey.

Fields

UNIT_VALUES

private static final Set<*I18nString*> **UNIT_VALUES**

3.25.11 eu.dzhw.fdz.metadatamanagement.variablemanagement.domain

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

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**

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**

Distribution

public class **Distribution** implements Serializable

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.

serialVersionUID

private static final long **serialVersionUID**

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.

FilterDetails

public class **FilterDetails** implements Serializable

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. Markdown is supported. 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. Can 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. Can be empty if and only if expression is empty. If present must be one of `FilterExpressionLanguages`.

serialVersionUID

private static final long **serialVersionUID**

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**

GenerationDetails

public class **GenerationDetails** implements Serializable
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. Markdown is supported. 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`.

serialVersionUID

private static final long **serialVersionUID**

Missing

public class **Missing** implements Serializable
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.

serialVersionUID

private static final long **serialVersionUID**

RelatedQuestion

public class **RelatedQuestion** implements Serializable

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*.

serialVersionUID

private static final long **`serialVersionUID`**

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`**

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**

Statistics

public class **Statistics** implements Serializable

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.

serialVersionUID

private static final long **serialVersionUID**

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.

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**

ValidResponse

public class **ValidResponse** implements Serializable

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.

serialVersionUID

private static final long **serialVersionUID**

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.

Variable

public class **Variable** extends *AbstractShadowableRdcDomainObject*

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. Markdown is supported. 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.

dataPackageId

private String **dataPackageId**

Id of the `DataPackage` to which this variable belongs.

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.

masterId

private String **masterId**

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`.

serialVersionUID

private static final long **serialVersionUID**

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.

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*), 136, 143
AbstractRdcDomainObject (*Java class*), 73
AbstractShadowableRdcDomainObject (*Java class*), 73
abstractSource (*Java field*), 123
accessWay (*Java field*), 95, 108
AccessWays (*Java class*), 132
accessWays (*Java field*), 144
Action (*Java enum*), 116
action (*Java field*), 115
additionalQuestionText (*Java field*), 118
AGGREGATION (*Java field*), 96
ALL (*Java field*), 83, 88, 89, 94, 95, 104, 105, 122, 125, 132, 133, 135, 138, 141
annotations (*Java field*), 85, 91, 101, 107, 118, 123, 127, 144
ANY (*Java field*), 141
AssigneeGroup (*Java enum*), 109
assigneeGroup (*Java field*), 112
authors (*Java field*), 81, 124

B

BUILTIN (*Java field*), 141
BYTECODE (*Java field*), 141

C

CAPI (*Java field*), 105
CATI (*Java field*), 105
CAWI (*Java field*), 105
CHAR (*Java field*), 141
CHARACTER (*Java field*), 141
citationHint (*Java field*), 81
client (*Java field*), 105
CLOSURE (*Java field*), 141
code (*Java field*), 75, 136
CollectionModes (*Java class*), 96
COMPILATIONSYNTHESIS (*Java field*), 96
COMPLEX (*Java field*), 141
Concept (*Java class*), 81
ConceptAttachmentMetadata (*Java class*), 82
ConceptAttachmentTypes (*Java class*), 83
conceptId (*Java field*), 82
conceptIds (*Java field*), 101, 118
ConceptInUseException (*Java class*), 84
Configuration (*Java class*), 109
configuration (*Java field*), 112
containsAnnotations (*Java field*), 120
CONTENTCODING (*Java field*), 96
CONTROLLED_SAMPLE_VOCABULARY (*Java field*), 132
Counter (*Java class*), 74
Country (*Java class*), 74
country (*Java field*), 126
CREATE (*Java field*), 116
CREATED (*Java field*), 107
createdBy (*Java field*), 73
createdDate (*Java field*), 73
CROSS_SECTION (*Java field*), 89
CROSSSECTION (*Java field*), 89
CROSSSECTIONADHOCFOLLOWUP (*Java field*), 90

D

DaraUpdateQueueItem (*Java class*), 111
data provider, 3, 36
DATA_PROVIDER (*Java field*), 109
DATA_SET_REPORT (*Java field*), 80
DataAcquisitionProject (*Java class*), 111
dataAcquisitionProjectId (*Java field*), 85, 87, 91, 93, 101, 103, 108, 115, 118, 120, 127, 129, 131, 144
dataCurators (*Java field*), 85
DataFormat (*Java enum*), 91
dataFormats (*Java field*), 95, 108
DataPackage (*Java class*), 85
DataPackageAttachmentMetadata (*Java class*), 87
DataPackageAttachmentTypes (*Java class*), 88

dataPackageId (*Java field*), 87, 92, 101, 118, 127, 144
dataPackageIds (*Java field*), 124
dataPackagesState (*Java field*), 109
dataProviders (*Java field*), 110
DataSet (*Java class*), 91
DataSetAttachmentMetadata (*Java class*), 93
dataSetId (*Java field*), 93, 144
dataSetIds (*Java field*), 124
dataSetNumber (*Java field*), 93, 145
dataSetsState (*Java field*), 110
DataSetTypes (*Java class*), 94
dataType (*Java field*), 127, 145
DataTypes (*Java class*), 125, 133
DATE (*Java field*), 133
Datenaufnahme, 3, 36
de (*Java field*), 75, 84, 89
derivedVariablesIdentifier (*Java field*), 145
description (*Java field*), 81, 82, 85, 87, 92, 93, 96, 101, 103, 126, 129, 135, 136
deviance (*Java field*), 139
displayName (*Java field*), 76
Distribution (*Java class*), 133
distribution (*Java field*), 145
DLP (*Java field*), 106
DOCUMENTATION (*Java field*), 83
doi (*Java field*), 81, 124
domainObjectId (*Java field*), 80
DONE (*Java field*), 79
doNotDisplayThousandsSeparator (*Java field*), 145
DOTDOTDOT (*Java field*), 141
DOUBLE (*Java field*), 142
DOWNLOAD_CUF (*Java field*), 132
DOWNLOAD_SUF (*Java field*), 132

E

en (*Java field*), 75, 84, 89
end (*Java field*), 76
ENVIRONMENT (*Java field*), 142
EPISODE_RECORD (*Java field*), 95
errorList (*Java field*), 78
errorMessage (*Java field*), 80
eu.dzhw.fdz.metadatamanagement.common.domain (*package*), 72
eu.dzhw.fdz.metadatamanagement.conceptmanagement.domain (*package*), 81
eu.dzhw.fdz.metadatamanagement.datapackagemanagement.domain (*package*), 85
eu.dzhw.fdz.metadatamanagement.datasetmanagement.domain (*package*), 91
eu.dzhw.fdz.metadatamanagement.instrumentmanagement.domain (*package*), 96
eu.dzhw.fdz.metadatamanagement.ordermanagement.doma (*package*), 105
eu.dzhw.fdz.metadatamanagement.projectmanagement.do (*package*), 109
eu.dzhw.fdz.metadatamanagement.questionmanagement.d (*package*), 117
eu.dzhw.fdz.metadatamanagement.relatedpublicationma (*package*), 123
eu.dzhw.fdz.metadatamanagement.surveymanagement.dom (*package*), 125
eu.dzhw.fdz.metadatamanagement.variablemanagement.d (*package*), 132
EXPERIMENT (*Java field*), 97
EXPERIMENT_FIELDINTERVENTION (*Java field*), 97
EXPERIMENT_LABORATORY (*Java field*), 97
EXPERIMENT_WEBBASED (*Java field*), 97
EXPRESSION (*Java field*), 142
expression (*Java field*), 135
expressionLanguage (*Java field*), 135
EXTERNALPTR (*Java field*), 142

F

FAILURE (*Java field*), 79
fieldPeriod (*Java field*), 127
fileName (*Java field*), 82, 87, 94, 103, 121, 129, 131
FilterDetails (*Java class*), 134
filterDetails (*Java field*), 145
FilterExpressionLanguages (*Java class*), 135
firstDate (*Java field*), 114
firstName (*Java field*), 77
firstQuartile (*Java field*), 139
FOCUSGROUP (*Java field*), 97
FOCUSGROUP_FACETOFACE (*Java field*), 97
FOCUSGROUP_ONLINE (*Java field*), 97
FOCUSGROUP_TELEPHONE (*Java field*), 97
Format (*Java class*), 95
format (*Java field*), 92
FreeResourceTypes (*Java class*), 113

G

GenerationDetails (*Java class*), 135
generationDetails (*Java field*), 145
GeographicCoverage (*Java class*), 126
geographicCoverages (*Java field*), 126
GRID (*Java field*), 122
grossSampleSize (*Java field*), 127

H

hasBeenReleasedBefore (*Java field*), 112
heightY (*Java field*), 77
hidden (*Java field*), 74
HIDE (*Java field*), 116
highWhisker (*Java field*), 139

I

I18nString (*Java class*), 75
id (*Java field*), 74, 79, 81, 82, 85, 87, 92, 94, 101, 103, 106, 107, 111, 112, 116, 118, 121, 124, 128, 129, 131, 145
ImageType (*Java enum*), 117
imageType (*Java field*), 121
ImmutableI18nString (*Java class*), 75
indexInConcept (*Java field*), 83
indexInDataPackage (*Java field*), 87
indexInDataSet (*Java field*), 94, 146
indexInInstrument (*Java field*), 103, 118
indexInQuestion (*Java field*), 121
indexInSurvey (*Java field*), 130
institutions (*Java field*), 85
instruction (*Java field*), 119
Instrument (*Java class*), 100
INSTRUMENT (*Java field*), 83
InstrumentAttachmentMetadata (*Java class*), 102
InstrumentAttachmentTypes (*Java class*), 104
instrumentId (*Java field*), 103, 119, 137
instrumentIds (*Java field*), 84, 124
instrumentNumber (*Java field*), 103, 119, 137
instrumentsState (*Java field*), 110
InstrumentTypes (*Java class*), 104
INTEGER (*Java field*), 142
INTERVAL (*Java field*), 138
INTERVIEW (*Java field*), 97, 105
INTERVIEW_EMAIL (*Java field*), 97
INTERVIEW_FACETOFACE (*Java field*), 97
INTERVIEW_FACETOFACE_CAPICAMI (*Java field*), 98
INTERVIEW_FACETOFACE_PAPI (*Java field*), 98
INTERVIEW_TELEPHONE (*Java field*), 98
INTERVIEW_TELEPHONE_CATI (*Java field*), 98
INTERVIEW_WEBBASED (*Java field*), 98
introduction (*Java field*), 119
isDataPackagesRequired (*Java field*), 114
isDataProviderReady (*Java field*), 113
isDataSetsRequired (*Java field*), 114
isInstrumentsRequired (*Java field*), 115
isPublicationsRequired (*Java field*), 115
isPublisherReady (*Java field*), 113
isQuestionsRequired (*Java field*), 115
isSurveysRequired (*Java field*), 115
isVariablesRequired (*Java field*), 115
ITEM_SET (*Java field*), 122

K

kurtosis (*Java field*), 139

L

label (*Java field*), 136, 143, 146
Language (*Java class*), 76
LANGUAGE (*Java field*), 142
language (*Java field*), 83, 87, 94, 103, 121, 123, 124, 130, 131
languageCode (*Java field*), 76
languageKey (*Java field*), 106
languages (*Java field*), 92
lastAssigneeGroupMessage (*Java field*), 112
lastDate (*Java field*), 114
lastModifiedBy (*Java field*), 73
lastModifiedDate (*Java field*), 73
lastName (*Java field*), 77
license (*Java field*), 81
LIST (*Java field*), 142
location (*Java field*), 79
LOGICAL (*Java field*), 142
LONG (*Java field*), 95
LONGITUDINAL (*Java field*), 90
LONGITUDINAL_COHORTEVENTBASED (*Java field*), 90
LONGITUDINAL_PANEL (*Java field*), 90
LONGITUDINAL_PANEL_CONTINOUS (*Java field*), 90
LONGITUDINAL_PANEL_INTERVAL (*Java field*), 90
LONGITUDINAL_TRENDREPEATEDCROSSSECTION (*Java field*), 90
lowWhisker (*Java field*), 139

M

masterId (*Java field*), 86, 88, 92, 94, 101, 104, 112, 119, 121, 128, 130, 131, 146
maximum (*Java field*), 139
maxNumberOfDecimalPlaces (*Java field*), 133
MDM (*Java field*), 107
meanDeviation (*Java field*), 140
meanValue (*Java field*), 140
MEASUREMENTSANDTESTS (*Java field*), 98
MEASUREMENTSANDTESTS_EDUCATIONAL (*Java field*), 98
MEASUREMENTSANDTESTS_PHYSICAL (*Java field*), 98
MEASUREMENTSANDTESTS_PSYCHOLOGICAL (*Java field*), 98
median (*Java field*), 140
METHOD_REPORT (*Java field*), 88
middleName (*Java field*), 77
minimum (*Java field*), 140
Missing (*Java class*), 136
missings (*Java field*), 134
MIXED_DATA (*Java field*), 113
mode (*Java field*), 140
MULTIPLE_CHOICE (*Java field*), 122

N

name (*Java field*), 96, 146

NOMINAL (*Java field*), 138
NOT_ACCESSIBLE (*Java field*), 132
NOTIFIED (*Java field*), 107
NULL (*Java field*), 142
number (*Java field*), 92, 101, 119, 128
numberOfObservations (*Java field*), 96
NUMERIC (*Java field*), 133

O

OBSERVATION (*Java field*), 98
OBSERVATION_COMPUTERBASED (*Java field*), 98
OBSERVATION_FIELD (*Java field*), 99
OBSERVATION_FIELD_NONPARTICIPANT (*Java field*), 99
OBSERVATION_FIELD_PARTICIPANT (*Java field*), 99
OBSERVATION_LABARATORY (*Java field*), 99
OBSERVATION_LABARATORY_NONPARTICIPANT (*Java field*), 99
OBSERVATION_LABARATORY_PARTICIPANT (*Java field*), 99
onBehalfOf (*Java field*), 80
ONSITE_SUF (*Java field*), 133
OPEN (*Java field*), 122
Order (*Java class*), 105
OrderAlreadyCompletedException (*Java class*), 106
OrderClient (*Java enum*), 106
ORDERED (*Java field*), 107
OrderedDataPackage (*Java class*), 107
OrderState (*Java enum*), 107
ORDINAL (*Java field*), 139
originalLanguages (*Java field*), 81, 102
OTHER (*Java field*), 84, 88, 99, 104
Other (*Java field*), 90

P

PAIRLIST (*Java field*), 142
PANEL (*Java field*), 89
panelIdentifier (*Java field*), 146
PAPI (*Java field*), 105
Period (*Java class*), 76
Person (*Java class*), 77
PERSONAL_RECORD (*Java field*), 95
pinToStartPage (*Java field*), 114
PNG (*Java field*), 117
Population (*Java class*), 126
population (*Java field*), 128
Product (*Java class*), 108
products (*Java field*), 106
projectContributors (*Java field*), 86
projectId (*Java field*), 111
ProjectState (*Java class*), 113
Projekt anlegen, 36
PROMISE (*Java field*), 142
publicationAbstract (*Java field*), 124
publicationsState (*Java field*), 110
PUBLISHER (*Java field*), 109
publishers (*Java field*), 110

Q

QUALITATIVE_DATA (*Java field*), 113, 125
QUANTITATIVE_DATA (*Java field*), 126
Question (*Java class*), 118
QUESTION_FLOW (*Java field*), 104
questionId (*Java field*), 121, 137
questionIds (*Java field*), 84, 124
QuestionImageMetadata (*Java class*), 120
QUESTIONNAIRE (*Java field*), 104
questionNumber (*Java field*), 137
questionsState (*Java field*), 110
questionText (*Java field*), 119
QuestionTypes (*Java class*), 122

R

R (*Java field*), 91, 138
RATIO (*Java field*), 139
RAW (*Java field*), 143
RECORDING (*Java field*), 99
RelatedPublication (*Java class*), 123
RelatedQuestion (*Java class*), 137
relatedQuestions (*Java field*), 146
relatedQuestionStrings (*Java field*), 137
relatedVariables (*Java field*), 146
relativeFrequency (*Java field*), 137, 143
Release (*Java class*), 113
release (*Java field*), 112, 116
REMOTE_DESKTOP (*Java field*), 133
Requirements (*Java class*), 114
requirements (*Java field*), 110
Resolution (*Java class*), 77
resolution (*Java field*), 121
responseRate (*Java field*), 128
rule (*Java field*), 136
ruleExpressionLanguage (*Java field*), 136
RuleExpressionLanguages (*Java class*), 138
RUNNING (*Java field*), 79

S

S4 (*Java field*), 143
sample (*Java field*), 128
sampleSize (*Java field*), 128
scaleLevel (*Java field*), 147
ScaleLevels (*Java class*), 138
SELFADMINISTEREDQUESTIONNAIRE (*Java field*), 99
SELFADMINISTEREDQUESTIONNAIRE_COMPUTERASSISTED (*Java field*), 99

SELFADMINISTEREDQUESTIONNAIRE_EMAIL (*Java field*), 99
SELFADMINISTEREDQUESTIONNAIRE_PAPER (*Java field*), 100
SELFADMINISTEREDQUESTIONNAIRE_SMSORMMS (*Java field*), 100
SELFADMINISTEREDQUESTIONNAIRE_WEBBASED (*Java field*), 100
SELFADMINISTEREDWRITINGSANDDIARIES (*Java field*), 100
SELFADMINISTEREDWRITINGSANDDIARIES_EMAIL (*Java field*), 100
SELFADMINISTEREDWRITINGSANDDIARIES_PAPER (*Java field*), 100
SELFADMINISTEREDWRITINGSANDDIARIES_WEBBASED (*Java field*), 100
seq (*Java field*), 74
serialVersionUID (*Java field*), 73–79, 82–84, 86, 88, 89, 92, 94, 96, 102, 104, 106, 108–117, 119, 121, 123, 124, 126, 128, 130, 131, 134–138, 140, 143, 147
shadow (*Java field*), 74
ShadowCopyCreateNotAllowedException (*Java class*), 78
ShadowCopyDeleteNotAllowedException (*Java class*), 78
ShadowCopyQueueItem (*Java class*), 115
ShadowCopyReleaseToDaraNotAllowed (*Java class*), 116
ShadowCopySaveNotAllowedException (*Java class*), 78
ShadowHidingNotAllowedException (*Java class*), 117
ShadowUnhidingNotAllowedException (*Java class*), 117
SIMULATION (*Java field*), 100
SINGLE_CHOICE (*Java field*), 122
skewness (*Java field*), 140
source (*Java field*), 123
sourceLink (*Java field*), 124
sourceReference (*Java field*), 125
SPECIAL (*Java field*), 143
SPEL (*Java field*), 135
sponsor (*Java field*), 86
SPSS (*Java field*), 91
standardDeviation (*Java field*), 140
start (*Java field*), 76
STATA (*Java field*), 135, 138
Stata (*Java field*), 91
state (*Java field*), 79, 106
Statistics (*Java class*), 139
statistics (*Java field*), 134
storageType (*Java field*), 147
StorageTypes (*Java class*), 141
STRING (*Java field*), 133
study (*Java field*), 109
studySeries (*Java field*), 86, 108
studySerieses (*Java field*), 125
SubDataSet (*Java class*), 95
subDataSets (*Java field*), 92
subtitle (*Java field*), 102
successorId (*Java field*), 74
successorNumbers (*Java field*), 120
successors (*Java field*), 120
SUMMARY (*Java field*), 100
Survey (*Java class*), 127
SURVEY_DATA (*Java field*), 113
SurveyAttachmentMetadata (*Java class*), 129
SurveyDataTypes (*Java field*), 108
surveyDesign (*Java field*), 86
SurveyDesigns (*Java class*), 88
surveyId (*Java field*), 130, 131
surveyIds (*Java field*), 93, 102, 125, 147
surveyMethod (*Java field*), 129
surveyNumber (*Java field*), 130, 131
surveyNumbers (*Java field*), 93, 102, 147
SurveyResponseRateImageMetadata (*Java class*), 130
SurveySampleTypeProvider (*Java class*), 131
surveysState (*Java field*), 110
SYMBOL (*Java field*), 143

T

Tags (*Java class*), 84, 89
tags (*Java field*), 82, 86
Task (*Java class*), 78
TaskErrorNotification (*Java class*), 80
TaskState (*Java enum*), 79
TaskType (*Java enum*), 80
taskType (*Java field*), 80
TechnicalRepresentation (*Java class*), 122
technicalRepresentation (*Java field*), 120
thirdQuartile (*Java field*), 140
TimeMethods (*Java class*), 89
TIMESERIES (*Java field*), 90
TIMESERIES_CONTINOUS (*Java field*), 90
TIMESERIES_DISCRETE (*Java field*), 90
title (*Java field*), 82, 83, 86, 88, 94, 102, 108, 125, 129, 130
topic (*Java field*), 120
totalAbsoluteFrequency (*Java field*), 134
totalValidAbsoluteFrequency (*Java field*), 134
totalValidRelativeFrequency (*Java field*), 134
TRANSCRIPTION (*Java field*), 100
type (*Java field*), 79, 83, 88, 93, 102, 104, 120, 123

U

UNDOCUMENTED (*Java field*), 122

UNHIDE (*Java field*), 116
unit (*Java field*), 127
UNIT_VALUES (*Java field*), 132
UnitValueProvider (*Java class*), 132
updateStartedAt (*Java field*), 111, 116
updateStartedBy (*Java field*), 111

V

validRelativeFrequency (*Java field*), 144
ValidResponse (*Java class*), 143
validResponses (*Java field*), 134
value (*Java field*), 144
Variable (*Java class*), 144
VARIABLE_QUESTIONNAIRE (*Java field*), 104
variableIds (*Java field*), 125
variablesState (*Java field*), 111
version (*Java field*), 73, 109, 114

W

wave (*Java field*), 129
WEAKREF (*Java field*), 143
WIDE (*Java field*), 95
widthX (*Java field*), 77
Word (*Java field*), 91

Y

year (*Java field*), 125