Creating an ODK form

Instruction Manual

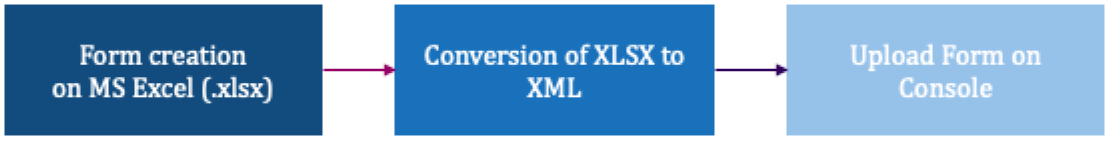
Table of Contents

1: GETTING STARTED

What is ODK?

The Open Data Kit software is an open source software that allows for collecting, managing, and using data in resource-constrained environments. It allows for the collection of data offline and submission of the data when internet connectivity is available. It allows users to aggregate data with full control over the collected data and the servers where this data is stored.

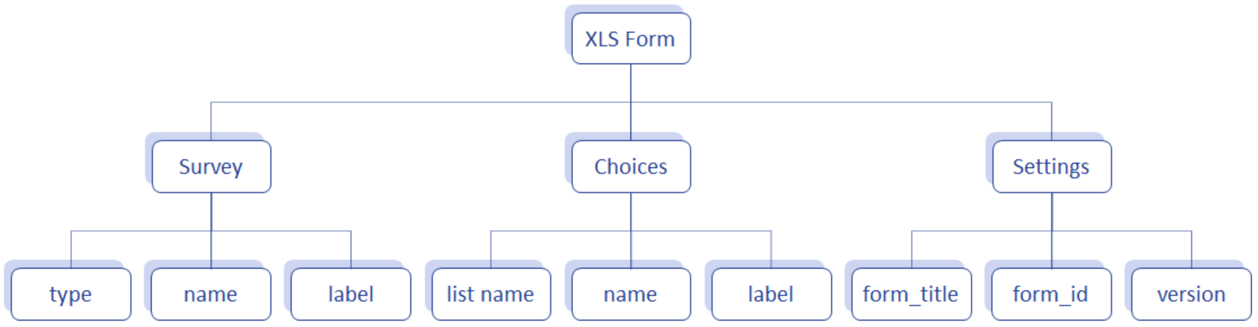
2: FORM CREATION ON ODK



Form Creation on MS Excel (.xlsx)

1. There is a common format that is to be followed to create the .xlsx form - Base XLS Form

2. There are **3 mandatory sheets** that need to be created in every excel form and each sheet as **certain mandatory columns** that need to be created within each sheet



The three mandatory sheets along with mandatory columns are as follows:

a. **Survey**: the three mandatory columns under this sheet are type, name and label
b. **Choices**: the three mandatory columns under this sheet are list name, name and label

c. **Settings**: the two mandatory columns under this sheet are form_title and form_id
d. **Some important rules**:
    i. The name of every sheet and column should be in lower case
    ii. Name of the excel file cannot start with a number
    iii. The name of the excel file cannot contain spaces

3. **Basic structure of the sheets**
    a. Survey sheet:
        i. Contains all the form contents such as the questions, the question type, the appearance of the questions, the constraints etc.
        ii. Column Descriptions -
            1. **Type**: enter the question type in this column (You can find all the different question types here)
            2. **Name**: give a unique name to each question, use lower case only and _ (underscore) as a separator
            3. **Label**: enter your question in this column
            4. Some additional useful columns are as follows:
                a. **Hint**: Enter instructions related to the question in this column
                b. **Required**: If the question is mandatory, enter "yes" in this column
                c. **Appearance**: commands related to appearance of the question appear in this column
        i. Additionally, for defining logic, operators and styling, refer to the following documentation -
            1. Form Logic
            2. Form Styling
            3. Form Operators

    b. Choices sheet:
        i. Contains the choices/options for all multiple choice questions
        ii. Column Descriptions -
            1. **List name**: Enter each of the list names that was created in the “survey” sheet
            2. **Name**: Give a unique name to each of your choices in the list
            3. **Label**: Enter each choice that will be visible to the user in this column

    c. Settings sheet:

i. Contains the form name, unique form id and form version

ii. Column Description -

1. **Form_title**: Enter the title of the form that will be displayed to the user
2. **Form_id**: Specifies the table name

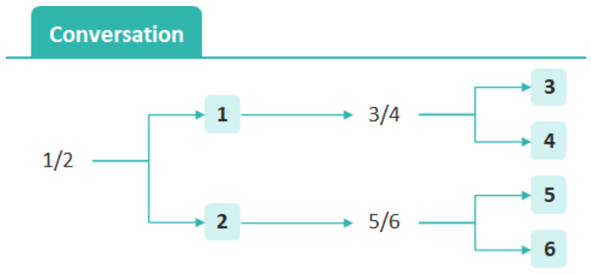
Other Details to Note

1. **Question Types used in UCI**
    a. **note**: for a note/text, doesn't produce a question
    b. **text**: for questions that have a free text response
    c. **integer**: for questions that have a whole number as response
    d. **select_one**: for MCQs that have only one answer. This is followed by the name of the list of options that will be presented to the user.
        a. In case of select_one, The name of the list cannot contain spaces, special characters other than – and _ (dash and underscore)

Note: ODK supports many more types of questions, however these are the question types that are currently enabled for the UCI use cases.

2. **Grouping Of Questions**
    a. This is useful to group questions together
    b. Begin group:
        i. Insert "begin group" row above the first question of the relevant group of questions
        ii. Assign name and label to the group
        iii. The text in the label column will appear above the group of questions
    c. End group:
        i. Insert "end group" row below the last question of the relevant group of questions
        ii. All other columns to be blank in the "end group" row

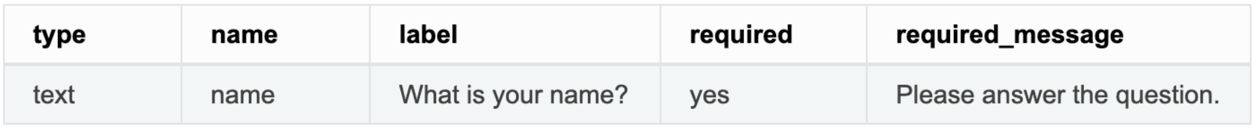
Note: Groups can be considered as a combination of 'Else IF' statements, where admin can define the complete trajectory of conversation based on user inputs.



3. **Validating And Constricting Responses**
    a. Adding contraints/validations -
        a. To validate or restrict response values, use the *'constraint'* column.
        b. You can add constraints in the following manner -
            i. Constraint expressions often use comparison operators and regular expressions.
            ii. The entered value of the response is represented in the expression with a single dot (.).
            iii. Examples -
                1. **. >= 18** (True if response is greater than or equal to 18)
                2. **. < 20 and . > 200** (True if the response is between 20 and 200)
                3. **.<${abc}** (True if the response is less than the response given for 'abc' question)

    b. Constraint message - The constraint expression will be evaluated when the user advances to the next screen. If the expression evaluates to True, the form advances as usual. If False, the form does not advance and the constraint_message is displayed.

4. **Requiring Responses**
    a. By default, users are able to skip questions in a form. To make a question required, put yes in the required column.
    b. Required questions are marked with a small asterisk to the left of the question label. You can optionally include a required_message which will be displayed to the user who tries to advance the form without answering the question.



type	name	label	required	required_message
text	name	What is your name?	yes	Please answer the question.

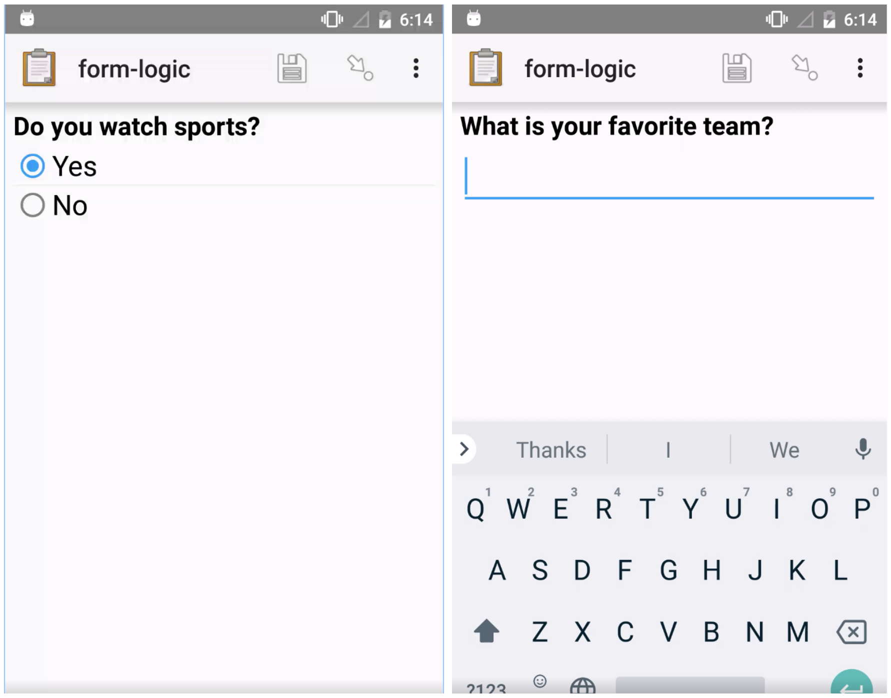
5. **Conditionally Showing Questions**
    a. The '*relevant'* column can be used to show or hide questions and groups of questions based on previous responses.
    b. If the expression in the relevant column evaluates to 'True', the question or group is shown. If 'False', the question is skipped.
    c. Often, comparison operators are used in relevant expressions. For example:
        i. **${age} <= 5** (True if age is five or less)

ii. **${has_children} = 'yes'** (True if the answer to has_children was yes)

d. Relevance expressions can also use functions. For example:

i. **selected(${allergies}, 'peanut')** (True if peanut was selected in the Multi select widget named allergies)

type	**name**	**label**	**relevant**
select_one yes_no	watch_sports	Do you watch sports?	
text	favorite_team	What is your favorite team?	${watch_sports} = 'yes'



form-logic

Do you watch sports?

[✓ Yes] No

form-logic

What is your favorite team?

6. Filtering Options In Select Questions

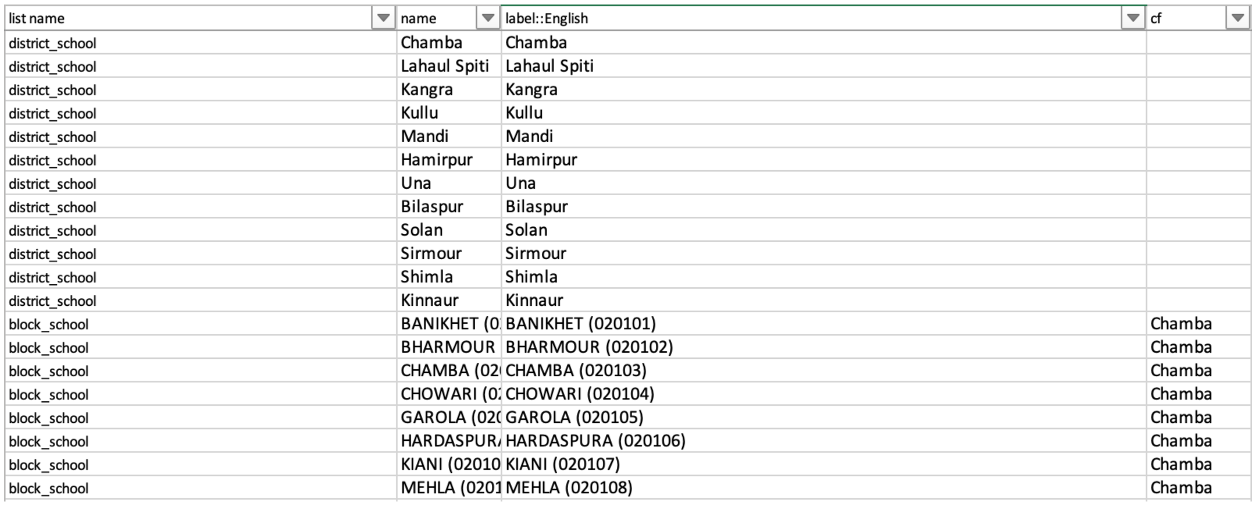
a. To limit the options in a select question based on the answer to a previous question, use a **choice_filter** row in the survey sheet, and filter key columns in the choices sheet.

b. For example, you might ask the user to select a state first, and then only display cities within that state. This is called a cascading select, and can be extended to any depth. This example form shows a two-tiered cascade: district and block

Survey :

type	name	label::English	choice_filter
begin group	**school_details**	**School Details**	
select_one district_school	district_school	District	
select_one block_school	block_school	Block	cf=${district_school}

Choices:

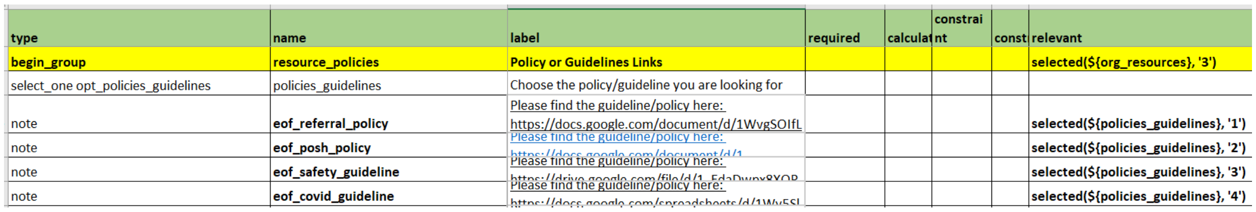


list name	name	label::English	cf
district_school	Chamba	Chamba	
district_school	Lahaul Spiti	Lahaul Spiti	
district_school	Kangra	Kangra	
district_school	Kullu	Kullu	
district_school	Mandi	Mandi	
district_school	Hamirpur	Hamirpur	
district_school	Una	Una	
district_school	Bilaspur	Bilaspur	
district_school	Solan	Solan	
district_school	Sirmour	Sirmour	
district_school	Shimla	Shimla	
district_school	Kinnaur	Kinnaur	
block_school	BANIKHET (0	BANIKHET (020101)	Chamba
block_school	BHARMOUR	BHARMOUR (020102)	Chamba
block_school	CHAMBA (02(	CHAMBA (020103)	Chamba
block_school	CHOWARI (0:	CHOWARI (020104)	Chamba
block_school	GAROLA (02(	GAROLA (020105)	Chamba
block_school	HARDASPUR/	HARDASPURA (020106)	Chamba
block_school	KIANI (02010	KIANI (020107)	Chamba
block_school	MEHLA (0201	MEHLA (020108)	Chamba

7. Additional instructions for creating a Bot form:

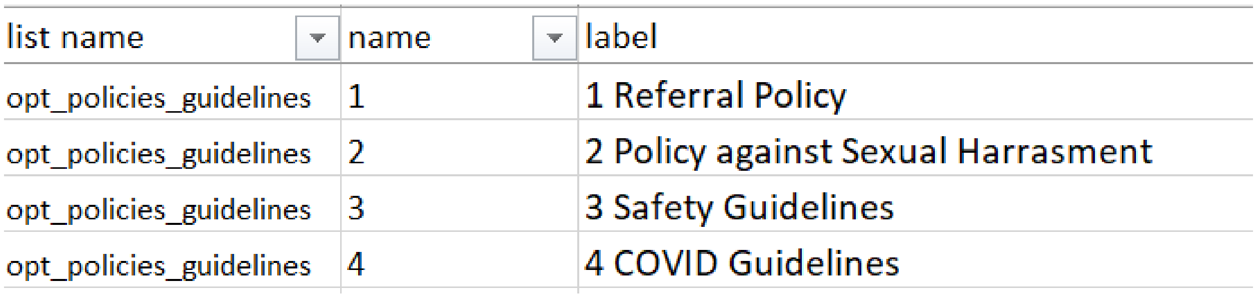
1. The last item in the branch *must be* of the question type text
2. The name of last item in the branch *must* start with 'eof_' (As shown in Reference Image 1)
3. In the choices sheet, the name of options *must be* the exact numerical value that we are asking users to select and the label for them should start with the same numerical value. (As shown in Reference Image 2)

Reference Image 1:



type	name	label	required	calculat	constrai nt	const	relevant
begin_group	**resource_policies**	**Policy or Guidelines Links**					**selected(${org_resources}, '3')**
select_one opt_policies_guidelines	policies_guidelines	Choose the policy/guideline you are looking for					
note	**eof_referral_policy**	Please find the guideline/policy here: https://docs.google.com/document/d/1WvgSOIfL					**selected(${policies_guidelines}, '1')**
note	**eof_posh_policy**	Please find the guideline/policy here: https://docs.google.com/document/d/1					**selected(${policies_guidelines}, '2')**
note	**eof_safety_guideline**	Please find the guideline/policy here: https://drive.google.com/file/d/1_FdaDwnx8XOP					**selected(${policies_guidelines}, '3')**
note	**eof_covid_guideline**	Please find the guideline/policy here: https://docs.google.com/spreadsheets/d/1WvFSL					**selected(${policies_guidelines}, '4')**

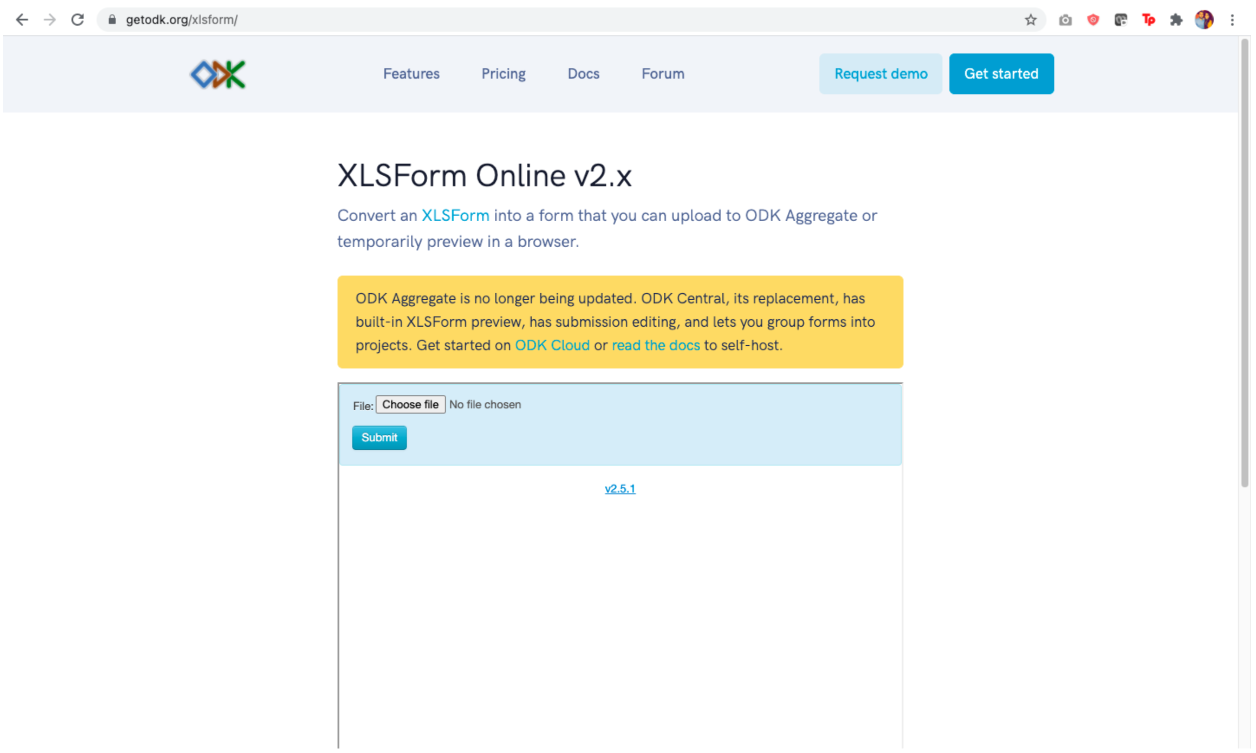
Reference Image 2:



list name	name	label
opt_policies_guidelines	1	1 Referral Policy
opt_policies_guidelines	2	2 Policy against Sexual Harrasment
opt_policies_guidelines	3	3 Safety Guidelines
opt_policies_guidelines	4	4 COVID Guidelines

3: FORM CONVERSION FROM XLS TO XML

a. Once the excel sheet is created and finalized, convert it to XML using this converter



XLSForm Online v2.x

Convert an XLSForm into a form that you can upload to ODK Aggregate or temporarily preview in a browser.

ODK Aggregate is no longer being updated. ODK Central, its replacement, has built-in XLSForm preview, has submission editing, and lets you group forms into projects. Get started on ODK Cloud or read the docs to self-host.

Sample XML would be as follows.

b. Click on 'Download XForm' to download the XML version of the ODK form

c. Click on 'Preview in Browser' to test the XML form