GLOBAL CHANGE PROGRAMME-JADAVPUR UNIVERSITY
WORKING PAPER SERIES

HOW TO USE OPEN DATA KIT (ODK):
A BRIEF TUTORIAL

Monalisa Ghosh
monalisaghosh16@gmail.com
and
Shyamasree Dasgupta
shyamasree.dasgupta@gmail.com

GCP/JU/15/02
Dated 23/11/2015

Global Change Programme
Jadavpur University
188 Raja S C Mallick Road, Kolkata 700032
Phone: +91 33 64147760

Website: http://www.juglobalchangeprogram.org

This working paper should be cited as:
Ghosh, M., Dasgupta, S., 2015. How to use Open Data Kit (ODK): A brief tutorial, Global Change Programme-Jadavpur University Working Paper # GCP/JU/15/02

Contents

1. What is Open Data Kit (ODK)?

Open Data Kit (ODK) enables users to capture and instantly digitize information of a variety of formats, eliminating the need for paper questionnaire surveys and data entry. It allows preparing a digitally programmed questionnaire, facilitates intense monitoring of the collection process and gathering of data immediately after survey in a format prepared for analysis. It thus eliminates the need for paper surveys and significantly reduces survey time and time required for data entry. The smartphones and similar devices are equipped to support ODK software.

2. Components of ODK

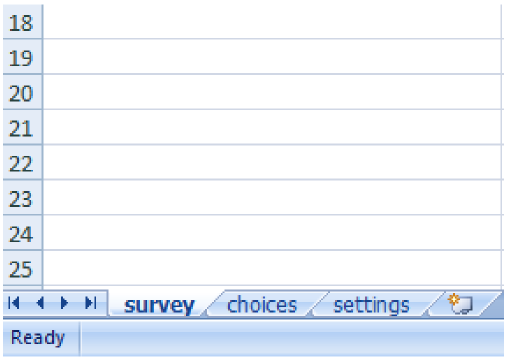
The ODK software suite consists of 3 different programs including **ODK Form, ODK Collect and ODK Aggregate.**

First we setup our server that is **ODK Aggregate** in our laptop. Then we search for the "**ODK Collect**" app from Google Play on the device. We install the latest version to the mobile device. For more information about ODK Collect and its requirements, visit the https://opendatakit.org/use/collect/. The questionnaires are subsequently saved to the phone's SD memory, where it can be accessed without internet connectivity. Questionnaire in ODK Collect is **ODK Form** which is an xml file. There are several ways to create forms for ODK. One of them used in our Project is XLSForm. After completion of XLSForm we convert it into xml file.

XLSForm is a standard form created to simplify the authoring of forms in Excel in a readable format even for non-technological persons. They are simple to get started with but also allow for complex XForms by someone familiar with the syntax described below. Here we have inscribed those syntaxes used in the context of our questionnaire. For further requirements, visit http://xlsform.org/.

2.1 ODK Form (Basic format of XLSForm)

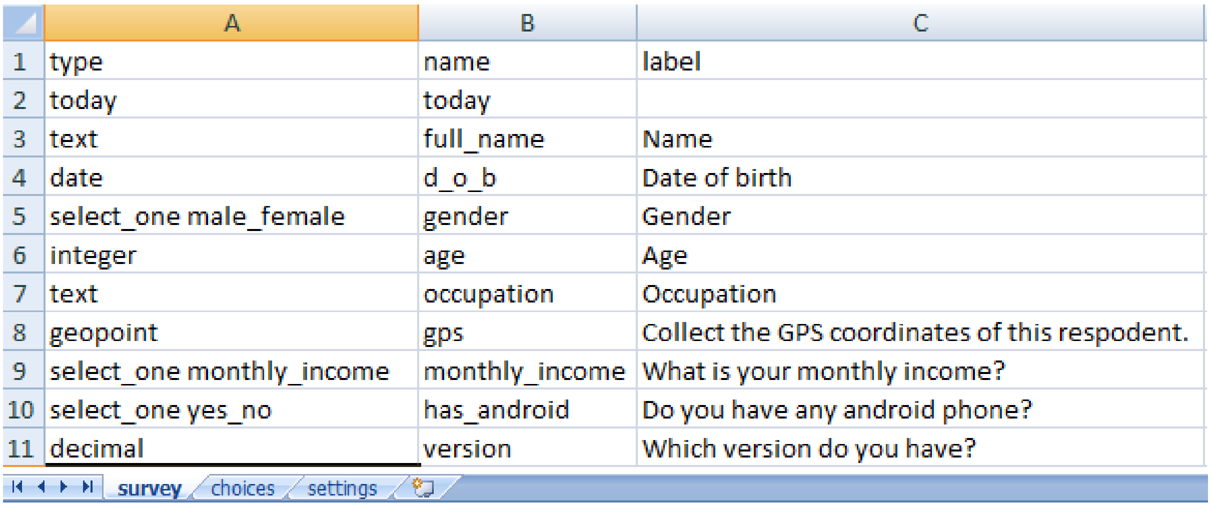
Each Excel workbook usually has three worksheets: **survey**, **choices** and **settings**.



2.1.1 The survey sheet

This worksheet gives the form its overall structure and contains most of the content of the form. It contains the full list of questions and information about how they should appear in the form. Each row usually represents one question; however, there are certain other features described below that one can add to the form which enables the user to incorporate complex type of questions (tables, ranking questions).

The **survey** worksheet has 3 mandatory columns: **type**, **name**, and **label**. The **type** column specifies the type of entry we are adding. Different types of questions are discussed under section Question types (Section 3). The **name** column specifies the unique variable name for that entry. No two entries can have the same name. The **label** column contains the actual text one see in the XLSForm.

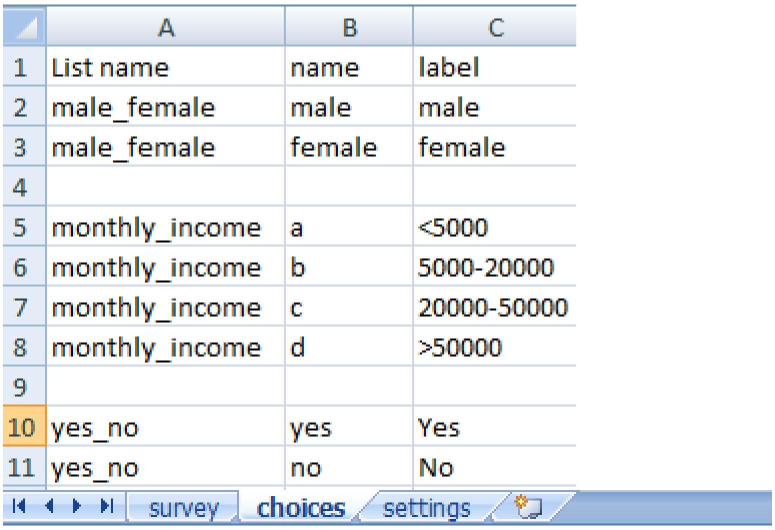


	A	B	C
1	type	name	label
2	today	today	
3	text	full_name	Name
4	date	d_o_b	Date of birth
5	select_one male_female	gender	Gender
6	integer	age	Age
7	text	occupation	Occupation
8	geopoint	gps	Collect the GPS coordinates of this respodent.
9	select_one monthly_income	monthly_income	What is your monthly income?
10	select_one yes_no	has_android	Do you have any android phone?
11	decimal	version	Which version do you have?

2.1.2 The choices sheet

This worksheet is used to specify the choices for multiple choice questions. Each row represents an answer choice. Answer choices with the same **list name** are considered part of a related set of choices and will appear together for a question. This also allows a set of choices to be reused for multiple questions (for example, yes/no questions).

The **choices** worksheet has 3 mandatory columns as well: **list name**, **name**, and **label**. The **list name** column lets one group together a set of related answer choices, i.e., answer choices that should appear together under a question. The **name** column specifies the unique variable name for that answer choice. The label column shows the answer choice exactly as one want it to appear on the form. This **choices** sheet corresponds to the **survey** sheet mentioned above.



	A	B	C
1	List name	name	label
2	male_female	male	male
3	male_female	female	female
4			
5	monthly_income	a	<5000
6	monthly_income	b	5000-20000
7	monthly_income	c	20000-50000
8	monthly_income	d	>50000
9			
10	yes_no	yes	Yes
11	yes_no	no	No

One should keep in mind when authoring forms in Excel is that the syntax used must be precise. For example, if one writes **Choices** or **choice** instead of **choices**, the form won't work.

2.1.3 The settings sheet

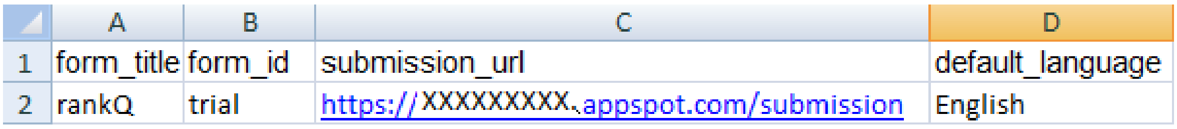
This worksheet allows one to further customize the form. An example of **settings** worksheet is below. The column headings in this example **settings** worksheet do the following:

form_title: The title of the form that is shown to users. The form title is pulled from **form_id** if **form_title** is blank or missing.

form_id: The name used to identify the form submission. The form id is pulled from the XLS file name if **form_id** is blank or missing.

submission_url: For encrypted forms, this url specifies the server where finalized forms are submitted to.

default_language: If the survey is to be undertaken in any local language then one needs to keep the "default_language" blank. If it is kept "English" then the question will always appear in English on the smart phone.



	A	B	C	D
1	form_title	form_id	submission_url	default_language
2	rankQ	trial	https://XXXXXXXXX.appspot.com/submission	English

2.2 ODK COLLECT

ODK Collect enables us to display the ODK Form on a smart phone in a predetermined logical system which can be used to carry out the survey. The program also supports multiple languages. Then the surveys taken on the phone can be sent wirelessly to the server ODK Aggregate via either wifi or a mobile internet connection.

Now we describe steps that we follow.

First we install ODK Collect application from Google Play Store on the smart phone. This creates a folder named “**odk”** in “File Manager” under the “SD Card”. There are four sub-folders under **odk.** We save XForm in the folder named “forms”.

Once the app is installed there will be an icon on the smart phone named “ODK Collect”. To begin the survey, click on “ODK Collect”. A window will open with multiple options. One can start the survey by touching **Fill Blank Form**. The questions will appear in sequence. To move on to the next question, simply swipe the screen. At the end one should be careful about **saving form** and then exit.

Multiple facilities are available in ODK Collect. We can edit saved forms, delete saved and blank forms.

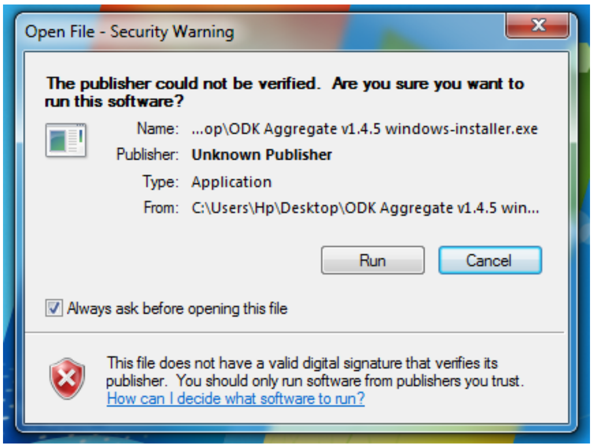
Finally we can send final forms to our server using internet connectivity.

2.3 ODK AGGREGATE

This is our server where we upload or store our data. To have a server we need to follow following steps.

First download ODK Aggregate file of latest version in laptop or PC from https://opendatakit.org/downloads/ for windows i.e. windows-installer.exe.

Next after double click upon above file; a box will appear that is given below.



Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name: ...op\ODK Aggregate v1.4.5 windows-installer.exe

Publisher: **Unknown Publisher**

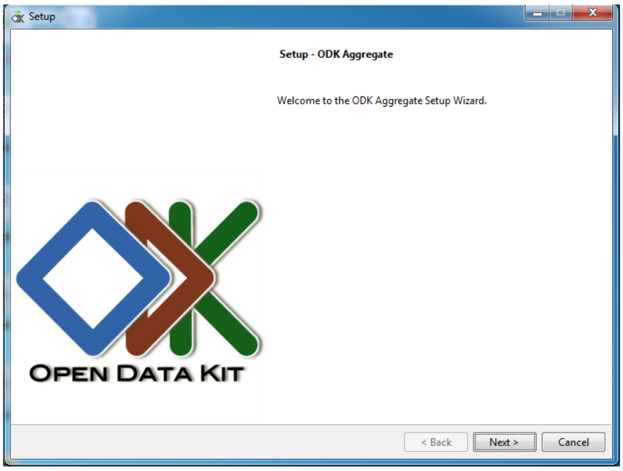
Type: Application

From: C:\Users\Hp\Desktop\ODK Aggregate v1.4.5 win...

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?

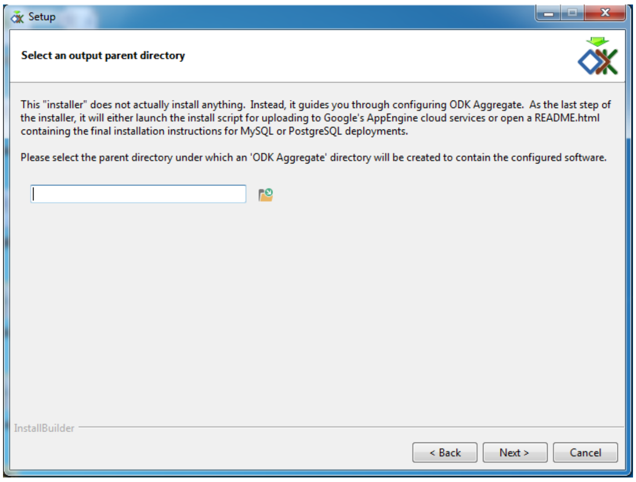
Now we run this software. After selecting Next button we accept license agreement.



Setup - ODK Aggregate

Welcome to the ODK Aggregate Setup Wizard.

Next a box will appear which ask for a parent directory where usually ODK Aggregate folder will be saved. Generally we select Desktop for our convenience but one can select a location of her own choice.

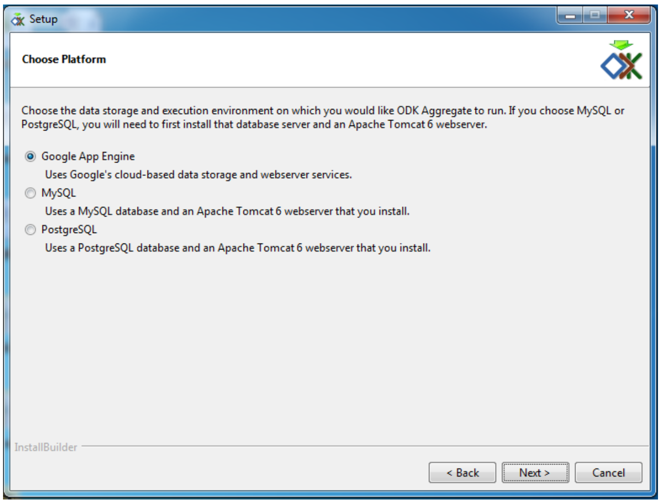


Select an output parent directory

This "installer" does not actually install anything. Instead, it guides you through configuring ODK Aggregate. As the last step of the installer, it will either launch the install script for uploading to Google's AppEngine cloud services or open a README.html containing the final installation instructions for MySQL or PostgreSQL deployments.

Please select the parent directory under which an 'ODK Aggregate' directory will be created to contain the configured software.

After that it’s required to choose a platform for our webserver. We have three options below among which we select Google App Engine.



Choose Platform

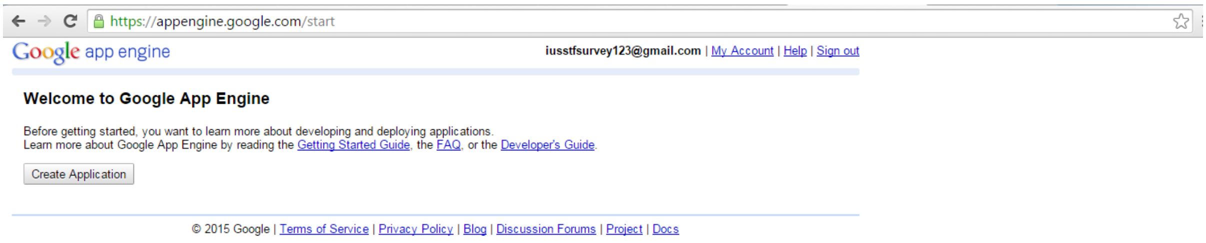
Choose the data storage and execution environment on which you would like ODK Aggregate to run. If you choose MySQL or PostgreSQL, you will need to first install that database server and an Apache Tomcat 6 webserver.

● Google App Engine
Uses Google's cloud-based data storage and webserver services.

○ MySQL
Uses a MySQL database and an Apache Tomcat 6 webserver that you install.

○ PostgreSQL
Uses a PostgreSQL database and an Apache Tomcat 6 webserver that you install.

But before that there is very important step. We need to create an application in Google. First go to https://appengine.google.com/ site and we need a gmail account to create an application. Due to upgradation of technology, appearance of the process of creating application may change from time to time. In our case, previously we just went to the above page and put an Application ID and select create application button. Now what we need to do is that we have to start with same https://appengine.google.com/ and through a gmail account we start creating an application.



https://appengine.google.com/start

Google app engine

iusstfsurvey123@gmail.com | My Account | Help | Sign out

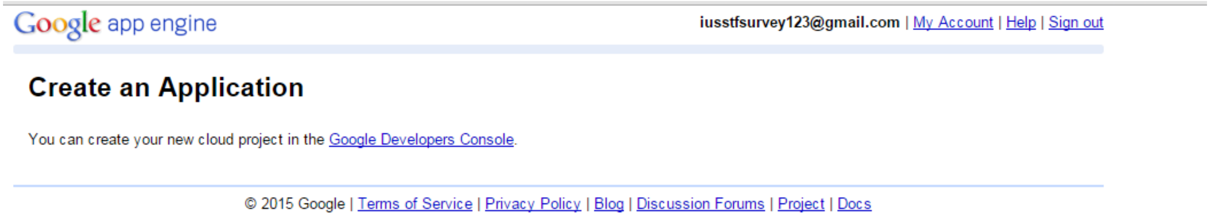
Welcome to Google App Engine

Before getting started, you want to learn more about developing and deploying applications.
Learn more about Google App Engine by reading the Getting Started Guide, the FAQ, or the Developer's Guide.

Create Application

© 2015 Google | Terms of Service | Privacy Policy | Blog | Discussion Forums | Project | Docs

When we select Create Application a page will appear like this



Google app engine

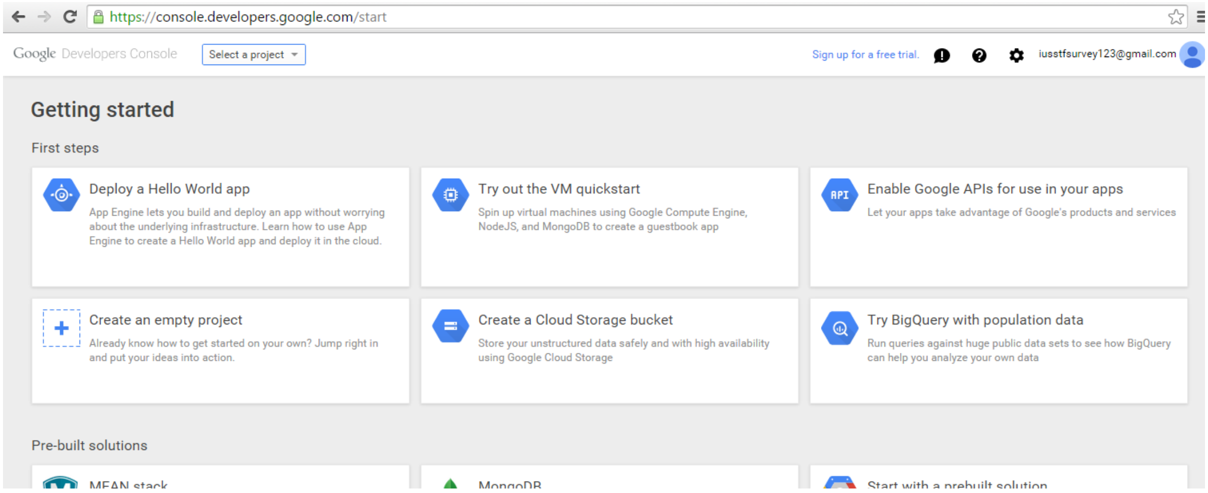
iusstfsurvey123@gmail.com | My Account | Help | Sign out

Create an Application

You can create your new cloud project in the Google Developers Console.

© 2015 Google | Terms of Service | Privacy Policy | Blog | Discussion Forums | Project | Docs

Here we select Google Developers Console and arrive at next page with several option will come out like this below



https://console.developers.google.com/start

Google Developers Console

Select a project

Sign up for a free trial.

iusstfsurvey123@gmail.com

Getting started

First steps

Deploy a Hello World app
App Engine lets you build and deploy an app without worrying about the underlying infrastructure. Learn how to use App Engine to create a Hello World app and deploy it in the cloud.

Try out the VM quickstart
Spin up virtual machines using Google Compute Engine, NodeJS, and MongoDB to create a guestbook app

Enable Google APIs for use in your apps
Let your apps take advantage of Google's products and services

Create an empty project
Already know how to get started on your own? Jump right in and put your ideas into action.

Create a Cloud Storage bucket
Store your unstructured data safely and with high availability using Google Cloud Storage

Try BigQuery with population data
Run queries against huge public data sets to see how BigQuery can help you analyze your own data

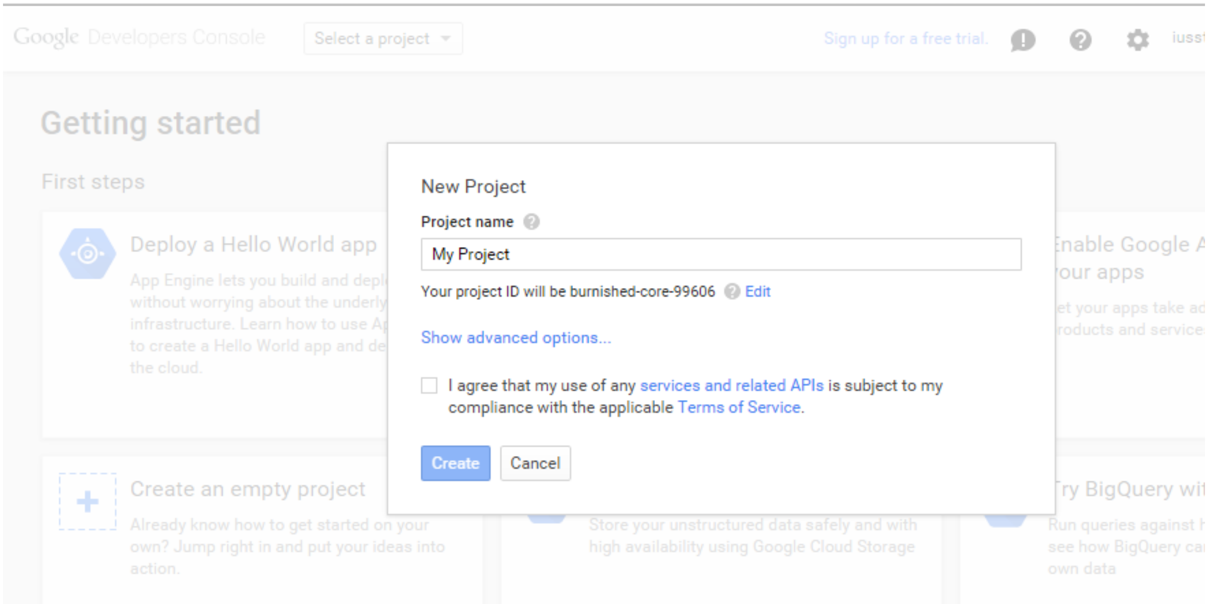
Pre-built solutions

MEAN stack

MongoDB

Start with a prebuilt solution

Next we get started with **Deploy a Hello world app**. A box pops-up like this below


Google Developers Console
Select a project
Sign up for a free trial.
iuss
Getting started
First steps
Deploy a Hello World app
App Engine lets you build and depl
without worrying about the underly
infrastructure. Learn how to use A
to create a Hello World app and de
the cloud.
Create an empty project
Already know how to get started on your
own? Jump right in and put your ideas into
action.
Store your unstructured data safely and with
high availability using Google Cloud Storage
Enable Google A
our apps
et your apps take ad
roducts and service
ry BigQuery wi
Run queries against
see how BigQuery ca
own data

New Project

Project name

My Project

Your project ID will be burnished-core-99606 Edit

Show advanced options...

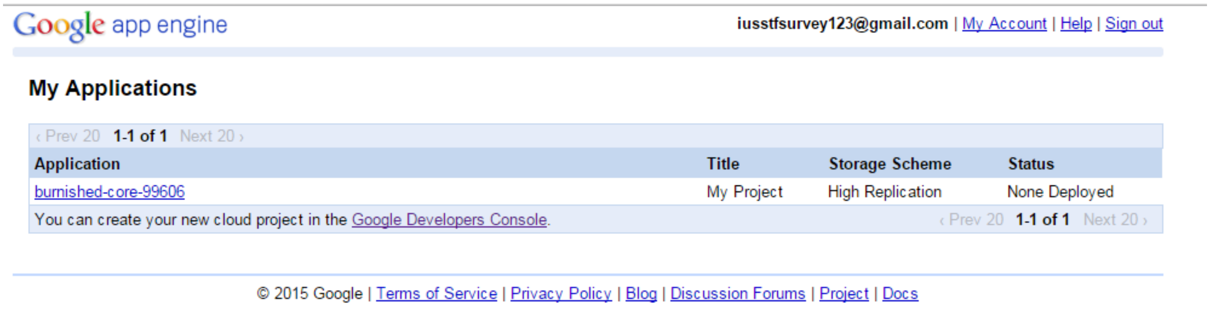
□ I agree that my use of any services and related APIs is subject to my compliance with the applicable Terms of Service.

Create Cancel

Next we write a project name of our choice. We agree with terms of services and select **Create** button.

Close the current tab and open https://appengine.google.com/ again.

You will see a page like this



iusstfsurvey123@gmail.com | My Account | Help | Sign out

My Applications

‹ Prev 20 **1-1 of 1** Next 20 ›

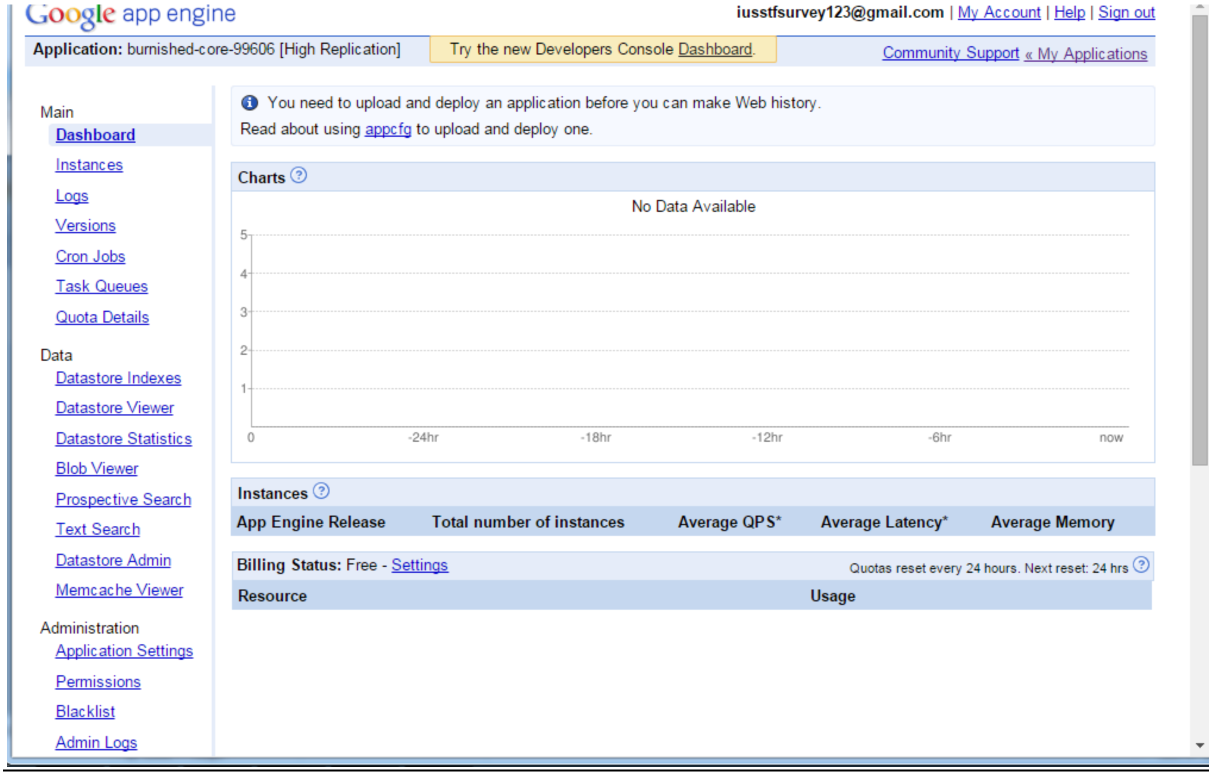
Application	Title	Storage Scheme	Status
burnished-core-99606	My Project	High Replication	None Deployed

You can create your new cloud project in the Google Developers Console.

‹ Prev 20 **1-1 of 1** Next 20 ›

© 2015 Google | Terms of Service | Privacy Policy | Blog | Discussion Forums | Project | Docs

If you click on burnished-core-99606, we arrive at the page below.



Application: burnished-core-99606 [High Replication]

You need to upload and deploy an application before you can make Web history.
Read about using appcfg to upload and deploy one.

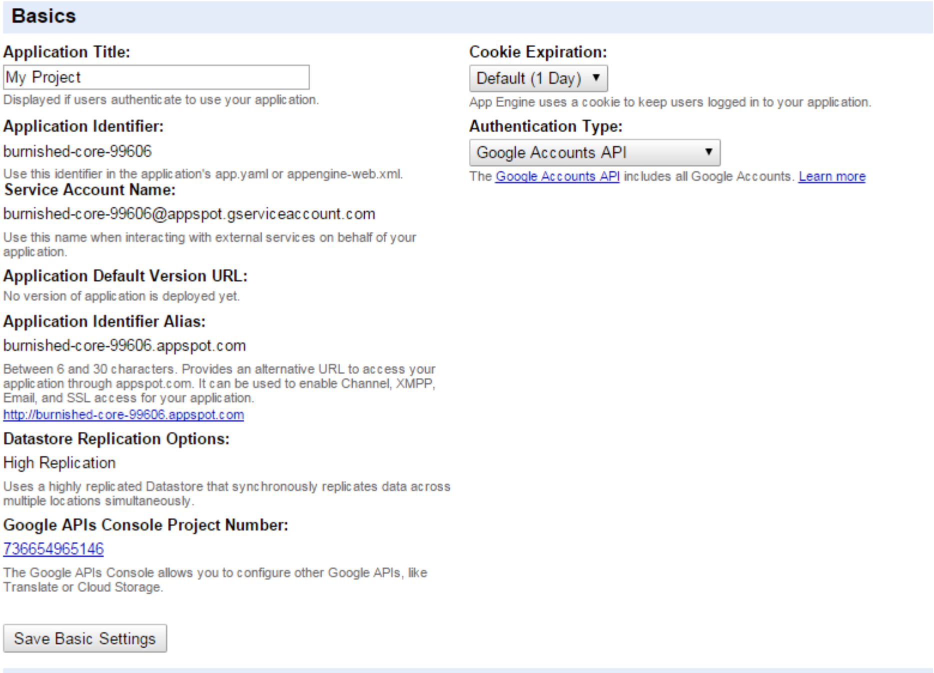
App Engine Release	Total number of instances	Average QPS*	Average Latency*	Average Memory

Billing Status: Free - Settings

Quotas reset every 24 hours. Next reset: 24 hrs

Resource	Usage

Here click on **Application Settings** under **Administration** and finally arrive at a page where at the end there is a **Create** button.



Application Title:
My Project
Displayed if users authenticate to use your application.

Application Identifier:
burnished-core-99606
Use this identifier in the application's app.yaml or appengine-web.xml.

Service Account Name:
burnished-core-99606@appspot.gserviceaccount.com
Use this name when interacting with external services on behalf of your application.

Application Default Version URL:
No version of application is deployed yet.

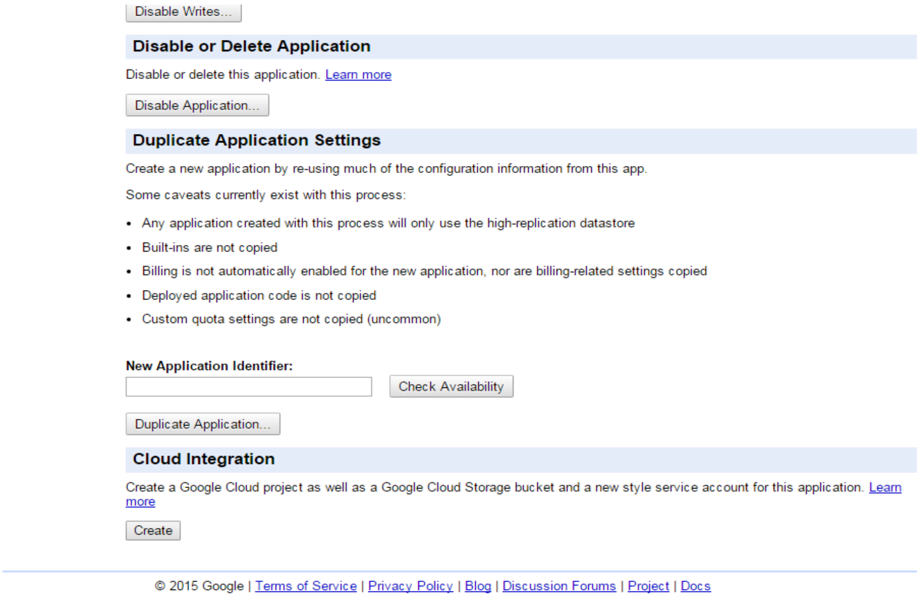
Application Identifier Alias:
burnished-core-99606.appspot.com
Between 6 and 30 characters. Provides an alternative URL to access your application through appspot.com. It can be used to enable Channel, XMPP, Email, and SSL access for your application.
http://burnished-core-99606.appspot.com

Datastore Replication Options:
High Replication
Uses a highly replicated Datastore that synchronously replicates data across multiple locations simultaneously.

Google APIs Console Project Number:
736654965146
The Google APIs Console allows you to configure other Google APIs, like Translate or Cloud Storage.

Cookie Expiration:
Default (1 Day) ▼
App Engine uses a cookie to keep users logged in to your application.

Authentication Type:
Google Accounts API ▼
The Google Accounts API includes all Google Accounts. Learn more



Disable or Delete Application

Disable or delete this application. Learn more

Duplicate Application Settings

Create a new application by re-using much of the configuration information from this app.

Some caveats currently exist with this process:

- Any application created with this process will only use the high-replication datastore
- Built-ins are not copied
- Billing is not automatically enabled for the new application, nor are billing-related settings copied
- Deployed application code is not copied
- Custom quota settings are not copied (uncommon)

New Application Identifier:

Cloud Integration

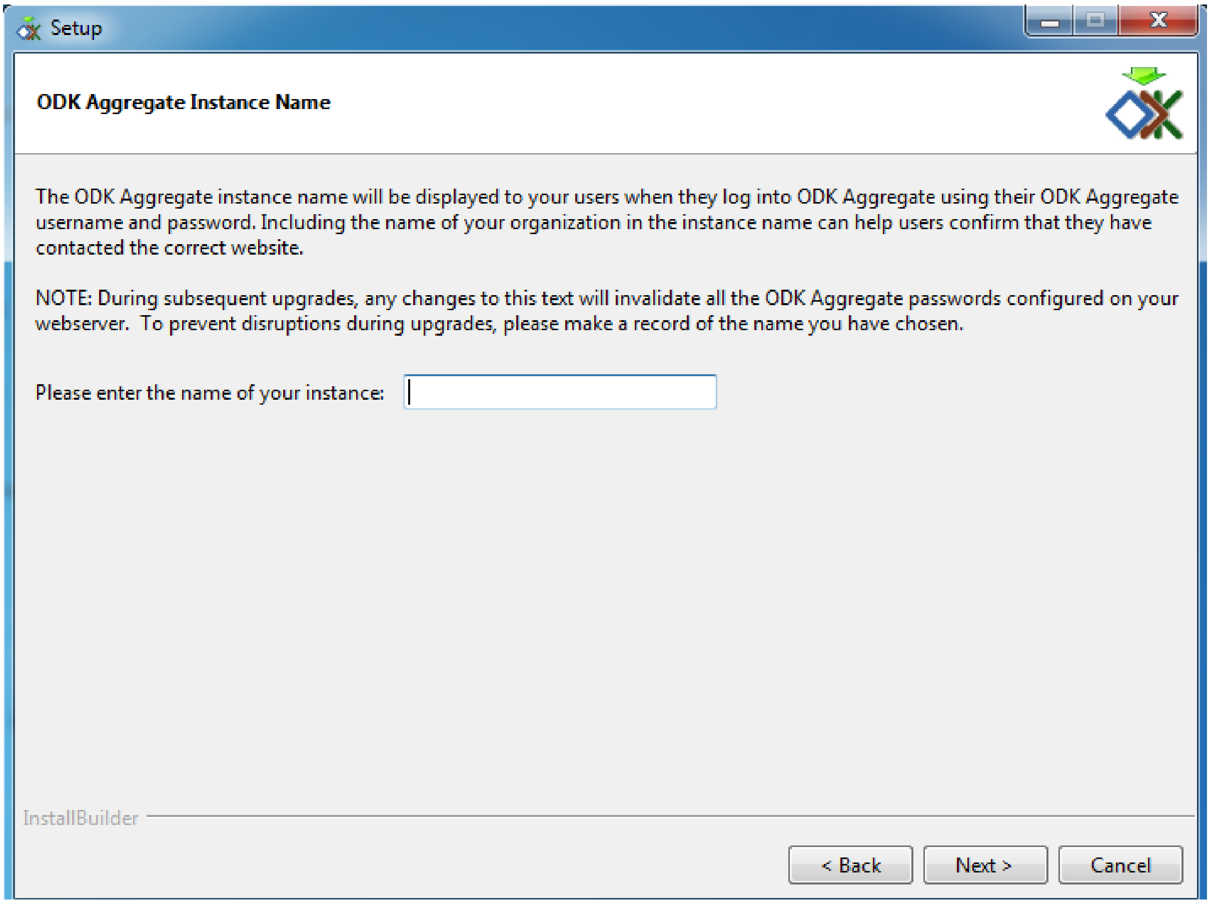
Create a Google Cloud project as well as a Google Cloud Storage bucket and a new style service account for this application. Learn more

© 2015 Google | Terms of Service | Privacy Policy | Blog | Discussion Forums | Project | Docs

Within few seconds application will be created after click an on **Create** button with a success message.

So, this is the process how we create an application through Google as we need this for our web server.

In ODK aggregate we are at a stage where we select **Google App Engine** as our platform. Next we enter a name of **instance**.



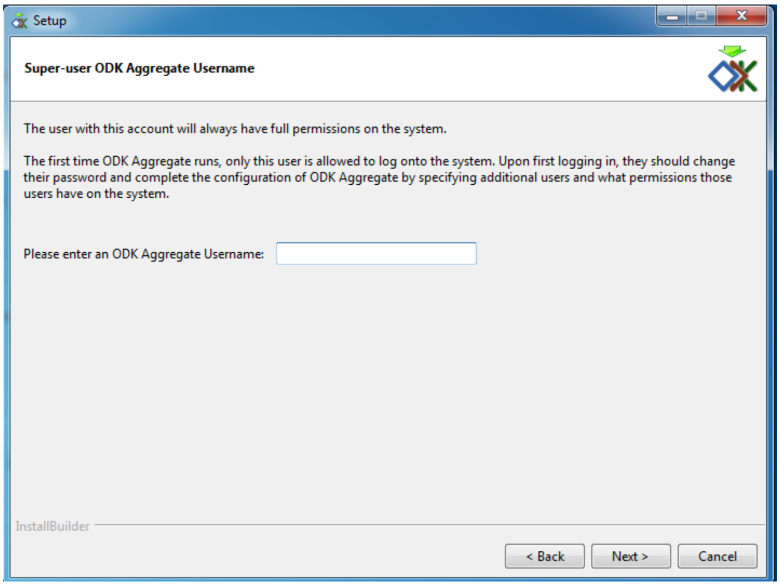
ODK Aggregate Instance Name

The ODK Aggregate instance name will be displayed to your users when they log into ODK Aggregate using their ODK Aggregate username and password. Including the name of your organization in the instance name can help users confirm that they have contacted the correct website.

NOTE: During subsequent upgrades, any changes to this text will invalidate all the ODK Aggregate passwords configured on your webserver. To prevent disruptions during upgrades, please make a record of the name you have chosen.

Please enter the name of your instance:

In the next page we have to put **ODK Aggregate Username** of our choice.



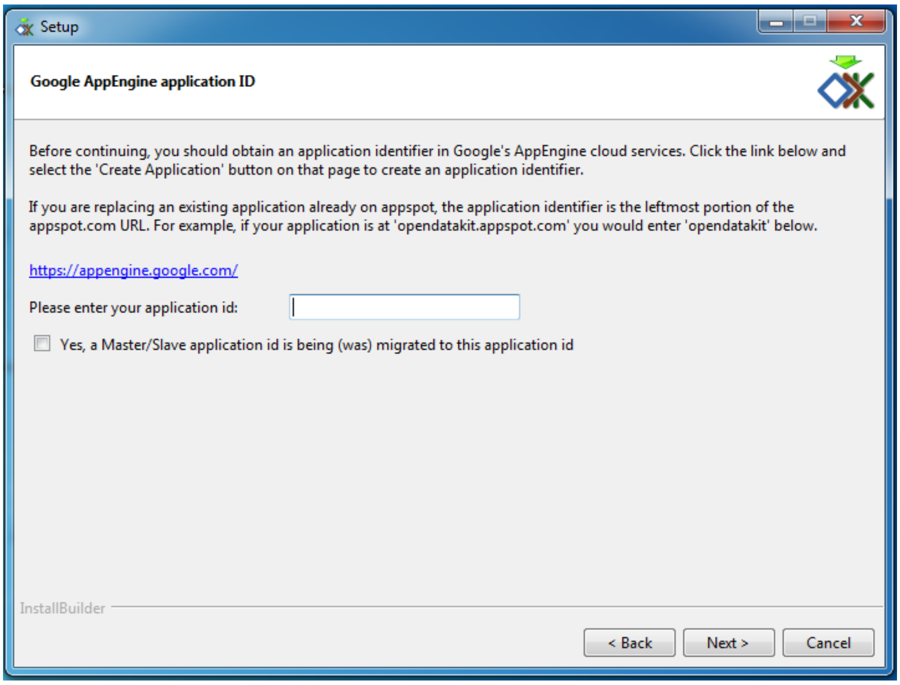
Super-user ODK Aggregate Username

The user with this account will always have full permissions on the system.

The first time ODK Aggregate runs, only this user is allowed to log onto the system. Upon first logging in, they should change their password and complete the configuration of ODK Aggregate by specifying additional users and what permissions those users have on the system.

Please enter an ODK Aggregate Username:

Next we required to enter the **application id** which we created before



Google AppEngine application ID

Before continuing, you should obtain an application identifier in Google's AppEngine cloud services. Click the link below and select the 'Create Application' button on that page to create an application identifier.

If you are replacing an existing application already on appspot, the application identifier is the leftmost portion of the appspot.com URL. For example, if your application is at 'opendatakit.appspot.com' you would enter 'opendatakit' below.

https://appengine.google.com/

Please enter your application id:

□ Yes, a Master/Slave application id is being (was) migrated to this application id

We already have this **application id** when we created application in appengine.google.com. In our above example it's **burnished-core-99606**. This is very crucial as it is going to be our server's url i.e. http://burnished-core-99606.appspot.com. We don't required to select the box above against which it is written, "Yes, a Master/slave..........application id".

Click on **Next** ODK Aggregate will start configuring. Then a page will ask for Email and Password where we insert our Gmail id and password which we created for Google app-engine and select **upload**.

Wait 3-10 minutes for the script to run. Once it's finished, we will get a "Success" message: *END-SCRIPT SUCCESS.*

If there is any mistake in any step, at the end it will show *END-SCRIPT-FAILURE*. That means process is hindered due to some wrong information. But the good thing is that the problem occurred is written in output message in this stage. After identifying the problem we have to re-run the Setup Wizard by going back to installed ODK Aggregate and double-click the installer to run it again.

Before re-running the wizard again we should delete the folder that has been created on our desktop as ODK Aggregate. Otherwise, we will be restricted when we insert destination of ODK Aggregate folder which we are supposed to do in our second step.

This is the theory through which we have tried to depict entire process of using ODK in details and hope, now you can apply ODK by yourself. But it may happen that you may found some difficulties when we practically try to do this.

3. Question types

This is the most important part in forming a questionnaire. XLSForm supports a number of simple question types. These are just some of the simple question type we use under **type** column in the **survey** worksheet:

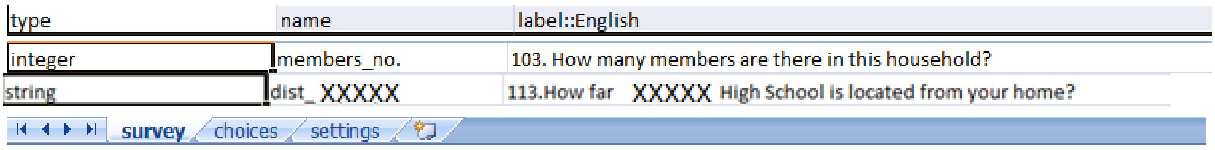
Question type	Answer input
integer	Integer (i.e., whole number) input.
decimal	Decimal input.
text	Free text response.
string	Under this input words, numbers, decimals all are allowed.
select_one [options]	Multiple choice question; only one answer can be selected.
select_multiple	Multiple choice question; multiple answers can be selected.
note	Display a note on the screen, takes no input.
date	Date input.
time	Time input.
today	Automatically captures date and time of the day of survey from android

For further types of question, visit http://xlsform.org/ .

3.1 Numbering (integer, decimal, string)

If the response is in number like 1, 2, 3, 4 etc, ‘integer’ question type is normally used. Remember this type won’t function in ODK Collect if one tries to give an input of any decimal like 1.3 etc. In that case

decimal type is used. So be careful, when inserting question type. If both integer and decimal appear in response, in this case one should use **'string'** question type. Under **string** type one can write in both words and numbers.

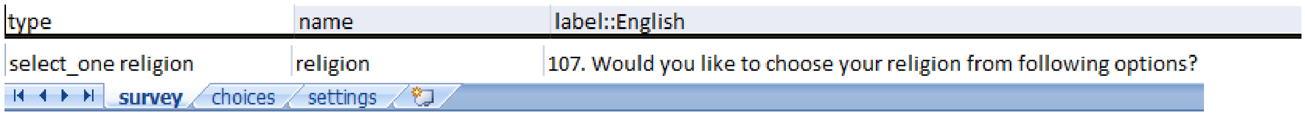


type	name	label::English
integer	members_no.	103. How many members are there in this household?
string	dist_ XXXXX	113.How far XXXXX High School is located from your home?

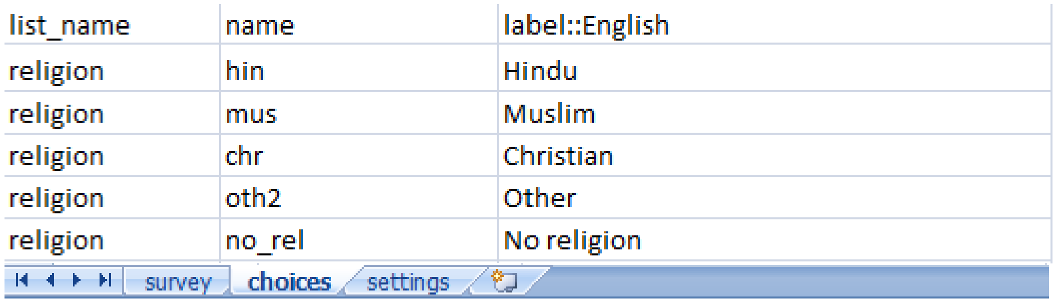
3.2 Multiple choice questions

select_one (select only one answer) and **select_multiple** (select multiple answers) questions both can be used in XLSForm. With this type of question we need to incorporate available options for that particular multiple choice question in our **choices** worksheet.

Here is an example of a **select_one** question:



type	name	label::English
select_one religion	religion	107. Would you like to choose your religion from following options?

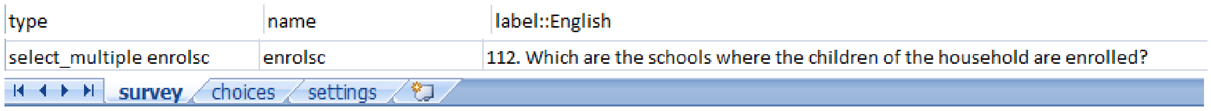


list_name	name	label::English
religion	hin	Hindu
religion	mus	Muslim
religion	chr	Christian
religion	oth2	Other
religion	no_rel	No religion

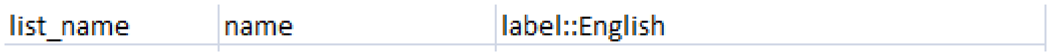
Note that **religion** in the **survey** worksheet must match the **religion** in the **list name** column in the **choices** worksheet i.e. options written after **select_one** under survey sheet and list name under **choices** sheet both must be same. This ensures that the form displays the correct list of answer choices for a particular question.

Example of **select_multiple** question:

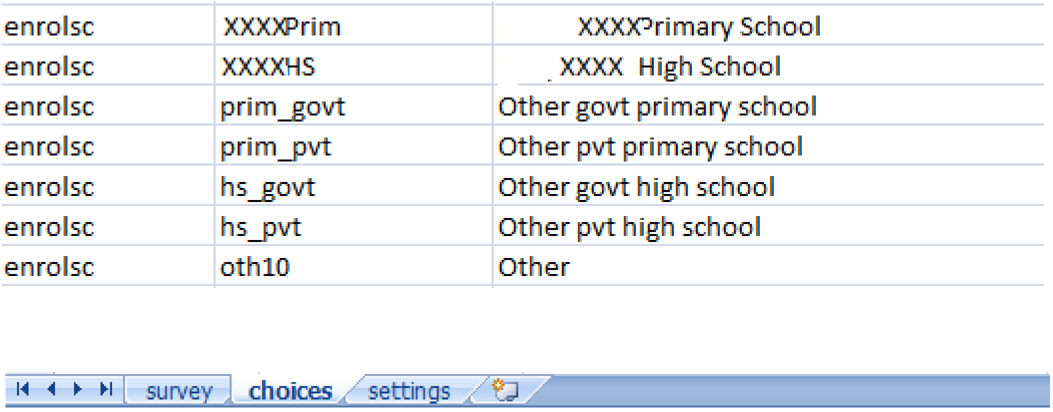
We can also add multiple choice questions that allow multiple answers to be selected, like so:



type	name	label::English
select_multiple enrolsc	enrolsc	112. Which are the schools where the children of the household are enrolled?



list_name	name	label::English

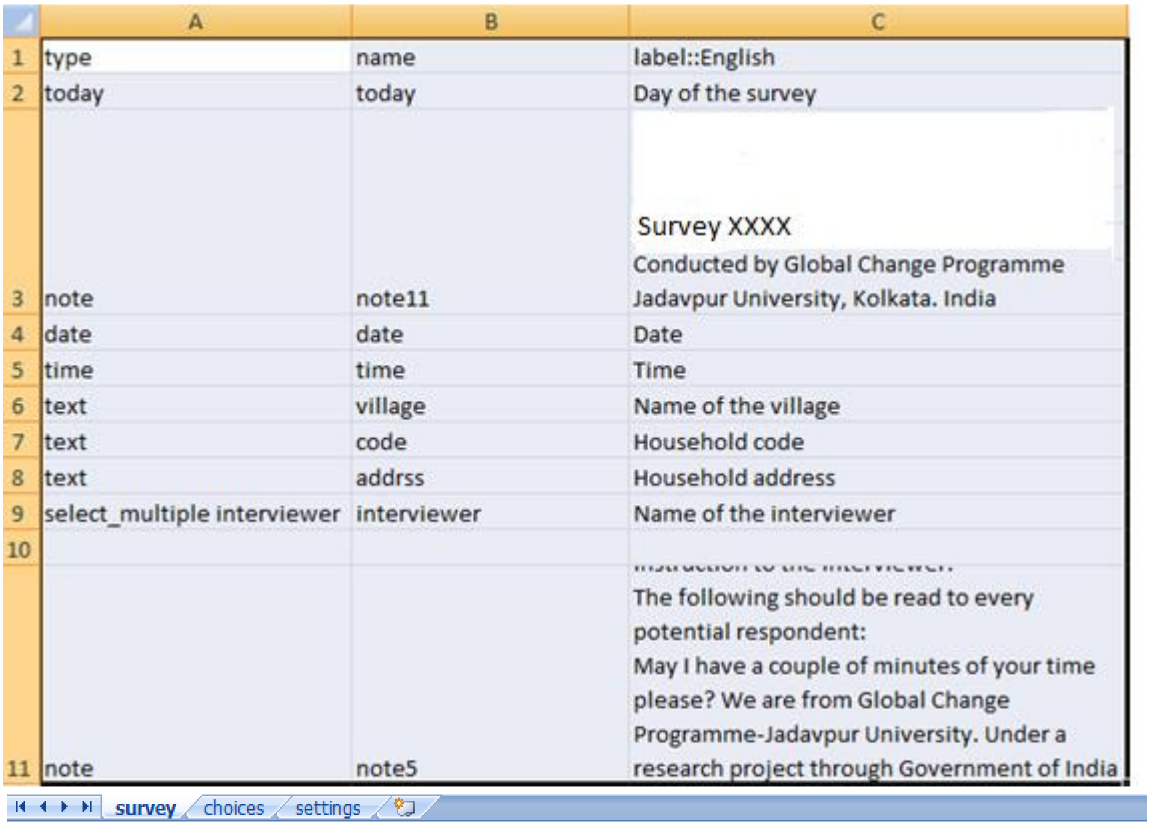


enrolsc	XXXXPrim	XXXXPrimary School
enrolsc	XXXXHS	XXXX High School
enrolsc	prim_govt	Other govt primary school
enrolsc	prim_pvt	Other pvt primary school
enrolsc	hs_govt	Other govt high school
enrolsc	hs_pvt	Other pvt high school
enrolsc	oth10	Other

3.3 Display of a text (note)

Sometime it is required to display note and we don't need any response against it in this case this question type is used which only display a note.

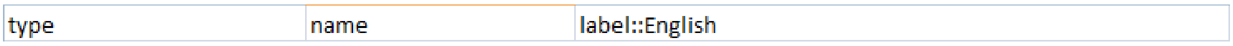
Example is given below.



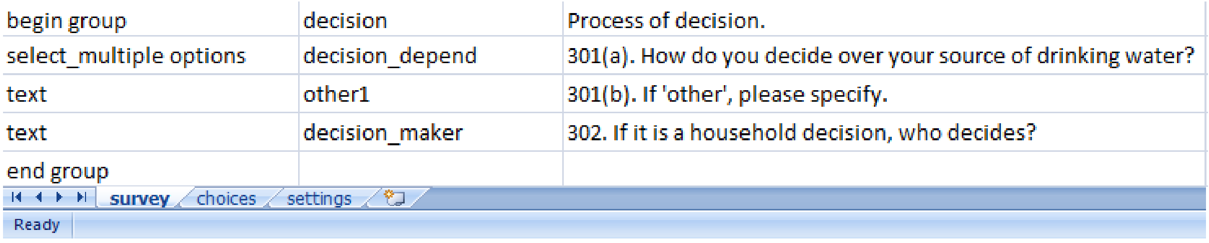
	A	B	C
1	type	name	label::English
2	today	today	Day of the survey
3	note	note11	Survey XXXX Conducted by Global Change Programme Jadavpur University, Kolkata. India
4	date	date	Date
5	time	time	Time
6	text	village	Name of the village
7	text	code	Household code
8	text	addrss	Household address
9	select_multiple interviewer	interviewer	Name of the interviewer
10			
11	note	note5	Instruction to the interviewer: The following should be read to every potential respondent: May I have a couple of minutes of your time please? We are from Global Change Programme-Jadavpur University. Under a research project through Government of India

3.4 Grouping questions

If multiple questions are related to one particular issue, then often they are put in a group. Example is given below



type	name	label::English

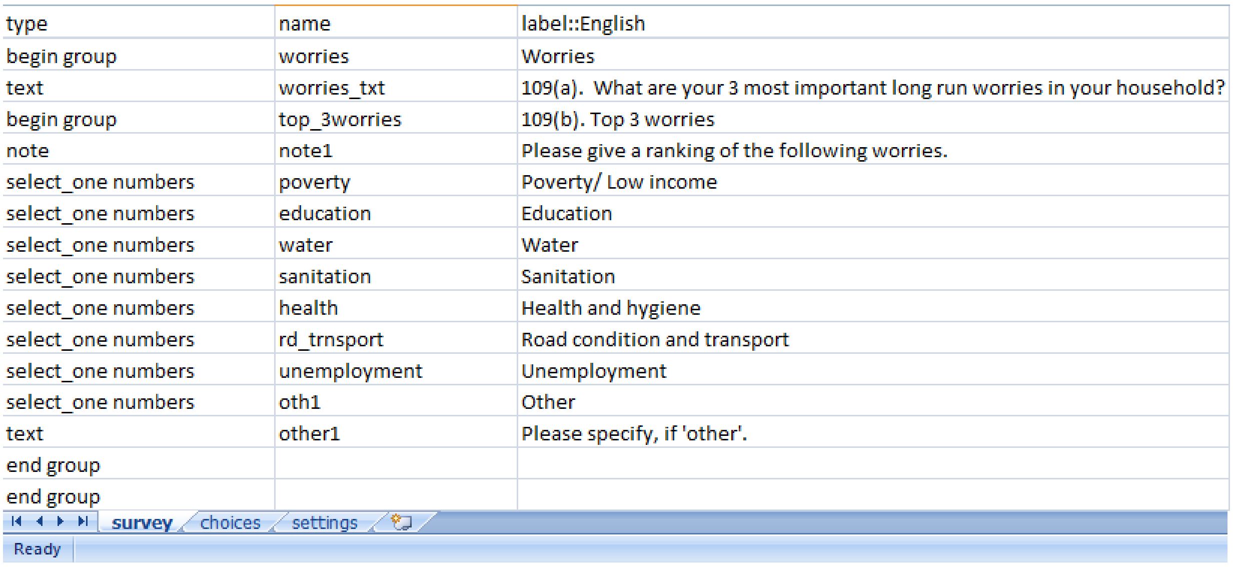


begin group	decision	Process of decision.
select_multiple options	decision_depend	301(a). How do you decide over your source of drinking water?
text	other1	301(b). If 'other', please specify.
text	decision_maker	302. If it is a household decision, who decides?
end group		

Notice how **end group** doesn’t require a name or label, because it is hidden in the form.

3.4.1 Nesting groups within groups

Groups of questions can be nested within one another:



type	name	label::English
begin group	worries	Worries
text	worries_txt	109(a). What are your 3 most important long run worries in your household?
begin group	top_3worries	109(b). Top 3 worries
note	note1	Please give a ranking of the following worries.
select_one numbers	poverty	Poverty/ Low income
select_one numbers	education	Education
select_one numbers	water	Water
select_one numbers	sanitation	Sanitation
select_one numbers	health	Health and hygiene
select_one numbers	rd_trnsport	Road condition and transport
select_one numbers	unemployment	Unemployment
select_one numbers	oth1	Other
text	other1	Please specify, if 'other'.
end group		
end group		

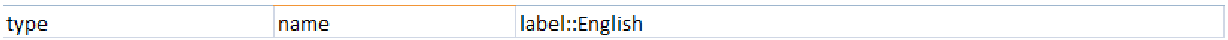
The first **end group** one see closes the **Top 3 worries** group, and the second one closes the beginning **Worries** group. When working with groups and one keep getting error messages when trying to converting the form in xml file, double-check that for each **begin group** one have one **end group**.

3.4.2 Skipping

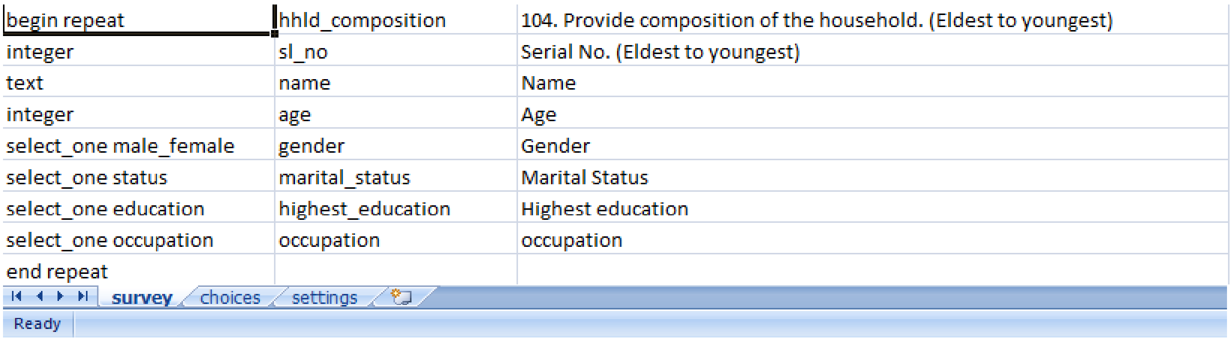
One feature of XLSForm is the ability to skip a group of questions by combining the group feature with relevant syntax. If one wants to skip a group of questions all at once, put the relevant attribute at the beginning of a group. We didn’t use this feature in our present survey even though we mentioned it as it comes under usages of **relevant function** (discussed under Functions)**.**

3.4.3 Repeats

A user can repeat a group of questions by using the **begin repeat** and **end repeat** construct:



type	name	label::English



begin repeat	hhld_composition	104. Provide composition of the household. (Eldest to youngest)
integer	sl_no	Serial No. (Eldest to youngest)
text	name	Name
integer	age	Age
select_one male_female	gender	Gender
select_one status	marital_status	Marital Status
select_one education	highest_education	Highest education
select_one occupation	occupation	occupation
end repeat		

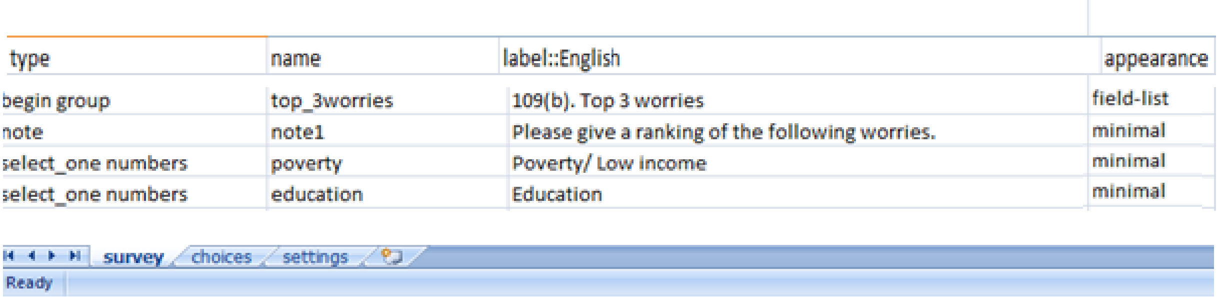
In this example, the Name, Age, Gender, Marital Status, Highest Education and Occupation are grouped together in a repeat group, and the user can repeat this group as many times as required by selecting the option in the form to start another repeat.

The **label** column is optional for **begin repeat**. Assigning a label to a repeat group will add the label as a title to the block of repeat questions in the form.

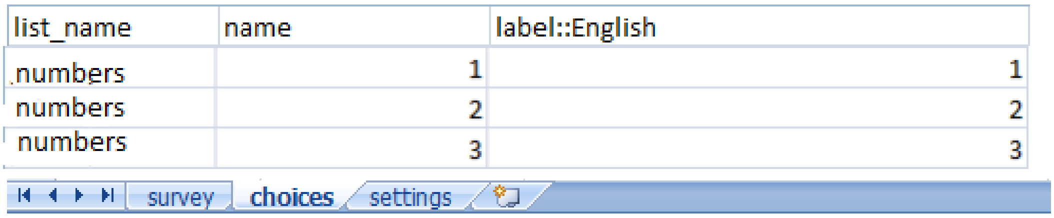
Instead of allowing an infinite number of repeats, one can specify an exact number of repeats. Details can be found in http://xlsform.org/.

3.5 Ranking questions

Sometimes we need to record respondent's preference ordering over few choices. This type of example is given below.



type	name	label::English	appearance
begin group	top_3worries	109(b). Top 3 worries	field-list
note	note1	Please give a ranking of the following worries.	minimal
select_one numbers	poverty	Poverty/ Low income	minimal
select_one numbers	education	Education	minimal



list_name	name	label::English
numbers	1	1
numbers	2	2
numbers	3	3

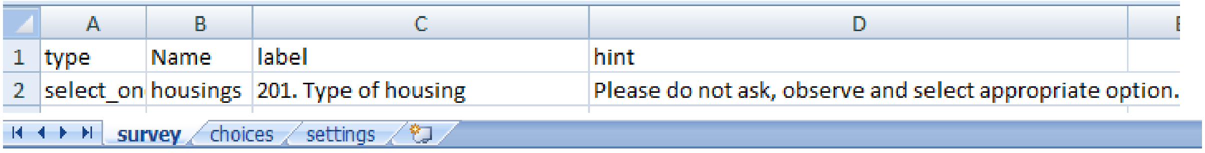
In above example respondent has to rank the 3 options according to her preference. We form it as a group question with appearance of list of numbers: **1, 2, 3** which are included in choices sheet against list_name **numbers**.

4. Functions

It has been written before that type, name and label, these three are mandatory in survey worksheet. Now we consider few more functions which are used to improve pattern of questionnaire such as **hints, constraint, relevant, require** etc.

4.1 Hints

Sometimes we want to add a small **hint** to a question on the form, instructing the user how to answer the question, but one doesn't want the hint to be part of the question itself. It's easy to add hints to questions in XLSForms. Simply add a **hint** column and add the hint message. In appears in italics. See below for an example.

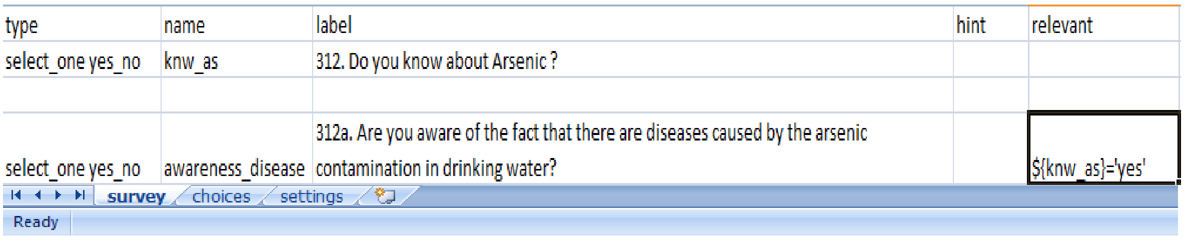


	A	B	C	D	E
1	type	Name	label	hint	
2	select_on	housings	201. Type of housing	Please do not ask, observe and select appropriate option.	

We didn't use any **constraint** in our XLSForm. Useful expressions to use in the **constraint** column can be found http://xlsform.org/ and http://opendatakit.org/help/form-design/binding/. Look under Operators.

4.2 Relevant

One great feature of XLSForm is the ability to skip a question or make an additional question appear based on the response to a previous question. Below is an example of how to do this by adding a **relevant** column for a **select_one** question, using example from our questionnaire before:



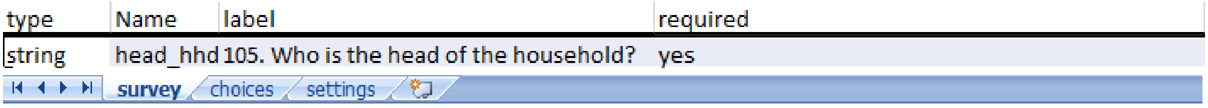
type	name	label	hint	relevant
select_one yes_no	knw_as	312. Do you know about Arsenic ?		
select_one yes_no	awareness_disease	312a. Are you aware of the fact that there are diseases caused by the arsenic contamination in drinking water?		${knw_as}='yes'

In this example respondent is asked, "Do you know about Arsenic?" If the answer is **yes**, then the next question appears whether he is aware of the diseases caused by the arsenic contamination in drinking water. ${ } around the variable name "knw_as" from previous question equals (=) 'yes' i.e the alternative which the variable takes from choice list.

4.3 Required

Another column 'required' is added to restrict the surveyor from moving to next question if she tries to skip any question, in other words marking them as required means the user will unable to move on to the next question or submit the form without entering an answer for that question.

To make questions required, see the example below:



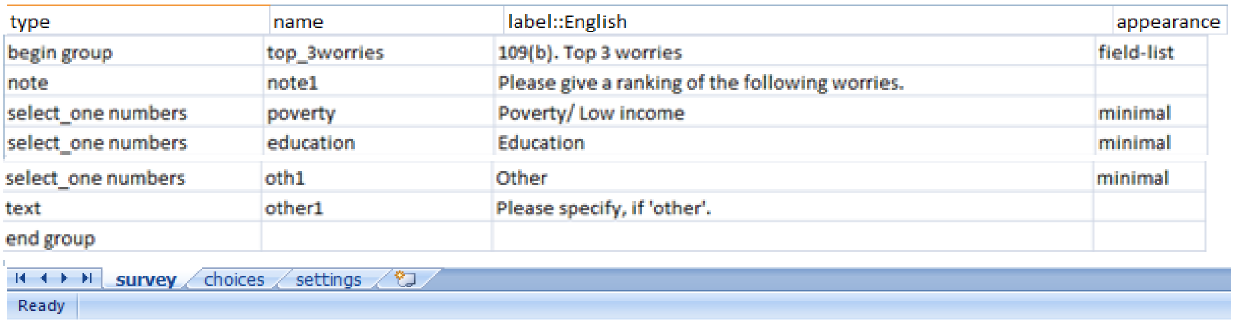
type	Name	label	required
string	head_hhd	105. Who is the head of the household?	yes

4.4 Appearance

The **appearance** column allows one to change the appearance of questions in the form. The following table lists the possible appearance attributes and how the question appears in the form which we have used in our questionnaire.

Appearance attribute	Question type	Description
Minimal	select_one, select_multiple	Answer choices appear in a pull-down menu.
field-list	Groups	Entire group of questions appear on one screen (for mobile clients only).

In our questionnaire we use this function as below:



type	name	label::English	appearance
begin group	top_3worries	109(b). Top 3 worries	field-list
note	note1	Please give a ranking of the following worries.	
select_one numbers	poverty	Poverty/ Low income	minimal
select_one numbers	education	Education	minimal
select_one numbers	oth1	Other	minimal
text	other1	Please specify, if 'other'.	
end group			

One can find many options of appearance attributes which are listed in http://xlsform.org/.

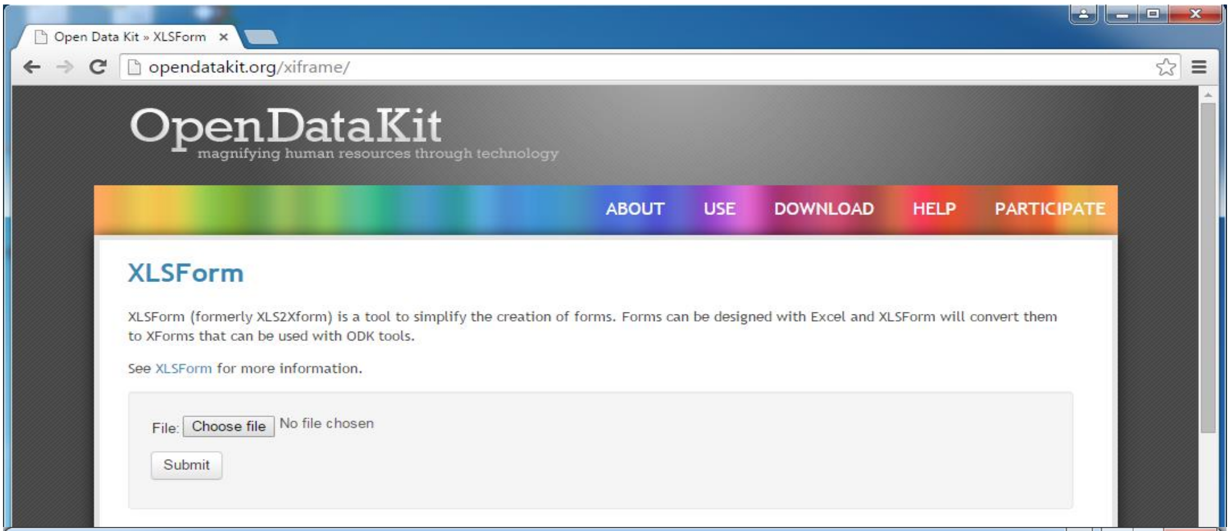
Functions that one can use in XLSForm according to requirements of questionnaire are **Formulas, Calculation**. It was not required for this present context, so we didn’t use any of these two. Details of these two functions can be found in http://xlsform.org/ link.

4.5 Multiple language support:

It is easy to add multiple languages to a form so that one can represent her own questionnaire in a language of her choice. This is one of the salient features of ODK. We have to simply name the **label::language1**, **label::language2**.

5. Conversion of XLSform to XML file

Finally when questionnaire is completed convert this final **XLSForm** into *.xml file which can be done by dint of http://opendatakit.org/xiframe/. Here one can convert by choosing questionnaire in excel and submit it. This website converts that xls file to xml file which is downloadable.

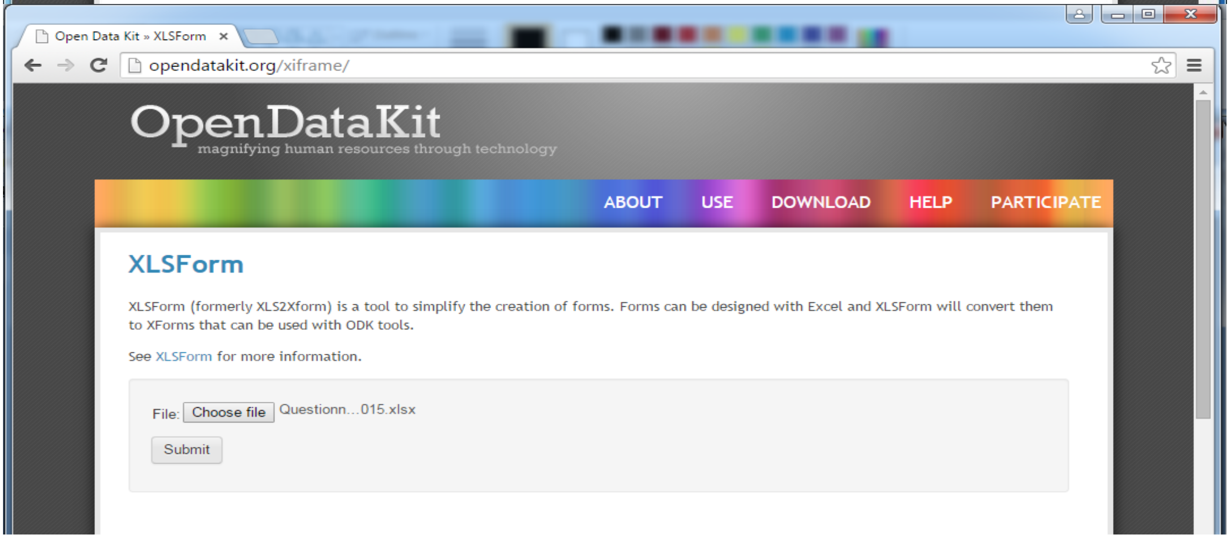


XLSForm

XLSForm (formerly XLS2Xform) is a tool to simplify the creation of forms. Forms can be designed with Excel and XLSForm will convert them to XForms that can be used with ODK tools.

See XLSForm for more information.

File: No file chosen



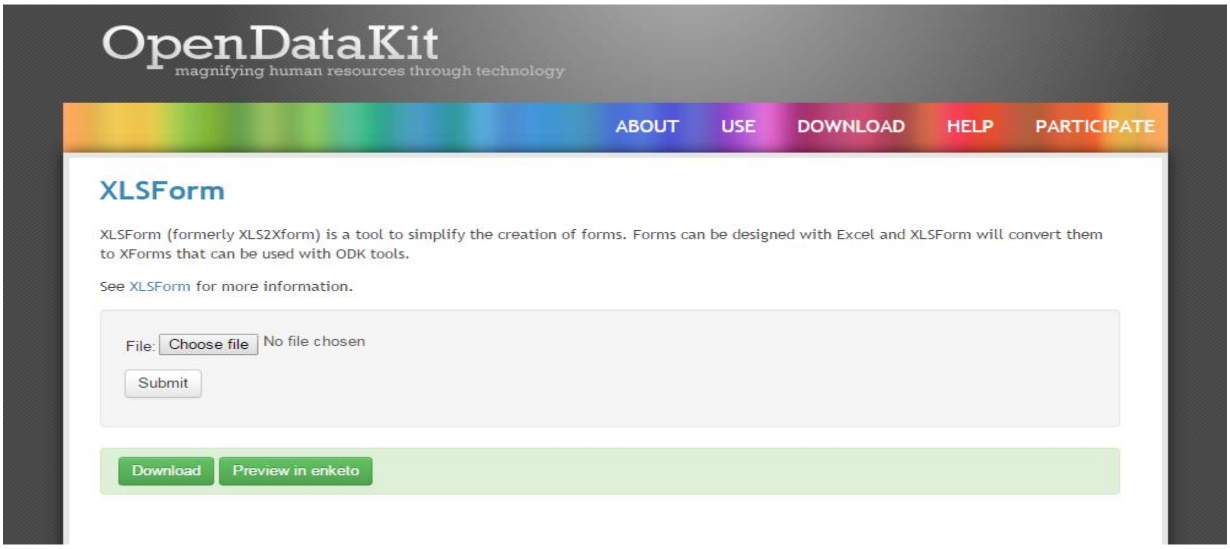
XLSForm

XLSForm (formerly XLS2Xform) is a tool to simplify the creation of forms. Forms can be designed with Excel and XLSForm will convert them to XForms that can be used with ODK tools.

See XLSForm for more information.

File: Questionn...015.xlsx

After submitting a page will appear like this below:



XLSForm

XLSForm (formerly XLS2Xform) is a tool to simplify the creation of forms. Forms can be designed with Excel and XLSForm will convert them to XForms that can be used with ODK tools.

See XLSForm for more information.

File: No file chosen

By selecting download we get our final XForm i.e.,**ODK Form** which through mail will be sent to ODK Collect.

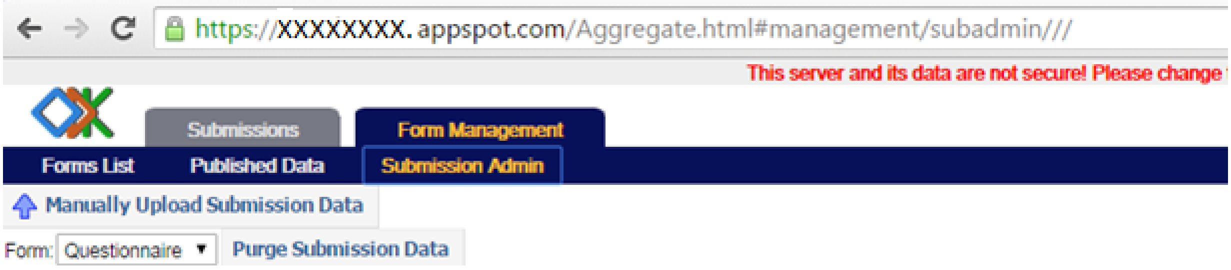
This xml file or XForm, is a popular open form standard that allows one to skip from complex functionality of logic in a consistent way.

6. How to upload forms on server

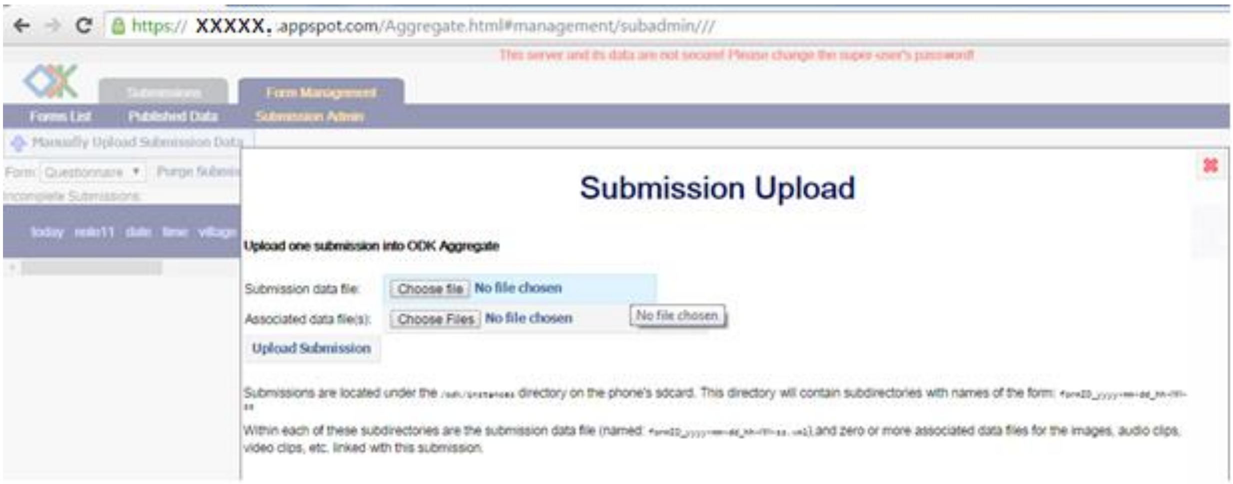
We can upload data by two methods. Either we can upload manually or just send all of finalized forms by using internet connection of our device.

6.1 Manually data upload

Final data is located in instances folder under main **odk** folder of SD Card. We have to send each xml file via gmail. Then download those files in your desktop or laptop. Then select Submission Admin under Form Management, a page will appear like below


https://XXXXXXXX. appspot.com/Aggregate.html#management/subadmin///
This server and its data are not secure! Please change
Submissions
Form Management
Forms List
Published Data
Submission Admin
Manually Upload Submission Data
Form: Questionnaire ▾
Purge Submission Data

Click on **Manually Upload Submission Data** and choose a file from the location where **dataxml files** are saved. Finally select **Upload Submission**and data will be uploaded on server.



Upload one submission into ODK Aggregate

Submission data file: No file chosen

Associated data file(s): No file chosen

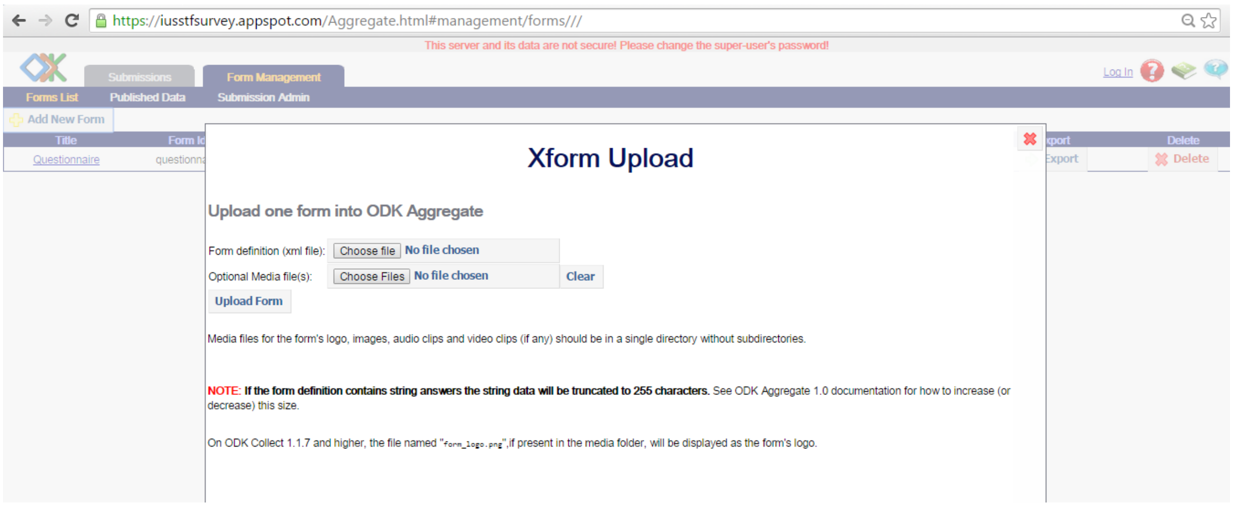
Submissions are located under the /odk/instances directory on the phone's sdcard. This directory will contain subdirectories with names of the form: formID_yyyy-mm-dd_hh-mm-ss

Within each of these subdirectories are the submission data file (named: formID_yyyy-mm-dd_hh-mm-ss.xml),and zero or more associated data files for the images, audio clips, video clips, etc. linked with this submission.

The **other way** is to upload data by sending from device. In ODK Collect we have to select all finalized forms and touch **Send Selected** button**.** While sending final forms we have to keep our server open. There is another way to upload that is done manually.

Note that we can send final forms if and only if our questionnaire is uploaded on server. How we upload our questionnaires given below:

and server will appear with 2 main headings **Submissions** and **Form Management.** Next select on **Form Management** and then click on **Add New Form** which will appear under **Forms List.** Now we can upload our questionnaire XForm by choosing downloaded XForm.



This server and its data are not secure! Please change the super-user's password!

Xform Upload

Upload one form into ODK Aggregate

Form definition (xml file):

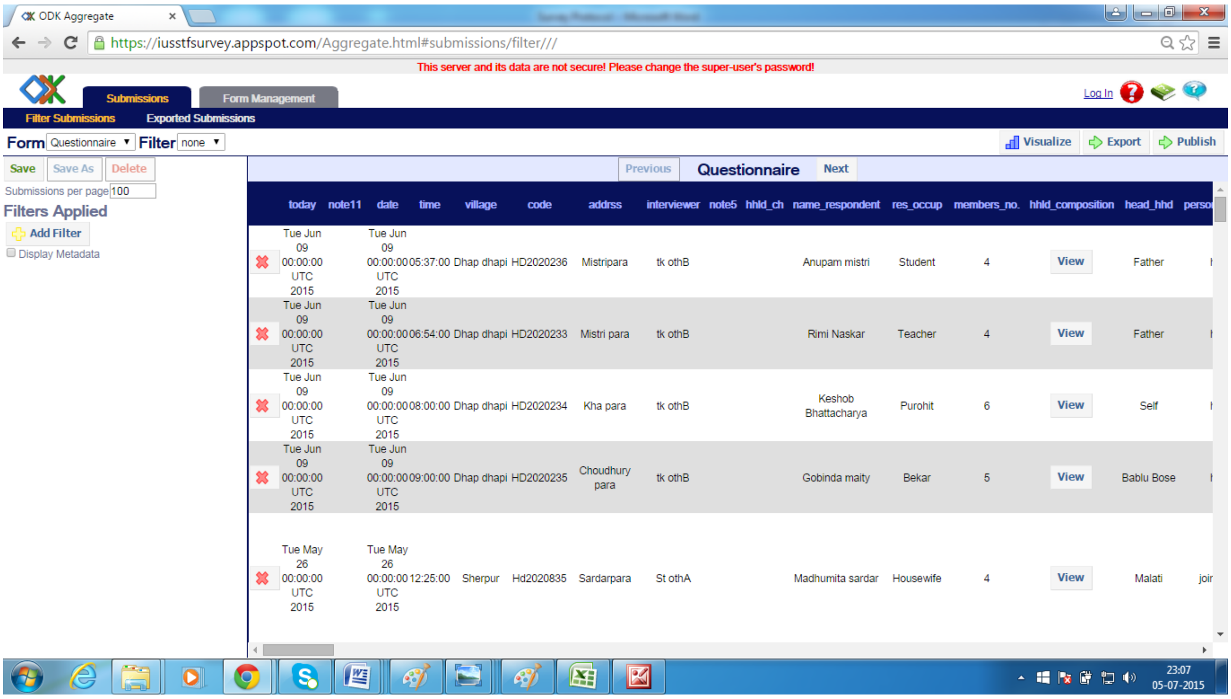
Optional Media file(s):

Media files for the form's logo, images, audio clips and video clips (if any) should be in a single directory without subdirectories.

NOTE: **If the form definition contains string answers the string data will be truncated to 255 characters.** See ODK Aggregate 1.0 documentation for how to increase (or decrease) this size.

On ODK Collect 1.1.7 and higher, the file named "form_logo.png",if present in the media folder, will be displayed as the form's logo.

When data will be uploaded, server will look like this below.



This server and its data are not secure! Please change the super-user's password!

Form Questionnaire ▾ Filter none ▾

Submissions per page 100

Filters Applied

□ Display Metadata

Questionnaire

	today	note11	date	time	village	code	addrss	interviewer	note5	hhld_ch	name_respondent	res_occup	members_no.	hhld_composition	head_hhd	perso
✖	Tue Jun 09 00:00:00 UTC 2015		Tue Jun 09 00:00:00 UTC 2015	05:37:00	Dhap dhapi	HD2020236	Mistripara	tk othB			Anupam mistri	Student	4	View	Father	I
✖	Tue Jun 09 00:00:00 UTC 2015		Tue Jun 09 00:00:00 UTC 2015	06:54:00	Dhap dhapi	HD2020233	Mistri para	tk othB			Rimi Naskar	Teacher	4	View	Father	I
✖	Tue Jun 09 00:00:00 UTC 2015		Tue Jun 09 00:00:00 UTC 2015	08:00:00	Dhap dhapi	HD2020234	Kha para	tk othB			Keshob Bhattacharya	Purohit	6	View	Self	I
✖	Tue Jun 09 00:00:00 UTC 2015		Tue Jun 09 00:00:00 UTC 2015	09:00:00	Dhap dhapi	HD2020235	Choudhury para	tk othB			Gobinda maity	Bekar	5	View	Bablu Bose	I
✖	Tue May 26 00:00:00 UTC 2015		Tue May 26 00:00:00 UTC 2015	12:25:00	Sherpur	Hd2020835	Sardarpara	St othA			Madhumita sardar	Housewife	4	View	Malati	joir

6.2 Send all of finalized forms

In ODK Collect we have an option **Send Finalized Form.** Under this all final forms are available. Select **Toggle all** and then **Send selected** buttons.

Now we have to **Export** data in *.csv format which save all data in Excel Sheet.

This is the **final output** of dataset which we need for our analysis.

7. Troubleshoots

First, after reading manuals and theories when we had started with ODK Aggregate we repeatedly ended up with *END-SCRIPT-FAILURE.* We couldn't understand our mistake from **output message**. After consulting with JU experts we get to know our mistake rather silly mistake. The process was hindered due to obstacle in internet connectivity. While setting up ODK Aggregate we used wifi connection of our Lab which restricted transferring information across web. So this problem was sorted out by using a dongle.

We have a question where respondent is asked to rank over three alternatives and another question where respondent has to select one level of satisfaction regarding criteria of drinking water from each source. Former has three options:1, 2, 3 for each alternative and the later has four levels of satisfaction for each criteria. Respondent has to select one among them. So we want them to appear in a drop down box after each alternative and each criteria. But we didn't perceive how to do this after trial and error of few methods finally we had to post our problem in ODK Community which is a open forum where people discuss about problems regarding ODK.

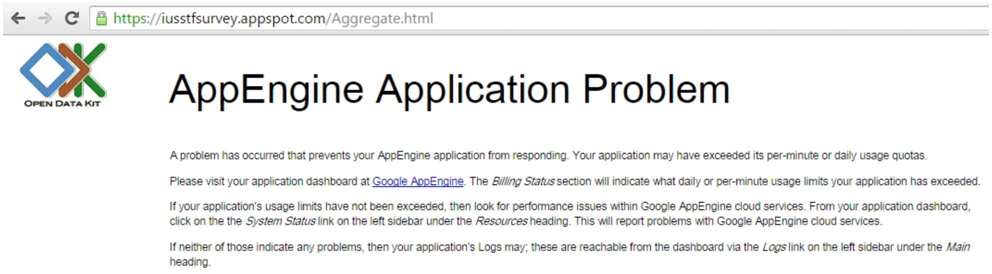
https://groups.google.com/forum/#!forum/opendatakit.

8. Limitations

We have already talked about advantages. But there are limitations too.

1. With full battery backup our device (Micromax) allows to conduct survey of maximum 6 households with disabled auto-rotation and lower brightness.
2. Server has restriction on daily usage. So it was not allowed to use after a certain time.

Server shows a message like below:



https://iusstfsurvey.appspot.com/Aggregate.html

AppEngine Application Problem

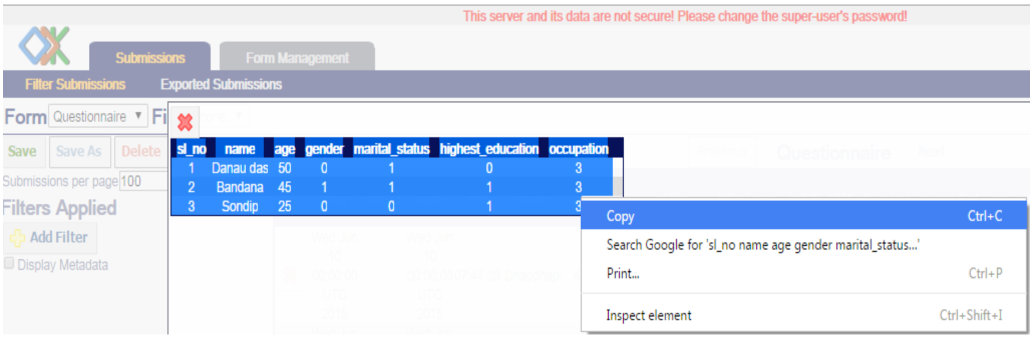
A problem has occurred that prevents your AppEngine application from responding. Your application may have exceeded its per-minute or daily usage quotas.

Please visit your application dashboard at Google AppEngine. The *Billing Status* section will indicate what daily or per-minute usage limits your application has exceeded.

If your application's usage limits have not been exceeded, then look for performance issues within Google AppEngine cloud services. From your application dashboard, click on the the *System Status* link on the left sidebar under the *Resources* heading. This will report problems with Google AppEngine cloud services.

If neither of those indicate any problems, then your application's Logs may; these are reachable from the dashboard via the *Logs* link on the left sidebar under the *Main* heading.

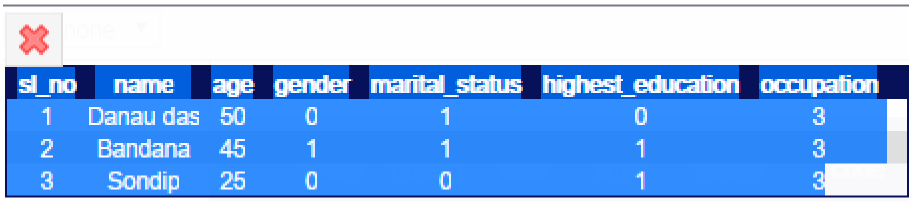
Problem is that we have 3 repeat questions for which we need to copy it from server and paste on excel sheet. But we are bounded by the available time span.



sl_no	name	age	gender	marital_status	highest_education	occupation
1	Danau das	50	0	1	0	3
2	Bandana	45	1	1	1	3
3	Sondip	25	0	0	1	3

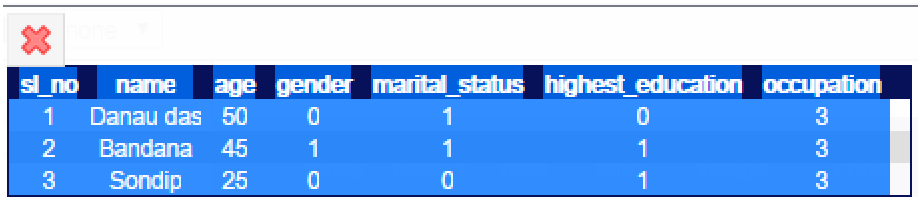
Note:

While copying select upto last digit does not drag cursor to the end of the box. Like this below:



sl_no	name	age	gender	marital_status	highest_education	occupation
1	Danau das	50	0	1	0	3
2	Bandana	45	1	1	1	3
3	Sondip	25	0	0	1	3

But **do not** select like this



sl_no	name	age	gender	marital_status	highest_education	occupation
1	Danau das	50	0	1	0	3
2	Bandana	45	1	1	1	3
3	Sondip	25	0	0	1	3

Notice that in above example we copied by selecting **Copy** option which will appear on **Left click** not by **ctrl+c.** As ctrl+c will not function in this context.

We figured out **two solutions**:

Try to keep submissions below 50, even though capacity per page is 100. Export all data and delete data from server when all informations are transported to excel sheet.

We create a second sever with different mail id and all necessary requirements. In our study we make second server and upload **manually.** We could not send it from device as **submission url** is different for new server and our questionnaire XForm includes previous submission url. All detail informations are given above with 1st server*.