NEMO Documentation

Release 8.0

The Carter Center

Feb 13, 2024

CONTENTS

NEMO is The Carter Center's open-source data collection and reporting system. Equipped with NEMO, enumerators can submit evaluations of a process via Android devices, SMS, or directly online in real-time to field or mission headquarters. NEMO's reporting system organizes enumerator findings, and is relied upon by The Carter Center missions around the globe to analyze and assess data.

The open source license for NEMO is Apache 2.0. NEMO can be hosted on personal servers and users can control who has access to the data.

CHAPTER ONE

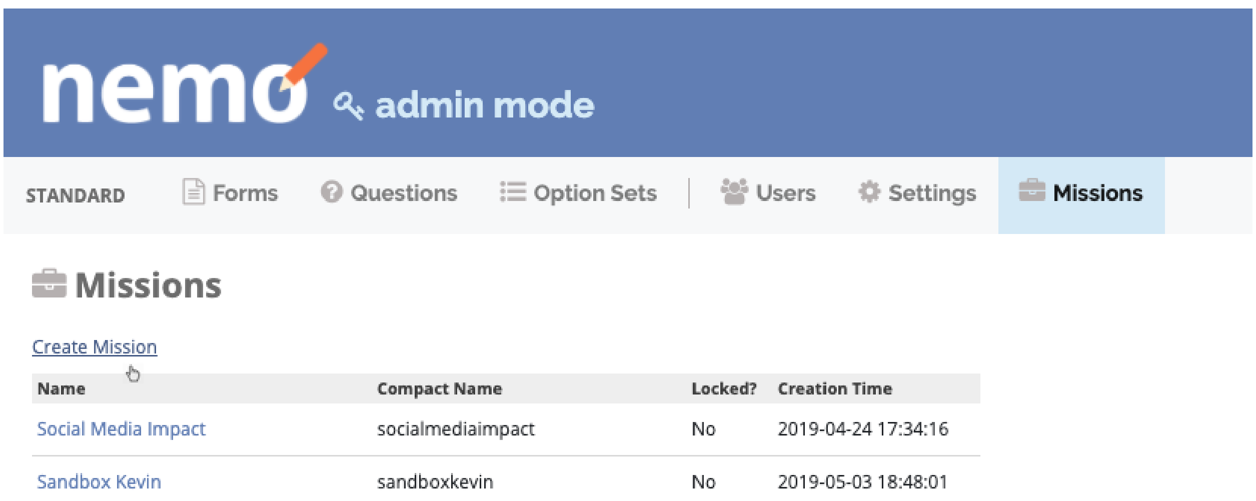
GETTING STARTED

Below are the steps required to create a new mission (Admin access required), create a new form, add questions to the form, then deploy the form.

1.1 Create a mission

Note: Only Admins can create missions.

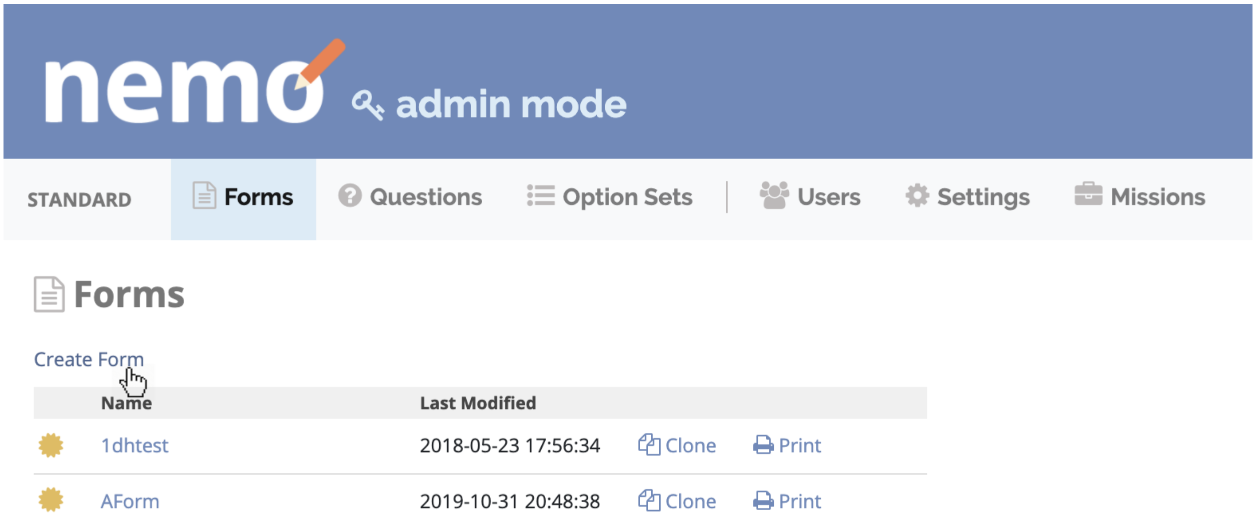
1. Click *Admin Mode* on the top right.
2. Click on *Missions* tab.
3. Click on *Create New Mission*.



Tip: Admins have the option to lock a mission. When a mission is locked, responses and forms cannot be created or edited, and users cannot be added or deleted. To lock a mission, edit the mission and check the box labeled *Locked*.

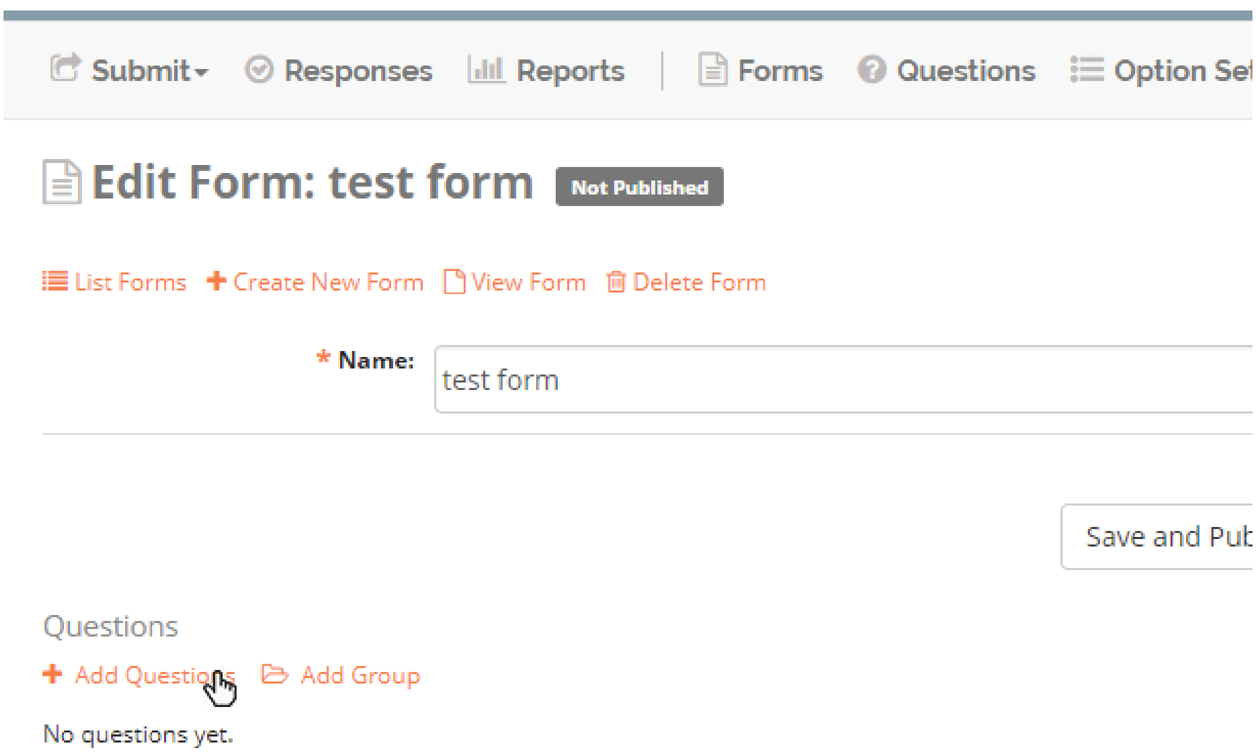
1.2 Create a new form

1. Exit Admin Mode and click on the *Forms* tab.
2. Click *Create New Form* to create your first form.



1.3 Add questions to form

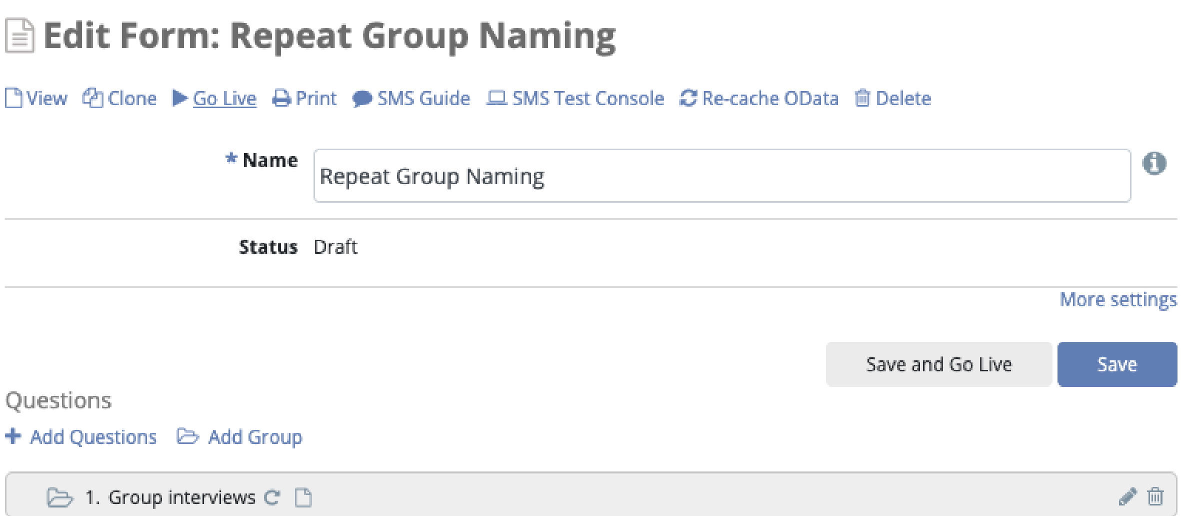
Click on + *Add Question*.



Tip: For more details on question types, see *Import new questions* section.

1.4 Go Live

After adding questions to your form, you need to **Go Live** in order to allow users to submit form responses. To do so click on *Go Live*.



1.5 Submit response

Form responses can be submitted in three different ways:

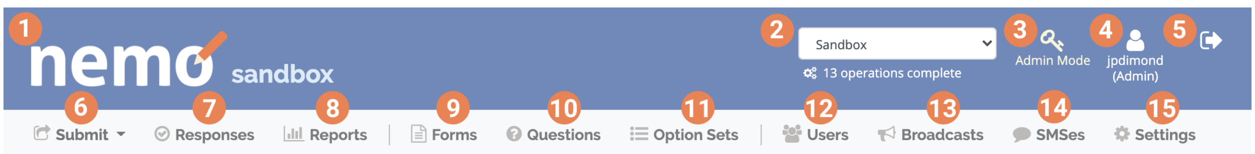
- On the *Web*.
- Using *ODK Collect* on an Android device.
- Via *SMS*.

CHAPTER

TWO

MENU BAR & HOME SCREEN

2.1 Viewing the header



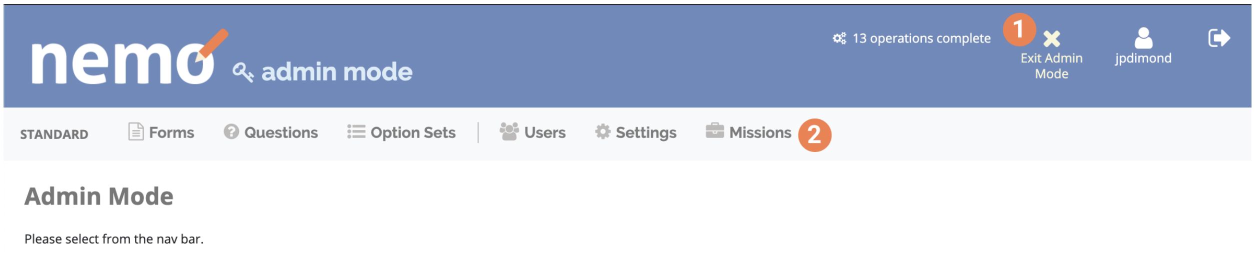
The header is the top section of the screen. Here are its components:

1. **NEMO Logo**. Return to the dashboard for the current mission.
2. **Mission Dropdown**. Change to a different mission.
3. **Admin Mode** (Admins only). Manage standard forms, manage users, create missions, and change system-wide settings.
4. **Edit profile**. Shows your username and role. Click to edit your profile.
5. **Log Out**. Securely end your session.
6. **Submit**. Select and fill out a published form. See *Web Submission* for details.
7. **Responses**. View submitted data.
8. **Reports**. Generate and view reports. See *Reporting and Data Analysis* for details.
9. **Forms**. Create, edit, and import *Forms*.
10. **Questions**. Create, edit, and import *Questions*.
11. **Option Sets**. Create, edit, and import *Option Sets*.
12. **Users**. Manage the users in the mission. See *User Management* for details.
13. **Broadcasts**. Send bulk messages to users. See *Broadcast Messages* Messages for details.
14. **SMS**. View a log of incoming and outgoing SMSes. See *SMS Submission* for details.
15. **Settings**. Change mission settings. See *Settings* for details.

This is the version that admins will see in Admin Mode:

There are only two new components:

1. The **Exit Admin Mode** icon. Click this to exit admin mode.
2. The **Missions** menu. Click this to go to the missions page, where you can create a new mission and see all missions that have already been created.



2.2 Viewing the footer

The footer is the bottom section of the screen. Here are its components:

1. Click **Change Language** to reveal a list of all languages available.
2. The time zone the user has selected. This can be changed on the settings page.
3. The outgoing SMS provider for the current mission (not shown in Admin mode). See *Settings*.
4. The current version number of NEMO.
5. Click on **About NEMO** to learn more about the software.

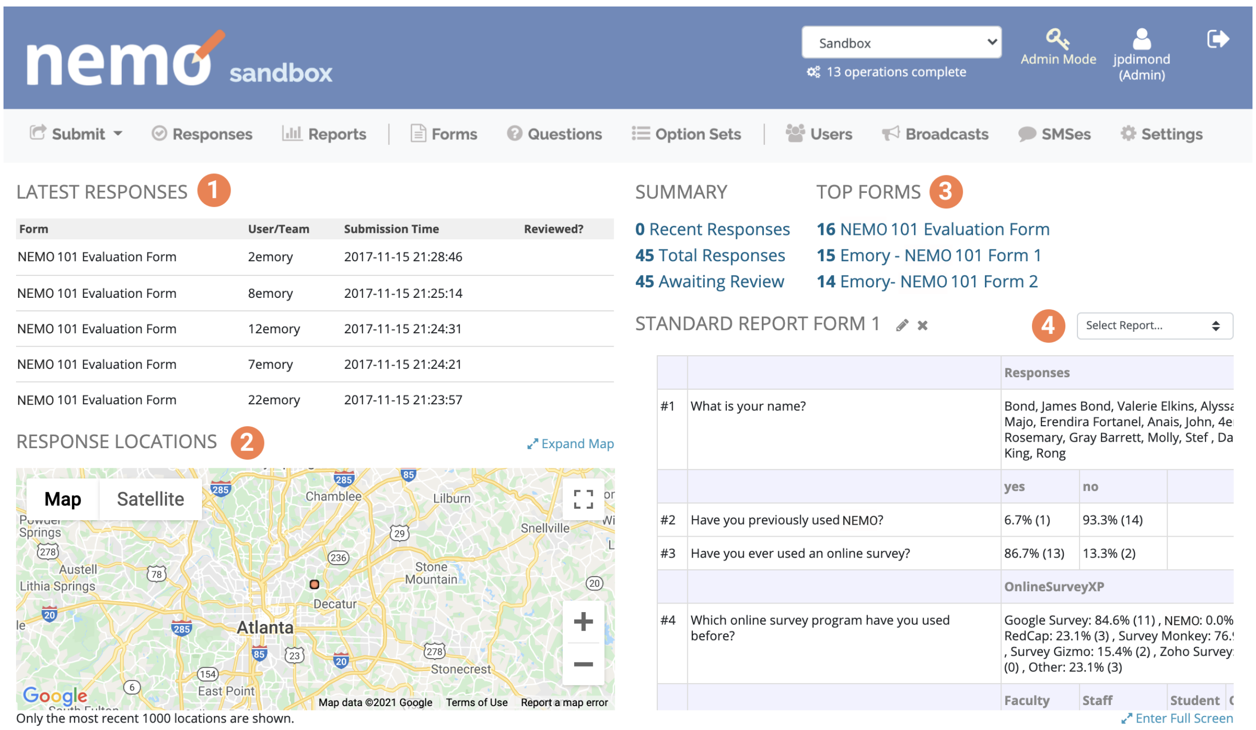
Note: Multilingual support is increasing through volunteer efforts. Right-to-left and left-to-right languages are both supported. When building forms, question and option names can be defined in any language in the ISO 639-2 standard.

2.3 Viewing the dashboard

The dashboard is the home screen for a mission. It gives an overview of collected data, including:

1. A listing of the latest responses to arrive.
2. A map of response locations.
3. Overview statistics on response activity.
4. A custom report created with NEMO's *Report* system.

Note: The report with the most views will be initially selected in the dashboard. You can change the selected report using the dropdown menu.


nemo sandbox
Sandbox
13 operations complete
Admin Mode
jpdimond (Admin)
Submit
Responses
Reports
Forms
Questions
Option Sets
Users
Broadcasts
SMSes
Settings
LATEST RESPONSES 1
Form
User/Team
Submission Time
Reviewed?
NEMO 101 Evaluation Form
2emory
2017-11-15 21:28:46
NEMO 101 Evaluation Form
8emory
2017-11-15 21:25:14
NEMO 101 Evaluation Form
12emory
2017-11-15 21:24:31
NEMO 101 Evaluation Form
7emory
2017-11-15 21:24:21
NEMO 101 Evaluation Form
22emory
2017-11-15 21:23:57
RESPONSE LOCATIONS 2
Expand Map
Map
Satellite
Chamblee
Lilburn
Snellville
Austell
Lithia Springs
Stone Mountain
Decatur
Atlanta
Stonecrest
East Point
Map data ©2021 Google
Terms of Use
Report a map error
Only the most recent 1000 locations are shown.
SUMMARY
0 Recent Responses
45 Total Responses
45 Awaiting Review
TOP FORMS 3
16 NEMO 101 Evaluation Form
15 Emory - NEMO 101 Form 1
14 Emory- NEMO 101 Form 2
STANDARD REPORT FORM 1
4
Select Report...
Responses
#1
What is your name?
Bond, James Bond, Valerie Elkins, Alyssa
Majo, Erendira Fortanel, Anais, John, 4e
Rosemary, Gray Barrett, Molly, Stef , Da
King, Rong
yes
no
#2
Have you previously used NEMO?
6.7% (1)
93.3% (14)
#3
Have you ever used an online survey?
86.7% (13)
13.3% (2)
OnlineSurveyXP
#4
Which online survey program have you used before?
Google Survey: 84.6% (11) , NEMO: 0.0%
RedCap: 23.1% (3) , Survey Monkey: 76.
, Survey Gizmo: 15.4% (2) , Zoho Survey
(0) , Other: 23.1% (3)
Faculty
Staff
Student
Enter Full Screen

CHAPTER

THREE

USER MANAGEMENT

3.1 User permission levels

NEMO has four user levels – Enumerator, Staffer, Coordinator, and Admin. Each user level has a different set of permissions based on the functions they are expected to perform during the missions.

Role	Responsibility	Privileges
Enumerator	Collects and submits data from the field.	• Fill form responses via android devices, via SMS, or NEMO online platform. • Edit own user information. • Generate reports (only from user's own submitted data).
Reviewer	Review collected data and performs real-time analysis.	• All Enumerator privileges. • Review responses submitted by users. • Generate reports from data submitted by users.
Staffer	Communicates with staff deployed to the field.	• All Reviewer privileges. • Broadcast SMS messages to users. • Delete responses.
Coordinator	Designs forms/checklists, reporting structures, and manages users for a specific mission.	• All Staffer privileges. • Create option sets, questions, and forms for the mission. • Create, edit and delete users within a mission.
Administrator	Designs standardized forms/checklists and manages processes accross several missions.	• All Coordinator privileges. • Create new missions. • Create standard option sets, standard questions, and standard forms to be shared/imported across all missions.

3.2 Create users

Create, delete, edit, and manage users on the users page. Administrators can create general users in Admin Mode and assign them to specific missions, or they can create users for specific missions in Mission Mode. Coordinators can only create new users in Mission Mode.

To create new users:

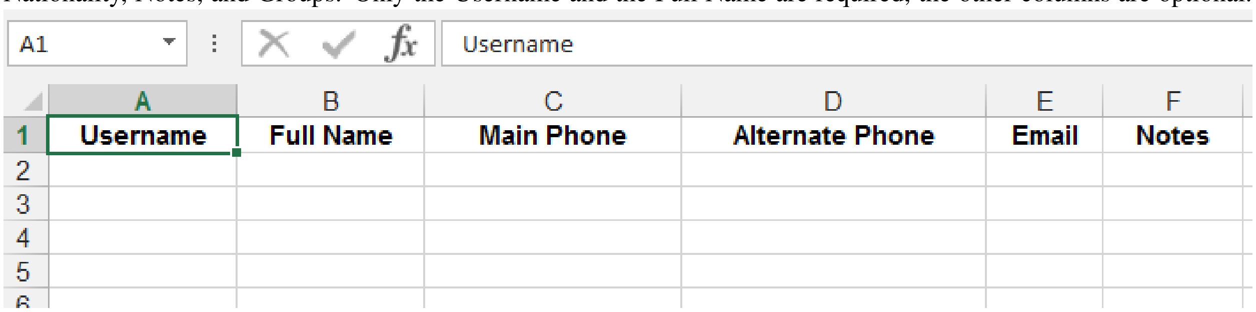
1. Determine whether to work in Admin Mode or Mission Mode.
2. Navigate to the users page by clicking *Users* on the main menu.
3. Click *Create User*.
4. Enter the new user's information.
5. Click *Save*.

3.3 Import users

You can import multiple users from a CSV file (you can export from XLS to CSV from any standard spreadsheet software).

1. From the Users page, click *Import Users*.
2. Click one of the links to download a template *CSV* file.
3. Assemble your user data in the template.
4. In your browser, select the spreadsheet file to upload.
5. Click *Import*.

Note: Available columns are: Username, Full Name, Main Phone, Alternate Phone, Email, Birth Year, Gender, Nationality, Notes, and Groups. Only the Username and the Full Name are required, the other columns are optional.



3.4 Manage existing users

To edit existing user information:

1. Find the user in the list of users.
2. Click located on the same line.

To delete existing users:

1. To delete a single user, click .
2. To delete multiple users, check the boxes to the left of their names and click *Delete Multiple Users*.

3.5 Export in vCard format

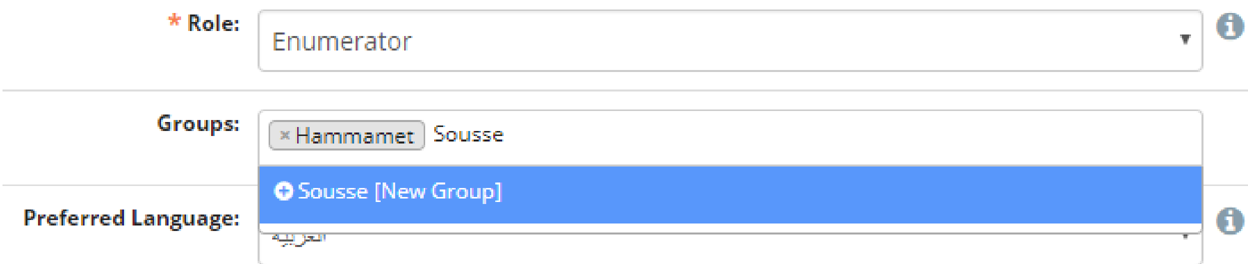
To export users information to a vCard, which is readable by most contact list applications:

1. Check the box next to the name of each user to be exported.
2. Click *Export as vCard*.

3.6 User groups

Placing users into groups makes it easier to search for users and send broadcasts to them. To add a user to a group:

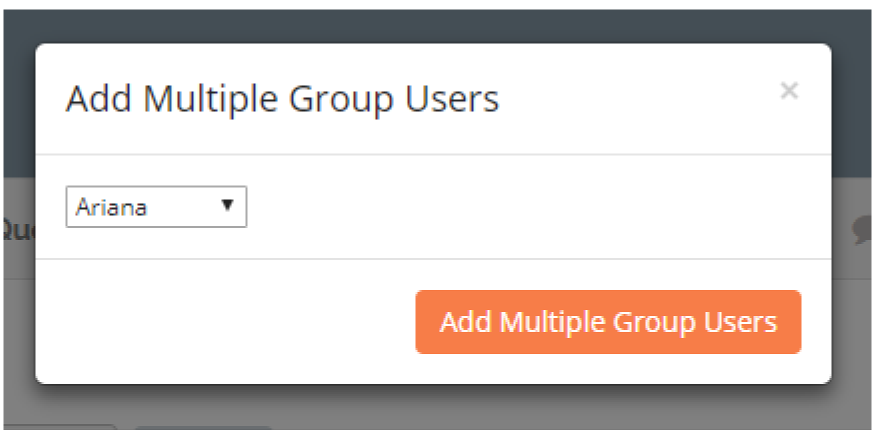
1. Click *Users* menu.
2. Click to edit a user.
3. In *Groups* input, select an existing group from the list or type the name of a new one then click **Enter**.



4. Click *Save* when finished editing.

To add multiple users to an existing group:

1. Click *Users* menu.
2. Check the box next to the users you want to add.
3. Click *Add Multiple Group Users*.
4. Select the group then click *Add Multiple Group Users*.



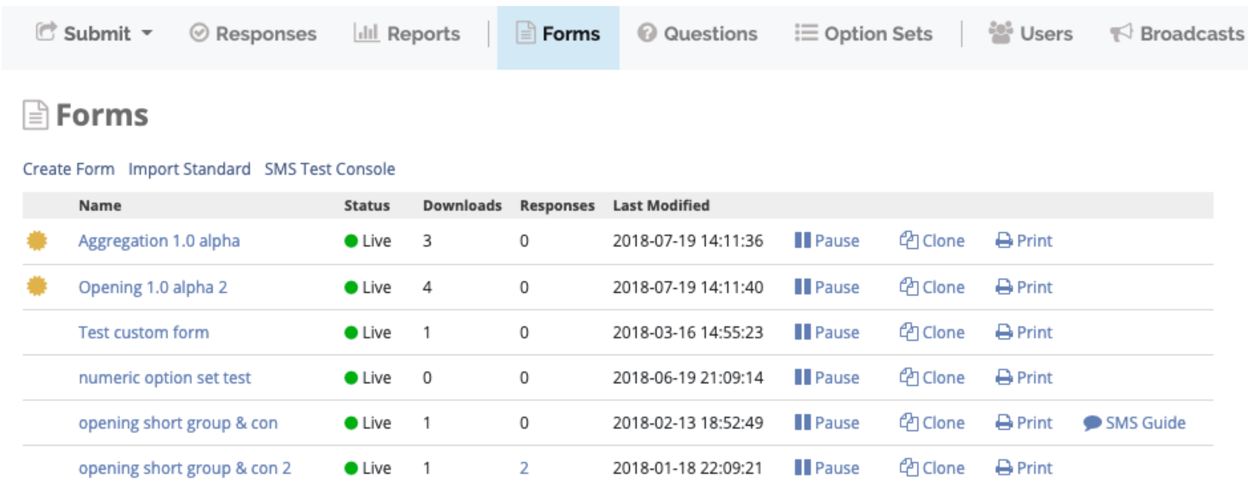
CHAPTER

FOUR

FORMS

4.1 Forms menu

The **Forms** menu lists all the available forms for a mission.



Submit Responses Reports Forms Questions Option Sets Users Broadcasts

Forms

Create Form Import Standard SMS Test Console

Name	Status	Downloads	Responses	Last Modified				
Aggregation 1.0 alpha	Live	3	0	2018-07-19 14:11:36	Pause	Clone	Print	
Opening 1.0 alpha 2	Live	4	0	2018-07-19 14:11:40	Pause	Clone	Print	
Test custom form	Live	1	0	2018-03-16 14:55:23	Pause	Clone	Print	
numeric option set test	Live	0	0	2018-06-19 21:09:14	Pause	Clone	Print	
opening short group & con	Live	1	0	2018-02-13 18:52:49	Pause	Clone	Print	SMS Guide
opening short group & con 2	Live	1	2	2018-01-18 22:09:21	Pause	Clone	Print	

Create New Form Click to *Create a general form*. *SMS Test Console* Click to test form submission via SMS. *Import Standard Forms* Click to *Import standard forms* to the mission. These forms can be edited within the mission.

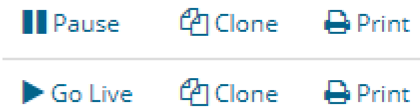
4.1.1 Form status

Forms have 3 different status:

- **Live**: form is available for enumerators to download and submit responses. Only minor changes to the form are permitted like form name, question title and hints.
- **Paused**: form is not available for enumerators to download and submit responses. Like live status, minor changes are permitted.
- **Draft**: form is not available for enumerators to download and all changes are permitted.

4.1.2 Action icons

For every form in the list, action icons are in the right and work as follows:



Pause form. Changes the form status to paused. Go live. Changes the form status to live. Clone. Click to create an identical copy of the form. Print. Click to print the form.

4.2 Create a general form

To create a new Form:

1. Click *Forms* menu.
2. Click *Create New Form*.
3. Type a name for the form in the text box.

To add questions to the Form:

1. Click + *Add Questions*.

- Select from existing questions in the question bank.
- Or create a new question to add to the Form (see *Questions* section below for detailed instructions).

2. Click *Save* when finished editing.

Form settings:

By clicking on *More Settings* you can:

- Set a *Default Response Name* for the form:

  Text entered here will be used as the default name of filled forms in ODK Collect only (it is not reflected in NEMO OData.) By default, the response name will be the title of the form.

  You can enter $QuestionCode to include the value of a question in the form.

  For example, entering: *Survey for $VillageNum-$Household*

  might name the form: *Survey for 7-Smith*

  This assumes you have questions with codes 'VillageNum' and 'Household'.

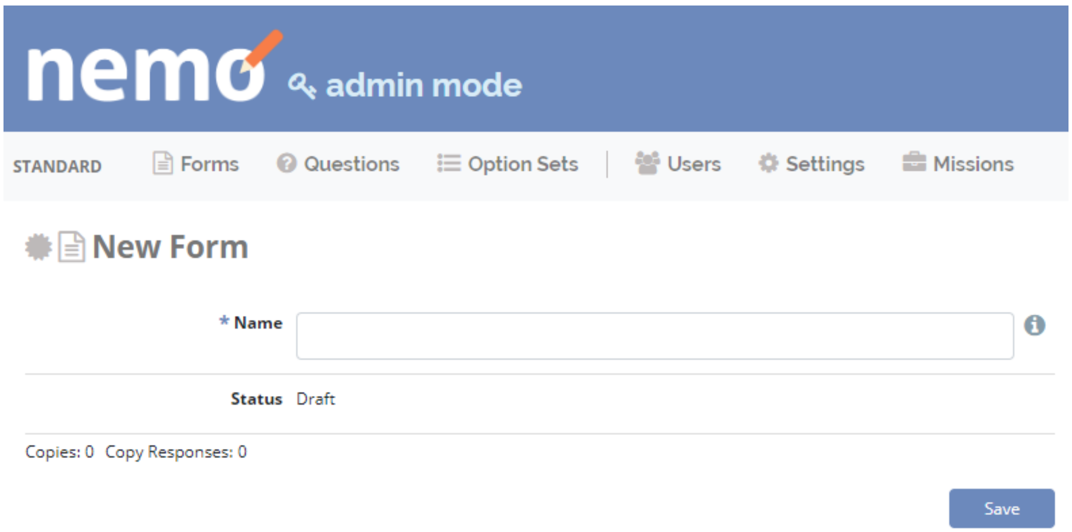
  You can also enter an XPath calculate expression by wrapping it with calc(): *calc($VillageNum + 100 div 2)*

- Check the *Allow Incomplete* box to allow forms to be submitted without required questions filled in.
- Check the *SMSable* box if you want to be able to receive responses to the form via SMS. For more information check *SMSable form*.

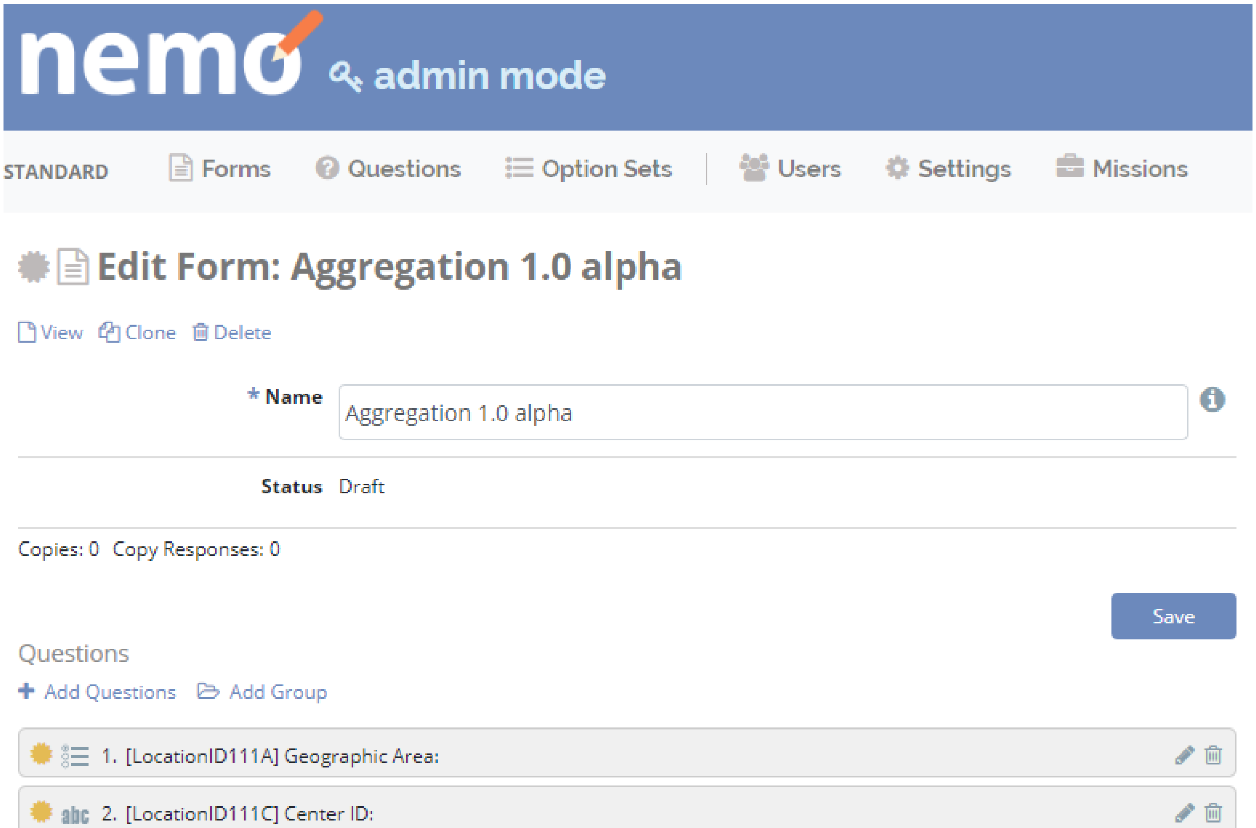
4.3 Create a standard form

Forms, questions, and option sets created in **Admin Mode** can be reused in multiple missions. If the Admin Mode link does not appear on the screen, permission is not granted to create standard forms.

1. Click *Admin Mode*.
2. Click *Forms* menu.
3. Click *Create New Form*.
4. Type a name for the form in the text box.
5. Click *Save*.



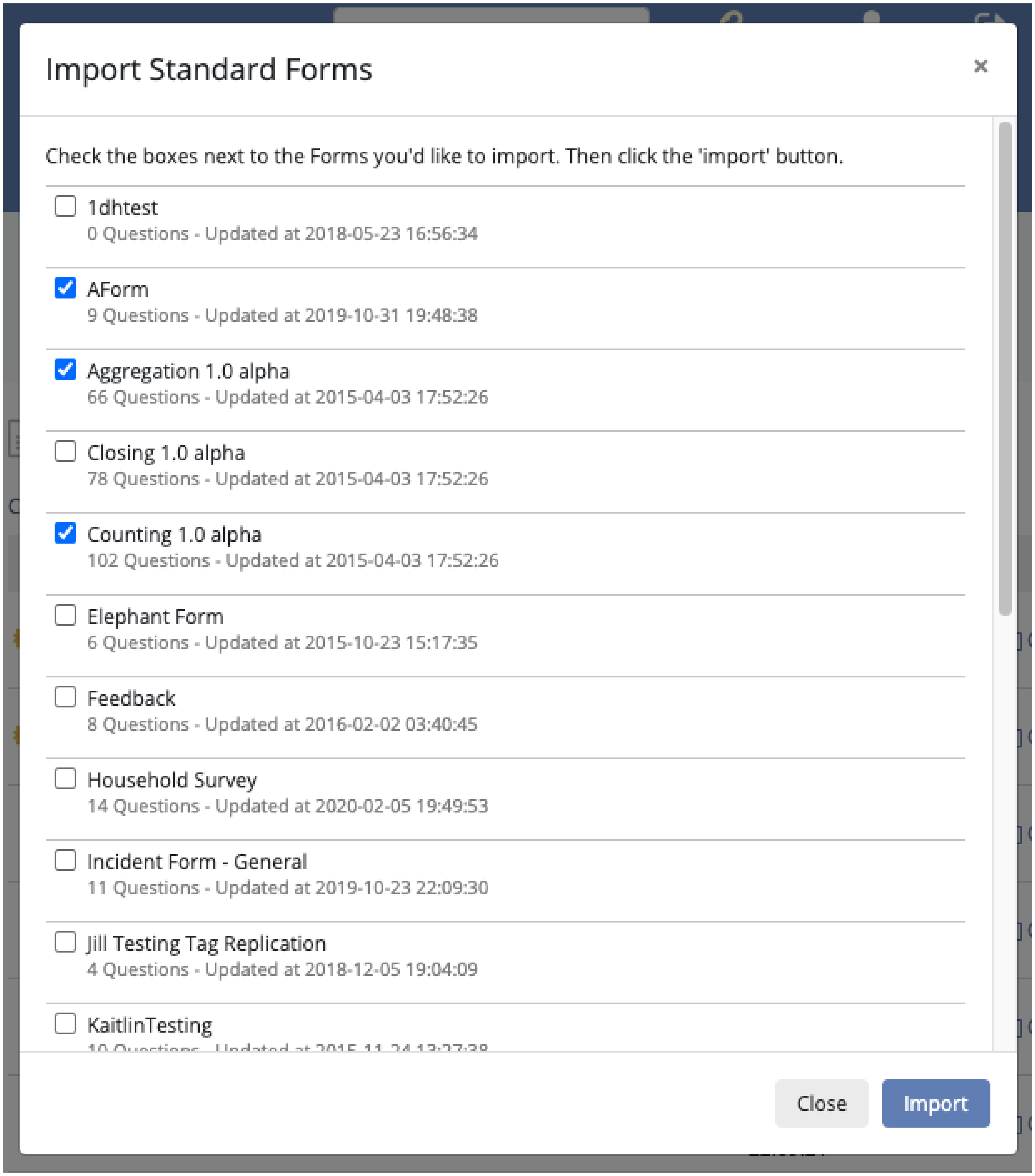
6. Click + *Add Questions* to add questions to the form.
7. To create a group of questions, click *Add Group*.
8. Click *Save*.



4.4 Import standard forms

To import a Standard Form:

1. Click *Forms* menu.
2. Click *Import Standard Forms*.
3. Check the boxes next to the forms you want to import.
4. Click *Import*.


Import Standard Forms
Check the boxes next to the Forms you'd like to import. Then click the 'import' button.
1dhtest
0 Questions - Updated at 2018-05-23 16:56:34
AForm
9 Questions - Updated at 2019-10-31 19:48:38
Aggregation 1.0 alpha
66 Questions - Updated at 2015-04-03 17:52:26
Closing 1.0 alpha
78 Questions - Updated at 2015-04-03 17:52:26
Counting 1.0 alpha
102 Questions - Updated at 2015-04-03 17:52:26
Elephant Form
6 Questions - Updated at 2015-10-23 15:17:35
Feedback
8 Questions - Updated at 2016-02-02 03:40:45
Household Survey
14 Questions - Updated at 2020-02-05 19:49:53
Incident Form - General
11 Questions - Updated at 2019-10-23 22:09:30
Jill Testing Tag Replication
4 Questions - Updated at 2018-12-05 19:04:09
KaitlinTesting
Close
Import

4.5 Create groups and grids

4.5.1 Groups

Grouping questions by context makes it easier for users to fill forms. Groups cannot be placed within questions or other groups.

To create a group:

1. Create or edit an existing form.
2. Create or add questions to the form.
3. Create at least one group.
   a. Click *Add Group*.
   b. Give the group a unique name.
   c. If you want to make the group of questions repeatable, check the box *Repeatable?*. Example: if entering details of multiple family members in a household.
   d. If you want to show the group of questions in the same screen in ODK Collect, check *Show on One Screen*.
   e. Choose the Display Logic.
   - Always display this group.
   - Display this group if all of these conditions are met.
   - Display this group if any of these conditions are met.
   f. Click *Save*.
4. Click and drag the questions in the desired order of appearance.
5. Click and drag groups in the desired order of appearance.
6. Drag questions intended for groups to the right so that they appear indented in relation to the group.
7. Click *Save* to save the form, or click *Save and Go Live* if the form is ready to be used.

4.5.2 Grids

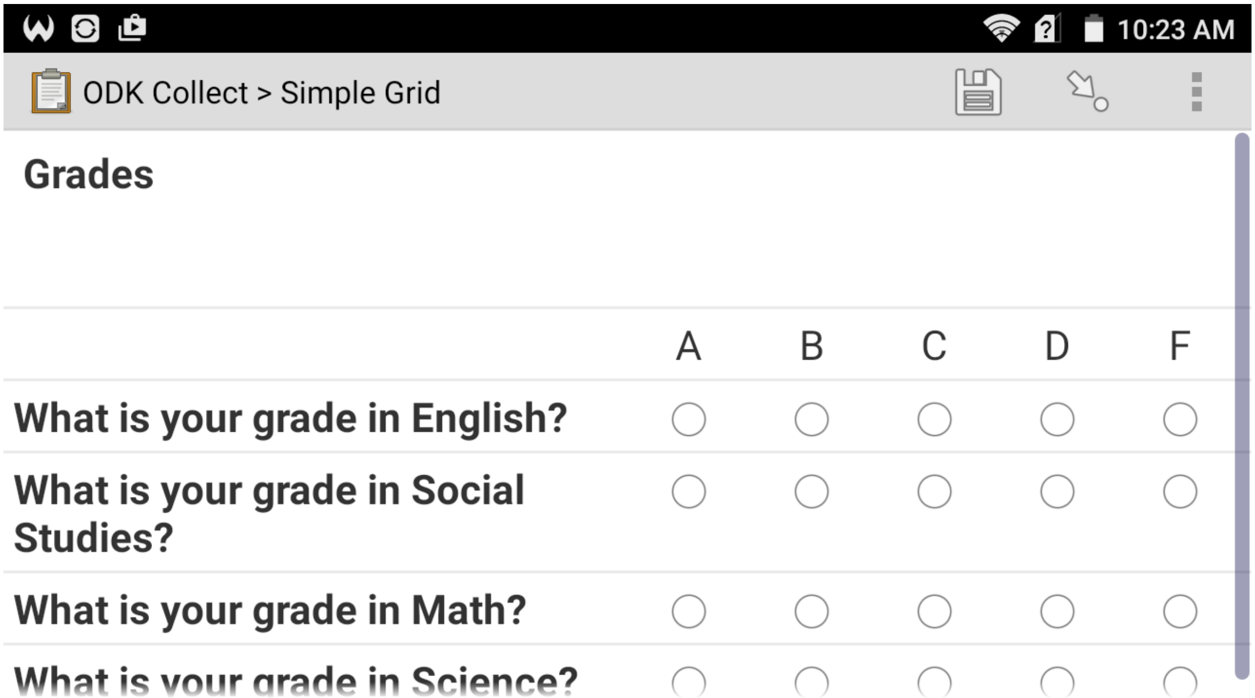
Groups can be used to create grids in ODK Collect, example:

Note: Grids can only be created under certain conditions:

- Questions must be in the same group.
- Questions must be of **Select One** type.
- Questions must have the same option set.

To create a grid:

1. Create or edit an existing form.
2. Create or edit an existing group.
3. Place **Select One** questions with the same option set in the group.

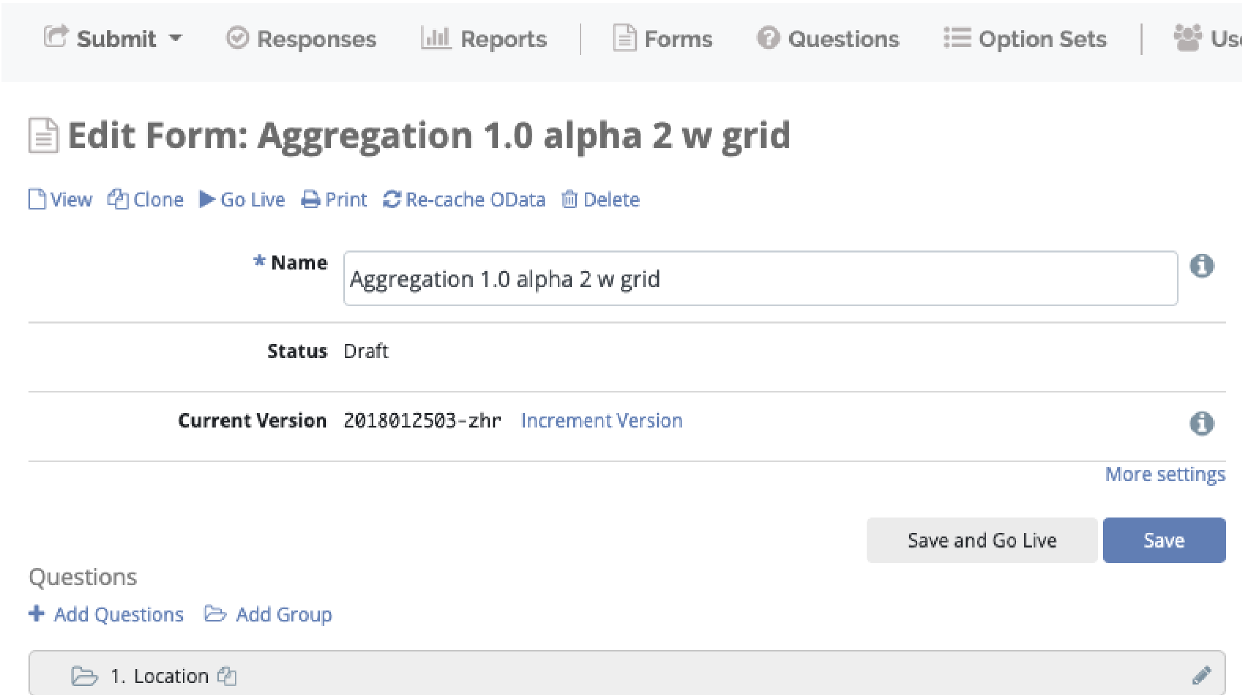


4. Click *Save* to save the form, or click *Save and Go Live* if the form is ready to be used.

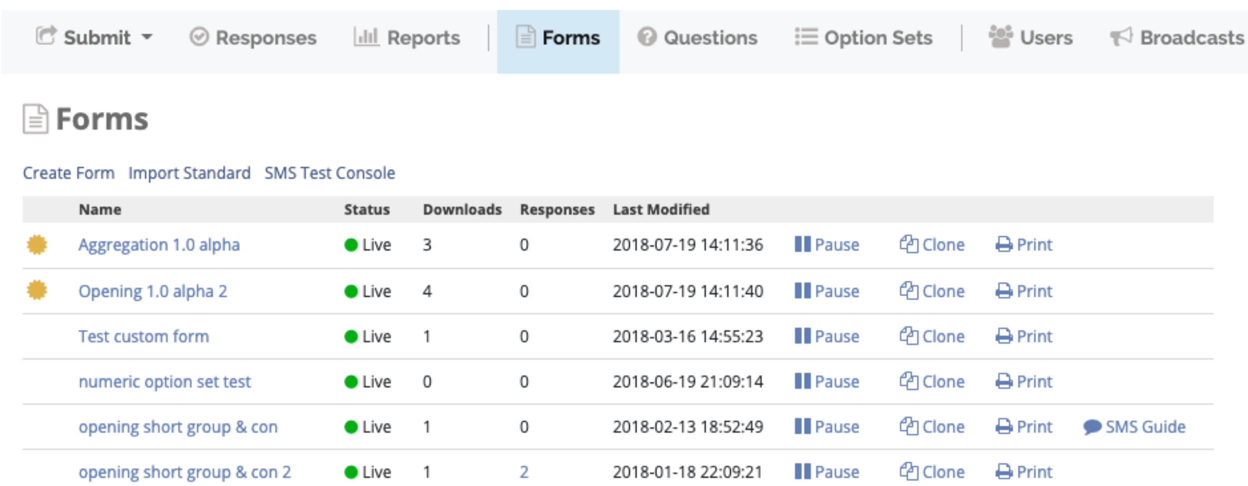
[2] Go live ——-

Once a form has been created, it must go live before users can submit responses. you can do so when editing a form by:

- Clicking Go Live on the top of the screen.
- Or clicking *Save and Go Live* button.



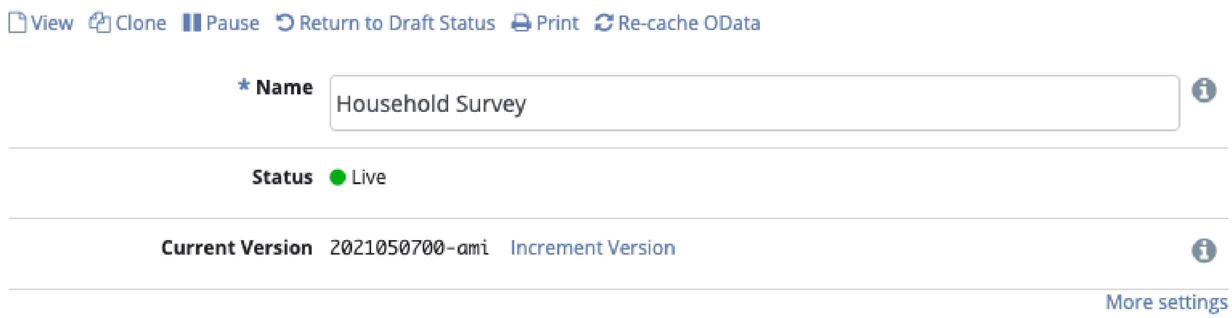
You can also Go Live from the forms list menu by clicking Go Live:



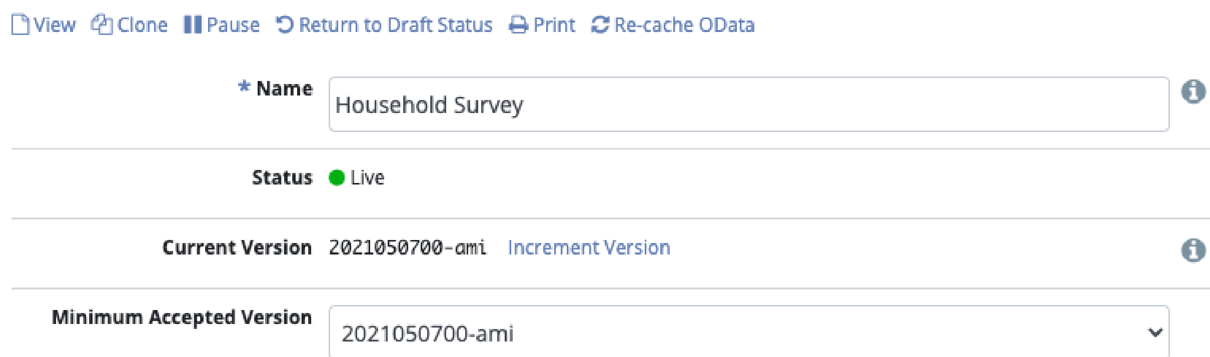
4.6 Form version

Each form version has a 10 digit ID and a 3 letters code (example: 2019092500-oyt). The form versioning system is intended to indicate small changes in the form design so that enumerators can update their copies of the form.

When editing a form, click *Increment Version* to increment the form version.



You can also choose the minimum accepted version of the form, to do so click *More settings* then *Minimum Accepted Version*



4.7 Print form

To print a hardcopy of a form:

1. Click *Forms* menu.
2. Select a form from the list.
3. Click *Print*.
4. A dialog will show up saying that you need to activate background colors and images. Click *OK* then activate these in your system/browser print options.

CHAPTER

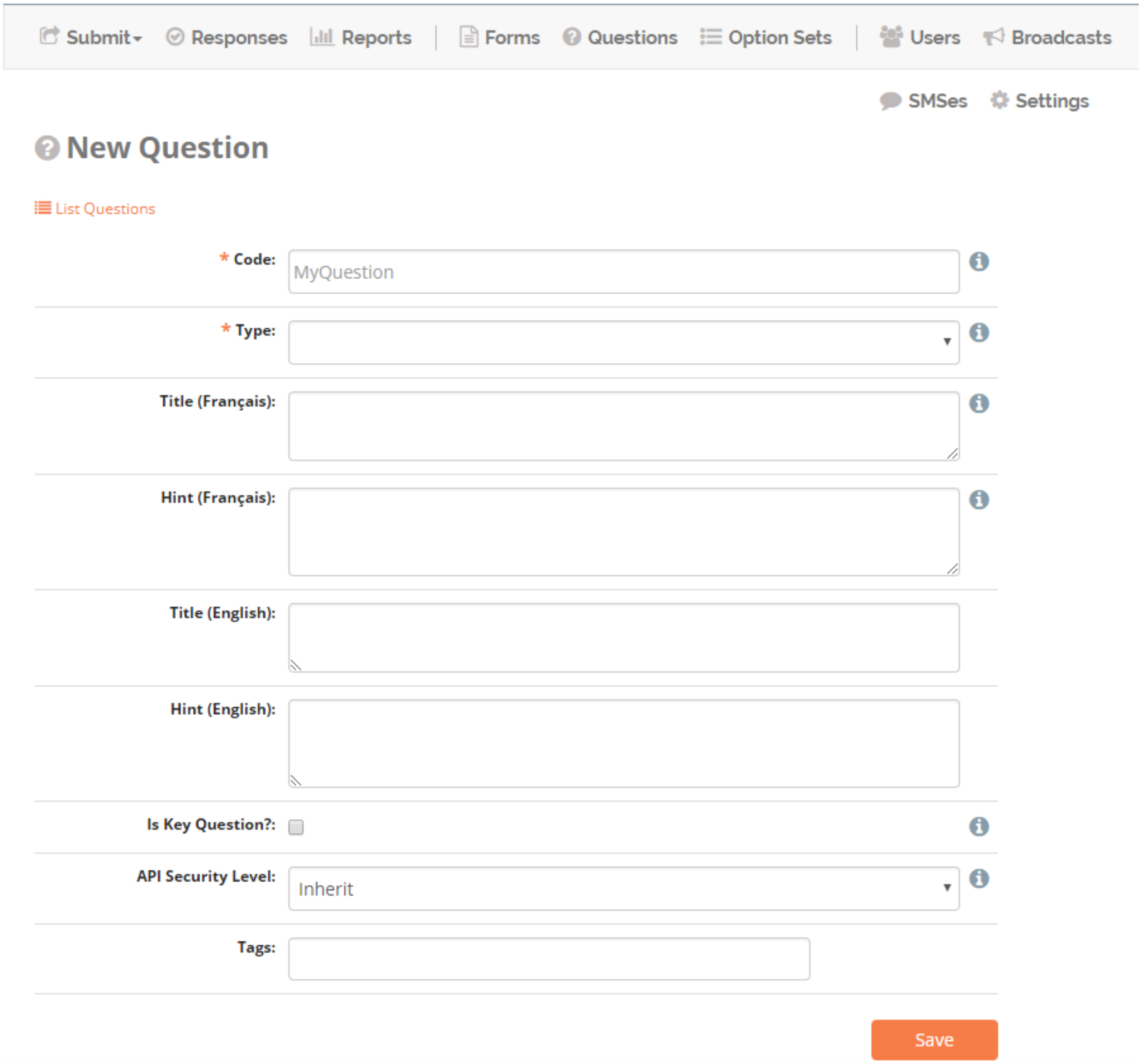
FIVE

QUESTIONS

5.1 Create a question

To create a question:

1. Click *Questions* in the menu bar.
2. Click *Create New Question*.



Code
: A unique name for the question.

Type
: Type of the question. More details can be found in *Import new questions* section.

Title
: Title of the question.

Hint
: Further instructions for the question. More details can be found in *Hints* section.

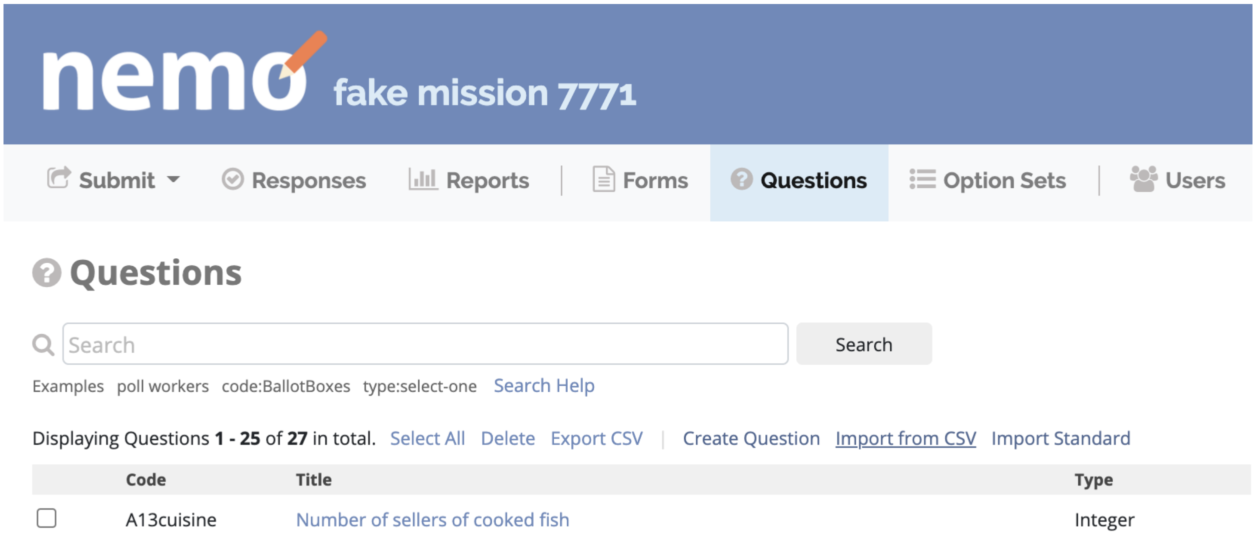
Is Key Question
: Answers to key questions help summarize responses and appear in the response listing. If this box is marked, a column showing these answers will be added in the Responses tab.

Tags
: Add *Tags* to this question.

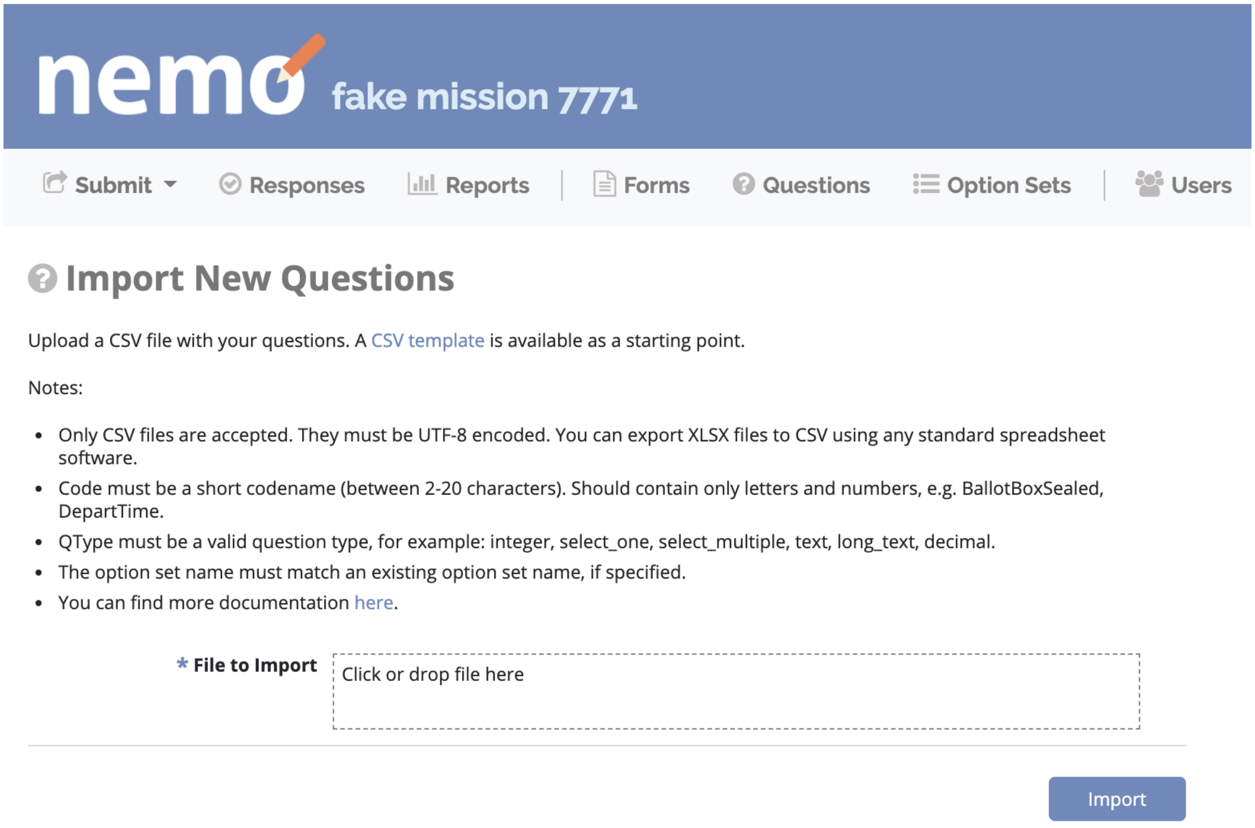
5.2 Import new questions

Upload a CSV to import questions. Importing from a CSV is very useful for questions with many translations.

1. Click *Questions* in the menu bar.



2. Click *Import from CSV*
3. Click the *CSV template* link to download a CSV template to use in Excel or other spreadsheet software. Note that the file you upload must be CSV format.
4. Upload the completed CSV file (description of columns below) and click *Import*.



The format of the CSV columns are as follows:

Code	QType	Option Set Name	Title[en]	Hint[en]	Title[fr]	Hint[fr]

- *Code* Code must be a short codename (between 2-20 characters). Should contain only letters and numbers, e.g. BallotBoxSealed, DepartTime.
- *QType* The question type. The question types that are supported for importing questions are the following (must follow underscore case): integer, select_one, select_multiple, text, long_text, decimal
- *Option Set Name* If using a select_one or select_multiple question type, include an existing option set name.
- *Title[en]* The title of the question in English or the default language of the mission. "en" represents the language code. Supported languages can be found in the mission settings where you can add or edit existing languages.
- *Hint[en]* The hint of the question in English or the default language of the mission. Hints are optional.
- *Title[fr]* Any additional languages that you would like to add (in this case French), should alternate with title and hints with the language shortcode in brackets. You can find view what languages are supported or add additional ones in the Mission settings.

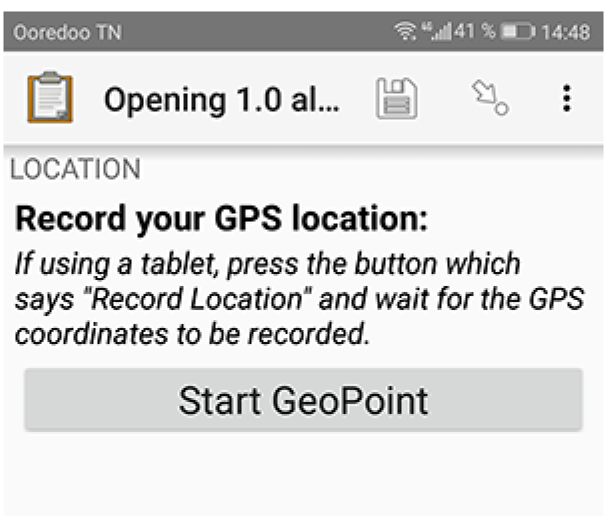
5.3 Types of questions

Type	Description
Text	Short text that shows in a single line of input.
Long Text	Long text that shows in multiple lines of input.
Integer	Numeric answer that must be a whole number without decimals.
Decimal	Numeric answer that allows for decimals.
Location	GPS location of the user.
Select One	Only one answer can be selected from a multiple choice option set.
Select Multiple	Multiple answers can be selected from a multiple choice option set.
Date/Time	Enter both date and time.
Date	Date only field.
Time	Time only field.
Image	The image should be at most 5MB. Accepted formats are: jpg, png.
Annotated Image	The image should be at most 5MB. Accepted formats are: jpg, png. You can add markup to the image.
Signature	Sign with a finger in ODK Collect or upload a signature image in NEMO.
Sketch	Sketch an image with a finger.
Barcode	Scan a barcode with ODK Collect.
Audio	Record or select a sound from your device. File size should be at most 10MB. Accepted formats are: mp3, ogg, webm, wav.
Video	Record or select a video from your device. File size should be at most 10MB. Accepted formats are: 3gp, mp4, webm, mpg, wmv, avi.

5.4 Hints

Hints are optional help texts used to provide additional instructions on the question.

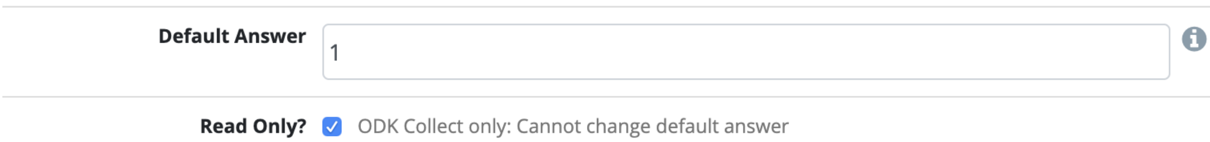
- On NEMO desktop, click on the right of the question to see the hint.
- On ODK Collect (NEMO Android app), the hint will be shown below the question as follows:



5.5 Read-Only

The NEMO development team is working on a read-only question type. In the meantime, you can still create read-only text on your form (ODK forms only). To create read-only text:

- Create a *Text* question.
- Enter the read-only text in the question's *Title*.
- Enter a *Default value* (value does not matter, as it will be read-only).
- Select the *Read-only* option. The read-only option will only appear if there is a default value.



5.6 Language translations

Question titles and hints can be translated into any language that has been set for the mission. To add a language to a mission, go to *Settings* menu and edit *Preferred Languages*.

When editing a question. Title and Hint will show up for selected languages:

Note: To view the translation, change the language by clicking *Change Language* in the footer. For example, the

French translation of an English question will appear once French is selected.

5.7 Tags

Tags are an easy way to organize information around a common keyword or theme. In NEMO, questions may be tagged with keywords or descriptors, which may be used to find all of the questions with the same tag. Tags are also used in Reports to help sort information assigned to questions.

To tag questions:

1. Open or create the question you wish to tag.
2. Type in the *Tags* box and click *Enter* to add a tag.

Note:

- You can add multiple tags.
- To delete a tag, click next to the tag.

3. Click *Save* after adding tags to save your changes.

5.8 Metadata type

Metadata type is a special value that can be pre-filled into a question. If chosen the question will be automatically hidden and not required, and any conditions will be removed. For now the Metadata is only available for Date/Time question type.

1. Create a new Date/Time question.
2. Click the *Metadata Type* dropdown.
3. Select the Metadata you want to record.

Note:

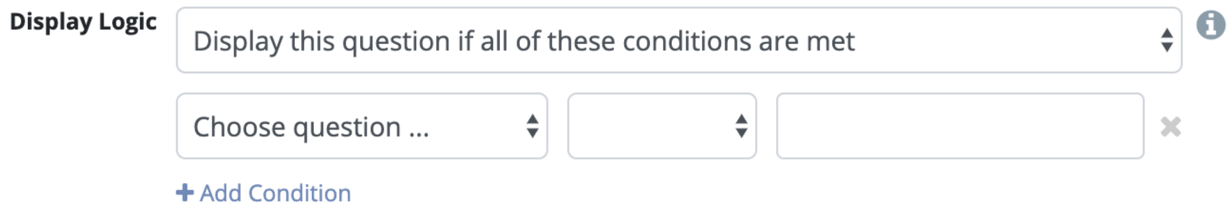
- Form Start Time: will record the time Enumerator started the form.
- From End Time: will record the time Enumerator ended the form.

5.9 Advanced options

The following features are only available for questions that are added to form.

5.9.1 Display logic

By default all questions are shown in the form. Display logic controls which question to show depending on conditions.



To edit display logic:

1. Click *Forms* menu.
2. Click next to the form you want to edit.
3. Click on the *Question* you want to edit.
4. Click on the display logic dropdown and choose between three options:

- Always display this question.
- Display this question if all of these conditions are met.
- Display this question if any of these conditions are met.

Note:

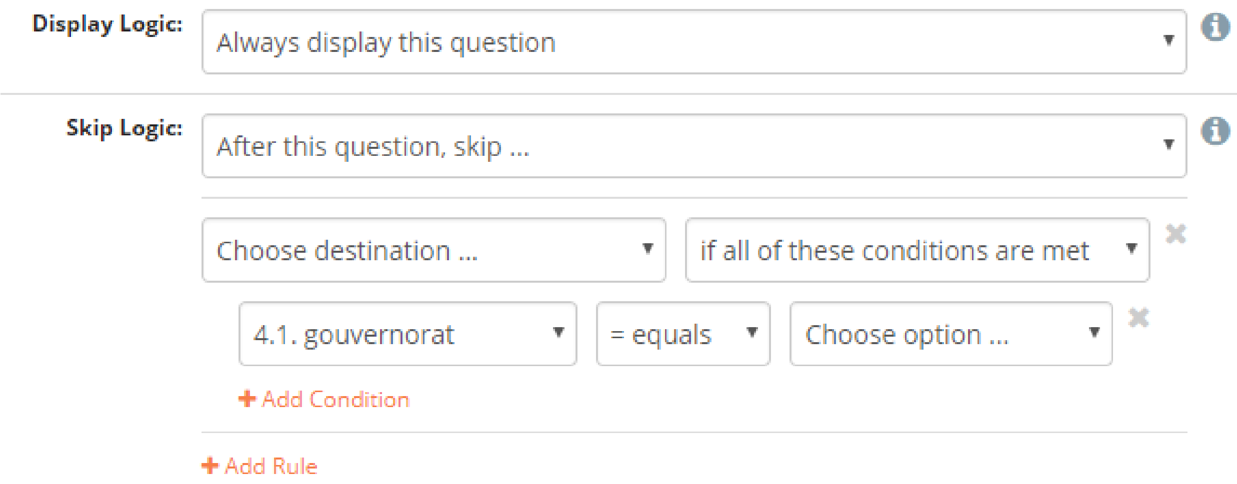
- Click + *Add Condition* if you want to add another condition for the same question.
- If you would like to use advanced expressions such as XPath, you should use default answer to perform the calculation, and then use display logic on a separate question. See Default answer and XPath Expressions below.

5.9.2 Skip logic

On ODK Collect (NEMO Android app), by default when you swipe left or click you will be redirected to the following question in the form. With the skip logic you can go to any question on the form if conditions are met.

To edit skip logic:

1. Click *Forms* menu.
2. Select a form from the list and click .
3. Click on the *Question* you want to edit.
4. Click on skip logic dropdown and select *After this question, skip* . . . .
5. Choose the destination and conditions to be met.

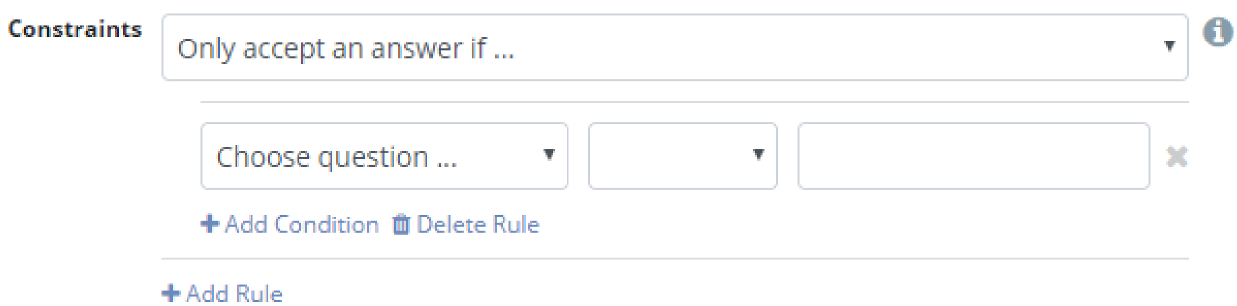


5.9.3 Constraints

Constraints are conditions that must be met in order for an answer to be accepted. This feature is only available on the mobile app **ODK Collect**.

To edit constraints:

1. Click *Forms* menu.
2. Select a form from the list and click .
3. Click on the *Question* you want to edit.
4. Click on constraints dropdown and select *Only accept an answer if* . . . .
5. Select the conditions and rules to be met.



Note:

- Constraints can only be added on previous questions.
- When editing an Integer question type you can also add a constraint about the *Minimum* and *Maximum* value.

5.9.4 Default answer

Text entered here will be pre-filled in the answer space (for ODK Collect only).

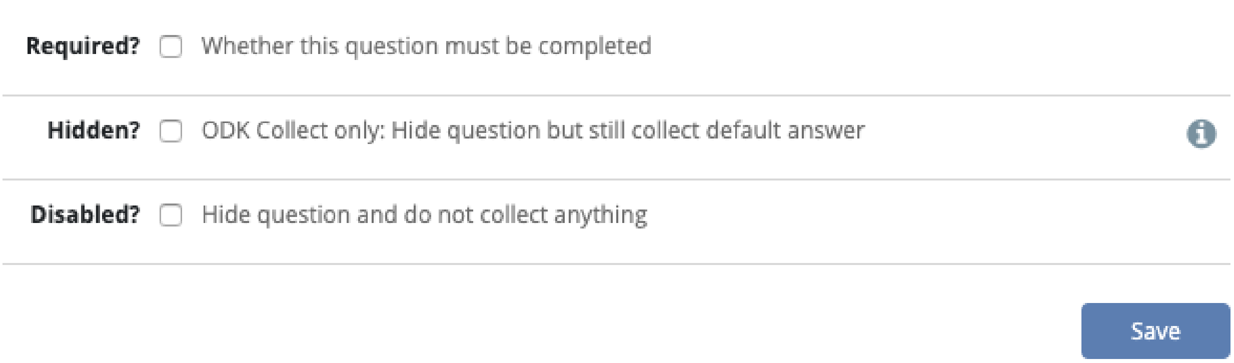
Previous answers: You can enter a `$QuestionCode` to include the value of a previous question. If the question has an option set and has values assigned to each option, you can use `$QuestionCode:value` to get the value of the answer. You can use the value to calculate a sum or average of the answers over multiple questions. Both of these types of expressions must be wrapped in a calc(). See the XPath expression section below for examples.

Repeat groups: When using repeat groups, you can access the number of the current item in the repeat group by using `$!RepeatNum`

For example, entering ID: `$Household-$!RepeatNum` would pre-fill the answer with ID: 176-2 for the second person in household 176, assuming you have a question with code 'Household'.

You can also enter an XPath expression by wrapping it with calc(). See the XPath expressions section below.

5.9.5 Required, hidden, and disabled options



- *Required*: Check *Required?* box to make this question required. Form cannot be submitted if not answered, unless an override code is provided.
- *Hidden*: Check *Hidden?* to hide question on the form but still collect default answer (ODK Collect only). For web and SMS forms, the question will behave as if disabled.
- *Disabled*: Check *Disabled?* to hide the question and not collect anything.

5.9.6 XPath expressions

An XPath expression is a function, operator, or value from previous responses that are dynamically generated as the form is filled out. XPath is a query language used to perform operations on XLSForms (the form standard that ODK Collect uses). Currently, only the default answer field on a question supports XPath expressions when used with ODK Collect. All XPath expressions must be wrapped in a calc() function in a question's default answer field.

Example `calc(($likert1:value * 2) + $likert2:value)`

All XPath expressions supported by ODK should be supported by NEMO. This includes boolean logic, string parsing, and number functions. Details on XPath operators (math and logic) can be found here. Complete documentation on XPath functions can be found here.

Common Scenarios

Conditionally display intervention

I want to display an intervention only if the sum of 2 likert questions is greater than 10.

1. Create Likert questions and an option set with values set as a number.
2. Create a question that calculates the score (e.g. put `calc($likert1:value + $likert2:value)` in the default answer.)
3. Create the intervention question and use display logic to only show the question if the default answer is > 2.

Using a Counter

While NEMO does not explicitly have a counter question type, you can access the number of an item within a repeat group. For example, you could put `calc($Household-$!RepeatNum)` in a default answer for a question. The answer would be with ID: 176-2 for the second person in household 176, assuming you have a question with code 'Household'.

Generate a Random String

I want to generate a random string to be associated with a response.

1. Create a text question and put `calc(uuid(8))` in the default answer. The number parameter is the length of the string you wish to generate.

CHAPTER

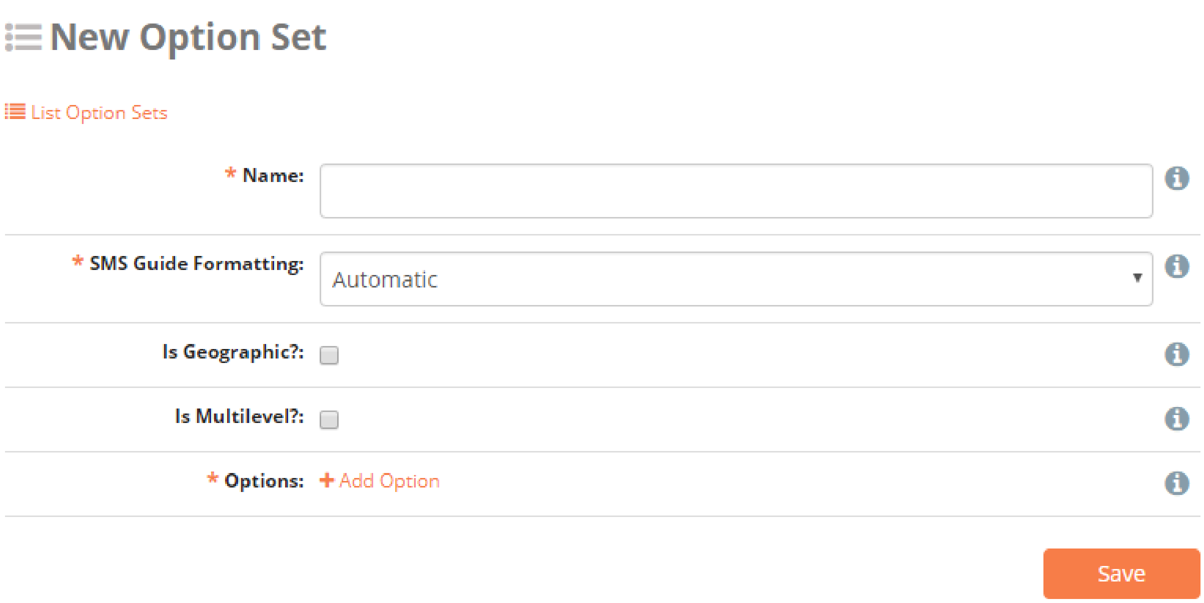
SIX

OPTION SETS

The Option Sets page is where the answers for **Select One** and **Select Multiple** question types can be created. Click *Option Sets* on the menu bar to create or edit existing Option Sets.

6.1 Create new option set

1. Click *Option Sets* menu.
2. Click *Create New Option Set*.



Name A unique name for the Option Set.

Is Geographic? Example, a set of provinces or regions. Responses that are geographic will appear on the map in the Dashboard. When checked, a new checkbox *With Coordinates?* is shown. If the options contain coordinates, check this box.

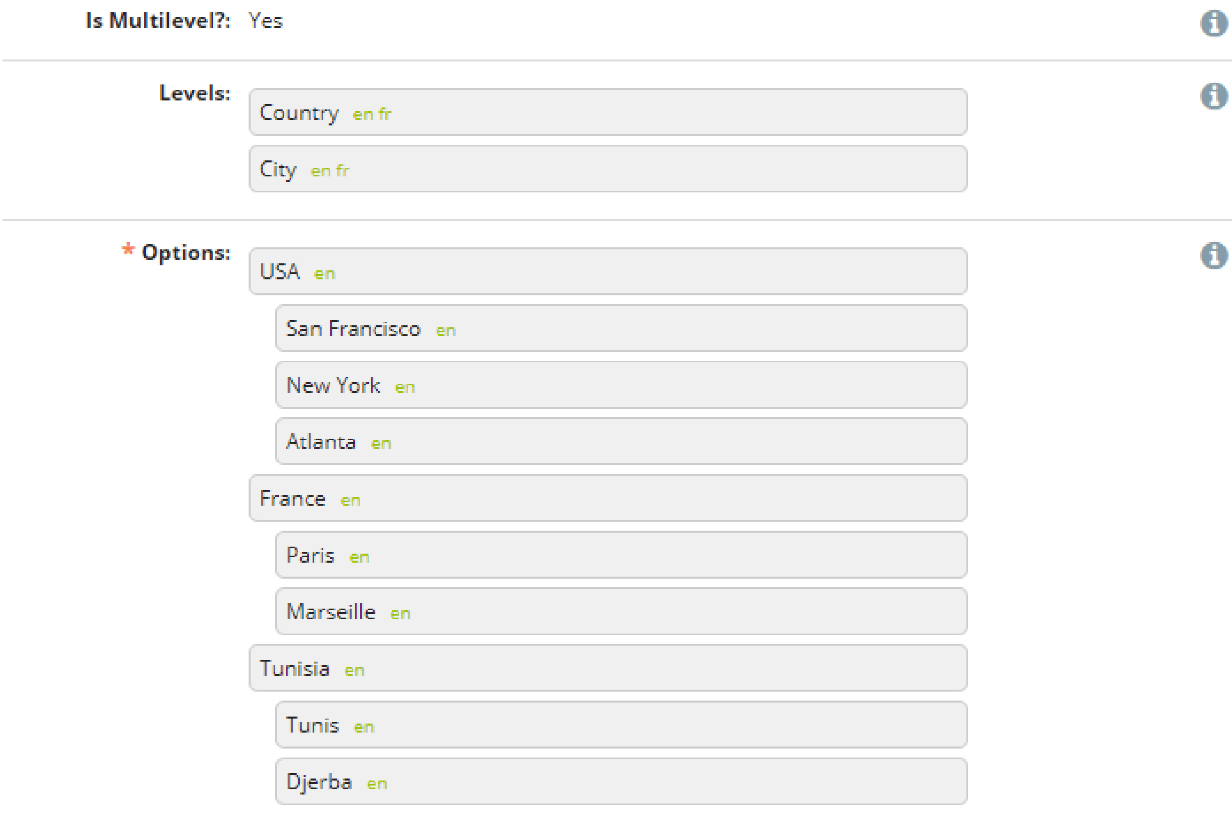
Is Multilevel? Check this box if you have *Multilevel options sets*.

Options Click *Add Option* to add a new option.

For each option added, you can add a value and relevant translations. To access or perform calculations with the option values, please see the sections on *Default answer* and *XPath expressions*.

6.2 Multilevel options sets

Checking *Is Multilevel?* box allows the use of hierarchically organized options. Multilevel option sets can only be used for **Select One** questions. Below is a classic example of countries and cities. Level 1 is **Country** and level 2 is **City**. The level 2 options needs to be indented compared to level 1 options. Drag and drop options to achieve the desired result.



6.3 Edit existing option set

1. Click *Option Sets* menu.
2. Select the Option Set to edit by clicking .
3. Click *Save*.

6.4 Import standard option set

1. Click *Import Standard Option Sets*.
2. Check the boxes next to the option sets to be imported.
3. Click *Import*.

6.5 Import new option set

Option Sets can be created with spreadsheet software like Excel, and uploaded directly into NEMO. You can import single level option sets with translations and multi-level option sets. Please note that multi-level option sets with translations are not supported.

Option sets with translations

1. Outside of NEMO, create a CSV or XLS file.
2. In the first row, for each language you would like, put a language code in brackets for each column header and an optional *Value* column header.

Example: occupation[en]|occupation[fr]|occupation[ht]|Value

3. Add the translations of the options.

	A	B	C	D
1	occupation[en]	occupation[fr]	occupation[ht]	Value
2	student	étudiante	elèv	0
3	doctor	médecin	doktè	1
4	teacher	professeure	pwofesè	2
5				
6				

4. In NEMO, click *Option Set* menu.
5. Click *Import New Option Set*.
6. Choose a name for the option set.
7. Choose the CSV file created in steps 1-3.
8. Click *Import*.

Note:

- If you are importing values, the Value column header must be capitalized.
- Note that the name (in the example, occupation) next to the language in brackets is optional.

Multi-level option sets

1. Outside of NEMO, create a CSV or XLS file.
2. In the first row, include the name for each level as an individual column header (example Company | Department | Name).

3. Add the names of the options.

	A	B	C
1	Company	Department	Name
2	Enterprise	A	Bob
3	Enterprise	A	Bill
4	Enterprise	B	Ted
5	Enterprise	B	Jane
6	Enterprise	C	Joe
7	Enterprise	C	Carl
8	Investco	A	Jeff
9	Investco	A	Sam
10	Investco	B	Emily
11	Investco	B	Nancy
12	Investco	C	Don
13	Investco	C	Rick

4. In NEMO, click *Option Set* menu.
5. Click *Import New Option Set*.
6. Choose a name for the option set.
7. Choose the CSV file created in steps 1-3.
8. Click *Import*.

Note: For large files, import can take some time. To see the status of the import go to **operations panel** by clicking on the link in the blue notice or going to https://yournemoinstance/en/operations .

6.6 Import option set with coordinates

To upload an option set with coordinates:

1. Outside of NEMO/ELMO, create a CSV file (you can export from XLS to CSV from any standard spreadsheet software).
2. In the first row, include the name for each level as an individual column header with **Coordinates** as the last column (example: Province | City | District | Coordinates).
3. Add the names of the options.
4. In the column for **Coordinates**, include both latitude and longitude in decimal format separated by a comma (example 0.054396, 18.259688).

D2 | f_x 0.054396, 18.259688

	A	B	C	D
1	**Province**	**City/District**	**Commune/Territory**	**Coordinates**
2	Équateur	Mbandaka	Mbandaka	0.054396, 18.259688
3	Équateur	Mbandaka	Wangata	0.030470, 18.232284
4	Équateur	Équateur	Basankusu	1.222329, 19.802599
5	Équateur	Équateur	Bikoro	-0.731622, 18.140574
6	Équateur	Équateur	Bolomba	0.351764, 19.226074
7	Équateur	Équateur	Bomongo	
8	Équateur	Équateur	Ingende	-0.310105, 18.938541
9	Équateur	Équateur	Lukolela	-1.060581, 17.175879
10	Équateur	Équateur	Makanza	1.613769, 19.107542
11	Mongala	Mongala	Bumba	2.184720, 22.470860
12	Mongala	Mongala	Bongandanga	1.507453, 21.073709
13	Mongala	Mongala	Lisala	2.148696, 21.513034
14	Nord-Ubangi	Gbadolite	Gbadolite	4.283803, 21.014185
15	Nord-Ubangi	Gbadolite	Molegbe	4.232063, 20.904815
16	Nord-Ubangi	Gbadolite	Nganza	

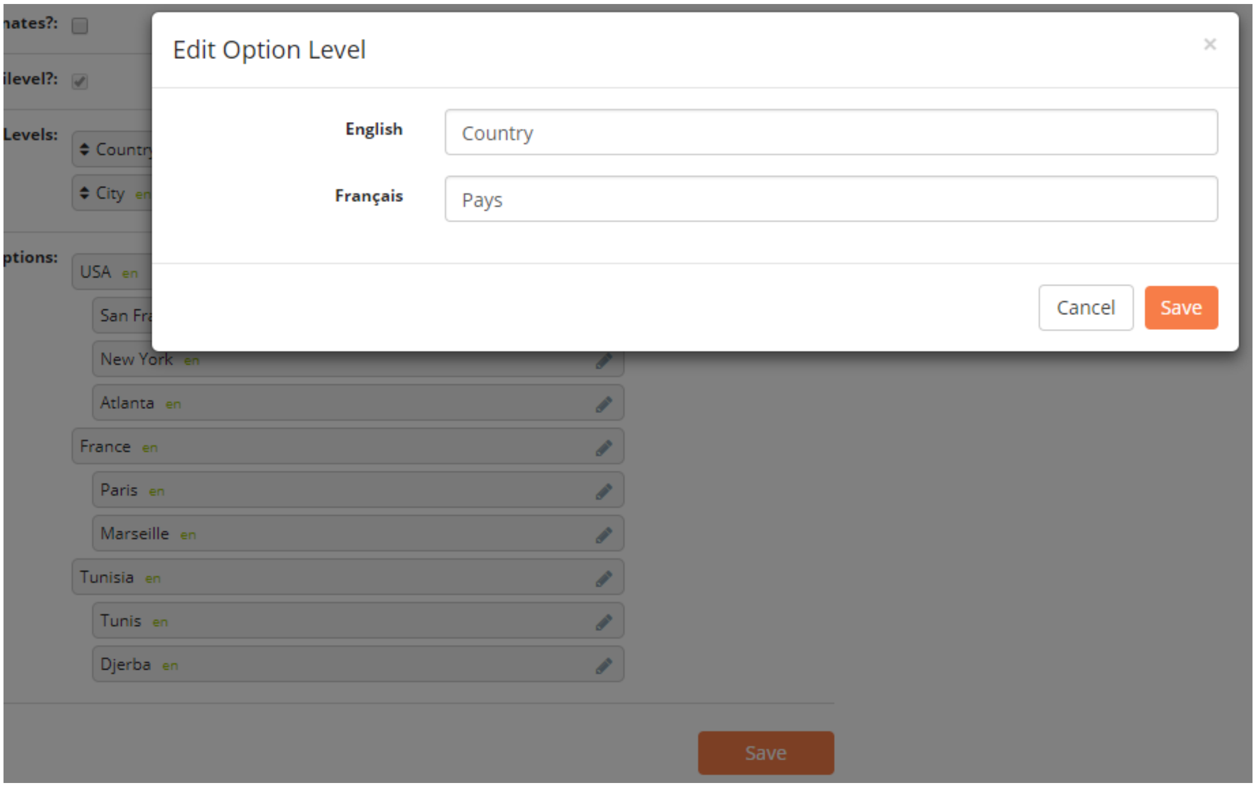
5. In the mission you are working on, click on the *Option Set* menu.
6. Click *Import New Option Set*.
7. Choose a Name for your option set.
8. Choose the CSV file you created.
9. Click *Import*.

6.7 Language translations

Options within an Option Set can be translated in a manner similar to translating questions. To translate an option:

1. Create or edit an existing option set.
2. Click next to the option to be translated.
3. Type the translation.
4. Click *Save*.

Note: The two-letter language code for every translation appears next to the option name.


Edit Option Level
English
Country
Français
Pays
Cancel
Save
Levels:
Country
City
USA
San Fr
New York
Atlanta
France
Paris
Marseille
Tunisia
Tunis
Djerba
Save

CHAPTER SEVEN

FORM STYLING

Titles, hints, and option sets can all be styled using Markdown, fonts and colors.

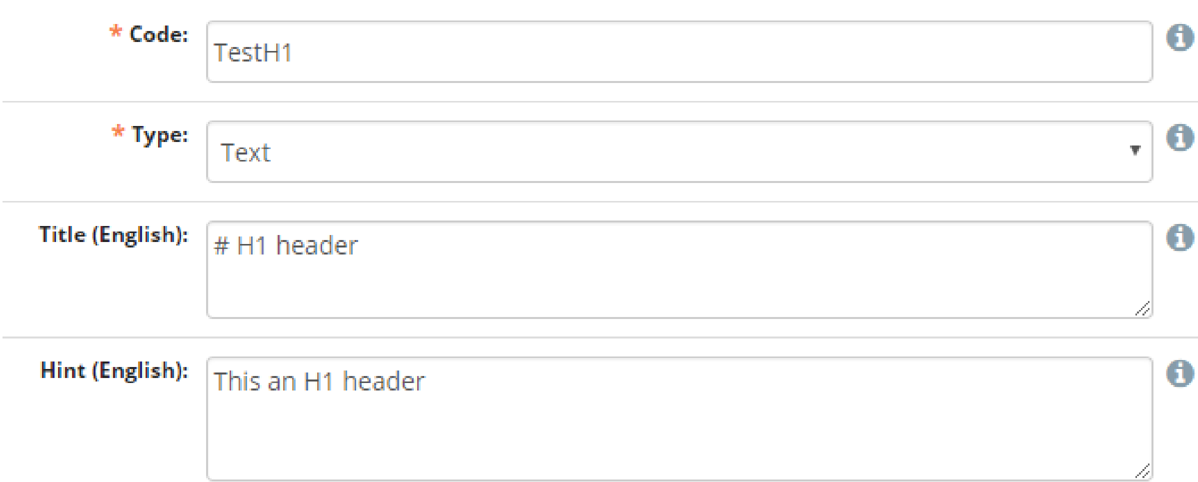
7.1 Markdown

7.1.1 Headers

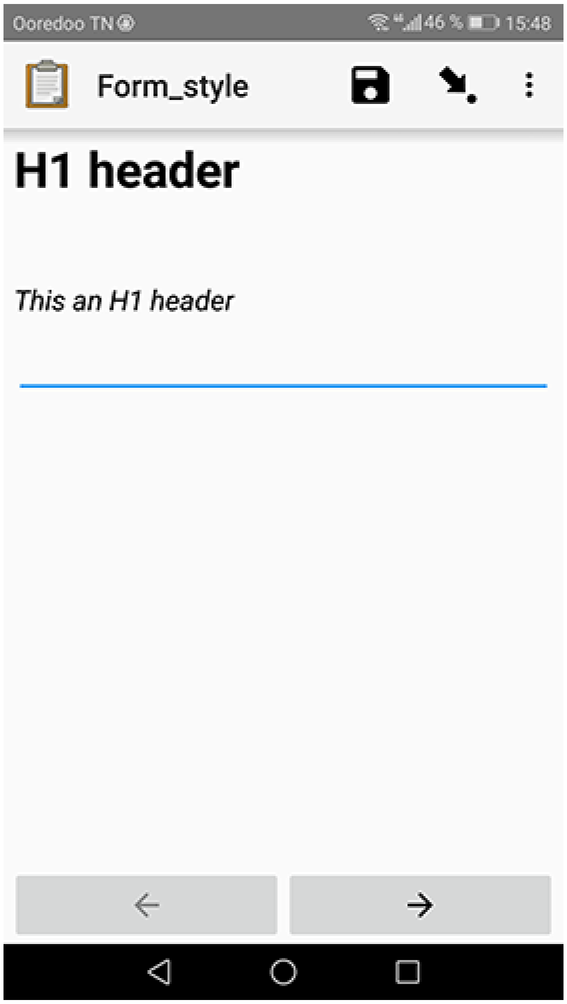
Titles and hints can be styled with one of six header levels.

```
# Header H1

## Header H2

### Header H3

#### Header H4

##### Header H5

###### Header H6
```

This is an example of how an H1 Header is set in NEMO:



And the result in ODK Collect:



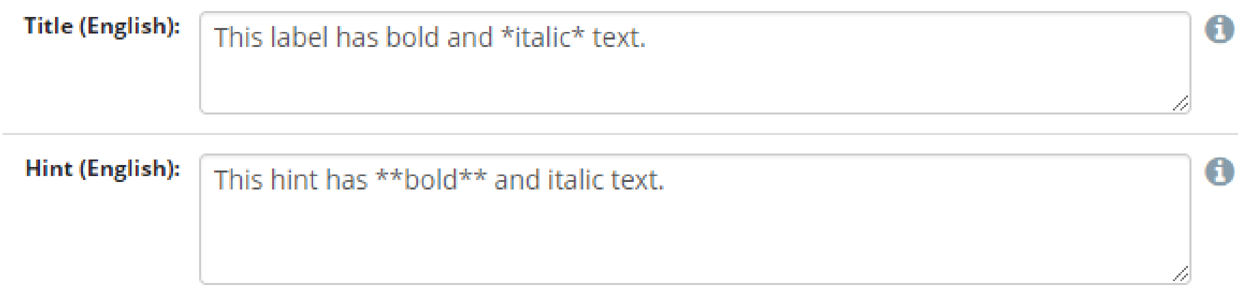
7.1.2 Emphasis

ODK Collect's Markdown support also includes **bold** and *italic* styling.

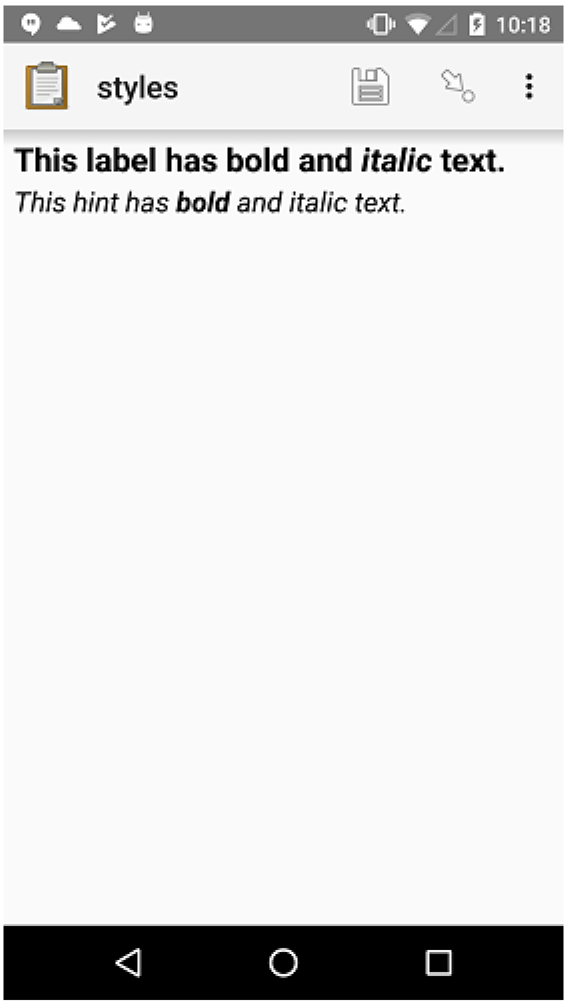
```
_italic_

*italic*

__bold__

**bold**
```

Note: The label of a form widget is already bold, so bolding text within the label has no effect. Similarly, the hint text of a form widget is already in italics, so italicizing text within the hint has no effect.

Example:



And the result in ODK Collect:

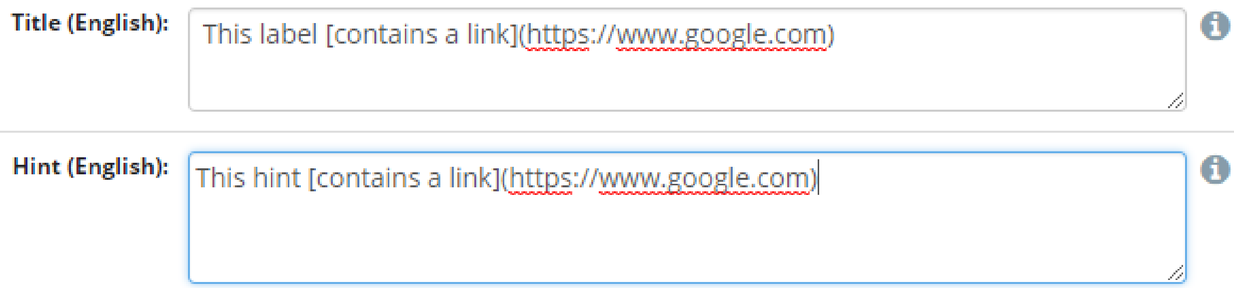


7.1.3 Hyperlinks

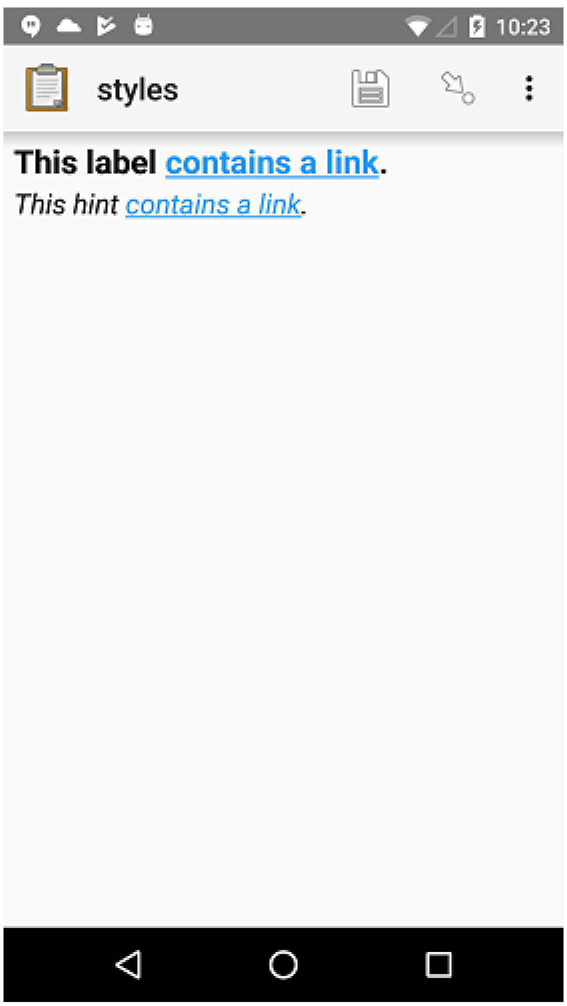
hyperlinks are also supported. When clicked they will open in the device's default browser. Below is the syntax:

```
[Link anchor text](link.url)
```

Example:



And the result in ODK Collect:



7.2 Fonts and colors

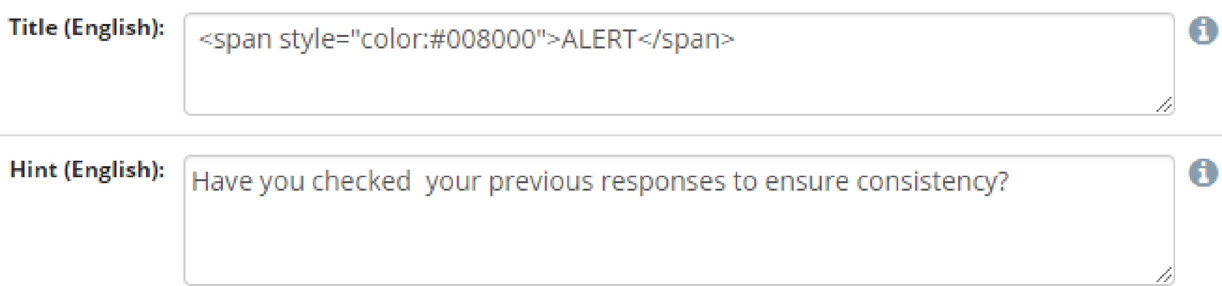
Use the style attribute on a <span> tag to add custom `color` and `font-family`.

- For `color`, try one of the named HTML color values or use a hex color.
- For `font-family`, it is best to use generic font categories rather than specific fonts:
  - serif
  - sans-serif
  - monospace
  - cursive
  - fantasy

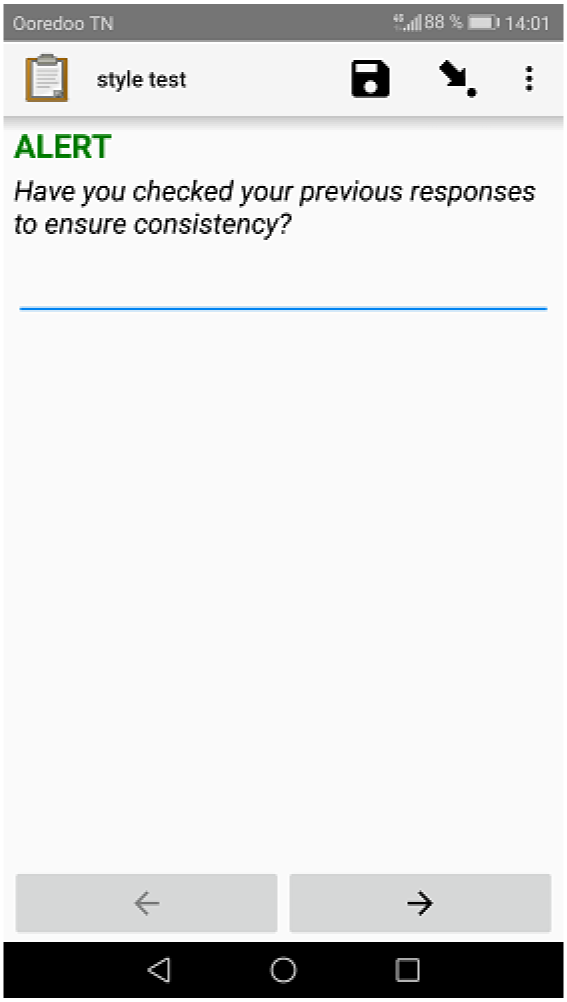
  This will ensure support across most devices. You can also use specific font choices, but you should test these on the actual devices being used.

Note: These two attributes, `color` and `font-family`, are the only style attributes supported in Collect.

Example:

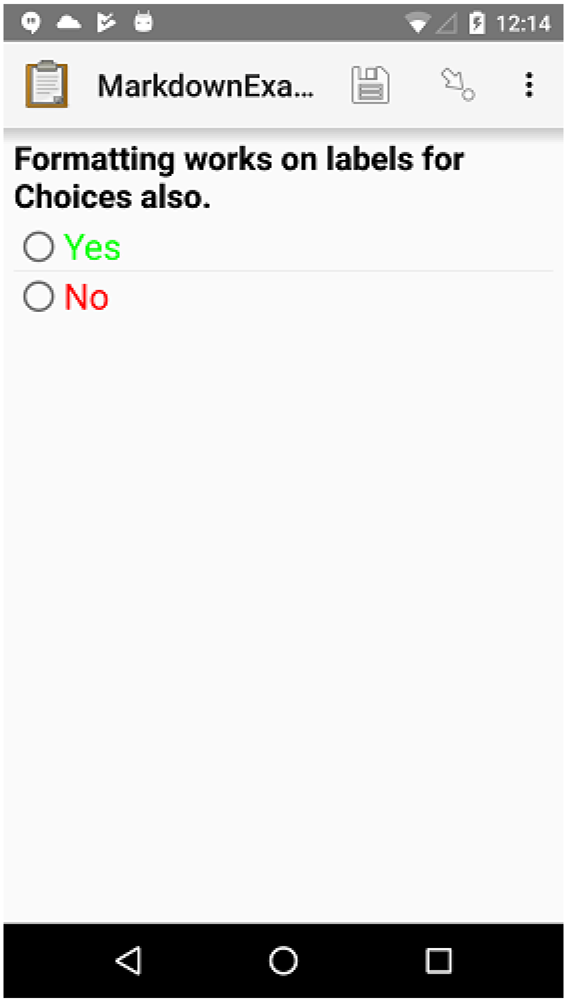


And the result in ODK Collect:



Font family example:

Option set example:



Note: This document is a derivative of the original Form Styling licensed under a Creative Commons Attribution 4.0 International License.

CHAPTER

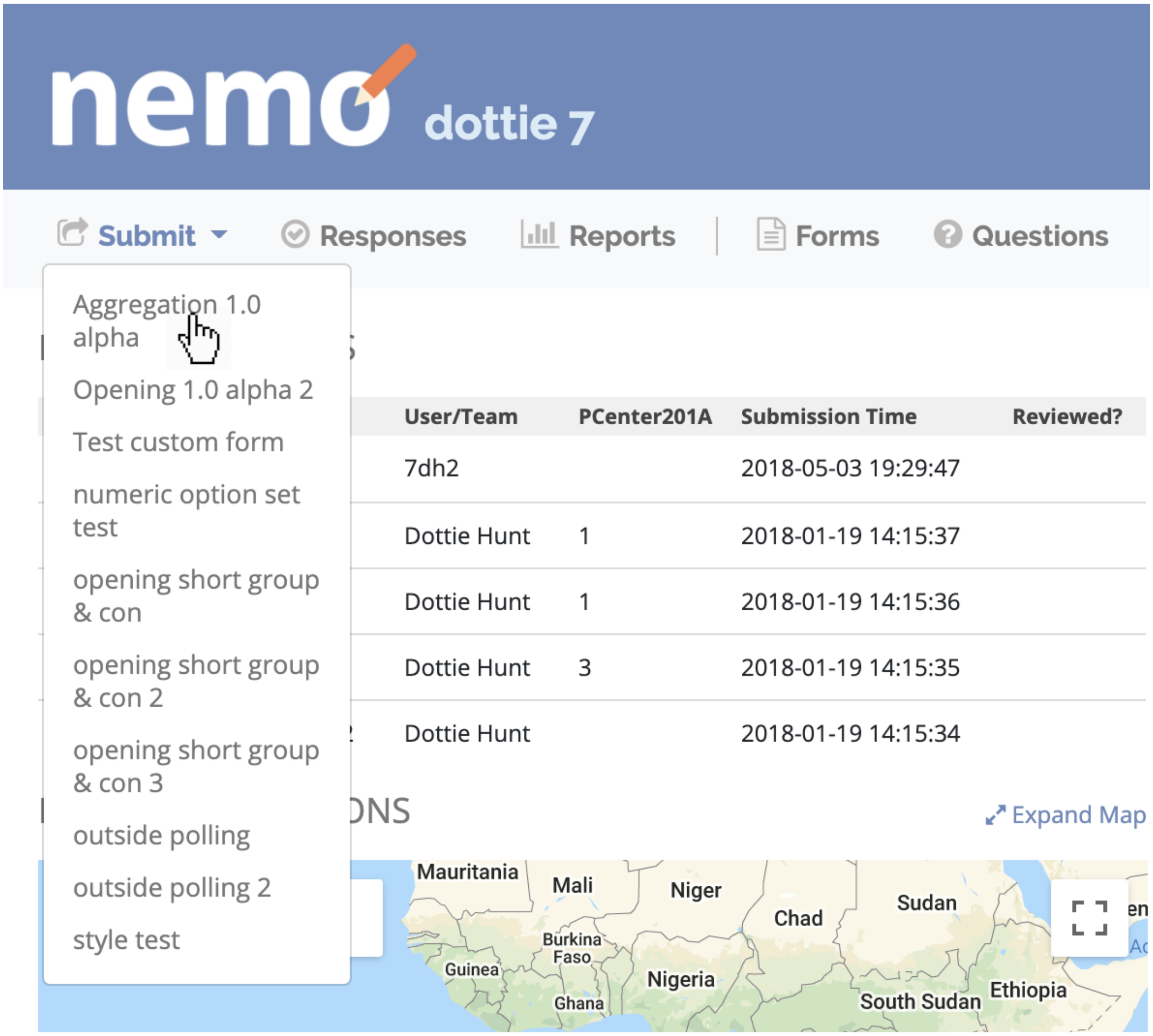
EIGHT

WEB

8.1 Submit form

To submit a form online:

1. Click *Submit* in the menu bar.
2. Select a form from the drop down list.
3. Fill the form using either the NEMO or Enketo editor.
4. Click *Save*.


nemo
dottie 7
Submit
Responses
Reports
Forms
Questions
Aggregation 1.0 alpha
Opening 1.0 alpha 2
Test custom form
numeric option set test
opening short group & con
opening short group & con 2
opening short group & con 3
outside polling
outside polling 2
style test
User/Team
PCenter201A
Submission Time
Reviewed?
7dh2
2018-05-03 19:29:47
Dottie Hunt
1
2018-01-19 14:15:37
Dottie Hunt
1
2018-01-19 14:15:36
Dottie Hunt
3
2018-01-19 14:15:35
Dottie Hunt
2018-01-19 14:15:34
Expand Map
Mauritania
Mali
Niger
Chad
Sudan
Burkina Faso
Guinea
Ghana
Nigeria
South Sudan
Ethiopia

CHAPTER

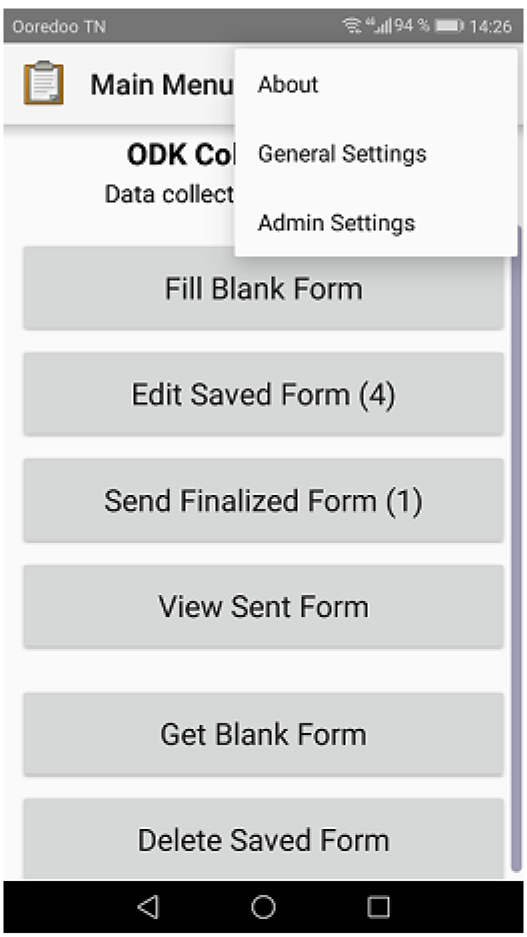
NINE

ODK COLLECT

NEMO forms can be submitted via Android devices with **ODK Collect**. The ODK Collect App is available in the Google Play store.

9.1 Setup ODK Collect

1. Open *ODK Collect* app from your Android device.
2. From the action button (), select *General Settings*.



3. Select *Server*.

4. Make sure *Type* is set to *ODK Aggregate*.
5. Type the *URL* of the mission, example https://example.getnemo.org/en/m/missionname.
6. Type your NEMO *Username* and *Password*.

9.2 Download forms

Forms must be downloaded to ODK Collect before they can be submitted. To download a form:

1. On the ODK Collect home screen, press *Get Blank Form*.

Note: A box may pop up asking for confirmation of username and password. If already setup in General Settings, press *OK*.

2. Check the box next to the forms you want to download.
3. Press *Get Selected*.

Note: When forms are updated in NEMO they need to be downloaded again to ODK Collect.

9.3 Submit forms

1. On the ODK Collect home screen, press *Fill Blank Form*.
2. Select a form from the list.
3. Fill the form.
4. On the last screen press *Save Form and Exit*.
5. On the ODK Collect home screen, press *Send Finalized Form*.
6. Check the box next to the forms you want to submit.
7. Press *Send Selected*.

Note:

- When obtaining GPS locations, stand outdoors. If indoors, stand by the nearest window.
- On the last screen, if you want to be able to edit the form again before submission, you need to uncheck the box *Mark form as finalized* before pressing *Save Form and Exit*.

9.4 Edit forms before submission

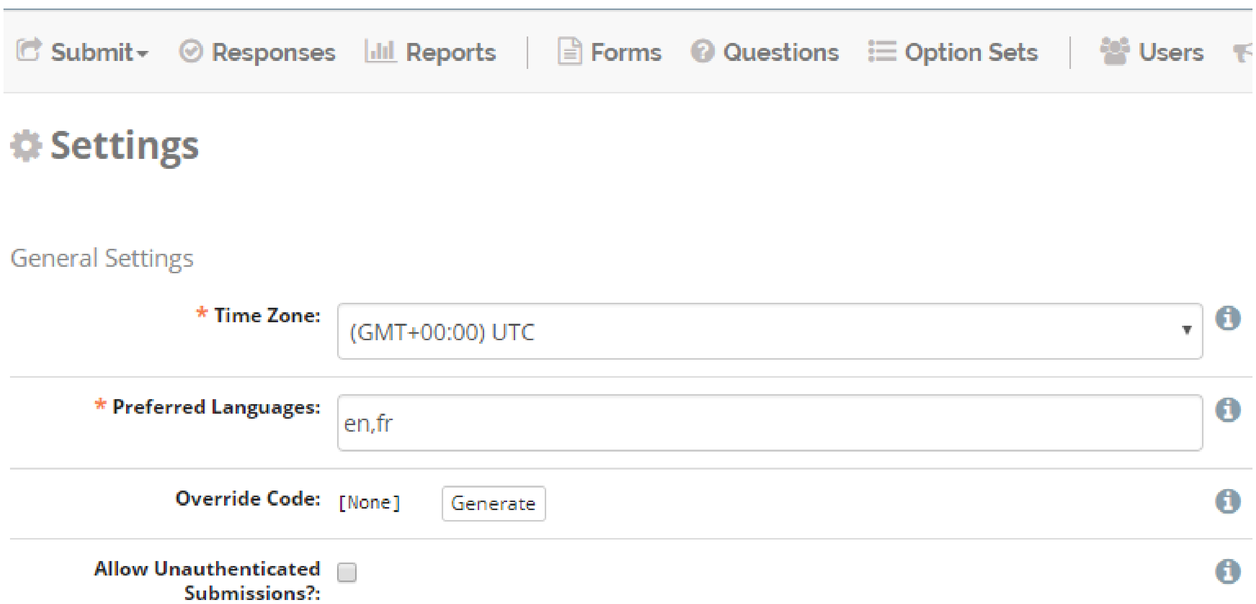
1. On ODK Collect home screen, press *Edit Saved Form*.
2. Select the form to be edited.
3. Select a question from the list to change its answer, or press *Go To Start* to review each question from the beginning.
4. When finished editing, check the box *Mark form as finalized* before pressing *Save Form and Exit*.

9.5 Override code

If you need to finalize and send forms having required questions not answered, you have to use an **override code**. This code is found on the settings page of each mission.

To generate an Override code:

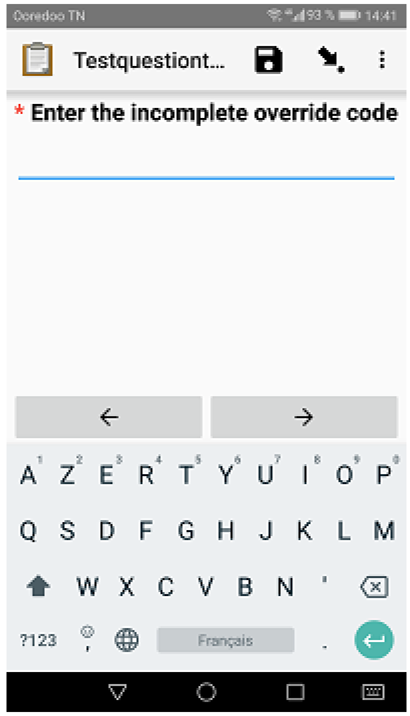
1. Click *Settings* menu.
2. If the *Override Code* is *None*, click *Generate* to generate a new code.



Note: In order to use the override code, forms must be set to allow for incomplete responses. When creating or editing a form, check the box **Allow Incomplete?**. If not initially set, forms have to be downloaded again to ODK Collect in order to take effect.

When submitting a form that allows incomplete responses a question will show up at the end saying **Is this form missing any required answers?**

- *I don't think so* will direct you to the next screen to save and exit the form.
- *Yes* you will be asked to enter the override code.



9.6 Import/Export settings

It is possible to share ODK Collect settings between multiple devices using QR code configuration. On your reference device:

1. Press the action button ().
2. Select *Admin Settings*.
3. Select *Import/Export settings*.

9.6.1 Scan code from other device

From the device that need to be configured press *Scan code from other device*. This will show a QR code scanner that you can use to scan the code from the reference device. Once the code is successfully scanned, Collect will return to the landing screen with a message saying settings were successfully loaded.

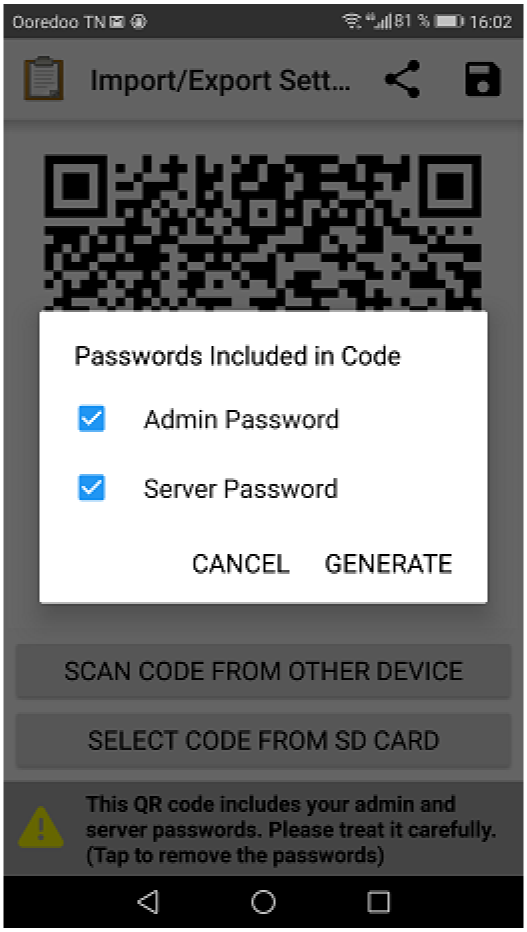
9.6.2 Share QR Code

You can share your settings QR code as an image, for this:

1. Tap on the top right of the screen.
2. Select an application from the list.

Note: The QR code contains all of the settings including your passwords. If you want to exclude these:

1. Tap the bottom of the screen.
2. Uncheck passwords.



9.6.3 Select code from SD card

If you already have the QR code settings as an image:

1. Tap *Select code from SD card*.
2. Select your QR code image.

Note: For more details about Import/Export settings see ODK Collect documentation.

CHAPTER TEN

SMS

Forms can be submitted via SMS. Any mobile phone with SMS capability and access to a cellular network can be used to submit data to NEMO.

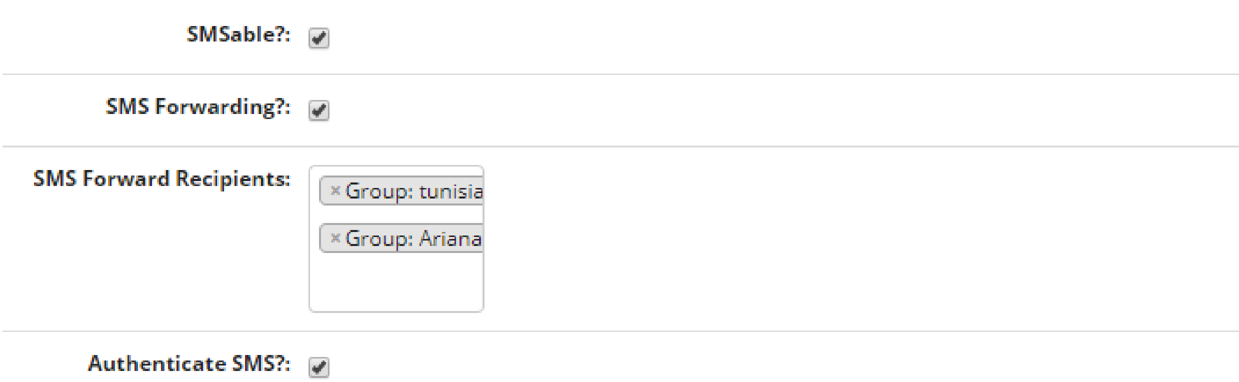
Given the space limitations of texts, SMS-based forms are most powerful when designed carefully. Certain types of questionnaires, such as qualitative ones with long-text answers, are usually not a good fit. However, SMS form submission allows the easy aggregations of large amounts of data across distances without good mobile or internet coverage. It also can be cheaper per user, and may suit certain monitoring and evaluation purposes, quantitative inquiries, or simple surveys.

10.1 SMSable form

By default, SMS form submission is not activated, to do so you need to:

1. Click *Forms* menu.
2. Create or edit an existing form.
3. Click *More settings*.

4. Check the box *SMSable?*.



When checking *SMSable?*, two new options are available:

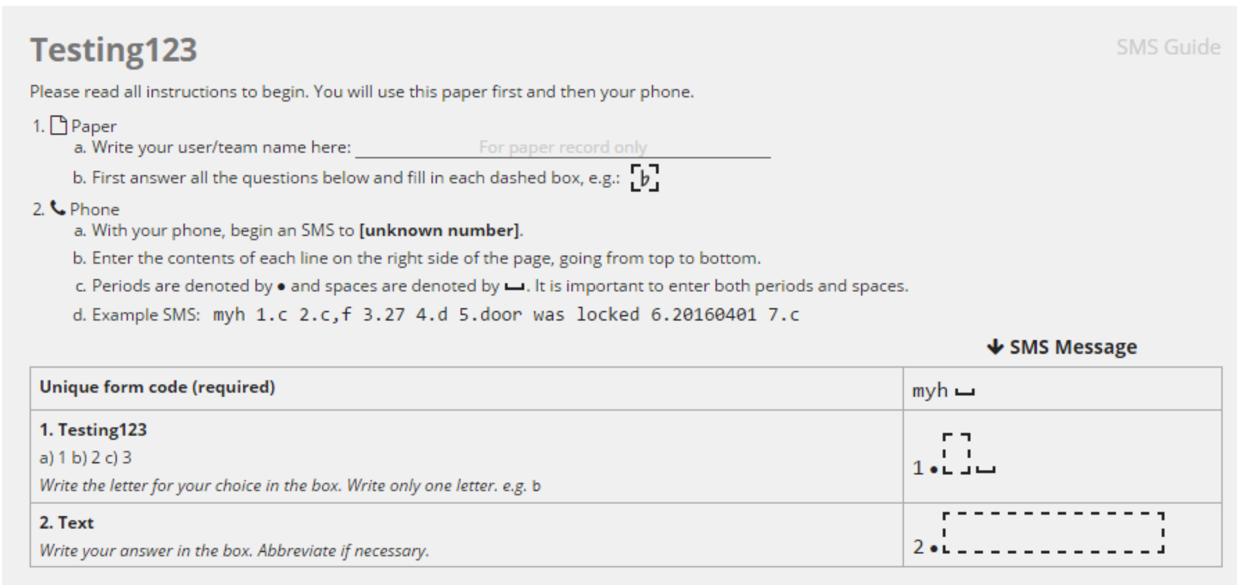
- *SMS Forwarding?* check this box if you want to forward all incoming SMS to a user or a group of users.
- *Authenticate SMS?* if checked, incoming SMS needs to contain users authentication code. More information can be found *below*.

10.2 SMS submission guide

An SMS submission guide is generated for all **SMSable** forms, to access the instructions:

1. Click *Forms* menu.
2. Select the form from the list.
3. Click on *View SMS Guide*.

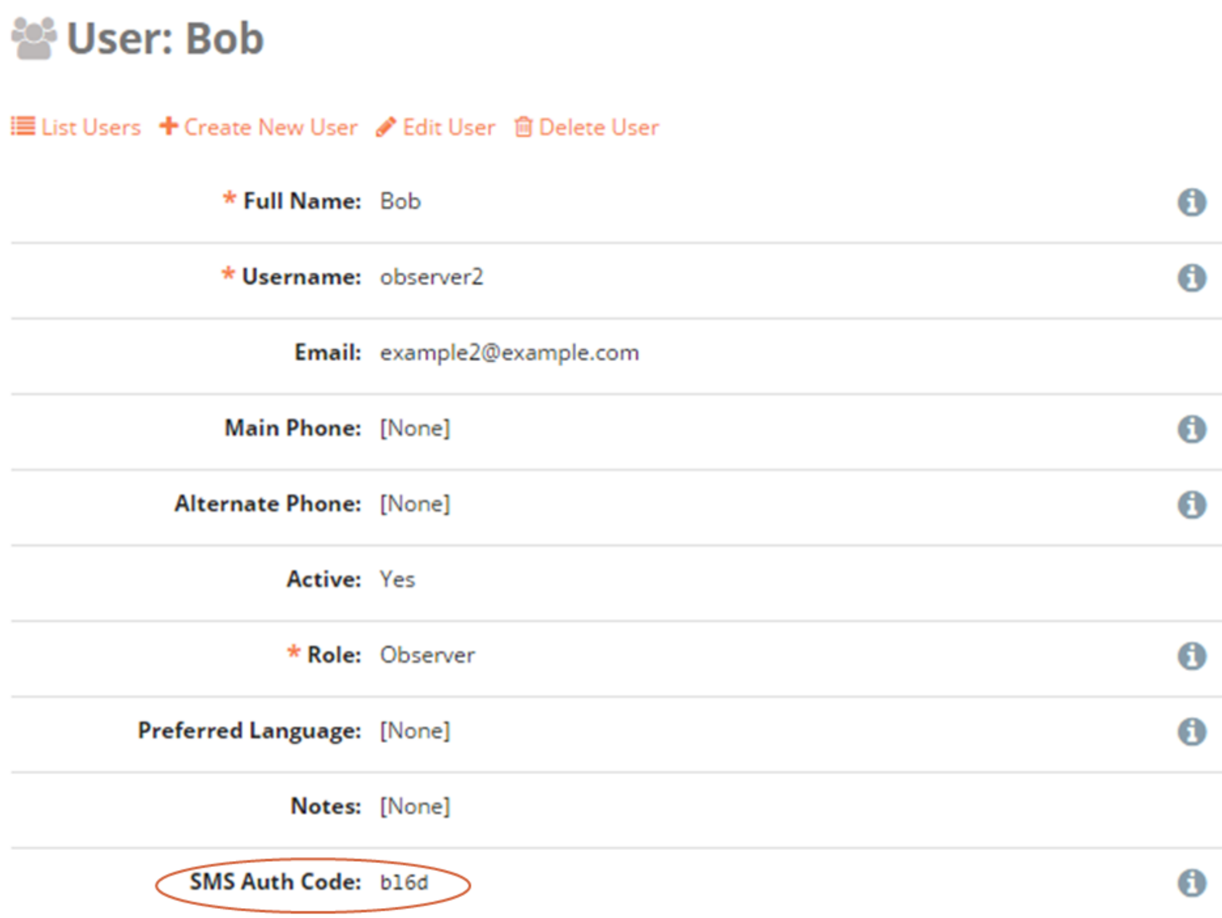
An example of the guide is shown below:



10.3 SMS authentication

NEMO only accepts form submissions from users having phone numbers pre-registered in their profiles. SMS Authentication feature, when enabled, works by requiring users to provide an additional code when submitting forms via SMS.

This four-character code is randomly assigned and unique to each user. It is generated when a user is created. The code can be found by clicking on the *Users* menu and then on the name of the user.



To generate a new authentication code:

1. Click *Edit User*.
2. Click *Regenerate*.
3. Click *Save*.

Activating SMS authentication adds extra security to NEMO by ensuring that the registered phone number and the authentication code belong to the same user. All SMS submissions that do not meet both these conditions are rejected.

10.4 Submitting SMS responses

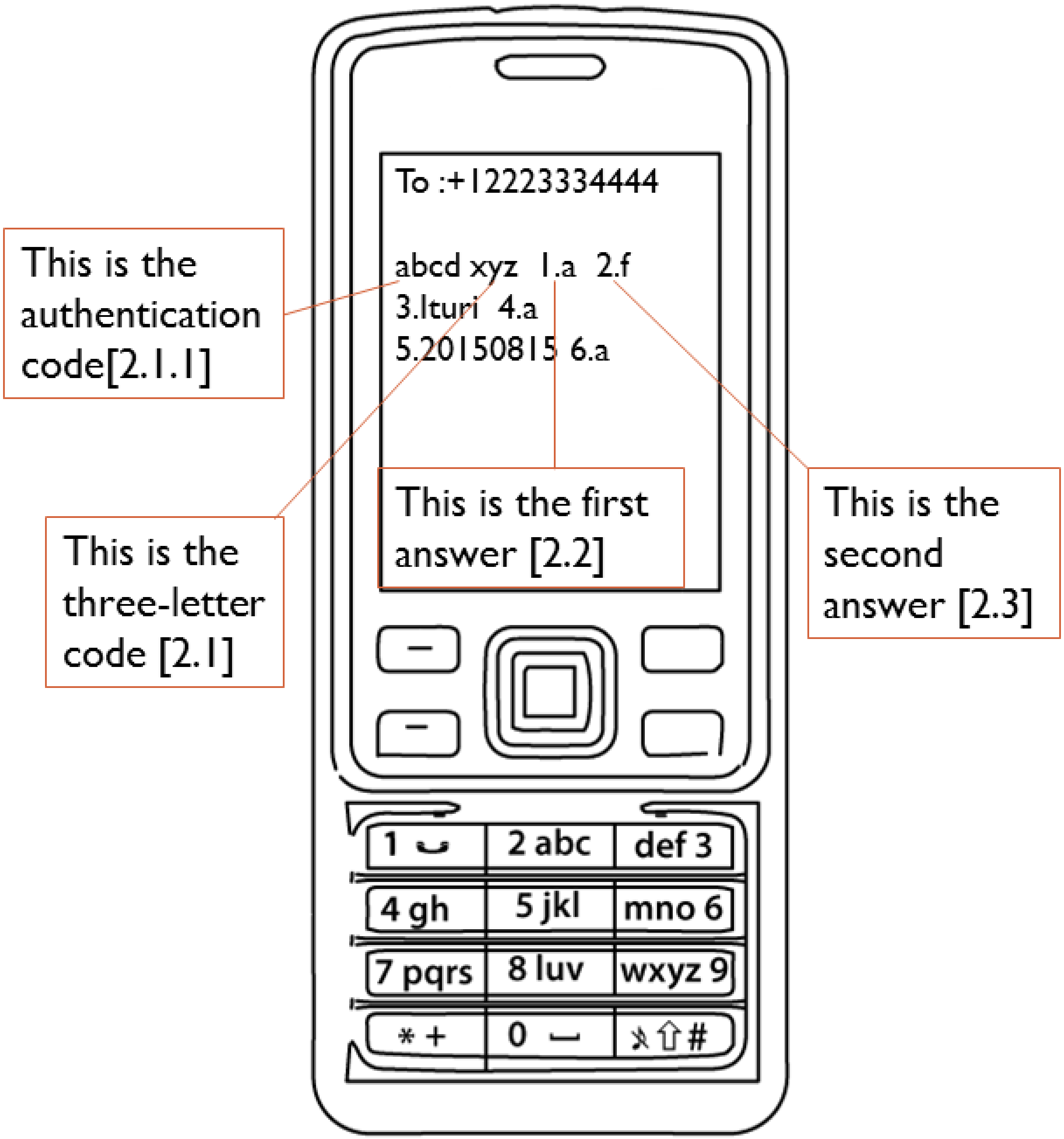
To submit forms to NEMO via SMS, the following is required:

- Mobile phone.
- Phone credit.
- *SMS-able NEMO form* with three letter unique code listed in the SMS Composition Guide.
- The NEMO number to which the SMS will be submitted.

To submit a form:

1. Type the NEMO phone number.
2. If the user authentication code is required, type the code followed by a space.

3. Type the form code followed by a space.
4. For each question, type the number of the question followed by the answer selected followed by a space.
5. **Send** the message.



Note: If not answering a question, skip that question number in the SMS. For example, if question 4 is not being answered, the message would look like *xyz 1.a 2.f 3.Ituri 5.20150815 6.a*.

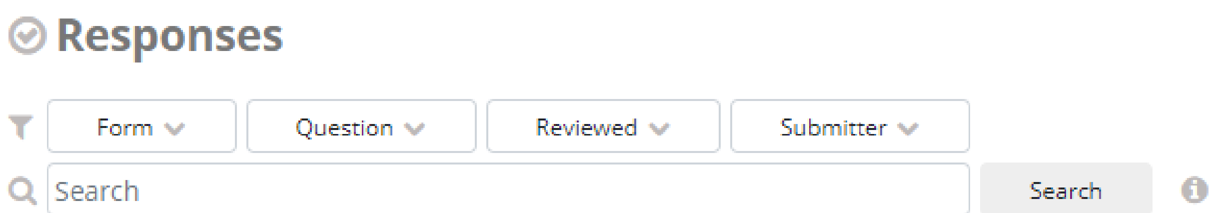
CHAPTER

ELEVEN

SEARCH

11.1 Responses filter component

Responses menu have a specific filter component with multiple dropdowns where you can filter by Form name, Question, Reviewed and Submitter.



- *Form* Select the name of the form.
- *Question* Filter by a specific value of a question, click *Add Condition* to add another question filter.
- *Reviewed* Select whether to show only reviewed responses, unreviwed responses, or both.
- *Submitter* Filter by a specic user or group.

11.2 Free search box

The free search box is available on responses, questions, users, and SMS menu. Below is the search box from users menu.

11.3 Operators

You can combine search terms with operators for a more accurate search.

Operators in NEMO are: **AND, OR, NOT**(! or -), **grouping** operator (parentheses), and **phrase** operator (“”).

Operator	Description
AND	By default, all terms in the search must be matched. For example, *apples oranges bananas* is equivalent to *apples AND oranges AND bananas*.
OR (|)	Matches when any of the terms match. Example *Household Form* | *Fruit Form* returns matches that are *Household Form* or *Fruit Form*.
NOT (!= or -)	Matches when the first term matches, but the second one does not. Example: *form != Household* returns the forms that are NOT Household. Another example: *apple -oranges* matches any response with an answer containing the word apples but NOT the word oranges.
(. . .)	Grouping parenthesis denotes the search terms boundaries. Example: *(red | green | blue) pixel* is equivalent to *(red OR Green OR blue) AND pixel*.
“. . . ”	Quotes match when search terms match an exact phrase. Example *submitter:”John Smith”*.

11.4 Qualifiers

A qualifier is a word you add to an expression to specify where to search. Example:

- **form: apples** within the responses menu will return all forms with the word **apples** in them.
- **type: long text** in the questions menu returns all questions of the long text type.

Available qualifiers depends on the menu you are in. They are listed below:

Responses menu

Qualifier	Function
form:	The name of the form submitted.
submitter:	The name of the user that submitted the response (partial matches allowed).
submit-date:	The date the response was submitted (example: **submit-date: 1985-03-22**).
reviewed:	Whether the response has been marked as **reviewed** (1 = yes or 0 = no).
source:	The medium via which the response was submitted (Web, ODK Collect, or SMS).
text:	Answers to textual questions.

Questions menu

Qualifier	Function
code:	The question code (partial matches allowed).
title:	The question title (partial matches allowed).
type:	The question type (text, long-text, integer, decimal, location, select-one, select-multiple, datetime, date, time).
tag:	Tags applied to the question.

Users menu

Qualifier	Function
name:	The user's full name.
login:	The user's username.
email:	The user's email address.
phone:	The user's phone number. No dashes or other punctuation, example: 1112223333.
group:	The user group that the user belongs to.

SMS menu

Qualifier	Function
content:	The message content (partial matches allowed).
type:	The message type: incoming, reply, or broadcast (partial matches allowed).
username:	The username of the sender or receiver (partial matches allowed).
name:	The full name of the sender or receiver (partial matches allowed).
number:	The phone number of the sender or receiver (partial matches allowed).
date:	The date the message was sent or received. Example date:2015-01-29.
datetime:	The date and time the message was sent or received. Use quotation marks and 24-hr time, example "2015-01-29 14:00".

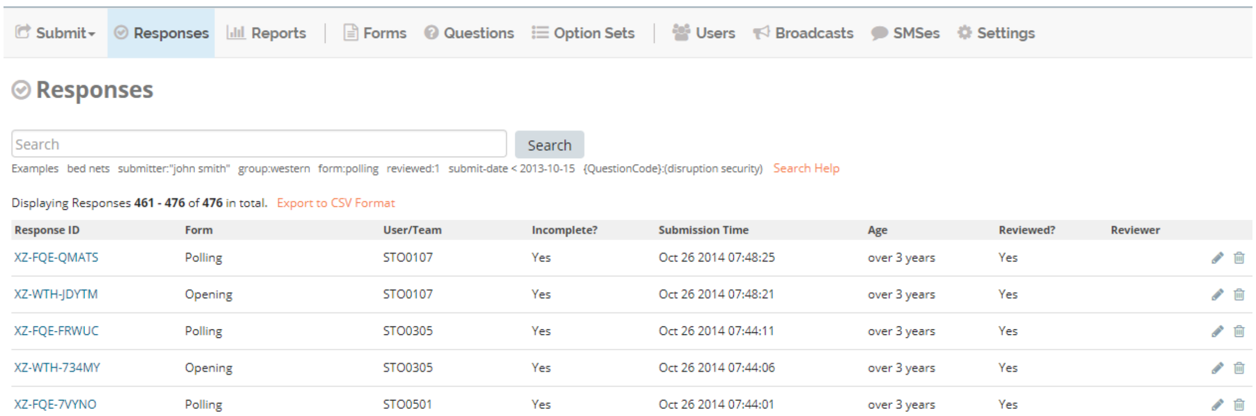
CHAPTER

TWELVE

REVIEW A RESPONSE

12.1 Basic flow

1. Click *Responses* menu.
2. Click on the same row as the response to be reviewed.

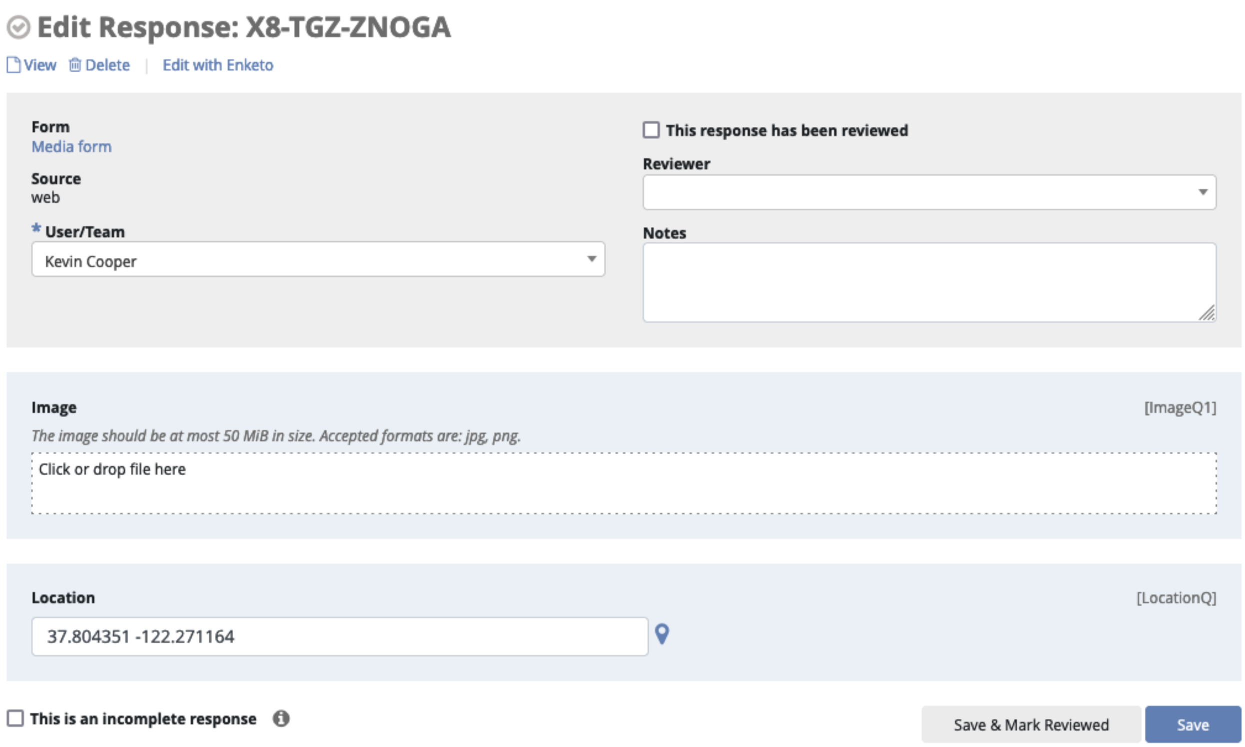


3. Make any necessary changes to the form.
4. Click *Save*.

12.2 NEMO Editor

In addition to editing the response answers, the NEMO editor also allows editing metadata about the submission:

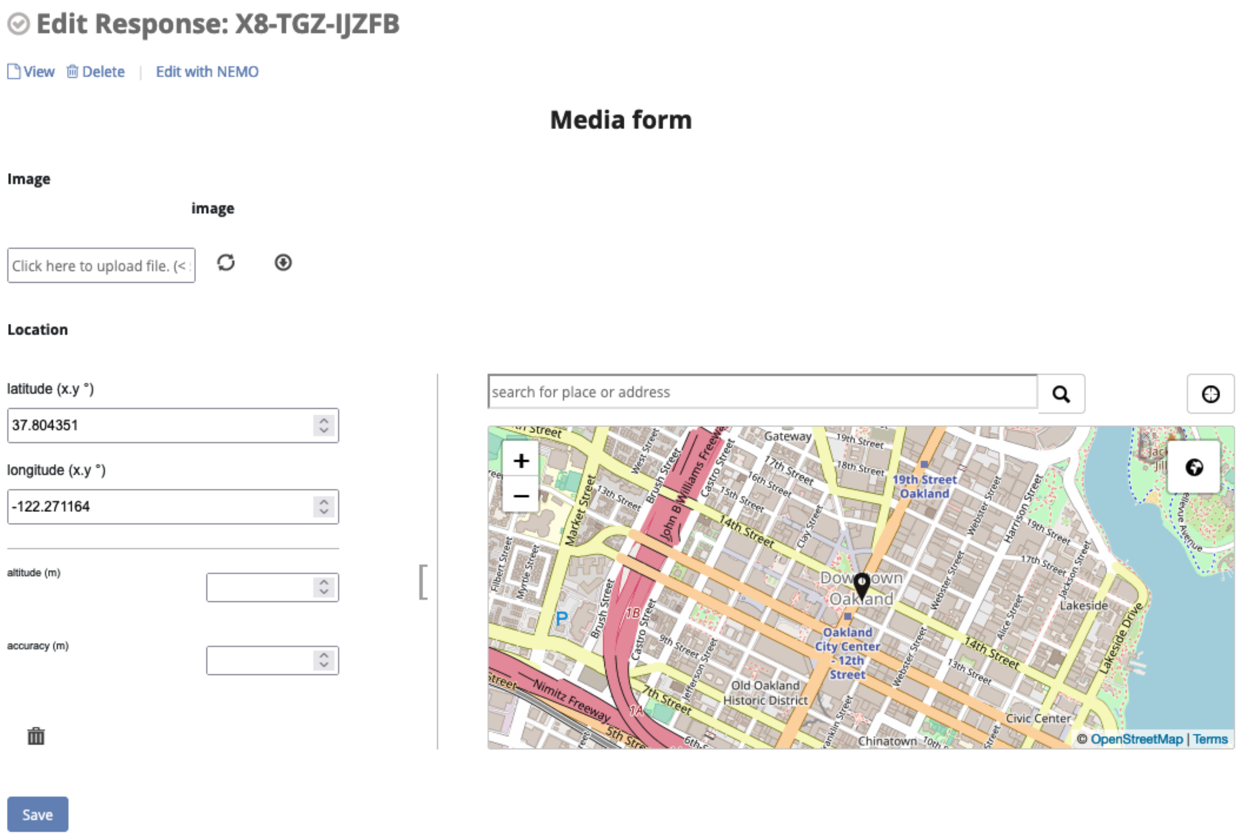
1. Optionaly check the box *Reviewed?*.
2. Optionaly add *Reviewer Notes*.



Note you will not be able to interact with conditional logic, such as display/skip conditions, in the NEMO editor.

12.3 Enketo Editor

The Enketo editor is an open-source tool which has been integrated into the NEMO app. Enketo supports conditional logic, such as display/skip conditions, even when reviewing existing responses.



12.4 Switching Between Editors

At any time, you can click the *View/Edit with Enketo* or *View/Edit with NEMO* link at the top of the page to switch between editors.

Note that Enketo currently only supports responses that have an associated XML file, which means it cannot be used to view responses made through the NEMO web form. Similarly, if you edit a response with the NEMO web editor, those edits will not be visible in Enketo and you will see an earlier version of the form; in this case a warning will appear at the top of the screen as a reminder.

CHAPTER

THIRTEEN

REPORTING AND DATA ANALYSIS

13.1 Internal report building

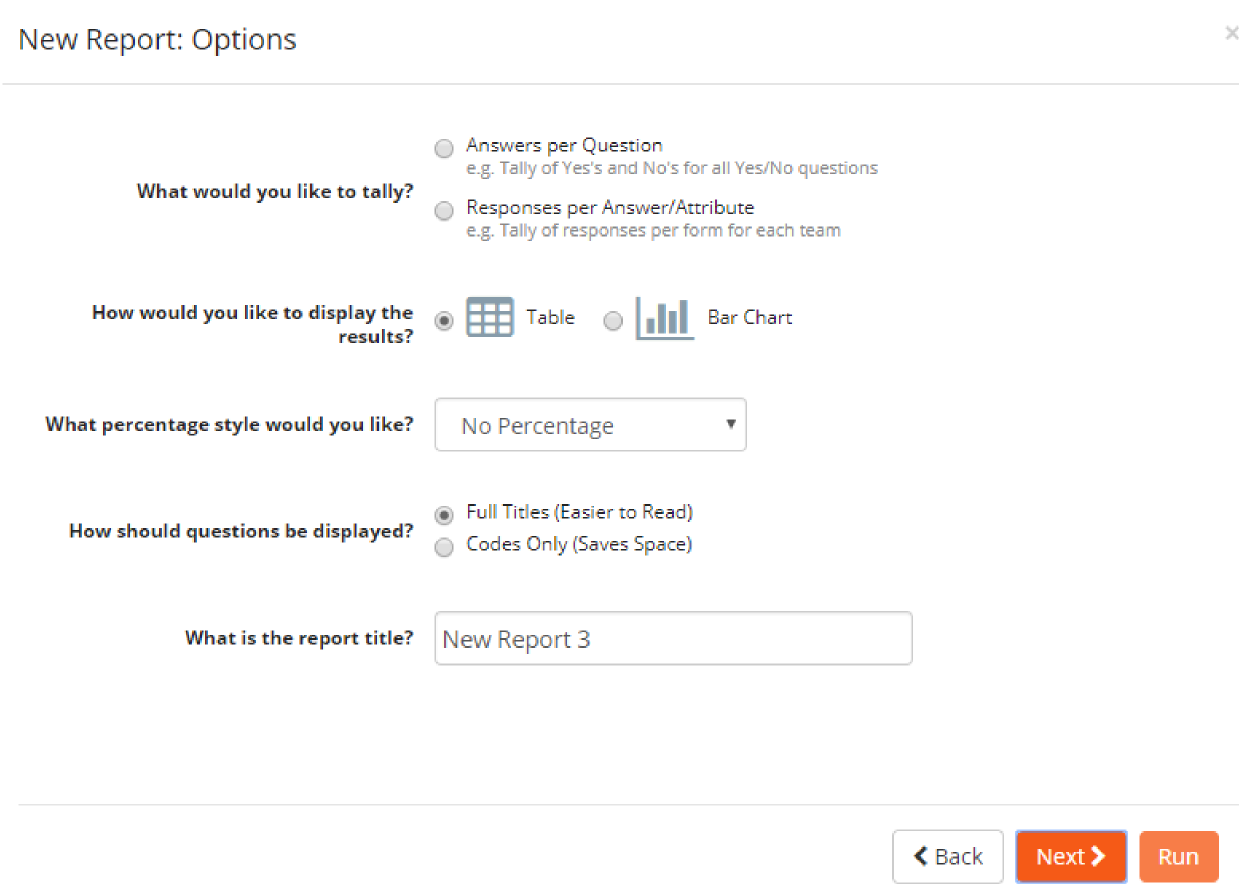
Submitted data can be visualized in the reports section of NEMO. Enumerators are only able to generate reports from their own submitted responses. Reviewers, staffers, coordinators and admins can generate reports from data submitted by all users.

To create a new report:

1. Click *Reports* menu.
2. Click *Create New Report*.

13.1.1 Tally report

Shows Aggregated tallies of answers or responses, grouped by attributes of interest. If selected, the following prompts/options are available:

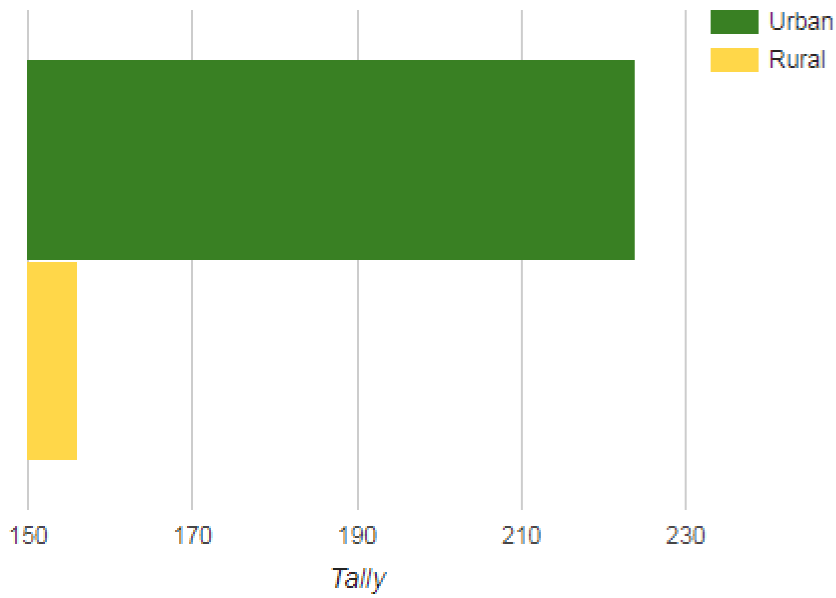


1. Select how you would like to tally.
2. Choose between *Table* or *Bar Chart* results display.
3. Select *How should questions be displayed*.
4. Type a name for the report.
5. Click *Next*.
6. Select the forms you want to include in your report then click *Next*.
7. Select the questions for row headers.
8. If needed, select the questions for column headers.
9. Click *Run*.

Example of answers per question Table display

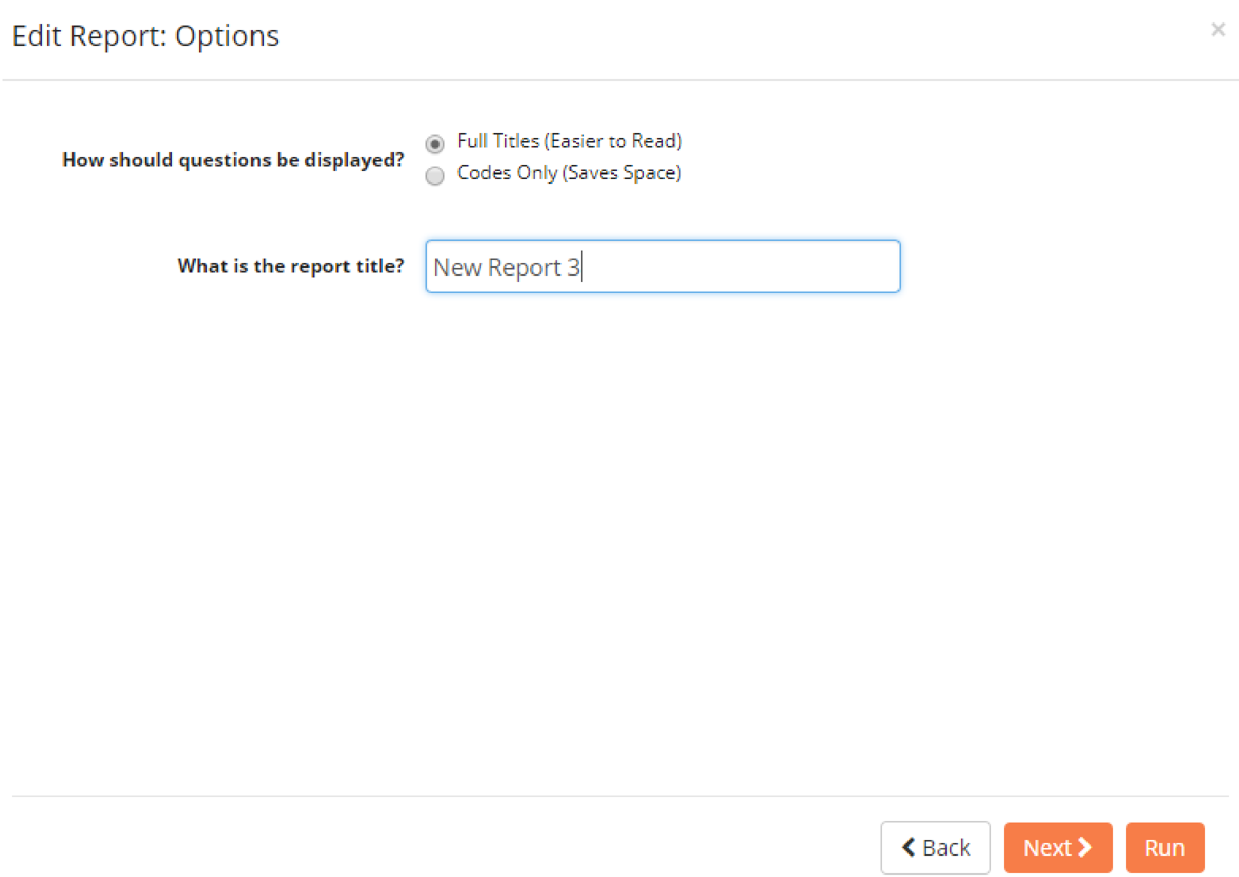
		Answers			
		[Blank]	Observed	Not observed	Total
Questions	Birds	1	4	3	8
	Reptiles	1	4	3	8
	Mammals	1	4	3	8
	Insects	2	4	2	8
	Total	5	16	11	32

Example of responses per answer/attribute Bar chart



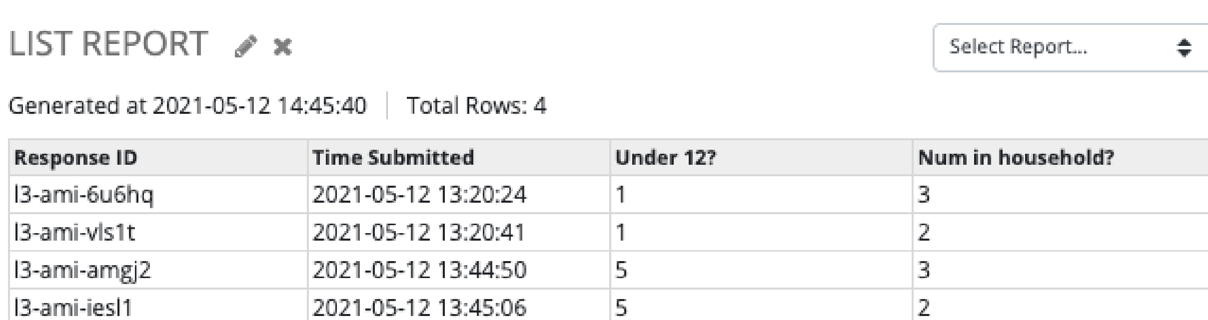
13.1.2 List report

A raw listing of answers and attributes for a set of responses. If selected, the following prompts/options are available:



1. Select *How should questions be displayed*.
2. Type a name for the report.
3. Click *Next*.
4. Select the forms you want to include in your report then click *Next*.
5. Select the columns you want to include then click *Run*.

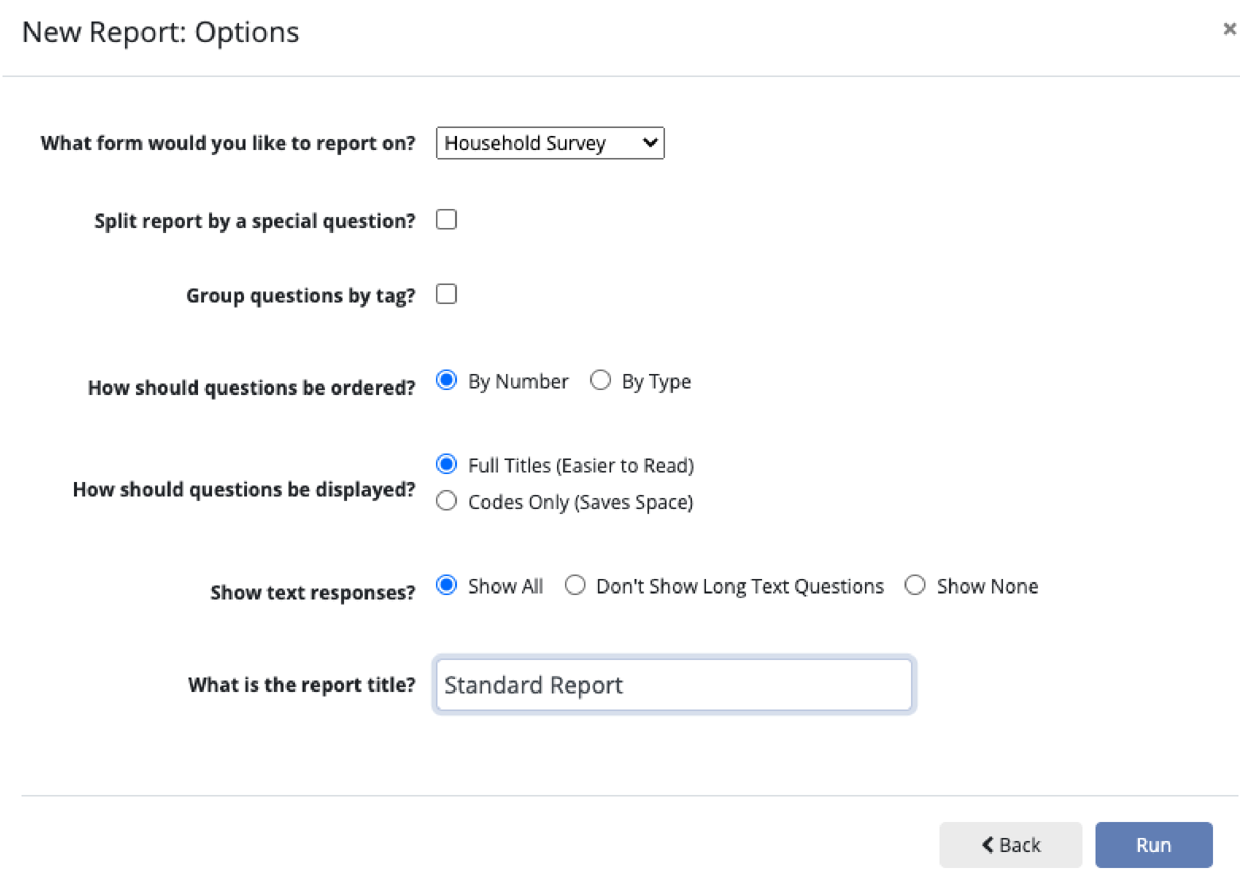
Example:



Response ID	Time Submitted	Under 12?	Num in household?
l3-ami-6u6hq	2021-05-12 13:20:24	1	3
l3-ami-vls1t	2021-05-12 13:20:41	1	2
l3-ami-amgj2	2021-05-12 13:44:50	5	3
l3-ami-iesl1	2021-05-12 13:45:06	5	2

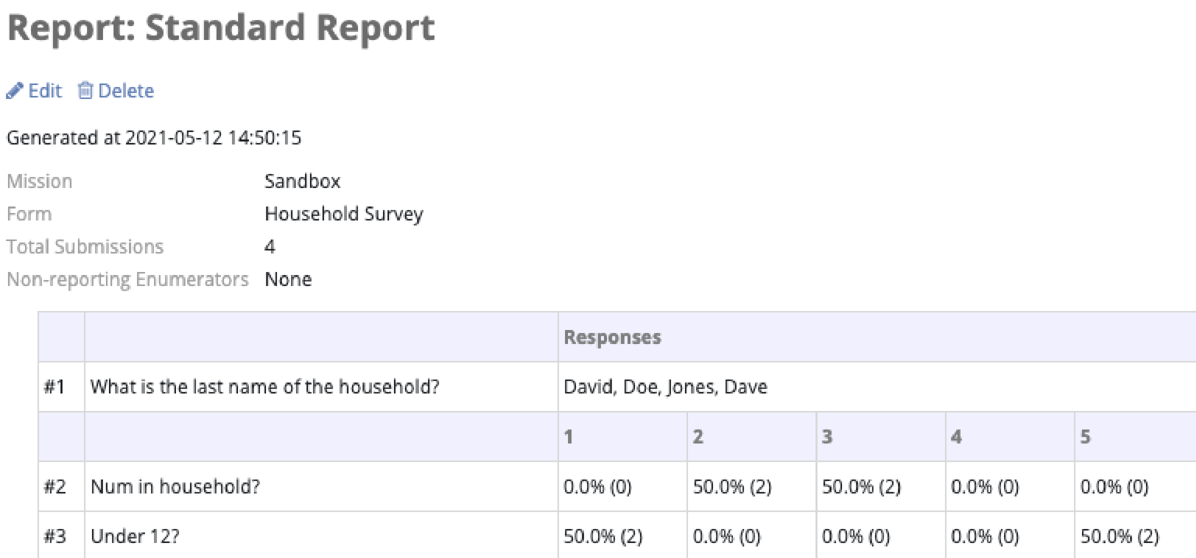
13.1.3 Standard report

A question-by-question summary of the responses for a specific form. The purpose of this report is to help give a ready-made overview of responses for a specific questionnaire. If selected, the following prompts/options are available:



1. Select the form you would like to report on.
2. If needed, check *Split report by a special question*.
3. If needed, check *Group questions by tag*.
4. Choose whether to order questions *By Number* (the order they appeared in the form) or *By Type*.
5. Select *How should questions be displayed*.
6. Choose how text responses should be displayed.
7. Type a name for the report.

Example:



Report: Standard Report

Edit Delete

Generated at 2021-05-12 14:50:15

Mission	Sandbox
Form	Household Survey
Total Submissions	4
Non-reporting Enumerators	None

		Responses				
#1	What is the last name of the household?	David, Doe, Jones, Dave				
		1	2	3	4	5
#2	Num in household?	0.0% (0)	50.0% (2)	50.0% (2)	0.0% (0)	0.0% (0)
#3	Under 12?	50.0% (2)	0.0% (0)	0.0% (0)	0.0% (0)	50.0% (2)

13.2 Exporting to spreadsheets (.csv)

If the options available within the Reports section do not meet mission needs for analyzing collected data, the data can be exported to a **CSV** file. To export data:

1. Click *Responses* menu.
2. Click *Export to CSV Format*.

Tally and **List** reports can also be exported to a CSV file:

1. Click *Reports* menu.
2. Click the report to be exported.
3. Click *Export Data To CSV Format*.

CHAPTER

FOURTEEN

CSV EXPORT

Response data can be exported to a downloadable CSV file (spreadsheet). To export data:

1. Click *Responses* menu
2. Click *Export* and select *CSV Format*

Tally and **List** reports can also be exported to a CSV file:

1. Click *Reports* menu.
2. Click the report to be exported.
3. Click *Export Data To CSV Format*.
4. The CSV file will download to your local machine.

CHAPTER FIFTEEN

XML EXPORT

Form data can be exported to a downloadable XML file. This makes it easy to import into other ODK apps such as ODK Central.

To export data for a single form:

1. Click *Form* menu and choose a particular form
2. Click *Export XML*

To export data for all published forms in a mission:

1. Click *Form* menu
2. Click *Export XML*

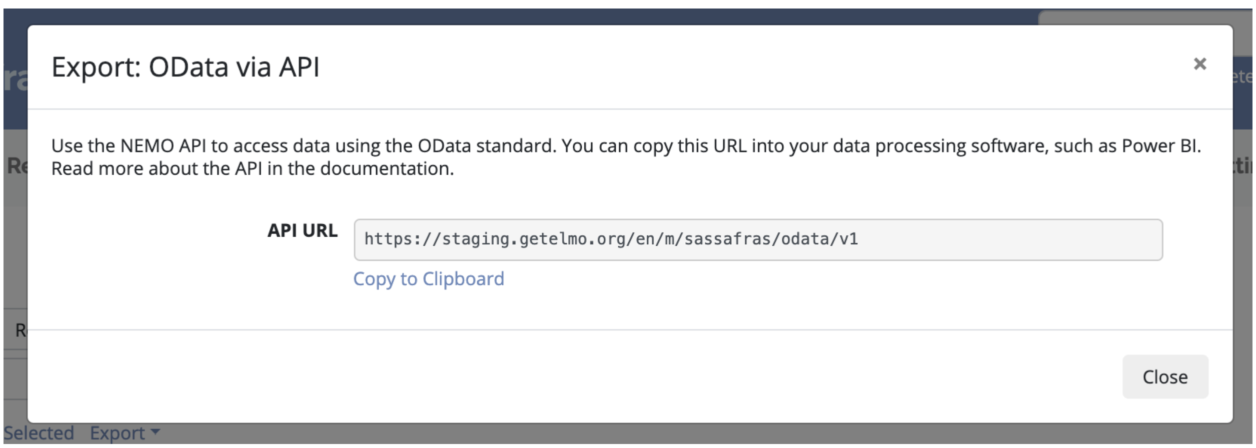
CHAPTER

SIXTEEN

ODATA VIA API

The API works with any external application that supports the OData standard (e.g. Power BI, ArcGIS, Tableau.)

1. Click *Responses* menu
2. Click *Export* and select *OData via API*
3. Click *Copy to Clipboard* to copy the API url.
4. Paste the API url in any external application that supports OData.



CHAPTER

SEVENTEEN

CLOUD

17.1 Set up a NEMO instance on the cloud

https://github.com/thecartercenter/nemo/blob/master/docs/production-setup.md

17.2 About NEMO Licenses

The open source license for NEMO code is Apache 2.0. Our code is located at GitHub. NEMO can be hosted on personal servers and users can control who has access to the data.

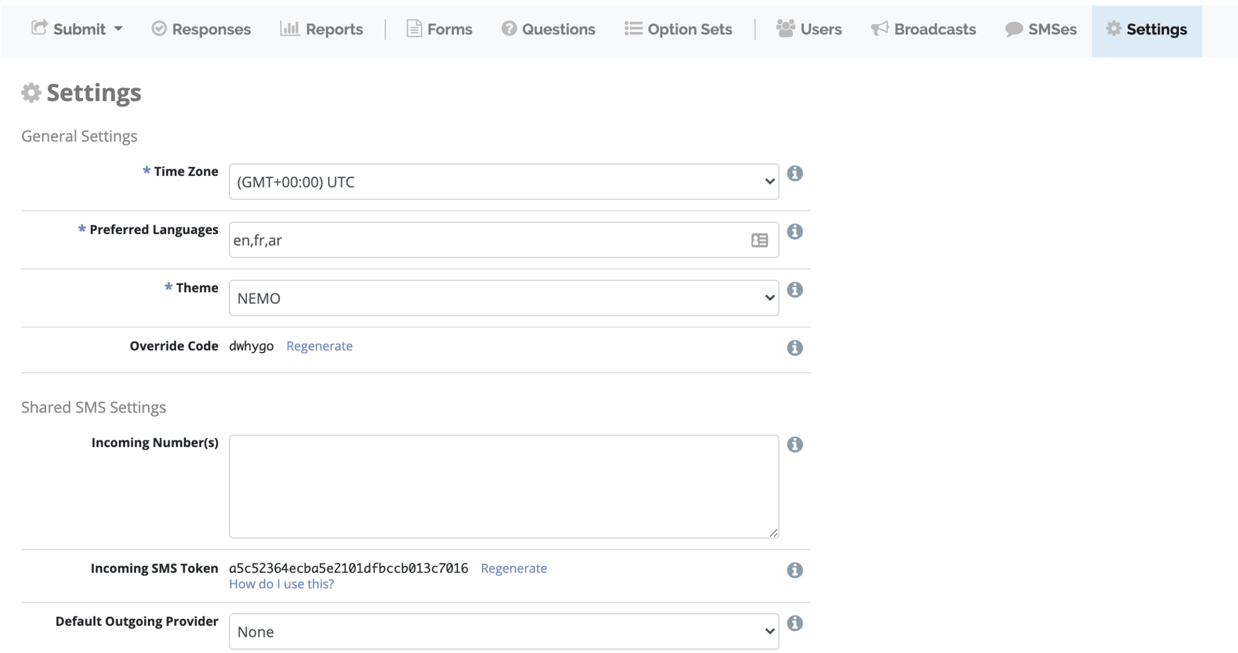
We're working on making this available in multiple languages, if this is something you are interested in contributing to, please contact us at info@getnemo.org!

CHAPTER

EIGHTEEN

SETTINGS

Settings are where you can define NEMO interface preferences and SMS information for each mission.



18.1 General Settings

18.1.1 Time zone

Time zone in which times are displayed throughout the site.

18.1.2 Preferred languages

Set the language(s) for the mission. This settings determines:

- The default language preference for new users.
- The default language for SMS replies.
- The languages for questions and options.

Enter the two-letter language code for the language (example: Arabic = ar; Chinese = zh). A list of compatible language codes can be found here. If multiple codes exist, type them in the preferred order of use and separate them with a comma (example: ar, zh).

In this example, the mission's primary language will be Arabic. Chinese will be used where Arabic is not available.

18.1.3 Theme

Choose a theme for the application. If you want to create a custom theme check the instructions.

18.1.4 Override code

Click *Generate* to set an override code. This code should be given to enumerators if the ability to send incomplete responses with ODK is needed. Check *Override code* for more details.

Click *Regenerate* to create a new override code. If generating a new code, please record the old code if there are previous live forms. The new code will only work for forms downloaded after the code is regenerated.

18.1.5 Allow Unauthenticated Submissions

If checked users will be able to send form responses without entering the authentication code.

18.2 Shared SMS Settings

18.2.1 Incoming Number(s)

Enter the phone number(s) including country code, separated by commas to which incoming SMSes for SMS forms should be sent. The phone number will be displayed on the SMS guide. Only needed if using SMS submissions.

18.2.2 Incoming SMS Token

This token is included in the URL used by the incoming SMS provider to prevent the submission of unauthorized messages. To get the URL click *How do I use this?*.

18.2.3 Default Outgoing Provider

The adapter used to send outgoing SMSes. More information can be found in *SMS Provider Setup* section.

18.3 External SQL

The SQL code used to extract data for use in external applications. Click *Select All* then copy content to paste it in the external application.

CHAPTER

NINETEEN

BROADCAST MESSAGES

Staffers, coordinators, and admins can send broadcast messages to a list of users. Broadcast messages can be sent via email and/or SMS.

Note: For SMS broadcast messages you need an SMS gateway with SMS credit established before messages can be sent.

To send a Broadcast Message:

1. Click *Broadcasts* menu.
2. Click *Send Broadcast*.
3. Select a *Medium* from the drop down list:

- **SMS preferred**: will try to send a SMS and then an email, if unsuccessful.
- **Email preferred**: will try to send an email and then an SMS if unsuccessful.
- **SMS only**: will send only SMS.
- **Email only**: will send only email.
- **Both SMS and Email**: will send both SMS and email.

4. Select *Recipients*.

- **All users in mission**: will send to all users in current mission.
- **All enumerators in mission**: will send to all enumerators in current mission.
- **Specif users/groups in mission**: type the name of the users and/or groups you want to send to.

5. Type a *Subject*.
6. Type your broadcast *Message*.
7. Click *Send*.

CHAPTER

TWENTY

SMS PROVIDER SETUP

NEMO needs to be synced with an SMS provider in order to use the SMS feature. Three options are available:

- *FrontlineSMS*
- *Twilio*
- *Generic SMS adapter*

20.1 FrontlineSMS

You can turn your Android smartphone or tablet into a gateway using FrontlineSMS. For this you need to sign up for a FrontlineCloud account.

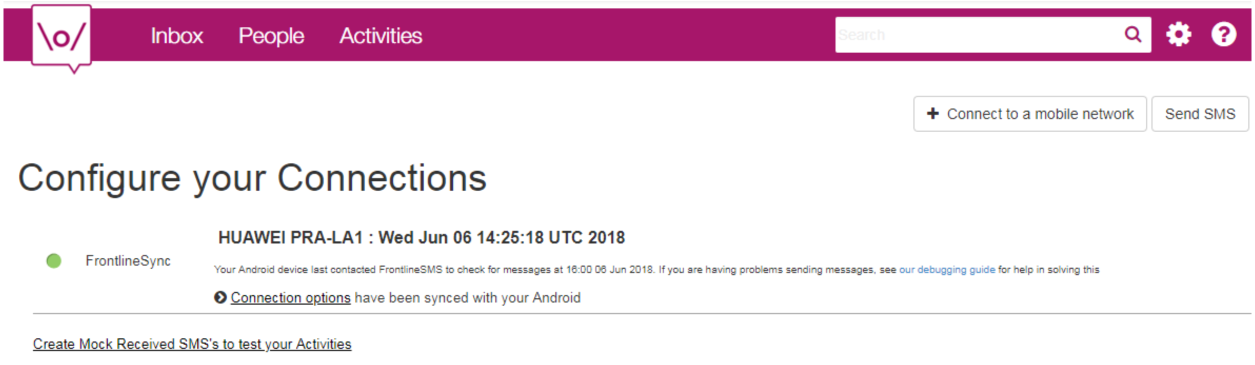
20.1.1 Synchronize FrontlineSMS with your Android device

1. Download FrontlineSync from the Play Store.
2. Open FrontlineSync app from your Android device.
3. Enter your FrontlineCloud email and password and press *CONNECT*.
4. Make sure *Send messages using this Androis* and *Upload incoming messages from FrontlineSync* are checked then press *UPDATE*.
5. Then press *DONE! START USING FRONTLINESYNC*.

To test your setup:

1. Open your FrontlineCloud account.
2. Click settings on the top right then click *Connections to mobile networks*.

You should now be able to see the device you have previously set up.



Note: For more details check Frontline documentation.

20.1.2 Synchronize FrontlineSMS with NEMO

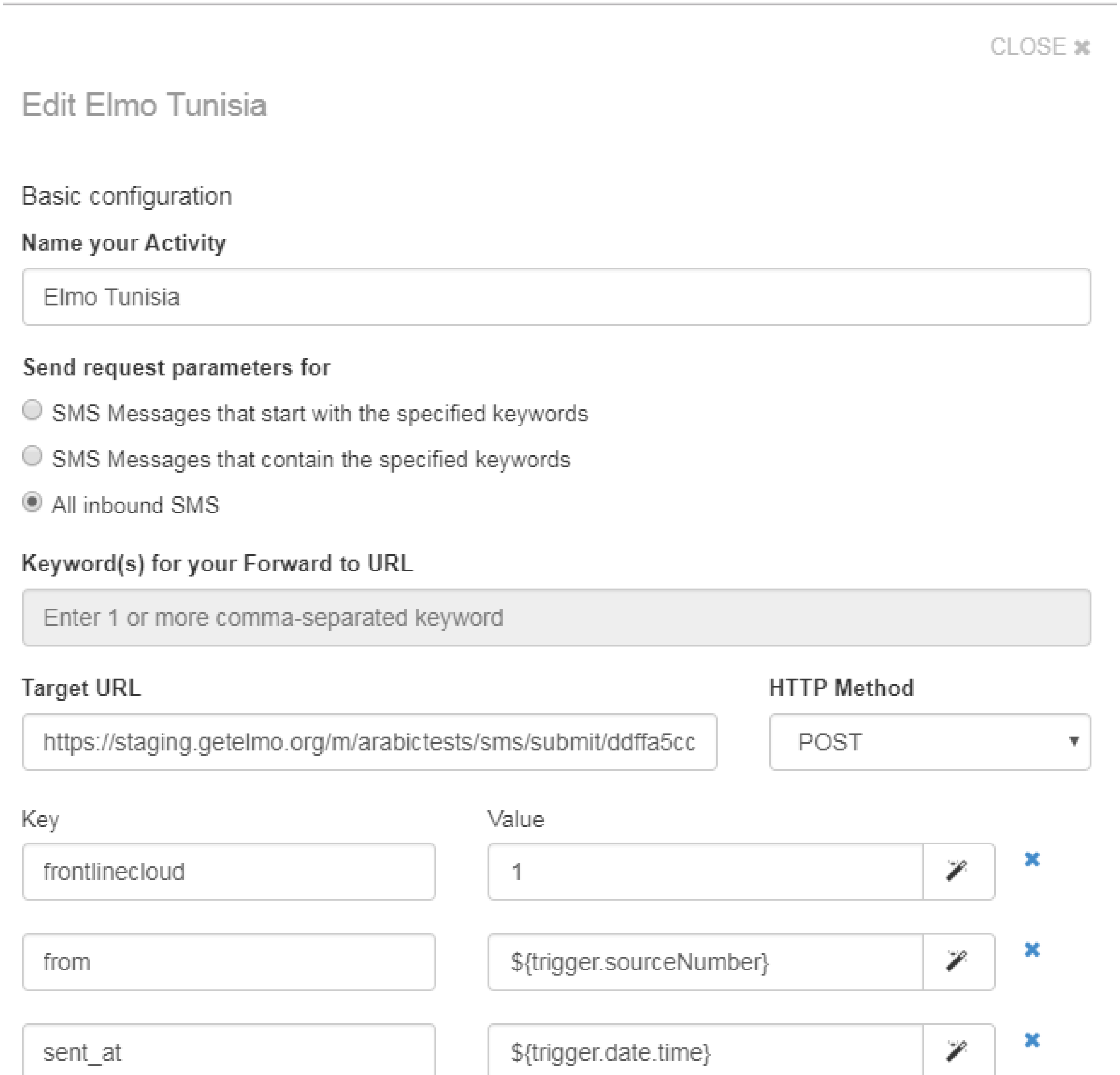
Now that you have synchronized your Android device with FrontlineCloud, you will need to synchronize FrontlineCloud with NEMO.

Set up a new activity

1. On FrontlineCloud click *Activities*.
2. Click *Create an Activity*.
3. Select *Forward to URL*.
4. Type a name for your activity.
5. Select *All inbound SMS*.
6. To get the *Target URL*, get back to your NEMO mission, click *Settings* then in the *Incoming SMS Token* section click *How do I use this?* and copy the URL that shows up in the dialog.
7. Make sure *HTTP Method* is set to **POST**.
8. Now set the following key-value pairs:

Key	Value
from	${trigger.sourceNumber}
frontlinecloud	1
sent_at	${trigger.date.time}
body	${trigger.text}

9. Click *Save*.



Generate an API Key

1. Click settings on the top right then click *API web services and Integrations*.
2. Click *Connect a web service*.
3. Select *Connect an external web service to your workspace*.
4. Enter a name for the web service.

A new row will appear on the screen with an API Key in the *Details*. We will need this API Key for the next step.

NEMO setup

1. Click *Settings* menu.
2. Add the SIM card number to the *Incoming Number(s)* field. If adding more than one number, separate the numbers with a comma.

3. Set *Default Outgoing Provider* to **FrontlineCloud**.
4. In *FrontlineCloud Settings* click *Change API Key*.
5. Paste the *API Key* that you previously generated in FrontlineCloud.

20.2 Twilio

If you are using Twilio as your SMS provider you need to set the following settings:

20.2.1 Twilio settings

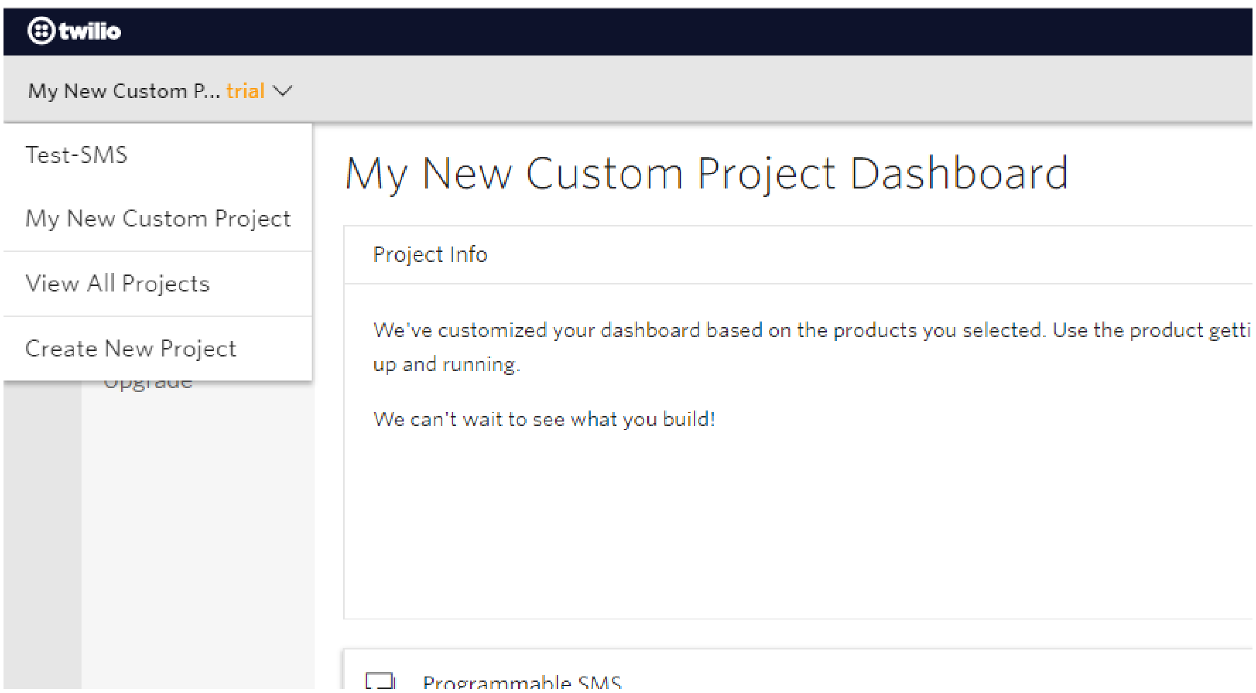
Once you have created your Twilio account you will need to:

- *Create a new Twilio project*.
- *Set up a Twilio phone number*.
- *Create a new Messaging service*.

Create a new Twilio project

To create a new Twilio project:

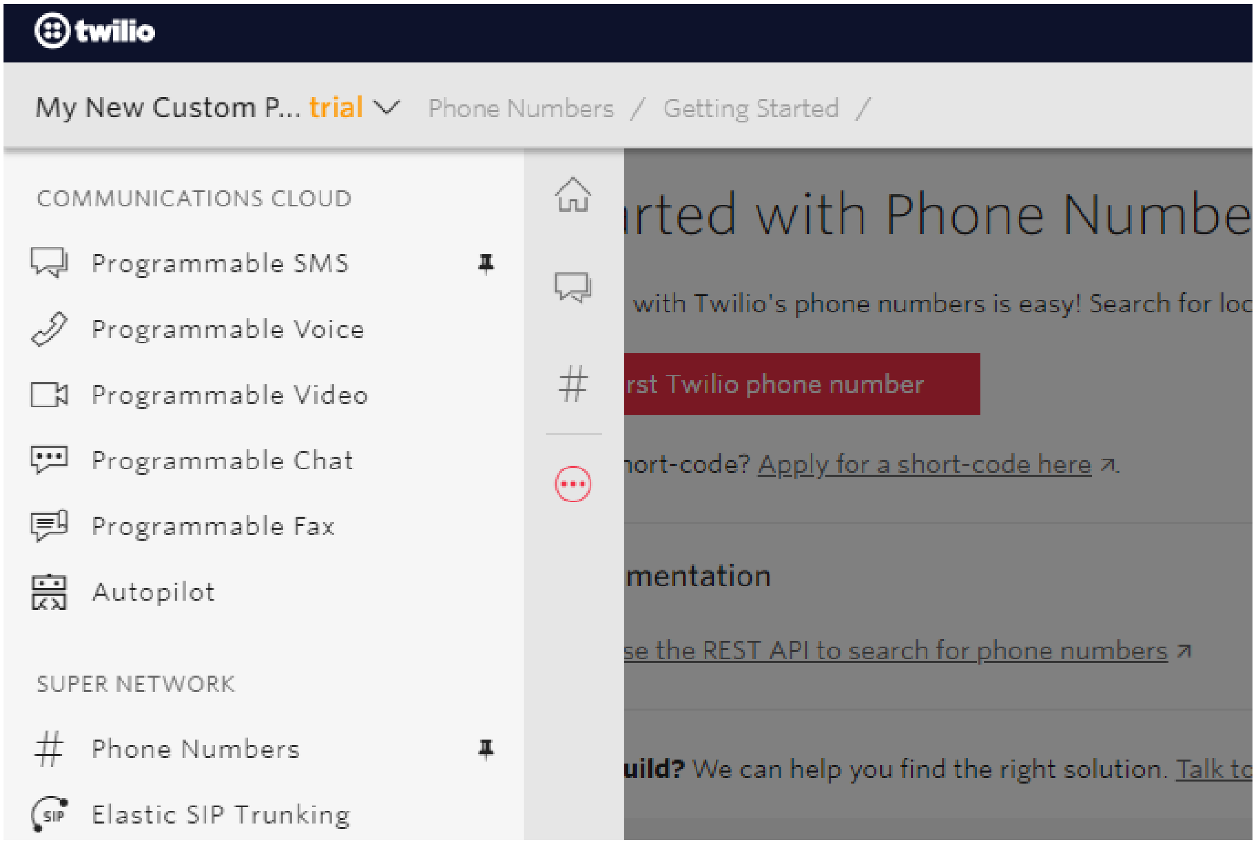
1. Click on the top left menu .
2. Select *Create New Project*.



3. Select *Products* then choose *Programmable SMS*.
4. Click *Continue*.

Set up a Twilio phone number

1. Click on the left navigation to see the list of products and services.
2. Select *Phone Numbers*.



3. Once in the Phone Numbers menu, you have three options:

- Get a free number from Twilio by going to *Getting Started* section.
- Buy a Twilio Number by going to *Buy a Number* section.
- Use your own Number by going to *Use Your Number* section.

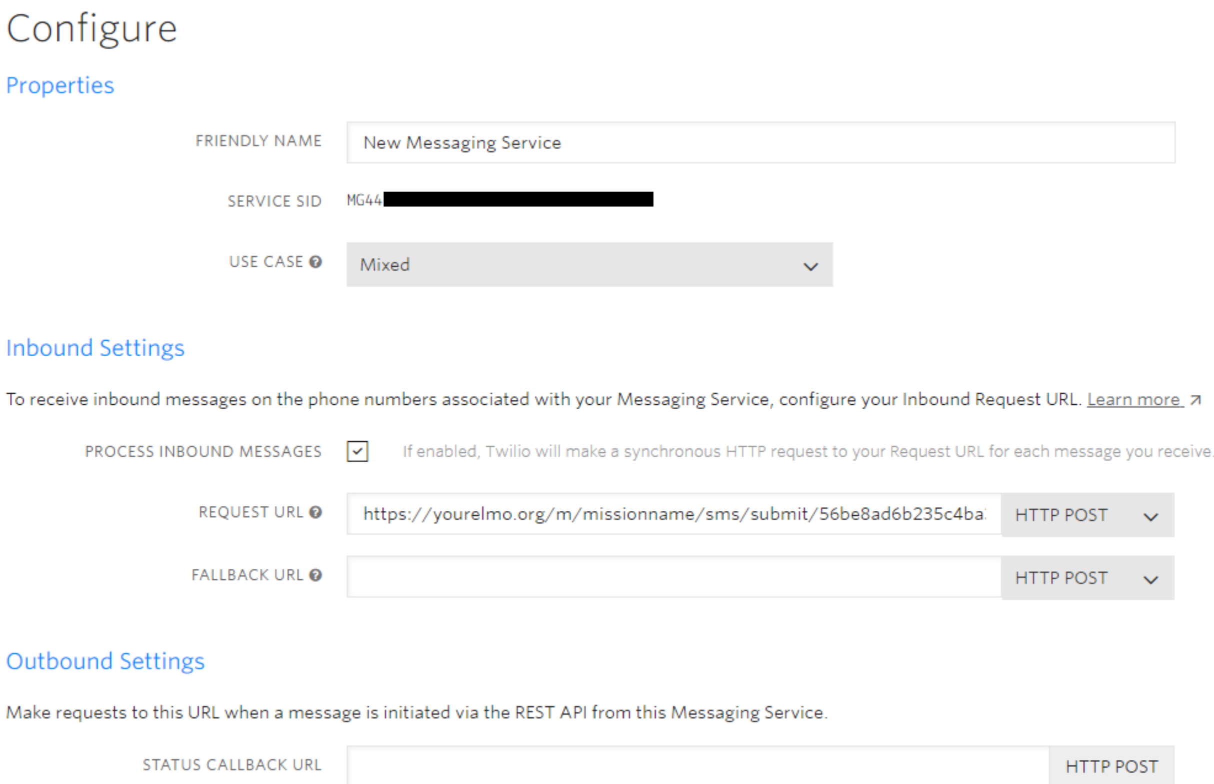
Create a new Messaging service

You need to create a new messaging service in order to forward all incoming SMSes to NEMO, for this:

1. Click *Programmable SMS* on the left navigation.
2. Click *SMS* on the left menu.
3. Click to create a new messaging service.
4. Choose a name for the service and set the *use case* to Mixed.
5. Click *Create*.
6. Under Inbound Settings, check *PROCESS INBOUND MESSAGES*.
7. To get the *REQUEST URL*, get back to your NEMO mission, click *Settings* then in the *Incoming SMS Token* section click *How do I use this?* and copy the URL that shows up in the dialog.

8. Outbound Settings should be left blank.

At the end you should have a configuration similar to this one:



9. Click *Save*.
10. Now you need to add a number to this messaging service, for this, Click *Numbers* on the left menu.
11. Click to add a number.

20.2.2 NEMO setup for Twilio

In your NEMO mission:

1. Click *Settings*.
2. Add the phone number from which you will receive SMSes to *Incoming Number(s)* field. If adding more than one number, separate the numbers with a comma.
3. Make sure the *Default Outgoing Provider:* is set to *Twilio*.
4. Scroll down to *Twilio Settings* section, set the *Outgoing Number* which is the phone number registered with Twilio. SMS replies and Outgoing SMS broadcasts won't work unless this number is owned by your Twilio account. This number must include the country code. Example: +25680344523.
5. Set the *Account SID* for your twilio account.
6. Click *Change Auth Token* to change the auth token for the Twilio account.
7. Click *Save*.

20.3 Generic SMS Adapter Settings

If you are using a provider other than Twilio and FrontlineSMS you will need to set up a Generic SMS Adapter.

1. Click *Settings* on your NEMO mission.
2. Make sure the *Default Outgoing Provider:* is set to *none*.
3. In *Generic SMS Adapter Settings* set a JSON formatted configuration string for the generic adapter. Example :

```
{
  "params": {
    "from": "num",
    "body": "msg"
  },
  "response": "<message>%{reply}</message>"
}
```

CHAPTER

TWENTYONE

DOCUMENTATION STYLE GUIDE

21.1 Spelling and grammar

21.1.1 American spelling and grammar

Whenever U.S. English and British (or other) English spelling or usage disagree, standard U.S. spelling and usage is preferred.

Wrong

```
The colour of the button is grey.
```

Right

```
The color of the button is gray.
```

21.1.2 Quote marks

- Quote marks should generally be avoided if possible.
- *Smart* quotes (also known as *curly quotes* or *directional quotes*) are not permitted in source files.

Avoid quote marks

Quote marks are used in prose writing to indicate verbatim text. This is rarely useful in technical writing, as verbatim text usually requires a more specific semantic markup.

Wrong

```
Click the button that says, "Save."
```

Right

```
Click :guilabel:`Save`.
```

Wrong

```
You may see an error message that says, "Something went wrong."
```

Right

```
You may get an error: ``Something went wrong.``
```

Straight quotes

Any time that you *do* need to use quotation marks, use straight (or *plain*) quotes. Sphinx and Docutils will output the typographically correct quote style.

21.1.3 Click instructions

When giving user interface click instructions always use **Click** rather than **Click on**.

Wrong

```
Click on :guilabel:`Create user`.
```

Right

```
Click :guilabel:`Create user`.
```

21.1.4 Serial comma

In a comma-delineated list of items, the penultimate item should be followed by a comma.

Wrong

```
Apples, oranges and pears.
```

Right

```
Apples, oranges, and pears.
```

A bulleted list is often more clear than an inline list.

Correct

```
You will need to be familiar with git, GitHub, and Python.
```

Possibly Better

```
You will need to be familiar with:

- git
- GitHub
- Python
```

There's no hard rule about which to use in any situation. Use your judgement: try it both ways and see which is more clear.

21.1.5 Ordered and unordered lists

An order list is numbered. It should be used when the order of the list is essential. For example, when enumerating a series of steps in a procedure.

Wrong

```
- First we do this.
- And then we do this.
- And the we do this.
```

Right

```
1. Do this.
2. Do this.
3. Do this.
```

An unordered list is bulleted. It should be used for a collection of items in which order is not essential.

Wrong

```
1. apples
2. oranges
3. bananas
```

Right

```
- apples
- oranges
- bananas
```

21.1.6 Avoid Latin

Several Latin abbreviations are common in written English:

At best, these present a minor barrier to understanding. This is often made worse by unintentional misuse.

Avoid Latin abbreviations.

Wrong

```
If you are writing about a specific process (e.g., installing an application)...
```

Right

```
If you are writing about a specific process (for example, installing an application)...
```

Etc.

Et cetera (or *etc.*) deserves a special mention.

Et cetera means "and all the rest," and is often used to indicate that there is more that could or should be said, but which is being omitted.

Writers often use *etc.* to gloss over details of the subject which they are not fully aware of. If you find yourself tempted use *etc.*, ask yourself if you really understand the thing you are writing about.

21.1.7 Avoid unneeded words

Adverbs

Adverbs often contribute nothing. Common offenders include:

- simply
- easily
- just

- very
- really
- basically

Wrong

```
To open the file, simply click the button.
```

Right

```
To open the file, click the button.
```

Wrong

```
You can easily edit the form by...
```

Right

```
To edit the form...
```

Filler words and phrases

Many words and phrases provide no direct meaning. They are often inserted to make a sentence seem more formal, or to simulate a perceived style of business communication. These should be removed.

Common filler phrases and words include:

- to the extent that
- for all intents and purposes
- when all is said and done
- from the perspective of
- point in time

This list is not exhaustive. These "canned phrases" are pervasive in technical writing. Remove them whenever they occur.

21.1.8 Semicolons

Semicolons are used to separate two independent clauses which could stand as individual sentences but which the writer feels would benefit by close proximity.

Semicolons can almost always be replaced with periods (full stops). This rarely diminishes correctness and often improves readability.

Correct

```
These "canned phrases" are pervasive in technical writing; remove them whenever they␣
↪occur.
```

Better

```
These "canned phrases" are pervasive in technical writing. Remove them whenever they␣
↪occur.
```

21.1.9 Pronouns

Third-person personal pronouns

Third-person personal pronouns are:

- he/him/his
- she/her/her(s)
- they/them/their(s)

Note: While some people consider *they/them/their* to be non-standard (or "incorrect") as third-person singular, it has gained wide use as a gender-neutral or gender-ambiguous alternative to *he* or *she*.

There are two issues with personal pronouns:

- gender bias
- clarity

To avoid gender bias, the third person gender-neutral *they/then/their(s)* is preferred over *he* or *she* pronouns when writing about abstract individuals.

Wrong

```
The enumerator uses his device.
```

Right

```
The enumerator uses their device.
```

Unfortunately, *they/them/their* is not a perfect solution. Since it is conventionally used as a plural pronoun, it can cause confusion.

Therefore, avoid the use of personal pronouns whenever possible. They are often out of place in technical writing anyway. Rewriting passages to avoid personal pronouns often makes the writing more clear.

Correct

```
When using Collect, first the enumerator opens the app on their device. Then they complete the survey.
```

Better

```
To use Collect:

- open the app
- complete the survey
```

"Same"

Same, when used as an impersonal pronoun, is non-standard in Modern American English. It should be avoided.

Wrong

```
NEMO is a data collection tool. The same can be used for...
```

Right

```
NEMO is a data collection tool. It can be used for...
```

Right

```
NEMO is a data collection tool that is used to...
```

21.1.10 Titles

Title case and sentence case

Document titles should be in `Title Case` – that is, all meaningful words are to be capitalized.

Section titles should use `Sentence case` – that is, only the first word should be capitalized, along with any proper nouns or other words usually capitalized in a sentence.

21.1.11 Verb forms

If a document or section describes a procedure that someone might do, use a verb ending in *-ing*. (That is, a gerund.) Do not use the "How to…" construction.

Wrong

```
How to install NEMO
--------------------------
```

Right

```
Installing NEMO
----------------------
```

If section title is a directive to do something (for example, as a step in a procedure), use an imperative.

```
Installing NEMO
------------------------

Download NEMO
~~~~~~~~~~~~~~~~~~~~~~

Section content here.
```

Section labels

Section titles should almost always be preceded by labels.

The only exception is very short subsections that repeat — like the **Right** and **Wrong** titles in this document.

In these cases, you may want to use the `rubric` directive.

Other titling considerations

- Do not put step numbers in section titles.
- Readers skim. Section titles should be clear and provide information.

Note: This document is a derivative of the original ODK style guide licensed under a Creative Commons Attribution 4.0 International License.

CHAPTER TWENTYTWO

CONTRIBUTING TO NEMO DOCS

22.1 Authoring Tools and Environment

Note: Developer and authoring tools have lots of options and alternatives. Local tools and workflows presented in this guide are what the author feels would be easiest for a non-coding contributor to set up and use.

22.1.1 Docs as code

NEMO Documentation follows (as much as possible) the Docs like Code philosophy. This means:

- Documentation source files are written in a plain text format. (We use reStructuredText.)
- Documentation source files are kept under version control. (We use git and Github.)
- Documentation is built from source to published output using a static site generator. (We use Sphinx.)

The 'Docs as Code' approach has many advantages, but we are aware that this approach can feel difficult for writers who aren't used to dealing with the command line. It can also be difficult for coders who are used to this approach, but who typically use simpler authoring tools (like Jekyll and Markdown).

This section of the Contributor Guide walks through our authoring and publishing workflow and toolchain, to make it as easy as possible for you to contribute.

22.1.2 Overview of workflow

When you first get started you'll need to:

- Fork to your own Github Account
- Clone it down to your local machine
- Install dependencies

And then each time you work you will:

- Make a branch for a specific task
- Make commits as you go
- Build and view the docs locally - Correct any errors and commit
- Push your branch to your Github fork
- Issue a pull request against the current working branch of the main repo (usually `master`)

- Pull latest back to your local machine from the main repo
- Repeat

22.1.3 Setting up your environment

Terminal (Command Line)

> **Warning:** This contributor guide is written primarily from a *nix (Bash Terminal) perspective, which is relevant to all flavors of Linux and MacOS. We consider the Bash terminal commands to be the "canonical" way to build and work with the docs.
>
> We have also provided explanations for how to adapt these commands to the Windows Command Prompt. (This is different than Windows Powershell, and not all the commands will work in Powershell. For more details on the Windows Command Prompt, see this article
>
> If you are on a Windows machine, you may prefer to use the adapted Windows instructions here. Alternatively, you can follow the Bash commands:
>
> - use the Linux subsystem (Windows 10)
> - use a bash terminal emulator, such as
>   - Cygwin
>   - gitbash
>
> The lead maintainers of this docs repo are not Windows users, so we rely on our contributor community to keep Windows-specific information complete and accurate. Contributions to this guide with explanations and help for Windows users are greatly appreciated.

Contributing to the docs requires interacting with git, Github, Python, and Sphinx, which requires the use of the Terminal. This is common among Linux users. Mac users unfamiliar with the Terminal can learn more from this tutorial.

Python

Python 3

If you don't know, check to see if you have Python 3 installed:

```
$ python3
```

On windows:

```
> python
```

If you get an error, you probably don't have it and will need to install Python 3.

On Windows make sure to select the option "Add python to the Path", while installing (see instructions) otherwise you need to add it manually .

If the Python command-line interpreter starts up, type `quit()` to exit.

Github and git

Git is a distributed version control system. It makes it possible to track changes in files over time, merge changes made by different contributors, compare different versions of the same file, and revert a file to an earlier point. Git can be very complicated, but you do not need to understand its advanced features or inner workings to use it.

GitHub is an online service that lets individuals and organizations host git repositories. It also provides additional collaboration tools like issue trackers. Open Data Kit uses GitHub for its public code and documentation projects.

You will need to:

- Install git
- Make sure that git is installed properly by typing (git) in the terminal or command prompt
  - On windows if you get any error check if environment variables are set up correctly(see instructions)
- Start a GitHub account

22.1.4 Getting ready to work

Clone the docs

Go to the NEMO Doc repo on Github, select the *Clone or download* button. Copy the URI from the text box that opens up. It will be something like: `https://github.com/your-gh-username/docs.git`

Open your terminal, and *cd* to your preferred directory. Then *git clone* the repo:

```
$ git clone https://github.com/your-github-username/docs.git
.
.
.
$ cd docs
```

The rest of the documentation assumes you are in the directory for the repo (the directory containing `conf.py` and `index.rst`).

Install dependencies

The first time you clone down the repo, you'll need to install the dependencies. Make sure you have your Python 3 virtual environment set up and activated in the docs repo and then:

```
$ pip install -r requirements.txt
```

Note: If you are working on the design, testing, or deployment of the docs, you might find the need to install an additional PyPi package. If you do, please update the requirements.txt file with `pip freeze > requirements.txt`. Pull Requests which change `requirements.txt` should include a note about why the new packages are needed.

Note: If you have problems when running the Sphinx commands (see below), you may have a dependency issue. Try running `pip install -r requirements.txt` again.

22.1.5 Workflow details

Make a new branch

Create a new branch in which you will work on a specific issue. The branch name should briefly describe what you are doing. For example, the original author of this contributor guide worked in a branch he called `contributing`. Also, make sure that all the branches are derived from the `master` branch to avoid intermixing of commits.

```
$ git checkout -b branch-name
```

Tip: Branch names should be short, lowercase, and use hyphens for separators.

Good branch names:

- `getting-started-guide`
- `contributing`
- `fix-issue-13`

Bad branch names:

- `getting started guide`
- `Getting started guide`
- `Getting_started_guide`
- `writing-the-getting-started-guide-adammichaelwood-july-2017-draft`

Work on the docs

Write and edit files in your favorite editor.

Build, view, and debug

To build the documentation into a viewable website:

```
$ sphinx-build -b dirhtml . build
```

This calls the sphinx-build utility. The `-b` switch specifies the builder, which in this case is `html` – as opposed to other builders like `pdf`. The `.` refers to the current directory (the build source) and `build` refers to the target of the build (the built files will be put into a directory labeled `build`).

When you run the build, you may see error or warning messages. These indicate potential problems with the documentation, like:

- syntax errors
- broken links
- terms not included in the glossary

Error and warning messages include a file name and line number for tracking them down. Try to resolve all your errors and warnings before issuing a pull request. However, if this is not possible, please add a note in your pull request so that we can help you debug the problem.

We will not merge Pull Requests that have warnings or errors in them.

Note: Because of a bug in Sphinx, the line numbers in error and warning messages will be off by the length of *rst_prolog* in `conf.py`.

To view the documentation in your web browser, you can use Python's built-in web server.

```
$ cd build
$ python -m http.server 8000
```

Then open your browser and go to http://localhost:8000.

Read through your doc edits in the browser and correct any issues in your source files. You'll need to shut down the web server (`CTRL C`) before rebuilding, then return to the main directory of the repo (`cd ..`).

It's a good idea to delete the `build` directory before each rebuild.

```
$ rm -rf build
$ sphinx-build -b dirhtml . build
```

Tip: The script `b.sh` is a utility script that can be run to build the directory. It not only saves typing effort but will also become the canonical build script for us, so it's good to get used to it from now.

Push your branch

Once your work on the issue is completed, add the files you've changed or created additionally, and write a relevant commit message describing the changes.

```
$ git add my_changed_files
$ git commit -m "A small but relevant commit message"
```

Then it's time to push the changes. The first time you do this on any branch, you'll need to specify the branch name:

```
$ git push origin branch-name
```

After that, you can just:

```
$ git push
```

Issue a pull request

A pull request (or PR) is a request from you to the NEMO Docs maintainers, for us to pull in your changes to the main repo.

Go the main docs repo on GitHub. You'll see a message there referencing your recently pushed branches. Select *Compare & pull request* to start a pull request.

Follow GitHub's instructions. The *Base fork* should be the main repo, and *base* should be `master`. Your repo and working fork should be listed beside them. (This should all populate by default, but you should double check.) If there is a green **Able to be merged** message, you can proceed.

You must include a PR comment. Things to include:

- A summary of what you did.

- A note about anything that probably should have been done, but you didn't do.
- A note about any new work this PR will create.
- The issue number you are working on. If the PR completes the issue, include the text `Closes #` and the issue number.
- A note about any errors or warnings, and why you did not or could not resolve them.
- A note justifying any changes to requirements.txt
- A note about any difficulties, questions, or concerns that came up while working on this issue.

Complete the pull request. The maintainers will review it as quickly as possible. If there are any problems the maintainers can't deal with, they will reach out to you.

Note: If you happen to rename any document file(`*.rst`), then be sure that you add the redirect in your PR.

To add the redirect go to `s3_website.yml`. Add a mapping from the old file name to the new file name below the **redirects:** line, one mapping per line. Several examples of how to format these are shown in the file.

For example you rename a file to `newcheck.rst` from `oldcheck.rst`, then to add the redirect:

```
redirects:
 oldcheck/index.html: /newcheck
```

Keep going

Once the PR is merged, you'll need to pull in the changes from the main repo into your local copy.

```
$ git checkout master
$ git pull master
```

Now you can find a new issue to work on, create a new branch, and get to work…

22.2 Writing in Sphinx

The NEMO documentation is built using Sphinx, a static-site generator designed to create structured, semantic, and internally consistent documentation. Source documents are written in reStructuredText, a semantic, extensible markup syntax similar to Markdown.

- reStructuredText Primer — Introduction to reStructuredText
  - reStructuredText Quick Reference
  - reStrcuturedTest 1-page cheat sheet
- Sphinx Markup — Detailed guide to Sphinx's markup concepts and reStructuredText extensions

---

Note: Sphinx and reStructuredText can be very flexible. For the sake of consistency and maintainability, this guide is *highly opinionated* about how documentation source files are organized and marked up.

22.2.1 Indentation

Indentation is meaningful in Sphinx and reStructured text.

- Use **spaces, not tabs**.
- Indent **two spaces**.

22.2.2 Documentation files

Sphinx document files have the `.rst` extension. File names should be all lowercase and use hyphens (not underscores or spaces) as word separators.

Normally, the title of the page should be the first line of the file, followed by the line of equal-signs.

```
Title of Page
================

Page content is here...
```

You can also wrap the title in two lines of asterisks.

```
*******************
Title of Page
*******************

Page content here.
```

The asterisks style is useful when you are combining several existing documents (and don't want to change every subsection headline) or when you are working on a document that might be split into separate documents in the future.

See *Sections and titles* for more details.

22.2.3 Custom CSS

You can add custom styling in `_static/css/custom.css`. Whenever you add any custom styling, add short comments describing the changes made and the PR number in which the changes were made.

For example:

```
/* Example css PR #xyx */

div[class^='example'] {
  color: black;
}
```

There are various sections in the `custom.css` file:

- Styling for rst roles and directives
- Responsive css
- Styling for JS implementation
- Utility classes

Each of these sections are enclosed between start and end comments. Make sure you add your code to the relevant section. If you don't find any section relevant, add a new section and add your code there.

For example:

```
/* New section starts */

/* Example css PR #xyx */

div[class^='example'] {
  color: black;
}

/* New section ends */
```

22.2.4 Table of contents

The `index.rst` file serves as a front-page to the documentation and contains the table of contents. The table of contents controls the documentation navigation menu. To add a new document to the table of contents, add the file new (without the `.rst` extension) to the list of file names in `index.rst`.

22.2.5 Sections and titles

Headlines require two lines: the text of the headline, followed by a line filled with a single character. Each level in a headline hierarchy uses a different character:

```
Title of the Page - <h1> - Equal Signs
=========================================


Major Section - <h2> - Hyphens
---------------------------------


Subsection - <h3> - Tildes
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Sub-subsection - <h4> - Double Quotes
""""""""""""""""""""""""""""""""""""""""


Sub-sub-subsection - <h5> - Single Quotes
'''''''''''''''''''''''''''''''''''''''''''''
```

If you need to combine several existing pages together, or want to start a single-page doc that you think might be split into individual pages later on, you can add a top-level title, demoting the other headline types by one:

```
************************************************
Page Title - <h1> - Asterisks above and below
************************************************

```

(continues on next page)

(continued from previous page)

```
Major Section - <h2> - Equal Signs
=======================================


Subsection - <h3> - Hyphens
---------------------------------


Sub-subsection - <h4> - Tildes
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Sub-sub-subsection - <h5> - Double Quotes
"""""""""""""""""""""""""""""""""""""""""""""

Sub-sub-sub-subsection - <h6> - Single Quotes
''''''''''''''''''''''''''''''''''''''''''''''''''
```

In either case, the underline of characters needs to be *longer than* the line of text. In the case of the asterisks, the two lines of asterisks need to be the same length.

Note: The exact order of underline characters is flexible in reStructuredText. However, this specific ordering should be used throughout the documentation.

Section labels

In order to facilitate efficient *Cross referencing*, sections should be labeled. This is done on the line above the section title. The format is:

- two dots
- underscore
- section label
  - lowercase
  - hyphen separators
- a single colon

```
.. _section-label:

Section Title
----------------

Lorem ipsum content of section blah blah.
```

The section label is a slugified version of the section title.

Section titles must be unique throughout the entire documentation set. Therefore, if you write a common title that might appear in more than one document (*Learn More* or *Getting Started*, for example), you'll need to include additional words to make the label unique. The best way to do this is to add a meaningful work from the document title.

```
 NEMO
===============

NEMO is a server application...

.. _aggregate-getting-started:

Get Started
-----------------
```

22.2.6 Basic markup

Note: Escaping Characters

Markup characters can be escaped using the \ characters.

```
*Italic.*

\*Not italic, surrounded by asterisks.\*
```

Italic.

Not italic, surrounded by asterisks.

Emphasis and inline literal

```
Single asterisks for *italic text* (``<em>``).

Double asterisks for **bold text** (``<strong>``).

Double back-ticks for ``inline literal text`` (``<code>``).
```

Single asterisks for *italic text* (`<em>`).

Double asterisks for **bold text** (`<strong>`).

Double back-ticks for `inline literal text` (`<code>`).

Note: The **bold**, *italic*, and `inline literal` styles do not carry semantic meaning. They should not be used when a more semantically appropriate markup construct is available; for example, when *writing about GUI text*.

Hyperlinks

External hyperlinks — that is, links to resources *outside* the documentation — look like this:

```
This is a link to `example <http://example.com>`_.
```

This is a link to example.

You can also use "reference style" links:

```
This is a link to `example`_.

.. _example: http://example.com
```

This may help make paragraphs with *a lot* of links more readable. In general, the inline style is preferable. If you use the reference style, be sure to keep the link references below the paragraph where they appear.

```
You can also simply place an unadorned URI in the text: http://example.com
```

You can also simply place an unadorned URI in the text: http://example.com

Lists

Unordered (bullet) lists

```
Bulleted lists ( ``<ul>`` ):

- use hyphens
- are unindented at the first level
- must have a blank line before and after

  - the blank line requirement means that nested list items will have a blank line␣
↪before and after as well

  - you may *optionally* put a blank line *between* list items
```

Bulleted lists ( ):

- use hyphens
- are unindented at the first level
- must have a blank line before and after
  - the blank line requirement means that nested list items will have a blank line before and after as well
  - you may *optionally* put a blank line *between* list items

Ordered (numbered) lists

```
Numbered lists ( ``<ol>`` ):

1. Start each line with a number and period
2. Can begin on any number
3. Must have a blank line before and after
4. Can have nested sub-lists

   a. nested lists are numbered separately
   b. nested lists need a blank line before and after

#. Can have automatic number with the ``#`` character.
```

Numbered lists (`<ol>`):

1. Start each line with a number and period
2. Can begin on any number
3. Must have a blank line before and after
4. Can have nested sub-lists

   a. nested lists are numbered separately

   b. nested lists need a blank line before and after

5. Can have an automatic number with the `#` character.

Definition lists

```
Definition list ( ``<dl>`` )
  a list with several term-definition pairs

Terms
  should not be indented

Definitions
  should be indented under the term

Line spacing
  there should be a blank line between term-definition pairs
```

Definition list (`<dl>` **)**
  a list with several term-defition pairs

Terms
  should not be indented

Definitions
  should be indented under the term

Line spacing
  there should be a blank line between term-definition pairs

Paragraph-level markup

```
Paragraphs are separated by blank lines. Line breaks in the source code do not create␣
↪line breaks in the output.

This means that you *could*, in theory,
include a lot of arbitrary line breaks
in your source document files.
These line breaks would not appear in the output.
Some people like to do this because they have been trained
to not exceed 80 column lines, and they like
to write .txt files this way.
Please do not do this.

There is **no reason** to put a limit on line length in source files for documentation,␣
↪since this is prose and not code. Therefore, please do not put arbitrary line breaks␣
↪in your files.
```

Paragraphs are separated by blank lines. Line breaks in the source code do not create line breaks in the output.

This means that you *could*, in theory, include a lot of arbitrary line breaks in your source document files. These line breaks would not appear in the output. Some people like to do this because they have been trained to not exceed 80 column lines, and they like to write .txt files this way. Please do not do this.

There is **no reason** to put a limit on line length in source files for documentation, since this is prose and not code. Therefore, please do not put arbitrary line breaks in your files.

Block quotes

```
This is not a block quote. Block quotes are indented, and otherwise unadorned.

  This is a block quote.
  - Adam Michael Wood
```

This is not a block quote. Block quotes are indented, and otherwise unadorned.

> This is a block quote. — Adam Michael Wood

Line blocks

```
| Line blocks are useful for addresses,
| verse, and adornment-free lists.
|
| Each new line begins with a
| vertical bar ("|").
|     Line breaks and initial indents
|     are preserved.
```

Line blocks are useful for addresses,
verse, and adornment-free lists.

Each new line begins with a
vertical bar ("|").
    Line breaks and initial indents
    are preserved.

Tables

Grid style

```
+------------+------------+-----------+
| Header 1   | Header 2   | Header 3  |
+============+============+===========+
| body row 1 | column 2   | column 3  |
+------------+------------+-----------+
| body row 2 | Cells may span columns.|
+------------+------------+-----------+
| body row 3 | Cells may  | - Cells   |
+------------+ span rows. | - contain |
| body row 4 |            | - blocks. |
+------------+------------+-----------+
```

<table>
<thead>
<tr><th>Header 1</th><th>Header 2</th><th>Header 3</th></tr>
</thead>
<tbody>
<tr><td>body row 1</td><td>column 2</td><td>column 3</td></tr>
<tr><td>body row 2</td><td colspan="2">Cells may span columns.</td></tr>
<tr><td>body row 3</td><td rowspan="2">Cells may span rows.</td><td rowspan="2">Cells
contain
blocks.</td></tr>
<tr><td>body row 4</td></tr>
</tbody>
</table>

Simple style

```
=====  =====  ======
   Inputs     Output
------------  ------
  A      B    A or B
=====  =====  ======
False  False  False
True   False  True
False  True   True
True   True   True
=====  =====  ======
```

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

CSV Table

The *csv-table* directive is used to create a table from CSV (comma-separated values) data. CSV is a common data format generated by spreadsheet applications and commercial databases. The data may be internal (an integral part of the document) or external (a separate file).

```
.. csv-table:: Example Table
 :header: "Treat", "Quantity", "Description"
 :widths: 15, 10, 30

 "Albatross", 2.99, "On a stick!"
 "Crunchy Frog", 1.49, "If we took the bones out, it wouldn't be
 crunchy, now would it?"
 "Gannet Ripple", 1.99, "On a stick!"
```

Table 1: Example Table

Treat	Quantity	Description
Albatross	2.99	On a stick!
Crunchy Frog	1.49	If we took the bones out, it wouldn't be crunchy, now would it?
Gannet Ripple	1.99	On a stick!

Some of the options recognized are:

`:widths:`

Contains a comma or space-separated list of relative column widths. The default is equal-width columns.

Note: The special value *auto* may be used by writers to decide whether to delegate the determination of column widths to the backend.

`:header:`

Contains column titles. It must use the same CSV format as the main CSV data.

`:delim:`

Contains a one character string used to separate fields. Default value is comma. It must be a single character or Unicode code.

```
.. csv-table:: Table using # as delimiter
  :header: "Name", "Grade"
  :widths: auto
  :delim: #

  "Peter"#"A"
  "Paul"#"B"

.. csv-table:: Table using # as delimiter
  :header: "Name", "Grade"
  :widths: auto
  :delim: #

```

(continues on next page)

(continued from previous page)

```
  "Peter"#"A"
  "Paul"#"B"
```

:align:

It specifies the horizontal alignment of the table. It can be *left* ,`right` or *center*.

```
.. csv-table:: Table aligned to right
  :header: "Name", "Grade"
  :align: right

  "Peter", "A"
  "Paul", "B"

.. csv-table:: Table aligned to right
  :header: "Name", "Grade"
  :align: right

  "Peter", "A"
  "Paul", "B"
```

:file:

Contains the local filesystem path to a CSV data file.

:url:

Contains an Internet URL reference to a CSV data file.

Note:

- There is no support for checking that the number of columns in each row is the same. However, this directive supports CSV generators that do not insert "empty" entries at the end of short rows, by automatically adding empty entries.

```
.. csv-table:: Table with different number of columns in each row
   :header: "Name", "Grade"

   "Peter"
   "Paul", "B"
```

Table 2: Table with different number of columns in each row

Name	Grade
Peter	
Paul	B

- Whitespace delimiters are supported only for external CSV files.

For more details, refer this guide on CSV Tables.

22.2.7 Sphinx-specific markup

Roles and directives

A *role* is an inline markup construct that wraps some text, similar to an HTML or XML tag. They look like this:

```
:rolename:`some text`
```

A directive is a block-level markup construct. They look like this:

```
.. directivename:: additional info or options here
  :option: optional-value
  :option: optional-value

  Content of block here, indented.

This is no longer part of the block controlled by the directive.
```

Most of the Sphinx-specific and NEMO-specific markup will use one or both of these constructs.

Cross referencing

Cross referencing is linking internally, from one place in the documentation to another. This is **not** done using the *Hyperlinks* syntax, but with one of the several roles:

```
:role:`target`
  becomes...
    <a href="target">reference title</a>

:role:`anchor text <target>`
  becomes...
    <a href="target">anchor text</a>
```

:doc:

- Links to documents (pages)
- *target* is the file name, without the `.rst` extension
- *title* is the first *headline* (<h1>) of the page

:ref:

- Links to *sections*
- *target* is the *Section labels*
- *title* is the *section title (headline)*

:term:

- Links to items in the glossary
- *target* is the term, in the glossary
- *title* is the term itself

To recap: If you do not include an explicit `<target>`, the text inside the role will be understood as the target, and the anchor text for the link in the output will be the title of the target.

For example:

```
- Link to this document:

  - :doc:`contributing`
  - :doc:`anchor text <contributing>`

- Link to this section:

  - :ref:`cross-referencing`
  - :ref:`anchor text <cross-referencing>`

- Link to a term:

  - :term:`participant`
  - :term:`anchor text <participant>`
```

- Link to this document:
  - *Contributing to NEMO Docs*
  - *anchor text*
- Link to this section:
  - *Cross referencing*
  - *anchor text*
- Link to a term:
  - participant
  - anchor text

Writing about user interface

Several roles are used when describing user interactions.

:guilabel:

Marks up *actual UI text* of form labels or buttons.

```
Press the :guilabel:`Submit` button.
```

:menuselection:

Marks up the *actual UI text* of a navigation menu or form select element.

```
Select :menuselection:`Help` from menu.
```

When writing about multi-level menus, use a single `:menuselection:` role, and separate menu choices with `-->`.

```
To save your file, go to :menuselection:`File --> Save` in the Main Menu.
```

Note: In some situations you might not be clear about which option to use from `:menuselection:` and `:guilabel:`, in which case you should refer to the following rule that we observe in our writing.

- Actual UI text will always receive `:guilabel:` role unless the text could reasonably be understood to be part of a menu.
- If the actual UI text could be understood as a menu, `:menuselection:` should be used.

`:kbd:`

Marks up a sequence of literal keyboard strokes.

```
To stop the local server, type :kbd:`CTRL C`.
```

`:command:`

Marks up a terminal command.

```
To build the documentation, use :command:`sphinx-build`.
```

`:option:`

Marks up a terminal command option.

```
The :option:`-b html` option specifies the HTML builder.
```

Other semantic markup

`:abbr:`

Marks up an abbreviation. If the role content contains a parenthesized explanation, it will be treated specially: it will be shown in a tool-tip in HTML.

`:dfn:`

Marks the defining instance of a term outside the glossary.

```
:dfn:`NEMO` (NEMO) is a data collection tool.
```

`:file:`

Marks the name of a file or directory. Within the contents, you can use curly braces to indicate a "variable" part.

```
is installed in :file:`/usr/lib/python2.{x}/site-packages`
```

In the built documentation, the `x` will be displayed differently to indicate that it is variable.

`:program:`

Marks the name of an executable program.

```
launch the :program:`NEMO Installer`
```

Images and figures

Image files should be put in the same directory as the `.rst` source file.

You must perform lossless compression on the source images. Following tools can be used to optimize the images:

- **ImageOptim** is a tool that allows us to optimize the images. It is not format specific which means it can optimize both jpeg as well as png images. You can download it from here . After launching ImageOptim.app, dragging and dropping images into its window gives you an in-place optimized file.
- **Pngout** is another option for optimizing png images. Installation and usage instructions can be found here .
- **Mozjpeg** can be used to optimize jpeg images. Installation and related information can be found on this link .

To place an image in a document, use the `image` directive.

```
.. image:: /img/{document-subdirectory}/{file}.*
  :alt: Alt text. Every image should have descriptive alt text.
```

Note the *literal* asterisk at the end *in place of a file extension*. Use the asterisk, and omit the file extension.

Use the `figure` to markup an image with a caption.

```
.. figure:: /img/{document-subdirectory}/{file}.*
  :alt: Alt text. Every image should have descriptive alt text.

  The rest of the indented content will be the caption. This can be a short sentence or␣
↪several paragraphs. Captions can contain any other rst markup.
```

Substitutions

Substitutions are a useful way to define a value which is needed in many places. Substitution definitions are indicated by an explicit markup start (".. ") followed by a vertical bar, the substitution text (which gets substituted), another vertical bar, whitespace, and the definition block. A substitution definition block may contain inline-compatible directives such as *image* or replace. For more information, refer this guide.

You can define the value once like this:

```
.. |RST| replace:: reStructuredText
```

and then reuse it like this:

```
We use |RST| to write documentation source files.
```

Here, `|RST|` will be replaced by reStructuredText

You can also create a reference with styled text:

```
.. |slack| replace:: **NEMO Slack**
.. slack: https://nemocommunity.slack.com
```

You can use the hyperlink reference by appending a "_" at the end of the vertical bars, for example:

```
You can ask about your problem in |slack|_.
```

You can ask about your problem in **NEMO Slack**.

The rst_epilog in conf.py contains a list of global substitutions that can be used from any file. The list is given below:

- If you want to create a hyperlink reference for NEMO Slack, you can use |nemo-slack|_.

  ```
  You can use |nemo-slack|_ to ask your questions.
  ```

  You can use **|nemo-slack|_** to ask your questions.

- To create a hyperlink reference for docs-related issues, use |docs-issue|_.

  ```
  If you find a problem, file an |docs-issue|_.
  ```

  If you find a problem, file an **|docs-issue|_**.

- To create a hyperlink reference for contributors guide, use |contrib-guide|_.

  ```
  
  ```

  Be sure to read the **|contrib-guide|_**.

  Be sure to read the **|contrib-guide|_**.

You can add inline images in the document using substitutions. The following block of code substitutes arrow in the text with the image specified.

```
The |arrow| icon opens the jump menu.

.. |arrow| image:: /img/{document-subdirectory}/{file}.*
           :alt: Alt text.
```

Image file names

Image file names should:

- be short yet descriptive
- contain only lower case characters
- have no spaces
- use hyphens as the separator

Good image file names:

- collect-home-screen.png
- build-data-export-menu.png

Bad image file names:

- Collect home screen.png

- `collect_home_screen.png`
- `3987948p2983768oh184692p094.jpg-large`

Tip: Be sure to obscure any personally-identifiable information from screen shots. Crop to the smallest relevant screen area. Annotate screen shots with arrows or circles to indicate relevant information.

Code samples

Use the `code-block` directive to markup code samples. Specify the language on the same line as the directive for syntax highlighting.

```rst
.. code-block:: rst

  Use the ``code-block`` directive to markup code samples.

.. code-block:: python

  print("Hello NEMO!")

.. code-block:: console

  $ python --version

.. code-block:: java

  public class HelloWorld {

      public static void main(String[] args) {
          // Prints "Hello, World" to the terminal window.
          System.out.println("Hello, World");
      }

  }
```

Note: **rst** code-blocks wrap overflow lines by default. To unwrap overflow lines, use **unwrap** class with **rst** code-blocks.

```rst
.. code-block:: rst
  :class: unwrap
```

Code-blocks for other languages don't wrap overflow lines. Instead of wrapping, you need to scroll side-ways. To wrap overflow lines with other code-blocks, use **wrap** class with them.

```rst
.. code-block:: python
  :class: wrap
```

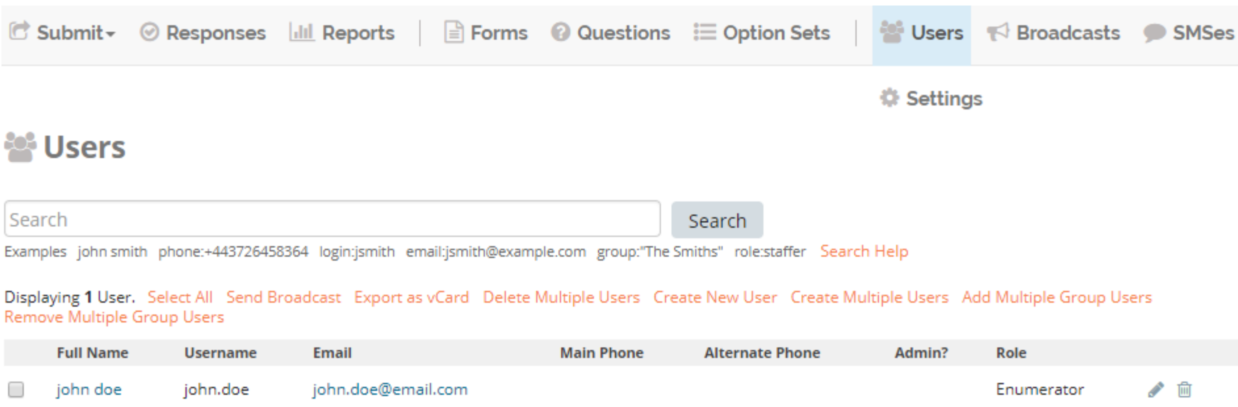
22.2.8 Font Awesome icons

Font Awesome icons are used in this project. For example to render the pencil icon the below syntax is used.

```
:fa:`pencil`
```

22.2.9 Screenshots

Whenever screenshots are used avoid any editing like adding numbers or arrows to the screenshot. Prefer using **guilabel** or **Font Awesome icons**.



:guilabel:

```
Click :guilabel:`Create New User`.
```

:fa:

```
Click :fa:`pencil` to edit user.
```

Note: This document is a derivative of the original Contributing to ODK Docs licensed under a Creative Commons Attribution 4.0 International License.

INDEX