A WSC guide to TextIt

www.textit.in and www.textit.in/api/v1

Basics:

TextIt is a service that we are using in the WSC project to collect answers to survey questions posed to smallholder farmers in Zambia and Kenya. The surveys are administered and collected via SMS (text message) on a weekly basis.

Here at IU we develop a series of questions for farmers that can help us get a better idea of the current ground situation in regards to crop, weather etc. These questions are strung together in system that is based on a 'question-response' cycle. This series of questions is what is referred to in the TextIt language as a **flow**.

Our flows are sent to farmers in Zambia and Kenya weekly and are often limited to about five questions, to help ensure the survey is finished before the farmer is interrupted. We try to develop questions that will be straightforward and understandable but will also give robust and dependable data.

Each individual with an account that is working with us will go through the weekly flow and their answers will be stored in variables as we have them defined. This set of variables for a particular contact through a flow is referred to as a **run**.

In the WSC project we have three major 'series' of flows that mirror the agricultural cycle. The series reflect some of the different ground variables at different times, we group our flows into 'planting', 'growing' and 'harvesting' series.

We currently enroll about 200 farmers in Kenya and less than 100 in Zambia. Response rates vary between the two countries, but in Kenya we consistently have 75-80% of our farmers complete the flows on a weekly basis.

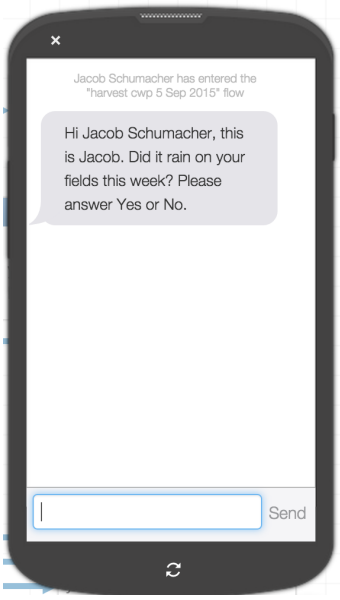
Developing new flows is easy and the TextIt flow-builder allows us to incorporate logic that can affect the questions or results for each user. This logic, along with some of TextIt's classifications systems, is what allows us to properly fit our questions to different subsets of our contacts.

As a farmer completes their run through the flow, their answers are stored by TextIt with accompanying metadata and identifications. We gather the data from TextIt using the TextIt API, a interface which allows us to access the data with nearly any programing language we like. As we continue to develop the ideal storage structure, the data will soon be available in an easy and manageable format for anyone to use.

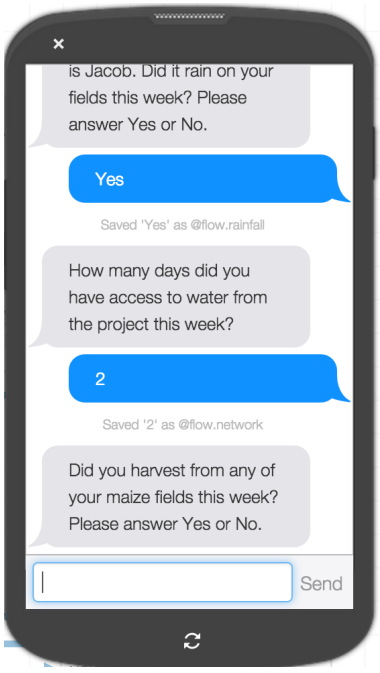
The TextIt development team is based out of Kigali, Rwanda and many of the projects that the application facilitates are related to health and agriculture. The development team has been extremely reliable and helpful in answering questions about a variety of topics related to development of flows, data structure and more. The WSC project was featured in one of the TextIt blog posts, see here: **http://blog.textit.in/crop-failure-early-warning-systems-powered-by-textit-a-case-study**

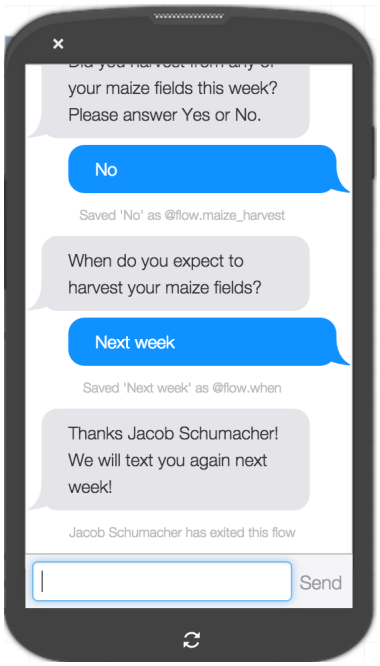
Example Run through a Flow:

At the beginning of the week the farmer will receive a message on his/her cell phone via SMS with the following content:



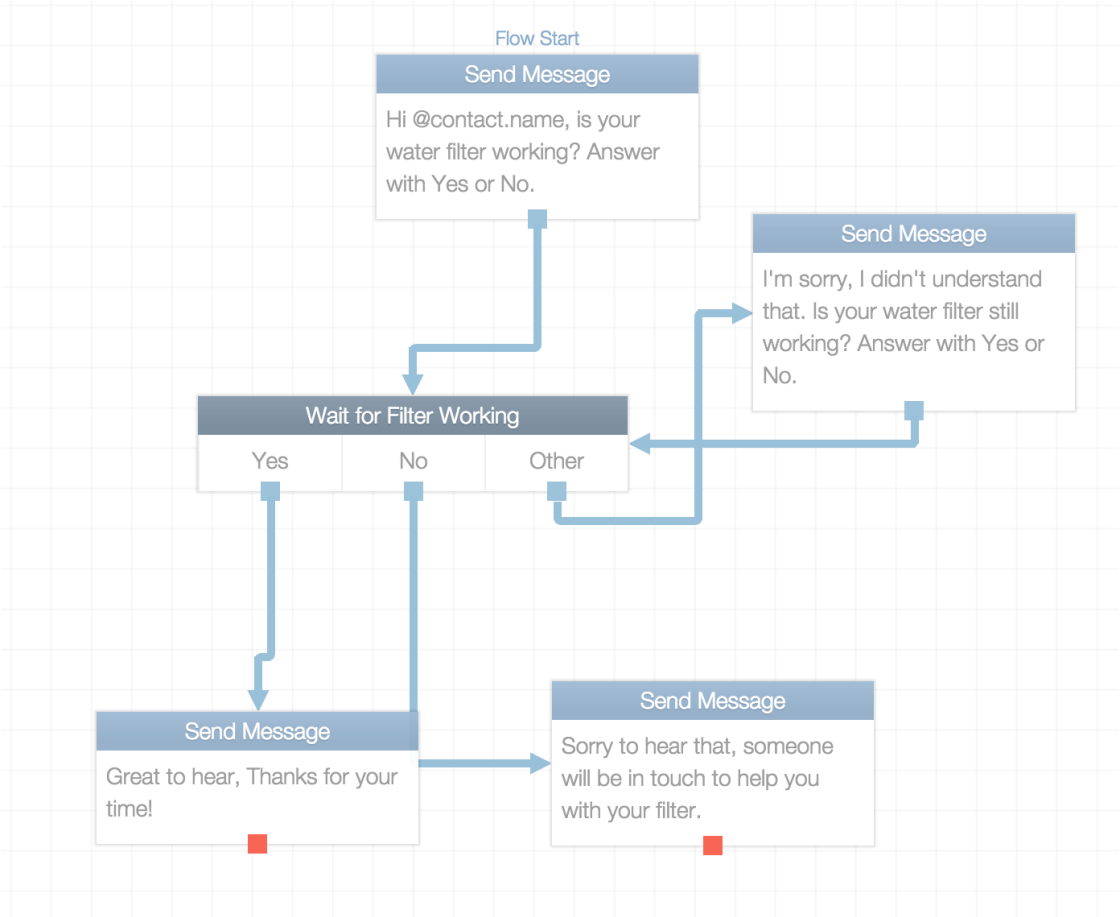
The farmer is then able to answer this question, after which the TextIt flow will automatically follow up with another question. The rest of the flow looks something like this…





Building a Flow:

Here is a simple example of a flow that TextIt provides to new users:



This flow shows some of the core functionality behind the TextIt platform and you can begin to understand the possibilities that the system offers. The forking of questions in a flow for instance is a very powerful tool that allows us to ask questions dynamically, but does not add any major complications to our data structure.

TextIt lists some of the additional functionalities offered in the flow-builder:

- Handling unexpected responses
- Testing flows in the simulator
- Managing users with groups
- Supporting additional responses
- Collecting numeric data
- Using variable subsitution
- Updating contacts inside flows
- Sending emails inside flows

Questions:

The following are the questions that we have asked farmers in the past as a part of the flow surveys:

Kenya Planting 2015
Did you plant any maize this week?
What seed variety did you plant?
How many kilograms of this seed variety did you plant?
Did you apply any fertilizer to this planting?
In the last 7 days, how many days did you irrigate your maize fields?

Kenya Growing 2015
Did it rain on your fields this week?
How many days did you work outside your farm for pay?
Did you weed any of your maize fields this week?
Did you apply fertilizer in any of your fields this week?
In the last 7 days, how many days did you have access to water from the network?
At this point, how many 90kg bags of maize do you expect to harvest at the end of this growing season?
How many 90kg bags of maize do you have remaining from last growing season's harvest?

Kenya Harvest 2015
Did it rain on your fields this week?
How many days did you have access to water from the project this week?
Did you harvest any of your maize fields this week?
How many maize cobs of green maize have you harvested this week?
How many 90kg bags of dry maize have you harvested this week?
Have you harvested all of your maize fields?

Zambia has a different set of survey questions that reflect slight differences in the farming practices and standards of the country. Some of the questions are the same as the Kenya questions, and some of the same questions are asked in multiple seasons:

Zambia Planting 2015
Did you plant any portion of your main field in the last week?
Did you plant your entire field, more than half your field or less than half your field?
Did you plant local maize, early maturing maize or medium maturing maize?
Is your entire field planted now?
Did it rain on your fields this week?

Zambia Growing 2015
Did you weed your main field in the last two weeks?
Did you apply any fertilizer to your main field in the last two weeks
Using the measuring stick provided, how tall is the maize in your main field? Please answer in cm.
At this stage how many 50kg bags do you expect to harvest at the end of the season from this main field?
Did you plant any garden crops in the last two weeks?
Did you do any piecework in the last two weeks?
How many 50kg bags of maize do you have left from last years harvest?

Zambia Harvest 2015
Have you harvested your main field?
How many 50kg bags of maize have you harvested from your main maize field?
Have you harvested your entire main maize field?
When do you expect to harvest your main maize field?

Data Structure:

As mentioned in the introductory section, TextIt stores our data for us until we call for it using the API. Knowing how that data is stored as it is waiting for us to retrieve it is important for those of us developing questions as well as those of us who will be working with the data directly as we develop its storage format and begin early analysis.

As you can see in the flow-builder example, TextIt allows us to set the response to a question as a variable (seen as 'Wait for [variable]' in the flow-builder). Essentially, the simplest level of our data are these key:value pairs between the response and defined variables. However, due to the nature of the SMS medium and the TextIt platform there are ways in which this simple data is confounded before we are able to use it.

In a simple flow such as this, the most obvious confounding of the stored data will be the metadata. This is not a bad confounding, but can very useful. Rather than just receiving the stored variable, the data for a flow has: identifications for the contact, the flow they are in, the time that each message was sent or received and additional information.

In more complex flows however, the metadata and additional data structure hierarchies can be difficult to navigate. For those of us working directly with the data it is important to know more about the structure before trying to extract its useful components.

The first thing to know about the data is that we'll be receiving it in a JSON format. If you're not familiar with JSON you can read about it here: **http://json.org/**

JSON is a simple organizational format used in data exchange. It may look a bit more rigid than the CSV format that some Excel data users may be reminded of, but it's inherently uncomplicated, as it focuses on the key:value pair as it's primary building block. What becomes more complicated in our work however are the immense hierarchies that are needed to express the content of a completed flow or run.

Helpfully, TextIt has provided keys to decrypting or at least understanding what comes through a request as a block of code. Here's a look at a simple set of values in JSON with none of the additional metadata or context:


```
"values": [
  { "label": "Water Source", "category": "Stream", "text": "from stream", "value": "Stream", "time": "2013-01-01T05:35:32.012" },
  { "label": "Boil", "category": "Yes", "text": "yego", "value": "Yes", "time": "2013-01-01T05:36:54.012" },
  { "label": "Household Members", "category": "All", "value": "15", "text": "15", "time": "2013-01-01T05:58:32.012" }
]
```


Our objective on the data processing side is to be able to take objects like these, with their identifying components and build them into a structure that is accessible, understandable and usable.

Data Retrieval and the API:

The TextIt API is the next step in understanding the full structure and potential of the data that we'll be working with. The API has excellent documentation which can be viewed here: **https://textit.in/api/v1**

The API has two major options for data access: the web hook and the RESTful call for batch data. The WSC team has worked with both options, which have their own merits, but for the most part at this stage we're focusing on batch processing as we explore some of the possible implementations of the data (storage, visualization, etc.). Both styles of data retrieval use REST methods to access the data.

In the API the best dataset that we can make a call for is the 'runs' dataset using the '/runs' endpoint in our URL. To do this we must send various parameters along with the request to get the right data. TextIt provides us with an API token to put into the header of the request. The parameter that we should be using to gather the runs is the 'flow_uuid' parameter. If we then make a call to the '/runs' endpoint with a 'flow_uuid' parameter set we will receive all of the runs through a particular flow.

For instance, if we're making the above call to a flow with 200 participants we will receive a JSON object with 200 objects in the results : [array], with each of these representing a single users messages in the flow. These results objects contain not only the response messages, but also the messages that we've sent. They also contain runs which are not complete. However, in this way the '/runs' endpoint is reliable because we always get an object that accounts for every user. Some metadata will help us to parse out unneeded data, such as the 'completed' boolean and the 'type' key which can identify sent or received messages.

The '/runs' endpoint is just one option for retrieving the data in the API, but it is the one that has been the focus for much of the batch processing work in WSC. This is partially because the structure of the '/runs' endpoint JSON objects is very similar to the structure of the web hook POST object. You can find an example of the full runs structure on the next page.

This example shows the first of 389 runs through this particular flow. This run first has the metadata (uuid, contact, etc.) then the 'steps' array' which shows the full expression of each message sent or received, and finally the 'values' array, where most of the data that we're interested in is stored.

Going forward with the work from this stage WSC members have mapped out two clear objectives for the data. The first is to prepare the data for storage in SEAD or another data curation service (SEAD2.0?). The ideal process here would be to use both the TextIt API and the SEAD API with some standard parsing operation in between. The second major objective is to be able to make simple visualizations of the data. These two objectives should in the long run give us the infrastructure to produce high quality scientific research from the data as well as crowd-pleasing communication of understanding in other mediums such as a website or conference presentations.

```
{
    "count": 389,
    "next": "/api/v1/runs/?page=2",
    "previous": null,
    "results": [
    {
        "uuid": "988e040c-33ff-4917-a36e-8cfa6a5ac731",
        "flow_uuid": "f5901b62-ba76-4003-9c62-72fdacc1b7b7",
        "contact": "09d23a05-47fe-11e4-bfe9-b8f6b119e9ab",
        "created_on": "2013-03-02T17:28:12",
        "expires_on": "2015-07-08T01:10:43.111Z",
        "expired_on": null
        "steps": [
            {
                "node": "22bd934e-953b-460d-aaf5-42a84ec8f8af",
                "left_on": "2013-08-19T19:11:21.082Z",
                "text": "Hi from the Thrift Shop!  We are having
specials this week. What are you interested in?",
                "value": null,
                "arrived_on": "2013-08-19T19:11:21.044Z",
                "type": "A"
            },
            {
                "node": "9a31495d-1c4c-41d5-9018-06f93baa5b98",
                "left_on": null,
                "text": "I want to buy a fox skin",
                "value": "fox skin",
                "arrived_on": "2013-08-19T19:11:21.088Z",
                "type": "R"
            }
        ],
        "values": [
            {
                "step_uuid": "9a31495d-1c4c-41d5-9018-06f93baa5b98",
                "category": "fox skins",
                "text": "I want to buy a fox skin",
                "label": "Interest",
                "value": "fox skin",
                "time": "2013-05-30T08:53:58.531Z"
            }
        ],
    },
    ...
}
```