

*m*TRACK-DHIS 2 Plugin

Samuel Sekiwere

June 18, 2012

Contents

1	Introduction	3
1.1	Background	3
1.1.1	<i>mTrack</i>	3
1.1.2	DHIS 2	3
1.1.3	Reason of Interoperability	3
2	How the two systems can communicate	4
2.1	Uniting Components	4
2.1.1	Health Facilities and Organisation Units	4
2.1.2	XForms and DataSets	4
2.1.3	XFormFields and DataElements	5
2.1.4	XFormSubmissionValues and CategoryOptionCombos	5
3	Design of mTrack-DHIS 2 Plugin	6
3.1	DHIS 2 Web API	6
3.1.1	Retrieving Data	6
3.1.2	Sending data	6
3.2	Data Mapping	7
3.2.1	The all_submission_values_view	7
3.2.2	The dhis2_mapping Table	8
4	Implementation	9
4.1	rapidsms_dhis2 application/submodule	9
4.1.1	Configurations	9
4.1.2	The postXmlForSubmission Function	9
5	References	11

1 Introduction

1.1 Background

1.1.1 *mTrack*

mTrack is a Health Management Information System(HMIS) used by the Ministry of Health in partnership with UNICEF and WHO to cover disease surveillance, essential medicine and nutrition, strengthening government capacity to collect and analyse and react to real-time information. *mTrack* can be used for mHealth data collection, processing, interpretation and analysis.

1.1.2 DHIS 2

The District Health Information Software – Version 2 (DHIS 2) is Free and Open Source Software (FOSS) HMIS designed and developed under a global research and development initiative (called Health Information Systems Project – HISP) originating from the Department of Informatics, University of Oslo, Norway.

DHIS 2 can be used for mHealth data collection, processing, interpretation and analysis.

1.1.3 Reason of Interoperability

Following the existence a variety of Health Management Information Systems, including *mTrack* and DHIS 2, the ministry of Uganda had to pick from this pool of applications, the primary application for the collection, processing, aggregation and analysis of the nation-wide mHealth data.

DHIS 2 was chosen as the primary system for this purpose, however, other systems like *mTrack* have strengths that need to be embraced to leverage DHIS 2. For instance, *mTrack*'s reliance on SMS from Mobile Phones gives it the ability to collect information even from the most remote places in the country.

It was for such reasons that these two systems had to work together so that we can take advantage of the strengths from each of them. This called for a way of feeding DHIS 2 with data collected by *mTrack*.

2 How the two systems can communicate

2.1 Uniting Components

Both systems(mTrack and DHIS 2) collect information from health centres across the country. This information is entered into predefined reports/forms whose format and fields are provided by the Ministry of Health.

The table below shows the major uniting components and how they are referred to in either system:

#	Component Name	mTrack equivalent	DHIS2 equivalent
1	Health Centres	Health Facilities	Organisation Units
2	Reports	XForms	DataSets
3	Report Fields	XFormFields	DataElements
4	Report Field Values	XFormSubmissionValues	categoryOptionCombos

Table 1: Table for uniting components

2.1.1 Health Facilities and Organisation Units

In mTrack, health facilities are saved as health units that are have associated locations they serve(usually referred to as catchment areas). It is these locations that give us an idea where the health facility falls in the locations hierarchy(that is; the district, health sub-districts, etc).

In DHIS2 we have what is called the organisational hierarchy. The organisational hierarchy defines the organisation structure of the DHIS2 instance, such as how the health facilities, administrative areas and other geographical areas are arranged with respect to each other.

Health facilities(HC II, HC III, HC IV, Hospitals) are the main reporting points for both systems, there is therefore need for uniquely identifying each facility from the rest. The following are used:

- code: uniquely indentifies a health facility in mTrack
- uid: a unique ID that differentiates organisation units in DHIS 2 from the rest

Please note that each component in DHIS 2 has a “uid”that differentiates it from the rest of the components within the same category.

2.1.2 XForms and DataSets

The RapidSMS xforms application provides an interactive web based form builder. Created forms support data being submitted to them via freehand formatted SMS, standard XForm HTTP posts

or structured SMS. mTrack uses XForms and their equivalent counterparts in DHIS2 are Datasets. All data entry in DHIS 2 is organised through the use of datasets. A dataset is a collection of data elements grouped together for data collection and data export between instances of DHIS 2

2.1.3 XFormFields and DataElements

An XForm is composed of one or more ordered fields. Currently four types are supported for each field, integer, decimal, string and coordinate. Each field must have a unique human readable name, as well as a shorter slug used to uniquely identify it. It is the XForm Fields that mTrack uses to signify the various indicators. A typical example is the “Malaria cases” which is part of the “Disease Report” XForm.

DHIS 2 on the other hand uses Data elements for this same purpose. Data elements define what is actually recorded in system, e.g. number of immunisations or number of cases of malaria.

Unique identification in both Systems:

System	Field	Identifier
mTrack	XFormField	slug
DHIS2	DataElement	id

DHIS 2 Data Elements can be further broken down into the following:

- **Data element groups:** these provide a mechanism for classifying related data elements into a common theme.
- **Data element categories:** can be used to disaggregate data elements into individual atomic components. Data element categories are typically a concept, such as Gender, Age or Disease Status. Data elements such as “Number of cases of confirmed malaria” are often broken into smaller component parts to determine, for instance, the number of confirmed malaria cases of particular age groups.
- **Data element category combinations:** These allow multiple data element categories to be combined into a related set. As an example, a data element “Number of new HIV infections” might be disaggregated according to the following categories.

2.1.4 XFormSubmissionValues and CategoryOptionCombos

All the information sent in to the mTrack using Xform submissions is processed and broken down into different XForm submission values for the different XForm fields.

An analogy for this in the DHIS 2 system are the “Data element category combinations”. Each category combination might have one or more options called “*CategoryOptionCombos*”, which will hold the actual values for the indicators.

3 Design of mTrack-DHIS 2 Plugin

3.1 DHIS 2 Web API

DHIS 2 ships with a Web API that complies with the rules of REST (Representational State Transfer) architectural style. The Web API provides a way of accessing the DHIS 2 data model through HTTP calls. For instance you can access a full list of data elements.

The Web API is used for both reading data from and sending data to the DHIS 2 data model. The data can be accessed/retrieved using popular formats for the web such as JSON, HTML, XML, PDF and PNG.

3.1.1 Retrieving Data

To use the WEB API to access data in DHIS 2, you only have to make an HTTP request, of course not forgetting authentication information with in the HTTP request. For example, the following HTTP call would aid you to access all the DataElements in XML format (assuming DHIS 2 is running under tomcat on port 8080):

```
curl -H "Accept: application/xml" -u username:password http://localhost:8080/api/dataElements
```

With the power the API provides, we can therefore read out the important information (from DHIS 2) that we need to use in binding the two systems together that were explained in chapter 2.

3.1.2 Sending data

With the Web API, you can send data values to DHIS 2 from third-party systems like mTrack. You send data values by sending a POST request with the data values in XML format defined by the <http://dhis2.org/schema/dxf/2.0> namespace:

```
<dataValueSet xmlns="http://dhis2.org/schema/dxf/2.0" dataSet="dataSetID"
completeDate="ISODate" period="periodISODate" orgUnit="orgUnitID">
<dataValue dataElement="dataElementID" value="1" />
<dataValue dataElement="dataElementID" value="2" />
<dataValue dataElement="dataElementID" value="3" />
</dataValueSet>
```

Note: We have omitted the *categoryOptionCombo* attribute as it is optional and not needed for this example.

From the example we can see that we need to identify the period, the data set, the org unit (facility) and the data elements for which to report. The dataValueSets resource description tells us that the identifier for monthly periods should be on the format yyyyMM which means that we will use 201201 for January 2012.

The following URL can be used to post the data values(You may replace localhost with the actual DHIS 2 server):
`curl -d @t.xml -u user:pass -H "Content-Type: application/xml" http://localhost:8080/api/dataValueSets`
The example assumes that the XML request has been saved in a file t.xml.

3.2 Data Mapping

Like we explained earlier in chapter 2, we need to clearly map out the different uniting components for both systems. A new database view (all_submission_values_view) and a table(dhis2_mapping) have been created with in the mTrack database, to help with the mapping and therefore make it easy to send the data values.

Both the view and the table have south migrations that can help you create them by issuing the following command:

```
python manage.py migrate dhis2
```

3.2.1 The all_submission_values_view

This view comes from joining the rapidsms_xform application tables and the eav application tables by exposing the relevant fields that are required for posting data values to DHIS2. This view exposes all the values in the xform submissions The fields in this view include the following:
(submissionid, date, keyword, report, name, slug, facility, reporter, phone, has_errors)

- submissionid: The id for the XFormSubmission
- date: The date of the submission
- keyword: The keyword for the XForm (helps to identify which report)
- report: The name of the XForm
- name: The name of the XFormField(read categoryOptionCombo)
- slug: The slug helps to tell which Xform field it is.
- value: The value for the Xformf field identified by the slug
- facility: This holds the *code* of the facility(Organisation Unit) for which the submission is made
- reporter: The name of the reporter for the submission
- phone: The phone number of the reporter
- has_erros: Whether the Xform submission had errors or not

This view can come in very handy should you choose to send the data values periodically(perhaps with cron) as opposed to sending them as soon an Xform submission comes in.

3.2.2 The dhis2_mapping Table

This table does the actual mapping of the important fields in mTrack to their DHIS 2 counterparts. This table includes the following: (name, field_slug, keyword, dhis2_uid, dhis2_dataelement)

- name: The name of the XFormField
- field_slug: The XFormField slug
- Keyword: The keyword for the XForm
- dhis2_uid: The actual UID of the component this xform field matches in DHIS2(In most cases the categoryOptionCombo for the case of the 033B HMIS report)
- dhis2_dataelement: The UID of data Element in DHIS 2 for which xform field would map

Note: This mapping table has to be fed with the actual UIDs in the DHIS 2 system, and if all the xform fields are mapped to their DHIS 2 counterparts, any kind of report can be sent from mTrack to DHIS 2. In other words values can be posted without having to worry about the report to which they belong.

Scripts have been created to automate the mapping and populate the values in these tables. However there is a limit to this automation because the name of the fields to match in both systems may vary since they were captured by different individuals and may not even follow a predefined standard. Better still you can modify these through the django admin site.

4 Implementation

Once the mapping that was discussed in the previous section has been done successfully, it becomes a lot easier for mTrak to “speak” with DHIS 2. All the above ideas were neatly bundled into an application that can be added as a sub-module(rapidsms-dhis2) to your project.

4.1 rapidsms_dhis2 application/submodule

This application has a similar structure to any typical django application and should be fairly easy to add to your project as a git submodule. It has an extra utils.py module which is a toolbox of functions to access and manipulate data values in DHIS 2. The access and modification functions make HTTP requests to DHIS 2 and they require authentication.

4.1.1 Configurations

To effectively make proper HTTP requests, you must have a DHIS2_CONFIG setting in your settings.py.

The DHIS2_CONFIG variable should have the following form:

```
DHIS2_CONFIG = {
'dhis2_url': 'http://localhost:8080/dhis/api/',
  'dhis2_user': 'username',
  'dhis2_passwd': 'password',
  'ctype': 'json', # when reading, json will be easier for python
}
# celery configurations follow
```

4.1.2 The postXmlForSubmission Function

The postXmlForSubmission function takes three arguments, that is; the XFormSubmission, the facility code (Orgunit ID), and the reporting period(YYYYMM). This function returns an XML string that you can post to DHIS 2 to capture all the the data values in the submission.

Once the XML string is returned, it is then posted to the DHIS 2 server using the *send_data(postXml)* function in the utils module.

Posting the data after to DHIS 2 tends to take some time and can significantly slow down the processing of XForm submissions. For this reason, it is better to have the calls to the *send_data* to be run as ‘celery tasks’. Below is a sample tasks.py that you would want to use for your celery instance:

```
from django.conf import settings
from celery.task import Task, task
from celery.registry import tasks
```

```
from dhis2.utils import send_data, postXmlForSubmission

@task
def sendSubmissionToDHIS2(submission, facility_code, week):
    postXml = postXmlForSubmission(submission, facility_code, week)
    try:
        resp = send_data(postXml)
    except:
        pass
```

You would then have to call the `sendSubmissionToDHIS2` task with the proper arguments to pass to `postXmlForSubmission`.

5 References

1. <http://www.dhis2.org/doc/snapshot/en/implementer/html/>
2. <http://www.dhis2.org/doc/snapshot/en/user/html/ch24s03.html>
3. <https://github.com/nyaruka/rapidsms-xforms/tree/master/docs>