Ronak Sutaria
Building methods and solutions which are economical, scalable and compatible with the need for…
Draft

Open Data Kit
android
ODK Collect 1.3 (1030)
Fill Blank Form
Send Finalized Form
Get Blank Form
Delete Saved Form
Drupal™

android/odk and drupal coming together

Automatic and scheduled importing of ODK Data into Drupal 7

Bringing the power of Android, ODK and Drupal together, to help Data reach the Wisdom layer

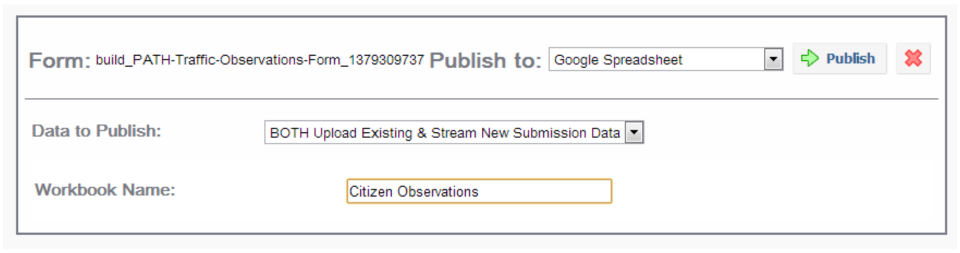
Android and Drupal are among the most powerful free and open source technology platforms available today. Open Data Kit (ODK) has been built by researchers to leverage the power of Android for Mobile Data collection.

This post outlines the steps that can be followed, to automatically and periodically import the ODK Data in to Drupal 7.

· · ·

Step 1.

In the ODK Aggregate, make sure that your ODK form is publishing data to Google Spreadsheets.



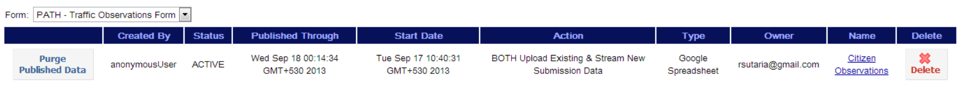
Form: build_PATH-Traffic-Observations-Form_1379309737 Publish to: Google Spreadsheet

Data to Publish: BOTH Upload Existing & Stream New Submission Data

Workbook Name: Citizen Observations

ODK Aggregate—Publish Data Settings

The Google Spreadsheet workbook is created by ODK, and it's link can be seen under the Published Data tab:



Form: PATH - Traffic Observations Form

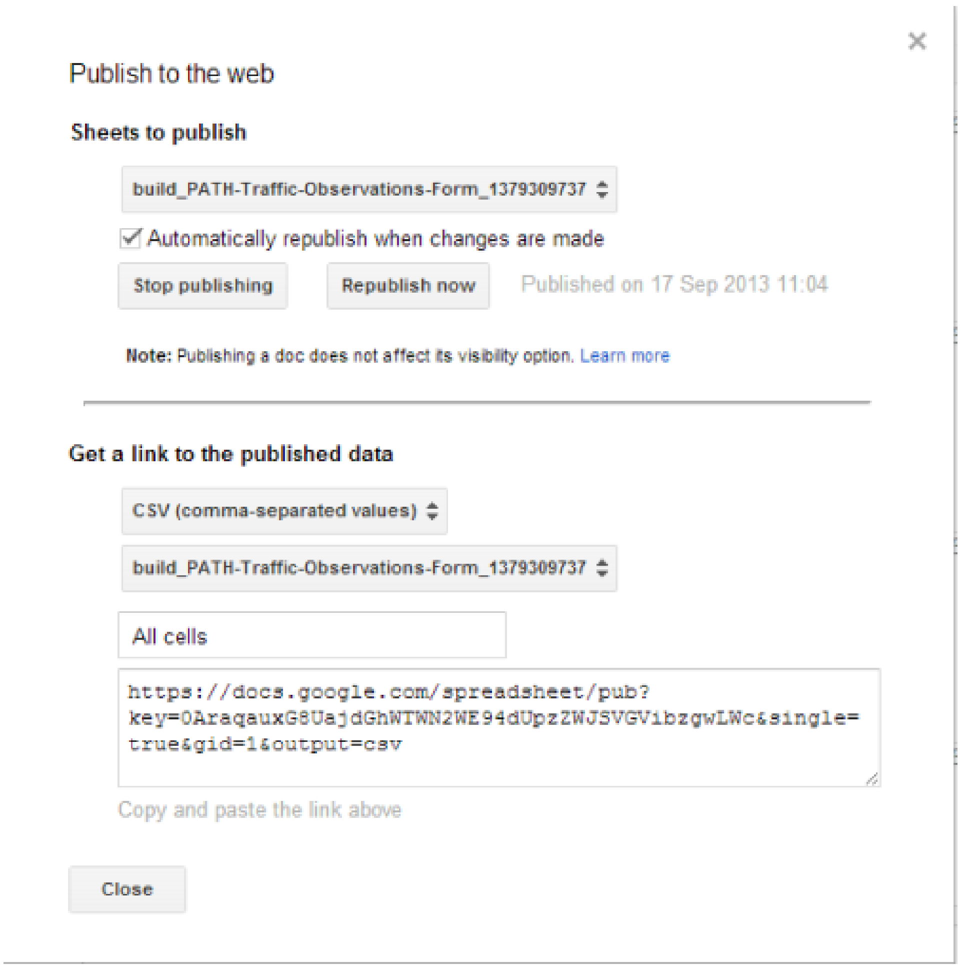
	Created By	Status	Published Through	Start Date	Action	Type	Owner	Name	Delete
Purge Published Data	anonymousUser	ACTIVE	Wed Sep 18 00:14:34 GMT+530 2013	Tue Sep 17 10:40:31 GMT+530 2013	BOTH Upload Existing & Stream New Submission Data	Google Spreadsheet	rsutaria@gmail.com	Citizen Observations	Delete

Link to Google Spreadsheet in ODK Aggregate

Step 2.

In the Google Spreadsheet—go to File -> Publish to the web.

Under "Publish to the web" select the "CSV" option, as follows:


Publish to the web

Sheets to publish

build_PATH-Traffic-Observations-Form_1379309737

✓ Automatically republish when changes are made

Stop publishing | Republish now | Published on 17 Sep 2013 11:04

Note: Publishing a doc does not affect its visibility option. Learn more

Get a link to the published data

CSV (comma-separated values)

build_PATH-Traffic-Observations-Form_1379309737

All cells

```
https://docs.google.com/spreadsheet/pub?
key=0AraqauxG8UajdGhWTWN2WE94dUpzZWJSVGVibzgwLWc&single=
true&gid=1&output=csv
```

Copy and paste the link above

Close

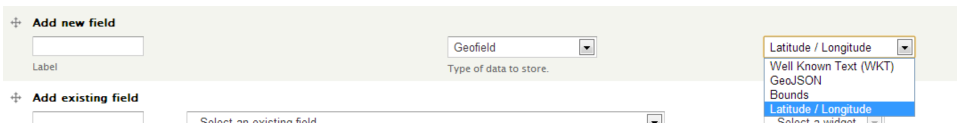
Google Spreadsheet—Publish to the web settings

Step 3.

In your Drupal 7 setup, create a “Content Type” having the fields which you would like to import from your ODK form into your Drupal node.

For the ODK “Image” media data (captured as a Photo or Image in ODK form), a corresponding “Image” field needs to be created in the content type. ODK sends image data as JPEG, so make sure jpg and jpeg extensions are allowed for this Image field.

For the ODK “Location” data (captured via “Record Location” field in ODK form), a corresponding Geofield field needs to be created in the content type. To correctly store Latitude/Longitude in Drupal, you need to install the Geofield module. The Geofield data type can store data in four different formats:


Add new field

Label

Geofield

Type of data to store.

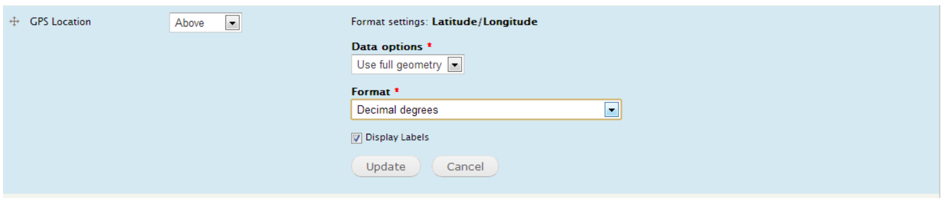
Latitude / Longitude

Well Known Text (WKT)
GeoJSON
Bounds
Latitude / Longitude

Add existing field

Location Data—Geofield settings for node field

ODK “Location” field sends location data as latitude, longitude, altitude and location accuracy. Hence, we will use latitude/longitude as the form element in Drupal.


GPS Location — Above

Format settings: **Latitude/Longitude**

Data options *
Use full geometry

Format *
Decimal degrees

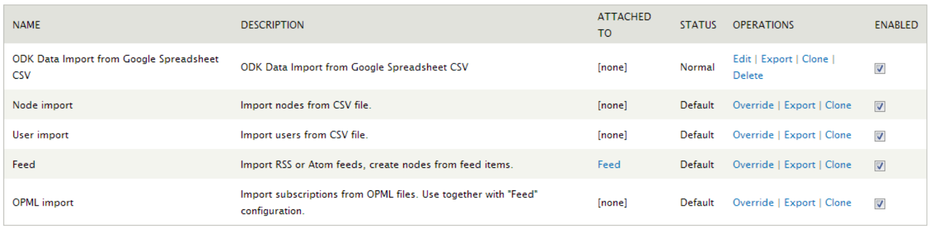
[✓] Display Labels

In the node “Manage Display” option, select “Use full geometry” and “Decimal degrees”.

Step 4.

Now, we will setup the actual ODK data import, which will result in each row of ODK data creating a separate node in Drupal. For this, we will use the amazing Feeds module.

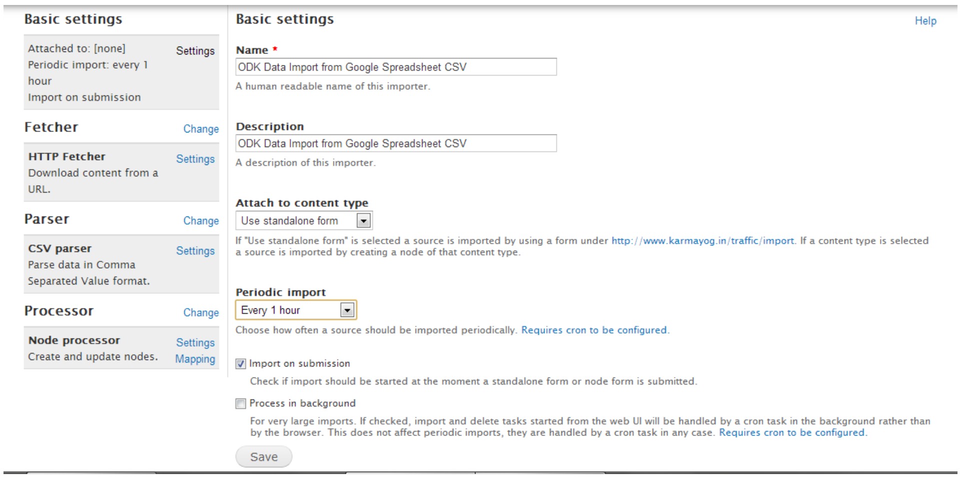
After enabling the Feeds module, click on Structure -> Feeds importer.



NAME	DESCRIPTION	ATTACHED TO	STATUS	OPERATIONS	ENABLED
ODK Data Import from Google Spreadsheet CSV	ODK Data Import from Google Spreadsheet CSV	[none]	Normal	Edit | Export | Clone | Delete	✓
Node import	Import nodes from CSV file.	[none]	Default	Override | Export | Clone	✓
User import	Import users from CSV file.	[none]	Default	Override | Export | Clone	✓
Feed	Import RSS or Atom feeds, create nodes from feed items.	Feed	Default	Override | Export | Clone	✓
OPML import	Import subscriptions from OPML files. Use together with "Feed" configuration.	[none]	Default	Override | Export | Clone	✓

Here, you click on the “Clone” link in the Node import (Import nodes from CSV file). Give a suitable name and description.

Now, click on “Edit” link of the node import that you have setup. You will get a screen as follows:


Basic settings

Attached to: [none]
Periodic import: every 1 hour
Import on submission — Settings

Fetcher — Change

HTTP Fetcher — Settings
Download content from a URL.

Parser — Change

CSV parser — Settings
Parse data in Comma Separated Value format.

Processor — Change

Node processor — Settings, Mapping
Create and update nodes.

Basic settings — Help

Name *
ODK Data Import from Google Spreadsheet CSV
A human readable name of this importer.

Description
ODK Data Import from Google Spreadsheet CSV
A description of this importer.

Attach to content type
Use standalone form
If "Use standalone form" is selected a source is imported by using a form under http://www.karmayog.in/traffic/import. If a content type is selected a source is imported by creating a node of that content type.

Periodic import
Every 1 hour
Choose how often a source should be imported periodically. Requires cron to be configured.

[✓] Import on submission
Check if import should be started at the moment a standalone form or node form is submitted.

[ ] Process in background
For very large imports. If checked, import and delete tasks started from the web UI will be handled by a cron task in the background rather than by the browser. This does not affect periodic imports, they are handled by a cron task in any case. Requires cron to be configured.

Save

Here, the most important settings is the 'Attach to content type', which has to be set to "Use standalone form". You should also setup a Periodic import to a fixed time, depending on how frequently your ODK form posts data. A corresponding cron will need to be setup to match the periodic import time configured here.

Next, click on "Fetcher" and select "HTTP Fetcher" (Download content from a URL).

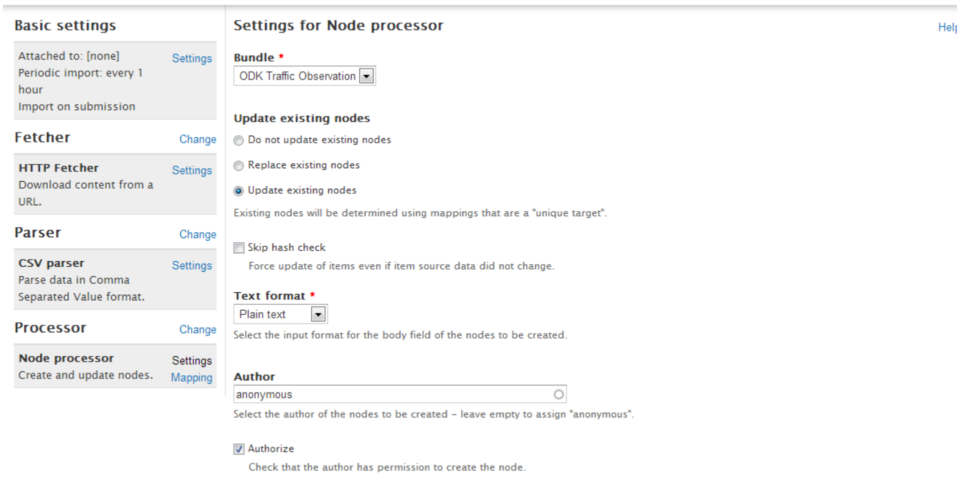
Next, click on "HTTP Fetcher" settings—no changes need to be made here, click on Save.

Next, click on "Parser" change—select the "CSV parser" here.

Next, click on "CSV parser" settings—and keep the default settings here.

Next, click on "Processor" change—select "Node processor" here.

Next, **click on "Node processor" settings.**


Basic settings

Attached to: [none]
Periodic import: every 1 hour
Import on submission — Settings

Fetcher — Change

HTTP Fetcher — Settings
Download content from a URL.

Parser — Change

CSV parser — Settings
Parse data in Comma Separated Value format.

Processor — Change

Node processor — Settings, Mapping
Create and update nodes.

Settings for Node processor — Hel

Bundle *
ODK Traffic Observation

Update existing nodes
( ) Do not update existing nodes
( ) Replace existing nodes
(●) Update existing nodes
Existing nodes will be determined using mappings that are a "unique target".

[ ] Skip hash check
Force update of items even if item source data did not change.

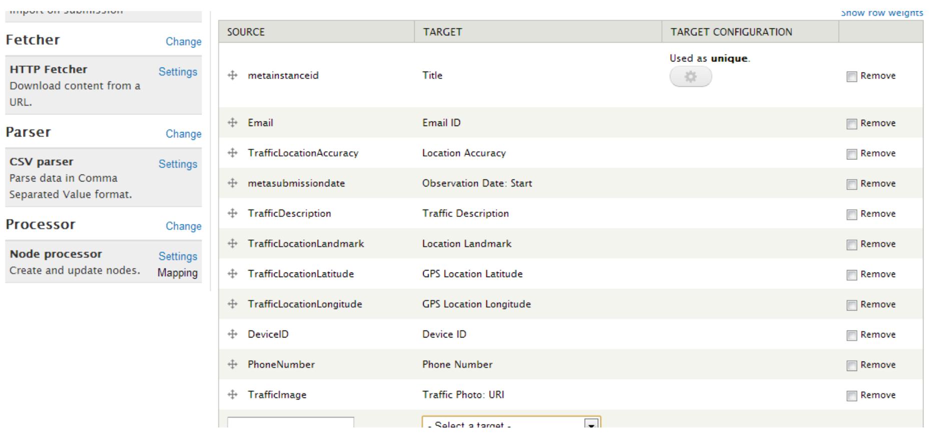
Text format *
Plain text
Select the input format for the body field of the nodes to be created.

Author
anonymous
Select the author of the nodes to be created - leave empty to assign "anonymous".

[✓] Authorize
Check that the author has permission to create the node.

This is an important setting, as here you assign the ‘content type’ you created earlier, to the CSV feed importer. In the bundle drop-down, select the content type which contains all the fields that match with the ODK form fields. You can select “Do not update existing nodes”, and keep the “Skip hash check” unchecked.

Next, **click on “Create and update nodes” Mapping.** This is the setting where you will map the fields from your ODK Google Spreadsheet to the content type fields in Drupal.


Fetcher — Change
HTTP Fetcher — Settings
Download content from a URL.
Parser — Change
CSV parser — Settings
Parse data in Comma Separated Value format.
Processor — Change
Node processor — Settings, Mapping
Create and update nodes.

SOURCE	TARGET	TARGET CONFIGURATION	
metainstanceid	Title	Used as **unique**.	Remove
Email	Email ID		Remove
TrafficLocationAccuracy	Location Accuracy		Remove
metasubmissiondate	Observation Date: Start		Remove
TrafficDescription	Traffic Description		Remove
TrafficLocationLandmark	Location Landmark		Remove
TrafficLocationLatitude	GPS Location Latitude		Remove
TrafficLocationLongitude	GPS Location Longitude		Remove
DeviceID	Device ID		Remove
PhoneNumber	Phone Number		Remove
TrafficImage	Traffic Photo: URI		Remove

In the Source field, give the name of the Spreadsheet headers and in the target give the corresponding content type fields.

In ODK, the metainstanceid is unique and in Drupal the Title field is unique, hence those are mapped to each other.

A particular field to take note of is the Latitude and Longitude. In Drupal, two sub-fields are automatically created for each Geofield data. In this case, GPS Location was the Geofield in Drupal, hence the two sub-fields created are GPS Location Latitude and GPS Location Longitude. These are mapped to the ODK Location fields—Latitude and Longitude.

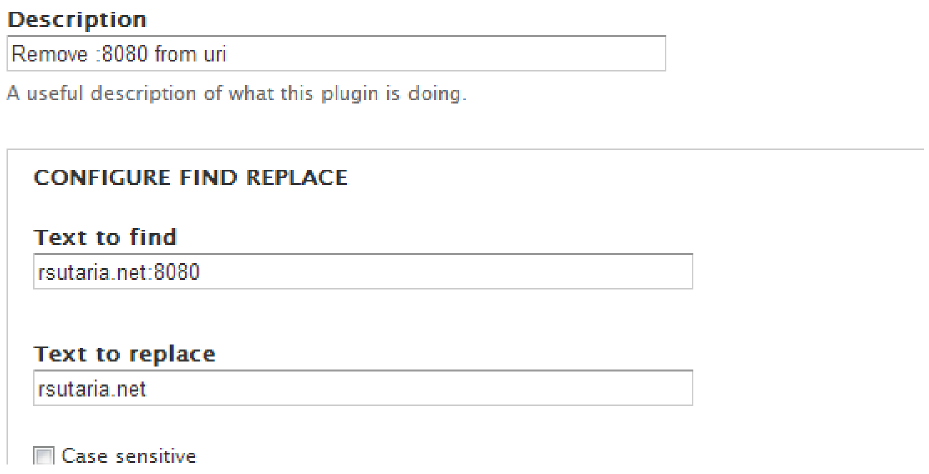
For the ODK Media “Image” field—the target field is the “Traffic Photo: URI” field.

***(Conditional)* Step 4b.**

If you have installed ODK Aggregate on Tomcat (maybe on a 64-bit Linux EC2 instance) and not on Google Appengine, the Image URL posted by ODK in the Google Spreadsheet contains the Tomcat port 8080. Hence, your Image URL will look like—http://<odk-aggregate-hostname>:8080/url-to-image….

A weird quirk (or bug) of the Drupal Feeds module is that it cannot process/import Images when the URL contains a port number. To overcome this problem, you will need to install the Drupal Feeds Tamper module, available here: https://drupal.org/project/feeds_tamper

On enabling this module, you will get a "Tamper" link on your Feeds Importer screen. On clicking the Tamper link, you will get a screen to "Add plugin" (which is basically a rule) for the Image URI field. Here, you can add a "Find and Replace" plugin, where you can give the string <your-odk-tomcat-hostname>:8080 in the Find field and <your-odk-tomcat-hostname> in the Replace field.


Description
Remove :8080 from uri
A useful description of what this plugin is doing.

CONFIGURE FIND REPLACE

Text to find
rsutaria.net:8080

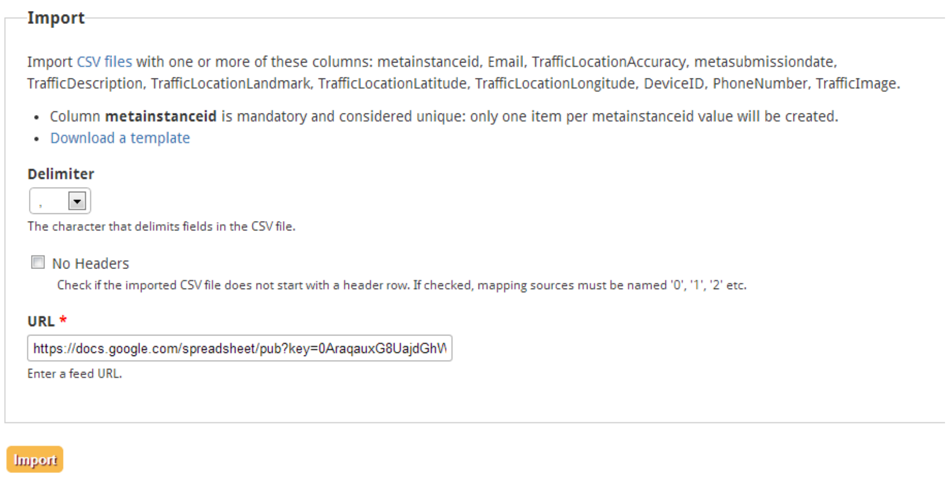
Text to replace
rsutaria.net

☐ Case sensitive

To make this setting work correctly, you need to make sure that your ODK Tomcat can function on both the port 80 and port 8080, and that port 80 is setup as a proxyPort for 8080.

Step 5.

To start importing the actual ODK data, go to the URL http://<drupal-url>/import. Here, you will see the CSV Importer listed, which was created in the above Step 4. On clicking that, you will get an option to enter the CSV Feed URL:


Import

Import CSV files with one or more of these columns: metainstanceid, Email, TrafficLocationAccuracy, metasubmissiondate, TrafficDescription, TrafficLocationLandmark, TrafficLocationLatitude, TrafficLocationLongitude, DeviceID, PhoneNumber, TrafficImage.

- Column **metainstanceid** is mandatory and considered unique: only one item per metainstanceid value will be created.
- Download a template

Delimiter

,

The character that delimits fields in the CSV file.

☐ No Headers

Check if the imported CSV file does not start with a header row. If checked, mapping sources must be named '0', '1', '2' etc.

URL *

https://docs.google.com/spreadsheet/pub?key=0AraqauxG8UajdGhW

Enter a feed URL.

CSV Feed Import URL settings

The URL to enter here is found in the Step 2 (shown in the screenshot*Google Spreadsheet—Publish to the web setting)*.

Now, Click on "Import"

If you have reached so far, time to congratulate yourself and have a nice, hot cup of tea :-)

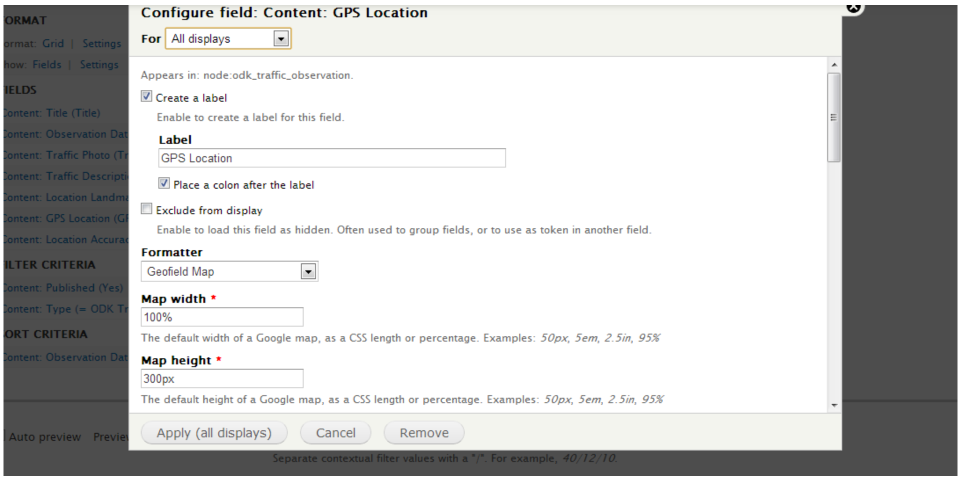
On successful import, you will get a message in the Status section saying "X imported items total".

You can verify the imported nodes by clicking on "Content". All the new nodes, imported from the CSV, will be listed here.

Step 6.

To view the newly imported nodes, a Drupal "View" can be created from Structure -> Views.

The Geofield module provides the feature to automatically show a Map of the location co-ordinates:


ORMAT
ormat: Grid | Settings
how: Fields | Settings
IELDS
Content: Title (Title)
Content: Observation Dat
Content: Traffic Photo (Tr
Content: Traffic Descripti
Content: Location Landm
Content: GPS Location (G
Content: Location Accura
ILTER CRITERIA
Content: Published (Yes)
Content: Type (= ODK Tr
ORT CRITERIA
Content: Observation Dat

Configure field: Content: GPS Location

For: All displays

Appears in: node:odk_traffic_observation.

☑ Create a label
Enable to create a label for this field.

Label
GPS Location

☑ Place a colon after the label

☐ Exclude from display
Enable to load this field as hidden. Often used to group fields, or to use as token in another field.

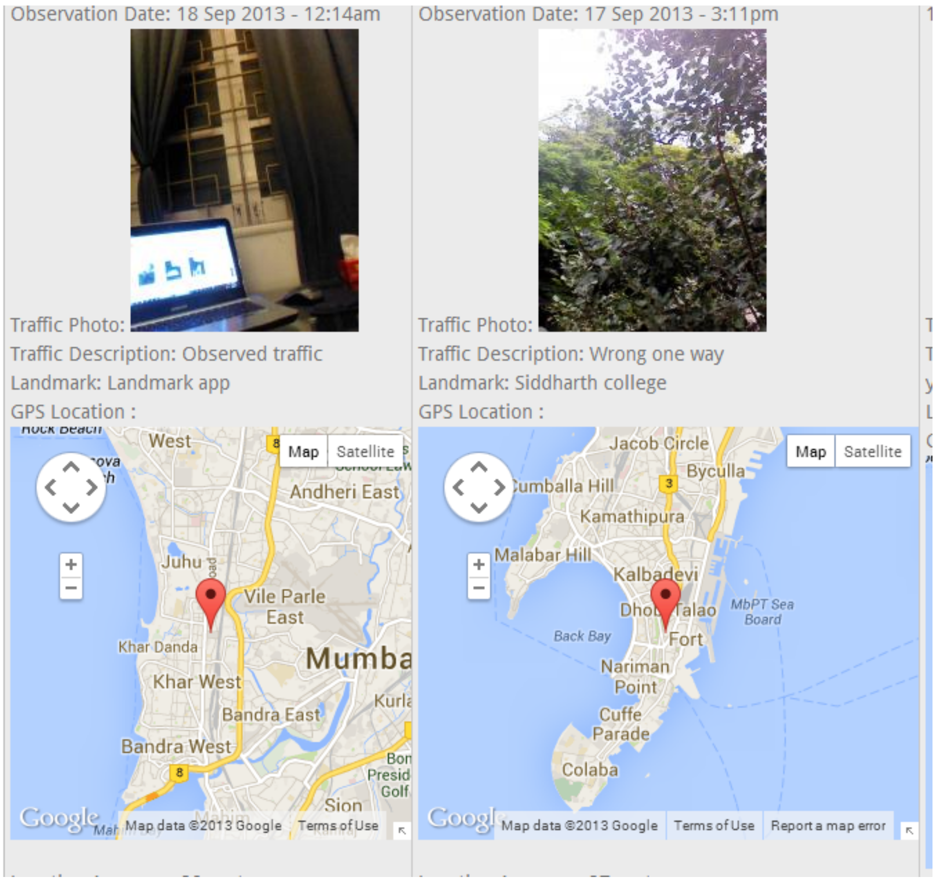
Formatter
Geofield Map

Map width *
100%
The default width of a Google map, as a CSS length or percentage. Examples: *50px, 5em, 2.5in, 95%*

Map height *
300px
The default height of a Google map, as a CSS length or percentage. Examples: *50px, 5em, 2.5in, 95%*

Separate contextual filter values with a "/". For example, *40/12/10*.

In the Formatter, select “Geofield Map”. You can also select Latitude/Longitude from the drop-down, if you would like to list the actual values instead of the Map.


Observation Date: 18 Sep 2013 - 12:14am
Traffic Photo:
Traffic Description: Observed traffic
Landmark: Landmark app
GPS Location :

Observation Date: 17 Sep 2013 - 3:11pm
Traffic Photo:
Traffic Description: Wrong one way
Landmark: Siddharth college
GPS Location :

The Drupal view will automatically show the Location of the GPS Latitude/Longitude captured via ODK on the Google Map in Drupal. The layout of the view can be made better by css/theme modifications.

Once the ODK data has been converted into Drupal nodes, powerful workflows can be built around that Data using the various other free Drupal modules.

. . .

Make sure to configure a Cron job on your server, to periodically import the ODK data in to your Drupal system.

. . .

Please feel free to leave comments and suggestions on this page, by hovering over the right margin and clicking on the “+” sign next to any paragraph.