

Connecting Ushahidi data to the HDX repository

I've been talking to the HDX team for some time now (well, since before HDX was a thing, but then so have many of us). [HDX](#) is a [data repository for humanitarian data](#): basically, it's a place to put machine- and human-readable datasets so that other people can use them too.

Ushahidi tools (Ushahidi platform instances, Crowdmap instances) often have datasets in them that could be useful to other people, so part of the conversation has been about how to share data from Ushahidi sites, both on the HDX site and in the HXL humanitarian standard.

Ushahidi CSV to HDX CSV

First let's look at how to share the CSV file that Ushahidi creates when you click on the "download reports" button. Before you do this, please, please, please read my post about mitigating potential risks to people from sharing your Ushahidi data.

Converting that CSV into HXL format is pretty easy: it's as simple as adding a row to your CSV file. Under the heading row ("#", "INCIDENT TITLE", "INCIDENT DATE", "LOCATION", "DESCRIPTION", "CATEGORY", "APPROVED", "VERIFIED", "LATITUDE", "LONGITUDE"), add another row with the HXL tags (#x_uniqueid, #x_reporttitle, #report_date, #loc, #x_description, #x_category, #x_approved, #x_verified, #lat_deg, #lon_deg). Erm... that's it.

Loading the CSV file into HDX is also pretty easy: you go to HDX, create an account (and an organisation, if you need to), then click on the 'upload data' button and follow the instructions.

Ushahidi API to HDX CSV

[I did this recently](#) for some Ushahidi location data (I used the API to pull out all latitude-longitude-placename rows from the Ushahidi instances used for the last Philippines typhoons, then wrote them to a CSV which I cleaned up a bit before posting manually to HDX).

If you can write (or [run](#)) Python, you can access a Ushahidi Platform API and get all the reports (or locations or categories etc) from an Ushahidi Platform instance - **IF** that instance has turned on its API (most have). This allows you to do more complex things, like get data from multiple Ushahidi instances, process it a bit (e.g. removing all those repeat locations) or merge it with other data (e.g. [Pcodes](#)).

Geeky bit (non-coders, look away now): This API to HDX transfer could be done as a 1-click button on the reports page, if we add an "HDX" plugin to the Ushahidi Platform. The design for the plugin is basically: a) allow users to add their HDX settings using the plugin's "settings" page, and b) add a button to the reports page, and use the existing Ushahidi CSV creation code to create a CSV

file, then push that file through the HDX API using the settings that the user's already added. This would be a pretty sweet Ushahidi plugin for a community coder to write: it's well-defined, and most of the code needed is already there.

Ushahidi API to HDX API call

HDX not only allows you to upload datasets, it also allows you to add a link to an API, so HDX sees the current information from a site (not a snapshot that you've turned into a CSV file). You can do this by adding one of the [Ushahidi API calls](#) to the HDX "link" field, e.g. if your site is called [www.yoursite.com](#),

- <https://www.yoursite.com/api?task=incidents&limit=1000> - for reports (NB this only grabs the first 1000 reports)
- <https://www.yoursite.com/api?task=categories> - for the categories list
- <https://www.yoursite.com/api?task=locations> - for the list of locations

Again, please please please read my notes about mitigating potential risks to people from sharing your Ushahidi data before doing any of these.

HDX to Ushahidi

The HDX datastores include files in CSV, Json, XML and Shapefile formats. You can import Ushahidi reports data from HDX into an Ushahidi instance (but be sure you have permission to do this before you do); you can also [upload shapefiles as layers](#) (converted to WMS or KML format: the [ogr2ogr tool](#) can do this for you) into an Ushahidi site.

Helping out with HDX and Ushahidi

HDX is an open-source project (<https://github.com/OCHA-DAP/>): if you're a Python coder, please consider helping out with its codebase.

The HDX team are also working on unlocking useful data that's still in PDFs. If you want a small task to help them with that, try this: [unlocking humanitarian planning data](#).

The Ushahidi Platform is also open-source (<http://github.com/ushahidi/>): if you're a Php coder, please consider helping out there too. The plugin specification above could be a nice place to start...