A quick-and-dirty API for data.un.org

An API is a piece of software on a website that allows programs on other websites and machines to access data held on that website. This means that coders can create programs that use third-party data (from e.g. Facebook, LinkedIn, Foursquare, the World Bank etc) as part of their applications. An example API is data.gov's broadband map API - an example call to which looks like "http://www.broadbandmap.gov/broadbandmap/provider?format=json". This provides data about broadband providers in a standard data format (json) to any program or application that needs it.

Without an API, datasite users are forced to access data through a series of mouseclicks leading eventually to either a webpage displaying the data, or to a file containing data that they can download.

Data.un.org does not have an API. Yet. But it would be very very easy to provide on their website. A trace of the calls made during a data.un.org file retrieval shows that the standard format for most of the datasets on data.un.org (the exception is datasets whose martids are digits) is as follows:

- Getting the ids needed to access a datafile:
  - The list of marts (agencies that provided data to data.un.org) is in a json file(http://data.un.org/Handlers/ExplorerHandler.ashx?t=marts) whose nodes all have the form martName, martId, childNodes; for example martName="Commodity Trade Statistics Database", martId="ComTrade", childNodes="".
  - The list of datafiles for each mart is in a json file (http://data.un.org/Handlers/ExplorerHandler.ashx?m=martId) where martId is the martId given above, e.g. http://data.un.org/Handlers/ExplorerHandler.ashx?m=ComTrade The important nodes in this file are: label, childNodes, martId, dataFilter, where dataFilter is the id for the datafile connected to that label. An example is label="Trade of good, US$, HS 1992, All commodities", martId="ComTrade", dataFilter="_l1Code%3a1" (these are really not intuitive names).
  - For almost all marts, the martId that you need to access a datafile is the same as the martId used to obtain the list above. The exception is martId="KI". This mart doesn't correspond to one agency, but is an aggregate of datasets provided by different agencies. The dataFilter for a KI file looks like this: "dataSetID%3aPopDiv%3bvariableId%3a12", where PopDiv is the true martId and variableId%3a12 is the true filterId. These "true" values can be used to access the datafile as described below.
- Accessing a datafile:
  - The description of a dataset is at URL http://data.un.org/Data.aspx?d=martId&f=dataFilter, e.g. http://data.un.org/Data.aspx?d=ComTrade&f=_l1Code%3a1. This won't download the file for you, but does give you a useful reference to the dataset that you're

downloading.

- Each datafile with a numerical martId (e.g. martId="10") can be accessed by deconstructing the dataFilter id, which has the form "docId:filenumber" (e.g. the dataFilter id for "Net Bilateral ODA, USD Millions" is "docID:164", i.e. filenumber is 164). At the moment, these are all Excel datafiles, and each of them can be accessed through the URL http://data.un.org/Handlers/DocumentDownloadHandler.ashx?t=bin&id=filenumber, e.g. http://data.un.org/Handlers/DocumentDownloadHandler.ashx?t=bin&id=164
- All other datafiles are available in 4 formats: XML, CSV, pipe-separated and semicolon-separated files. These can be accessed directly using the URL http://data.un.org/Handlers/DownloadHandler.ashx?DataMartId=martId&DataFilter=dataFilter&Format=format, where format is one of "csv", "xml", "psv" or "scsv", e.g. http://data.un.org/Handlers/DownloadHandler.ashx?DataMartId=Comtrade&DataFilter=_l1Code%3a1&Format=csv will download the CSV version of the Commodity Trade Statistics Database dataset "Trade of good, US$, HS 1992, All commodities".
- Finally, all the datafiles are downloaded wrapped in a .zip file whose name doesn't bear any resemblance to the dataset itself, e.g. "file UNdata_Export_20120619_164851092.csv in zip file UNdata_Export_20120619_164851092.zip" for a file downloaded on 2012-06-19 (19th June 2012). Most modern web languages have zipfile-handling libraries that can easily handle this.

In summary, I've described how to access a data.un.org datafile directly, using only the calls already provided by this website. Propagating this information and potentially also using this information to add an API line to each description file in data.un.org is a trivial exercise that will make a large difference to coders' ability to include UN data in their applications and websites.

I've attached two files that will help anyone who can't do the "getting the ids" part of these instructions to just go straight to the "accessing a datafile" ones.

This is part of a 3-note set: the api instructions, correcting GIS references in the data.un.org dataset and accessing the UN's other online data. This should be enough to get people using static UN data (as in not datastreams like Twitter feeds, map reports etc) more widely.

Addendum: as I visualized GIS data for the data.un.org dataset, some of the counts I was getting for countries seemed a bit off. For instance, countries (Australia, Canada) with names at the start of the alphabet had much larger counts than countries (e.g. Yemen) at the end of the alphabet. I at first put this down to collection bias (e.g. data on developed countries was much easier to obtain than data on developing ones), but decided to investigate why Australia had 50000 references in

the data. The answer is that we need to do a bit more work on the API (not impossible: there's a filter value that can be used to select blocks of data, and the limit is also variable). UNStats has limited the downloads from its site to no more than 50000 rows of each dataset, but there are several datamarts with datafiles that exceed these limits (as in each datafile contains 1000s of values per country), with the Commodity Trade Statistics Database, from UNSTATS (code: COMTrade) being a consistent offender.

Basically, if your data comes back with approximately 50000 rows in it, then you probably don't have the whole dataset, and anything over 40000 rows is suspect. I tried looking at the filesize as a quick rule of thumb, i.e. filesizes between 5Mb and 6Mb might be affected, filesizes above 6Mb are definitely affected, but when I compared filesize against number of rows, there were smaller files (around 3Mb) that had the same problem. If you're downloading UNstats data, you need to be aware of this issue, but you probably won't see it very often: the number of rows per dataset is a long-tail distribution (see the figure above), and only 159 files out of 7000+ have more than 40000 lines in them.