

GIPO Tool API

Index:

[Introduction](#)

[API Usage](#)

[Search Parameters](#)

[Code sample](#)

[Performance and Security](#)

[Reporting Bugs and Requesting Enhancements](#)

Introduction

The acronym “API” stands for “Application Programming Interface”. The GIPO Tool offers a REST API to programmatically access the search server.

Programmers use the API to make applications, websites, widgets, and other projects that interact with GIPO Tool contents. Users can query it via HTTP GET and receive results in XML (default), JavaScript Object Notation (JSON), or CSV format among other formats.

This document provides a brief overview of access to the server and use of its API.

API Usage

Search Parameters

Queries are executed by creating a URL that contains all the appropriate query parameters (although some client applications may hide this detail to users).

Note that many characters in the query syntax are special characters in URLs, so when constructing request URLs manually, you must properly URL-Encode¹ these characters.

q - main query

For instance this URL will respond with all items:

[http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=*](http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=)

GIPO’s Tool server returns results in XML format by default. There are some online tools² you can use to beautify/format the XML or display it in tree view. The <result> element contains a series of <doc> elements (10 if the rows parameter is not specified) that represent the search results.

start - zero-based hit starting point. This parameter defines the count of document from which you are interested. This parameter helps in pagination of results.

¹ HTML URL Encoding Reference http://www.w3schools.com/tags/ref_urlencode.asp

² XML viewer from Code Beautify: <http://codebeautify.org/xmlviewer>
XML viewer from Text Tools: <http://countwordsfree.com/xmlviewer>

rows - maximum number of "hits" to return. If not specified, the default value is "10".

http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=*&start=100&rows=50

If the start is 100 and rows is 50, the result XML will have 50 documents starting from 100th result.

fl - comma-separated field list

http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=*&rows=20&fl=tstamp,title

GIPO Field List:

API Field Name	Common Name	Type	Description
title	Title	string	The descriptive heading of the original item.
url	URL	string	The Web address of the original item in its source.
lang	Language	string	A two-letter code (en, es, fr, etc.) representing the name of the language of the item. This field follows the ISO 639-1 standard ³ that defines identification codes for most of the major languages of the world.
gipo_aspects	Issues	array of strings	A list of Internet governance issues under which the item has been classified.
tags	Tags	array of strings	A list of keywords or relevant terms extracted from the description of the item. They usually correspond to companies, people, places, topics, etc.
gipo_world_regions	World Regions	array of strings	A list of countries, continents or geographic areas mentioned in the description of the item.
tstamp	Date	date	Combined date and time in UTC when the item was collected by the tool. Represented by concatenating a complete date expression, the letter T as a delimiter, and a valid time expression. For example, "2016-01-29T15:17:48Z".

hl - set to "true" enables highlighted snippets to be generated in the query response

³ Language codes - ISO 639 http://www.iso.org/iso/home/standards/language_codes.htm

hl.fl - comma-separated field list for which to generate highlighted snippets

API Field Name	Common Name	Type	Description
content_XX <i>[where XX is an ISO 639-1 code]</i>	Description	string	<p>A snippet of text from the original publication or document.</p> <p>Different fields are used for every language.</p> <p><i>The default field to use will be the one whose name results from joining the value of the lang field at the end of the "content_" string. If lang field value is "unknown" the content_en field should be used.</i></p> <p><i>For example: if lang field value is either "en" or "unknown", the content_en field will have the description of the item in English language.</i></p>

For example:

http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=ipv6&fl=title&hl=true&hl.fl=content_en

fq - filter queries (multiple values supported) that are additional constraints to satisfy besides the query

For example the following URL will respond with all items, except sourced from Twitter, that have been classified under Legal issue and where Uganda is mentioned:

http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=-Tweeted&fq=gipo_aspects:%22Legal%22&fq=gipo_world_regions:uganda

wt - writer type - format of Solr response

These are the most common settings⁴ for the wt parameter: csv, json, php, phps, python, ruby, smile, velocity, xml, xslt

http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=*&rows=100&wt=json

sort - specify sort criteria either by field(s) or function(s) in ascending (asc) or descending (desc) order

http://observatory.giponet.org/api/solr-proxy/rest/proxy/select?q=*&rows=20&fl=tstamp,title&sort=tstamp%20asc

⁴ The available response formats are fully documented in Apache Solr Reference Guide - Response Writers <https://cwiki.apache.org/confluence/display/solr/Response+Writers>

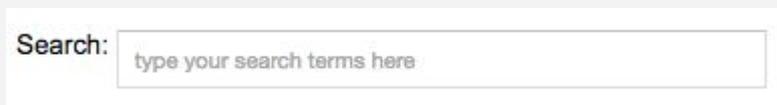
Code sample

This is just an example form created using HTML. It illustrates an extremely simple front-end for the API call. The GET method sends the data as part of the URL.

This code:

```
<form method="get" action="http://observatory.giponet.org/api/solr-proxy/rest/proxy/select">
  <label>
    <span>Search: </span>
    <input name="q" size="50" maxlength="200" value="" type="text" placeholder="type your search terms here">
    <input name="rows" value="20" type="hidden">
    <input name="fl" value="tstamp,title,lang,url,gipo_aspects,tags,gipo_world_regions" type="hidden">
    <input name="fq" value="gipo_world_regions:africa" type="hidden">
    <input name="wt" value="xml" type="hidden">
  </label>
</form>
```

produces this search box:



For example searching for “access” term will generate a response similar to this XML code:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">0</int>
    <lst name="params">
      <str name="q">access</str>
      <str name="fl">tstamp,title,lang,url,gipo_aspects,tags,gipo_world_regions</str>
      <str name="fq">gipo_world_regions:africa</str>
      <str name="rows">3</str>
      <str name="wt">xml</str>
    </lst>
  </lst>
  <result name="response" numFound="94" start="0">
    <doc>
      <str
name="url">http://www.sbs.ox.ac.uk/cybersecurity-capacity/content/lifting-barriers-internet-development-africa-suggestions-improving-connectivity</str>
      <str name="title">Lifting Barriers to Internet Development in Africa: Suggestions for Improving Connectivity</str>
      <date name="tstamp">2016-03-06T15:28:00Z</date>
      <arr name="gipo_world_regions">
        <str>africa</str>
      </arr>
      <arr name="tags">
        <str>broadband</str>
        <str>telegraphy</str>
      </arr>
      <str name="lang">en</str>
    </doc>
    <doc>
      <str name="url">http://www.techzim.co.zw/2016/04/multichoice-take-legal-action-subscribers-operators-illegally-accessing-dstv-south-africa/</str>
      <str name="title">MultiChoice to take legal action against subscribers & operators involved in access of DStv South Africa</str>
      <date name="tstamp">2016-04-28T10:52:09Z</date>
      <arr name="tags">
        <str>television in south africa</str>
        <str>dstv</str>
        <str>multichoice</str>
      </arr>
    </doc>
  </result>
</response>
```

```
</arr>
<str name="lang">en</str>
<arr name="gipo_world_regions">
  <str>south africa</str>
  <str>africa</str>
</arr>
</doc>
<doc>
  <str name="url">http://www.itnewsafrika.com/2016/05/airtel-and-facebook-to-bring-basic-internet-services-to-nigerians/</str>
  <str name="title">Airtel and Facebook to bring basic Internet services to Nigerians</str>
  <date name="timestamp">2016-05-11T06:00:07Z</date>
  <arr name="gipo_aspects">
    <str>Development</str>
  </arr>
  <arr name="tags">
    <str>facebook</str>
    <str>bse sensex</str>
    <str>cnx nifty</str>
    <str>airtel africa</str>
    <str>national broadband plan</str>
  </arr>
  <str name="lang">en</str>
  <arr name="gipo_world_regions">
    <str>rwanda</str>
    <str>kenya</str>
    <str>zambia</str>
    <str>malawi</str>
    <str>nigeria</str>
    <str>africa</str>
    <str>ghana</str>
  </arr>
</doc>
</result>
</response>
```

This example shows only the raw request and response. Your client code should be the one launching the request and processing the response in a way that suits your needs.

Performance and Security

Applications that make use of the API should be designed to the extent possible to reduce load on the index. We recommend that you cache API responses in your application or on your site if you expect high-volume usage.

For example, don't try to call the API on every page load of your website. Instead, call the API from time to time and save the response to your local server, displaying your cached version on your site.

Applications must take into account best practices such as the filtering of queries passed to the index in order to prevent malicious access; this filter code may provide a convenient point at which to add a cache.

Reporting Bugs and Requesting Enhancements

Before filing an issue to report a bug please examine the HTTP response closely: the status code you received back, the message that accompanies that code, the HTTP headers in the response, and the response body itself.

If you are still seeing problems you can send us a message via contact form (<http://observatory.giponet.org/contact>) but make sure that what you're filing is a material issue with the platform.

File an issue with as exhaustive detail as possible to reproduce your problem. Make sure to include the exact URLs you're executing, the exact GET bodies & HTTP headers you're sending and any other relevant details.

The procedure for making an enhancement request (to add a missing feature) is nearly the same as the procedure for reporting a bug. Just mark your message as an "enhancement" in some part of it.