

Globodox REST API

(Help Manual)

Table of Contents

Foreword	0
Part I Introduction to REST API	3
1 Working With Globodox REST API.....	4
2 Login to Globodox.....	5
3 Getting a List of DBs.....	7
4 Getting Stack\Document type List.....	9
5 Getting List of Stack Items.....	12
6 Getting a count of Documents related to a Stack Item.....	14
7 Getting details for Documents related to a Stack Item.....	16
8 Download a Document.....	19
9 Upload a Document.....	21
10 Log out from Globodox.....	23
11 Response Status Codes.....	24
Part II Advanced Topic	25
1 How to get JSON output instead of XML.....	25
Index	0

1 Introduction to REST API

After going through this section you will have a basic understanding of REST API.

REST calls are made via the HTTP protocol just like calls made during a web browsing session. So when you type `www.itaz.com/index.htm` in your web browser, the browser makes an HTTP call to the `itaz.com` server and requests for a page called `index.htm`. The data returned by the `itaz.com` server is HTML text which the browser parses, interprets and displays on the screen.

In a normal web browsing scenario, the web server will respond with the HTML. But in a REST API scenario, the web server responds with data in the form of XML or JSON. That is the main difference. So if you type the REST API URL in a web browser, the browser will show you XML or JSON.

To use the REST API in any programming language, the developer must know the following...

- 1) How to make an HTTP call from their application?
- 2) How to receive the XML or JSON response that the server sends?
- 3) How to parse and understand the received XML or JSON data?

If you know how to use any REST based API from any programming language on any operating system then you can talk to the Globodox REST API.

1.1 Working With Globodox REST API

We chose to create a REST based API for Globodox because it is the best way to do a cross platform API. It can be called from any programming language from any operating system including Windows, Mac and Linux. The Globodox REST API needs Windows and IIS to be installed **only** on one server machine (on the same LAN or even on the Internet). All the client machines can be run on any operating system. In order to communicate with Globodox REST API a user will need to make following calls.

Actions	
Login to Globodox	Getting a List of Globodox DBs
Getting Globodox Stack\Document type List	Getting List of Globodox Stack Items
Getting Document count in Globodox Stack	Getting Documents related to a Stack Item
Download a Document	Upload a Document
Log out from Globodox	Response Status Codes

1.2 Login to Globodox

Description

Returns an authentication session token. The token is used as an identity of the client to avoid revalidation of credentials for subsequent REST requests made by the client for that session.

URL:

```
POST /api/signin/post.ashx
```

Header Parameters:

Parameter Name	Description
Username	The Globodox user name you want to login as
Password	The password for the Globodox user
AccessLevel	Use any one of the following values: <ul style="list-style-type: none">• <i>default</i> - To let Globodox decide the access level as specified in the user's profile• <i>viewonly</i> - To login with view-only access• <i>full</i> - To login with full access
RememberMe	true or false
Action	Specifies what action to take. For now specify this as "Signin".

Response:

Response Type	Description
On Success	A <i>Consumer Token</i> is returned.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpWebRequest request = WebRequest.Create("http://localhost/api/signin/post.ashx") as
HttpWebRequest;
request.Method = "POST";

//set the required header parameter
request.Headers.Add("Username", "user1");
request.Headers.Add("Password", "password");
request.Headers.Add("AccessLevel", "default");
request.Headers.Add("RememberMe", "false");
request.Headers.Add("Action", "Signin");

// Get response

try
{
    HttpWebResponse response = request.GetResponse() as HttpWebResponse;
    {
        string sToken = string.Empty;

        // Get the authentication token
        sToken = response.Headers["Consumer-Token"];

        // Consumer token will be considered as authentication token used as identity of the
        client.
    }
    catch(Exception poErr)
    {
        MessageBox.Show(poErr.Message);
    }
}
```

1.3 Getting a List of DBs

Description

Returns a list of Globodox DBs that the currently logged in user is allowed to see.

URL:

```
GET /api/database/all/full/get.ashx
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

Response:

Response Type	Description
On Success	List of Globodox DBs will be returned in the XML format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

```
C#  
  
// Create the web request  
HttpWebRequest request = WebRequest.Create("http://localhost/api/database/all/full/get.ashx"  
    ) as HttpWebRequest;  
  
request.Method = "GET";  
  
//set the required header parameters  
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call  
to login  
  
// Get response  
  
string lsResponse = string.Empty;  
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)  
{  
    // Get the response stream  
    StreamReader reader = new StreamReader(response.GetResponseStream());  
    lsResponse = reader.ReadToEnd();  
}  
  
//The response can now be used to populate a list
```

XML Sample Output:

On successful execution of the request following XML text will be received showing the Globodox Databases.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <databases>
    <database>
      <oid>4</oid>
      <name>Db2.gxdb</name>
      <displayname>Db2</displayname>
      <type id="0">dbAccess</type>
    </database>
    <database>
      <oid>3</oid>
      <name>DB1.gxdb</name>
      <displayname>DB1</displayname>
      <type id="0">dbAccess</type>
    </database>
    <database>
      <oid>2</oid>
      <name>Sample DB.gxdb</name>
      <displayname>Sample DB</displayname>
      <type id="0">dbAccess</type>
    </database>
    <database>
      <oid>1</oid>
      <name>MainDB.gxdb</name>
      <displayname>MainDB</displayname>
      <type id="0">dbAccess</type>
    </database>
  </databases>
  <summary listtype="databases" start-index="1" max-results="10" returned="4" />
</response>
```


1.4 Getting Stack\Document type List

Description

Returns a list of Stack\Document types in Globodox.

URL:

```
GET /api/entitymetadata/{db}/all/full/get.ashx?entitytype={entitytype}&iscustomentity={iscustomentity}
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

URL Parameters:

Parameter Name	Description
db	Value of the oid of the database received after getting the list of Globodox databases.

Query Parameters:

Parameter Name	Description
entitytype	Any of the following values: <ul style="list-style-type: none">• 0 - For Stack Types.• 1 - For Document Types.
iscustomentity	<i>optional.</i> Any of the following values: <ul style="list-style-type: none">• true - To get the list of user defined entities• false - To get the list of system defined entities

Response:

Response Type	Description
On Success	List of Globodox Stack\Document types will be returned in the XML format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpWebRequest request = WebRequest.Create("http://localhost/api/entitymetadata/1/all/full/
get.ashx?EntityType=0&iscustomerentity=true") as HttpWebRequest;

request.Method = "GET";

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call
to login.

// Get response

string lsResponse = string.Empty;
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());
    lsResponse = reader.ReadToEnd();
}

//The response can now be used to populate a list
```

XML Sample Output:

On successful execution of the request following XML text will be received showing the stack types.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <entities>
    <entity>
      <oid>58</oid>
      <name>tblVendor</name>
      <displayname>Vendor</displayname>
      <collectionname>Vendors</collectionname>
      <logicalname>vendor</logicalname>
      <entitytypecode>1001</entitytypecode>
      <type>etDataEntity</type>
      <alias>Vend</alias>
      <issecured>True</issecured>
      <creator typecode="18" oid="1">Super Administrator</creator>
      <createddate>2008-10-17T14:02:09Z</createddate>
      <modifieddate>2008-10-18T13:35:23Z</modifieddate>
      <associatedentitylinkmask>2</associatedentitylinkmask>
      <description>Use this Stack Type to create a list of all your vendors.</
description>
      <isaudited>False</isaudited>
      <ischildentity>False</ischildentity>
      <iscustomerentity>True</iscustomerentity>
      <ishidden>False</ishidden>
      <isintersect>False</isintersect>
      <islookupentity>False</islookupentity>
      <ispkplatformgenerated>True</ispkplatformgenerated>
      <isvirtual>False</isvirtual>
      <linkedto>0</linkedto>
      <publishstate>Published</publishstate>
    </entity>
  </entities>
  <summary listtype="entities" start-index="1" max-results="10" returned="1" />
</response>
```

1.5 Getting List of Stack Items

Description

Returns a list of Globodox Stack items.

URL:

```
GET /api/{stacktype}/{db}/all/full/get.ashx?[start-index={startindexvalue}][&max-results={maxresultsvalue}]
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

URL Parameters:

Parameter Name	Description
stacktype	Name of the Stack type.
db	Value of the oid of the database received after getting the list of Globodox databases.

Query Parameters:

Parameter Name	Description
startindexvalue	<i>optional.</i> Value of the starting index of the rows to be returned. Specify 0 to begin start at the first document.
maxresultsvalue	<i>optional.</i> Value of the count of rows to be returned.

Response:

Response Type	Description
On Success	List of Globodox Stack Items will be returned in the XML format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpWebRequest request = WebRequest.Create("http://localhost/api/vendor/1/all/full/get.ashx?
start-index=1&max-results=50") as HttpWebRequest;

request.Method = "GET";

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call
to login

// Get response
string lsResponse = string.Empty;
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());
    lsResponse = reader.ReadToEnd();
}

//The response can now be used to populate a list
```

XML Sample Output:

On successful execution of the request following XML text will be received showing the stack items.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <vendors>
    <Vendor>
      <oid>14</oid>
      <name>Acme Corp. </name>
    </Vendor>
    <Vendor>
      <oid>15</oid>
      <name>ORM Office Supplies</name>
    </Vendor>
    <Vendor>
      <oid>16</oid>
      <name>Zuvixo Office Furnitures</name>
    </Vendor>
    <Vendor>
      <oid>18</oid>
      <name>Smith and Co. </name>
    </Vendor>
    <Vendor>
      <oid>19</oid>
      <name>Western Gas & Electric</name>
    </Vendor>
  </vendors>
  <summary listtype="vendors" start-index="1" max-results="50" returned="13" />
</response>
<summary listtype="entities" start-index="1" max-results="10" returned="2" />
</response>
```

1.6 Getting a count of Documents related to a Stack Item

Description

Returns the count of documents related to a Stack Item

URL:

```
GET /api/document/{db}/all/count/-/{stacktype}/get.ashx?ParentDataEntityObjectID={stackitemid}
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

URL Parameters:

Parameter Name	Description
db	Value of the oid of the database received after getting the list of Globodox databases.
stacktype	Name of the Stack type.

Query Parameters:

Parameter Name	Description
stackitemid	Value of the oid of stack item received after requesting for the list of Stack items.

Response:

Response Type	Description
On Success	Document count will be returned in the XML format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpWebRequest request = WebRequest.Create(
    "http://localhost/api/document/1/all/count/-/vendor/get.ashx?ParentDataEntityObjectID=14")
as HttpWebRequest;

request.Method = "GET";

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call
to login

// Get response
string lsResponse = string.Empty;
string lsCount = string.Empty;
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());
    lsResponse = reader.ReadToEnd();
    lsCount = lsResponse.Substring(lsResponse.IndexOf("<count>")); // "<Count>N</Count>..."
    lsCount = lsCount.Substring(7); // "N</Count>..."
    lsCount = lsCount.Substring(0, lsCount.IndexOf("<")); // ">N</Count>..."
}

//The response can now be used to populate a list
```

XML Sample Output:

On successful execution of the request following XML text will be received showing the Document count in Globodox stack.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <documents>
    <document>
      <count>1</count>
    </document>
  </documents>
  <summary listtype="documents" start-index="1" max-results="1" returned="1" />
</response>
```

1.7 Getting details for Documents related to a Stack Item

Description

Returns the details for documents related to a Stack Item.

URL:

```
GET /api/document/{db}/all/full/{stacktype}/get.ashx?ParentDataEntityObjectID={stackitemid}
[&start-index={startindexvalue}][&max-results={maxresultsvalue}]
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

URL Parameters:

Parameter Name	Description
db	Value of the oid of the database received after getting the list of Globodox databases.
stacktype	Name of the Stack type.

Query Parameters:

Parameter Name	Description
stackitemid	Value of the oid of the stack item received after requesting for the list of Stack items.
startindexvalue	<i>optional</i> Value of the starting index of the rows to be returned. Specify 0 to begin start at the first document
maxresultsvalue	<i>optional</i> Value of the count of rows to be returned

Response:

Response Type	Description
On Success	Information about the requested document will be returned in XML format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:**C#**

```
// Create the web request
HttpWebRequest request = WebRequest.Create(
    "http://host/api/document/1/all/full/-/vendor/get.ashx?
    ParentDataEntityObjectID=14&start-index=1&max-results=20") as HttpWebRequest;

request.Method = "GET";

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call
to login

// Get response

string lsResponse = string.Empty;
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());
    lsResponse = reader.ReadToEnd();
}
```

XML Sample Output:

On successful execution of the request following XML text will be received.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <documents>
    <document typecode="2">
      <oid>1</oid>
      <title>Agreement Acme</title>
      <creator typecode="18" oid="1">Super Administrator</creator>
      <modifiedby typecode="18" oid="1">Super Administrator</modifiedby>
      <owner typecode="18" oid="1">Super Administrator</owner>
      <owninggroup typecode="31" oid="1">Administrators</owninggroup>
      <source>disk</source>
      <filestore typecode="22" oid="1">Default File Store</filestore>
      <path>DefaultFileStore\FolderBE2</path>
      <createddate>2008-10-17T14:22:51Z</createddate>
      <modifieddate>2008-10-18T14:50:25Z</modifieddate>
      <majorversion>1</majorversion>
      <minorversion>0</minorversion>
      <state id="1">Checked-In</state>
      <doctype id="2000">Agreement</doctype>
      <boundtostack typecode="1001" typecodename="Vendor" oid="14">Acme Corp. </
boundtostack>
      <isinworkflow>False</isinworkflow>
      <isencrypted>False</isencrypted>
      <onhold>False</onhold>
      <lastlockedon>1899-12-30T00:00:00Z</lastlockedon>
      <lockholder typecode="18" oid="" />
      <version_comment />
      <version_createdon>2008-10-17T14:22:51Z</version_createdon>
      <version_creator typecode="18" oid="1">Super Administrator</version_creator
>
      <version_displaymask>0</version_displaymask>
      <version_isindexed>True</version_isindexed>
      <version_indexedon>2008-10-17T15:56:01Z</version_indexedon>
      <version_filename>Agreement acme. tif</version_filename>
      <version_filesize>148252</version_filesize>
      <version_filetype mimetype="">TIF</version_filetype>
      <securitylabel typecode="30" oid="" />
    </document>
  </documents>
  <summary listtype="documents" start-index="1" max-results="20" returned="1" />
</response>
```

1.8 Download a Document

Description

Downloads the specified document from a Globodox DB

URL:

```
GET api/download/document/{db}/{oid}/get.ashx
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

URL Parameters:

Parameter Name	Description
db	Value of the oid of the database
oid	Value of the oid of the document you wish to download

Response:

Response Type	Description
On Success	Document will be downloaded
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpWebRequest request = WebRequest.Create("http://localhost/api/download/document/2/1/get.ashx") as HttpWebRequest;

request.Method = "GET";

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call to login

// Get response
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    string lsFilePath = "c:\\temp\\download\\invoice.tiff";
    byte[] lBuffer = new byte[4096];
    int liBlockSize;
    using (FileStream loFS = new FileStream(lsFilePath, FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.None))
    {
        Stream loSrcStream = response.GetResponseStream();

        do
        {
            liBlockSize = loSrcStream.Read(lBuffer, 0, 4096);
            if (liBlockSize > 0) loFS.Write(lBuffer, 0, liBlockSize);
        }
        while (liBlockSize > 0);

        loFS.Close();
    }
}

//The downloaded file can now be used for further processing
```

1.9 Upload a Document

Description

Uploads the document to a Globodox DB

URL:

```
POST /api/upload/{doctype}/{db}/post.ashx
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login
Upload-Content-FileName	File name of the uploaded document.
Upload-Content-Length	File size in bytes.
Upload-Content-Type	File mime type

URL Parameters:

Parameter Name	Description
doctype	Any of the following values: <ul style="list-style-type: none">• <i>document</i> - For un-typed document.• <i>Doctype name (defined in the Globodox DB)</i> - For typed document e.g. Invoice,Agreement,Check,Email,Letter.
db	Value of the oid of the database.

Response:

Response Type	Description
On Success	Information about the uploaded document will be returned in XML format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpRequest request = WebRequest.Create(
"http://localhost/api/upload/document/2/post.ashx") as HttpRequest;

request.Method = "POST";

request.Method = "POST";
request.KeepAlive = false;
request.Timeout = 300000;

string lsFilePath = "c:\\temp\\upload\\invoice.tiff";
FileInfo loFileInfo = new FileInfo(lsFilePath);

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call
to login
request.Headers.Add("Upload-Content-FileName", System.IO.Path.GetFileName(lsFilePath));
request.Headers.Add("Upload-Content-Length", loFileInfo.Length.ToString());
request.Headers.Add("Upload-Content-Type", GetMimeType(loFileInfo.Extension));

request.Headers.Set("Pragma", "no-cache");
request.ContentType = "application/octet-stream";

byte[] array;
using (FileStream fileStream = new FileStream(lsFilePath, FileMode.Open, FileAccess.Read))
{
    array = new byte[(int)fileStream.Length];
    fileStream.Read(array, 0, array.Length);
    fileStream.Close();
}

request.ContentLength = (long)array.Length;
Stream requestStream = loRequest.GetRequestStream();
requestStream.Write(array, 0, array.Length);
requestStream.Close();

// Get response
using (HttpWebResponse response = loRequest.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());
    lsResponse = reader.ReadToEnd();
}
```

XML Sample Output:

On successful execution of the request following XML text will be received.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <documents>
    <document>
      <oid>234</oid>
      <name>invoice</name>
    </document>
  </documents>
  <summary listtype="documents" listitemtype="document" start_index="1"
max_results="1" returned="1" />
</response>
```

1.10 Log out from Globodox

Description

Logs out from Globodox.

URL:

POST /api/signout/POST.ashx

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login
Action	Specifies what action to take. For now specify this as "Signout".

Response:

Response Type	Description
On Success	Status code 200/OK is returned
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

```
C#

// Create the web request
HttpRequest request = WebRequest.Create("http://localhost/api/signout/POST.ashx") as
HttpRequest;
request.Method = "POST";

//set the required header parameter

request.Headers.Add("Consumer-Token",sToken); // Token obtained and stored after the call to
login
request.Headers.Add("Action", "Signout");

// Get response

try
{
    HttpResponse response = request.GetResponse() as HttpResponse;
}
catch(Exception poErr)
{
    MessageBox.Show(poErr.Message);
}
```

1.11 Response Status Codes

Any one of the status codes mentioned in the table below will be returned as a part of the response to any request made to the Globodox REST API.

Code	Meaning	Description
200	Ok	The request returned a successful response.
201	Created	Following a POST command, this indicates success.
204	No Response	Server has received the request but there is no information to send back.
308	Incomplete	Response code that the REST API server sends in case of a multi session http request
400	Bad Request	The request had bad syntax or was inherently impossible to be satisfied.
401	Unauthorized	The resource could not be created/accessed because the client submitting the request either has not provided authentication credentials, or authentication has failed (e.g. due to an invalid user name or password).
403	Forbidden	The request is for something forbidden. Authorization will not help.
404	Not Found	The resource requested for in the URL does not exist.
500	Internal Server Error	A service error prevented the list of resources from being read.

2 Advanced Topic

[How to get JSON output instead of XML](#)

2.1 How to get JSON output instead of XML

By default the output of the REST API call is in XML format. In order to get the output of a REST API call in the form of JSON instead of XML use **alt** optional query parameter along with the REST API web request. Following example shows how to get the list of Globodox databases in the form of JSON format.

URL:

```
GET /api/database/all/full/get.ashx?[alt=json]
```

Header Parameters:

Parameter Name	Description
Consumer-Token	Token obtained and stored after the call to login

Query Parameters:

Parameter Name	Description
alt	json - To view the output in JSON format.

Response:

Response Type	Description
On Success	List of Globodox databases will be returned in the JSON format.
On Failure	A status code corresponding to the reason of the failure is returned.

Sample Code:

C#

```
// Create the web request
HttpWebRequest request = WebRequest.Create(
    "http://localhost/api/database/all/full/get.ashx?alt=json") as HttpWebRequest;

request.Method = "GET";

//set the required header parameter
request.Headers.Add("Consumer-Token", sToken); // Token obtained and stored after the call
to login

// Get response

string lsResponse = string.Empty;
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());
    lsResponse = reader.ReadToEnd();
}

//The response can now be used to populate a list
```

JSON Sample Output:

On successful execution of the request following JSON text will be received.

```
{
  "response":{
    "Databases":[
      {
        "name":"Sample DB.gxdb",
        "oid":2
      },
      {
        "name":"MainDB.gxdb",
        "oid":1
      }
    ],
    "summary":{
      "listtype":"Databases",
      "maxresults":10,
      "returned":2,
      "startindex":1
    }
  }
}
```