

DevResults API

Last Modified on 10/11/2018 12:40 pm EDT

An API (Application Program Interface) lets you access components of one application for use in another. DevResults' API allows you to pull information from your DevResults site to use or display in another application or website.

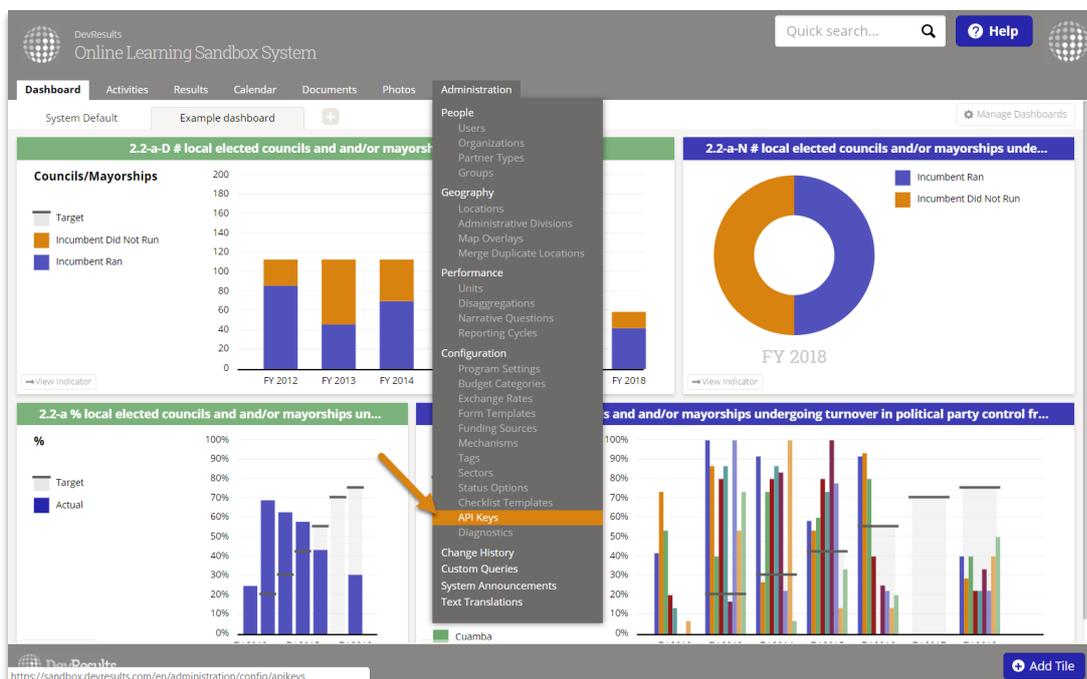
How do I use the API?

There are several steps that you'll need to take to ensure that you can authenticate and interact with the API properly.

1. Configure API Access

This step is completed inside the DevResults application. You'll need to have high enough access to be able to create new users before proceeding.

Once you have access and are logged in to the application, navigate to API Keys in the **Administration** menu:



The screenshot shows the DevResults Online Learning Sandbox System interface. The 'Administration' menu is open, and 'API Keys' is highlighted with an orange arrow. The interface includes a search bar, a 'Help' button, and several dashboard widgets. The 'Administration' menu lists various system settings and user management options.

- People
 - Users
 - Organizations
 - Partner Types
 - Groups
- Geography
 - Locations
 - Administrative Divisions
 - Map Overlays
 - Merge Duplicate Locations
- Performance
 - Units
 - Disaggregations
 - Narrative Questions
 - Reporting Cycles
- Configuration
 - Program Settings
 - Budget Categories
 - Exchange Rates
 - Form Templates
 - Funding Sources
 - Mechanisms
 - Tags
 - Sectors
 - Status Options
 - Checklist Templates
 - Diagnostics
- Change History
- Custom Queries
- System Announcements
- Text Translations

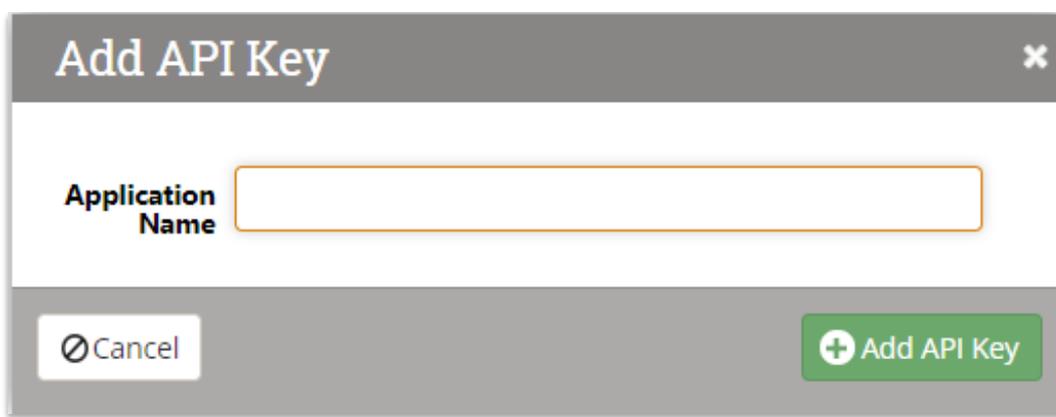
You will now be presented with a list of any previously configured applications. DO NOT re-use previously configured application keys. These keys are used to track which application made changes in DevResults. If a key is re-used, you will lose the ability to track the source of changes.

Next, you'll want to create a new API Key for your application. To do so, click the *Add API*

Keybutton:



You will be presented with a dialog and prompted to input the application name:



Choose an application name and click the **Add API Key** button to finalize the creation of the new API Key. Once the key is created, you will be taken to the API Key Details page where you will be able to modify the name of your application, see both the API Secret and API Token, as well as regenerate a new API Token if needed.

2. Signing an API Request

Once you've created your API Key in DevResults, you'll have everything you need to make requests to the DevResults API. Each request will require several pieces of information to construct:

1. Your API Token
2. Your API Secret
3. The Current Unix Epoch Time in Milliseconds. This is simply the number of milliseconds that have passed since January 1, 1970.

You will then use that information to build up the API Request URL. There are several steps to building the complete URL. For our example, we are going to use **demo.devresults.com** as our site. To build the URL, follow these steps (we've provided several samples below in various languages):

1. **Build the base API Request URL**

This will consist of the API REST URL with two query string parameters appended. The first is going to be the API token using `t` as the parameter name. The second will be the number of milliseconds past since the Unix Epoch time using `ms` as the parameter name. The completed URL will look something like

```
http://demo.devresults.com/api/awards?t=yourToken&ms=123456789
```

2. Build the signature

Get the query parameters, sort them alphabetically by key, and build a signature using the following format: `key1|value1|key2|value2|keyn|valuen`. The result for the above URL would be `ms|123456789|t|yourToken|`.

3. Hash the signature

Hash the generated signature using a Hash-based message authentication code (HMAC) with SHA256 as the hashing algorithm and your secret as the key.

4. Build the final URL

Generate the final URL by appending the hashed signature to the URL using `s` as the parameter name. The final URL would be

```
http://demo.devresults.com/api/awards?  
t=yourToken&ms=123456789&s=yourHashedSignature
```

Samples

You can download code samples from the [DevResults GitHub](#) to get started, or use the samples below.

- [Python](#)
- [PHP](#)
- [C#](#)
- [Ruby](#)
- [NodeJS](#)

Python

```

import hashlib
import hmac
import calendar
import time
import urlparse

secret = bytes("~~yourApiSecret~~").encode("utf-8")
token = "~~yourApiToken~~"
ms = calendar.timegm(time.gmtime()) * 1000

apiUrl = "http://demo.devresults.com/api/awards?t=%s&ms=%s" % (token, ms)
)
uri = urlparse.urlparse(apiUrl)
queryValues = urlparse.parse_qs(uri.query)

sortedKeys = queryValues.keys()
sortedKeys.sort()

signature = reduce(lambda x, y: x + y,
    map(lambda key: key + "|" + queryValues[key][0] + "|", sortedKeys))
sigbytes = bytes(signature).encode("utf-8")

hashedSignature = hmac.new(secret, sigbytes, digestmod=hashlib.sha256).hexdigest()
apiUrl += ("&s=%s" % hashedSignature)

```

PHP

```

$secret = "~~yourApiSecret~~";
$token = "~~yourApiToken~~";
$ms = time() * 1000;

$apiUrl = "http://demo.devresults.com/api/awards?t={$token}&ms={$ms}";
parse_str(parse_url($apiUrl, PHP_URL_QUERY), $queryValues);

$sortedKeys = array_keys($queryValues);
sort($sortedKeys);

$map = array_map(function($key) use (&$queryValues) {
    return $key . "|" . $queryValues[$key] . "|";
}, $sortedKeys);
$signature = array_reduce($map, function($x, $y) { return $x . $y; });
$hashedSignature = hash_hmac("sha256", $signature, $secret, false);

$apiUrl .= "&s={$hashedSignature}";

```

C#

```

var secretBytes = System.Text.Encoding.UTF8.GetBytes("~~yourApiSecret~~"
);
var token = "~~yourApiToken~~";
var ms = (DateTime.UtcNow - new DateTime(1970, 1, 1, 0, 0, 0)).TotalMilli
seconds;
var urlBase = string.Format("http://demo.devresults.com/api/awards?t={0}
&ms={1}", token, ms);
var uriBuilder = new UriBuilder(urlBase);

byte[] hashBytes;

var queryValues = System.Web.HttpUtility.ParseQueryString(uriBuilder.Que
ry);
var sortedKeys = new SortedDictionary(StringComparer.OrdinalIgnoreCase);
foreach (string param in queryValues)
{
    sortedKeys.Add(param, queryValues[param]);
}

var signature =
    sortedKeys.Aggregate("", (s, kvp) => s + string.Format("{0}|{1}|", k
vp.Key, kvp.Value));
var signatureBytes = System.Text.Encoding.UTF8.GetBytes(signature);
using (var algorithm = HMAC.Create("HMACSHA256"))
{
    algorithm.Key = secretBytes;
    hashBytes = algorithm.ComputeHash(signatureBytes);
}

var hashedSignature = BitConverter.ToString(hashBytes).ToLower().Replace
("-", "");

queryValues.Add("s", hashedSignature);
uriBuilder.Query = queryValues.ToString();
var apiUrl = uriBuilder.Uri.ToString();

```

Ruby

```
require 'net/http'
require 'openssl'
require 'cgi'

secret = "~~yourApiSecret~~"
token = "~~yourApiToken~~"
ms = Time.now.to_i * 1000

apiUrl = "http://demo.devresults.com/api/awards?t=#{token}&ms=#{ms}"
uri = URI.parse(apiUrl)
queryValues = CGI::parse(uri.query)

sortedKeys = queryValues.keys.sort

signature = ""

sortedKeys.each { |k|
  signature << "#{k}|#{queryValues[k][0]}|"
}

hashedSignature = OpenSSL::HMAC.hexdigest("sha256", secret, signature)

apiUrl << "&s=#{hashedSignature}"
```

NodeJS

```
var crypto = require('crypto');
var url = require('url');

var secret = '~~yourApiSecret~~';
var token = '~~yourApiToken~~';
var ms = new Date().getTime();

var apiUrl = 'http://demo.devresults.com/api/awards?t=' + token + '&ms='
+ ms;
var parsedApiUrl = url.parse(apiUrl ,true);
var query= parsedApiUrl.query;

var keys = [];
var sigBase = '';

for(var key in query) {
    if(query.hasOwnProperty(key)) {
        keys.push(key);
    }
}

keys.sort();

for(var i = 0; i < keys.length; i++) {
    sigBase += keys[i];
    sigBase += '|';
    sigBase += query[keys[i]];
    sigBase += '|';
}

var sig = crypto.createHmac('sha256', secret).update(sigBase).digest('hex');

apiUrl += '&s=' + sig;
```

Didn't answer your question? Please email us at help@devresults.com.

Related Articles
