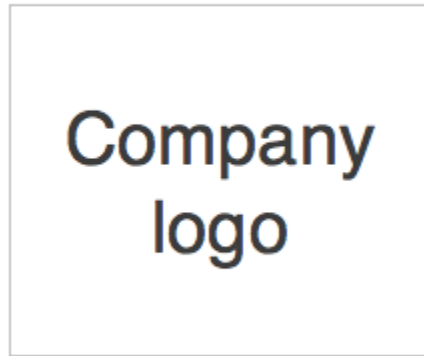




Weebly Cloud

APIs

Last generated: May 19, 2017



© 2016 Your company. This is a boilerplate copyright statement... All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

Table of Contents

- Use the REST APIs
 - [About the REST APIs](#)
 - [Page API](#)

About the Weebly Cloud REST APIs

Weebly APIs are provided using REST over HTTPS and utilize the HTTP verbs GET, POST, PUT, PATCH and DELETE. We provide a number of client libraries to help get you started. See the following sections for more general information about using our APIs.

Client Libraries

Looking for an easy way to get started with our API rather than having to piece it all together yourself? Install one of our client libraries! These libraries make it easy to interact with our API:

- Language-specific calls that process the HTTP for you so you can work directly with resources in your code. For example, our PHP library has a `createUser` method that you can use without having to provide HTTP header info. The returned new user is a resource you can work with directly, instead of having to process an HTTP response.
- Easier authorization.
- Fully documented, with robust examples.

Download our libraries on GitHub here:

- [PHP](#)
- [Ruby](#)
- [Python](#)
- [Java](#)

Endpoint

The endpoint is: `https://api.weeblycloud.com`

Schema

All API access is over HTTPS, and accessed from the `api.weeblycloud.com` domain. All data is sent and received as JSON. Blank fields are included as null instead of being omitted. Dates and times are Unix Epoch timestamps.

List View

When retrieving a list of resources, a summary list view of each resource is returned. In some cases, this is akin to a compound primary key. In most root resource cases, like `product`, the list view contains enough information to make a workable list view in an application without making calls for each resource.

Detail View

When retrieving a single resource, you will receive a detailed view typically resulting in all the attributes of the resource. In addition, sub resources (like a product's skus) that are owned by this resource are inlined to the result. Other resources (like categories) which are associated but not owned by this resource provide the ids needed to look them up separately.

HTTP Redirects

We may issue redirects. Any call may result in a redirect, at which point you should make an identical call but to the newly provided resource URL.

HTTP Verbs

You can use the following:

- **GET** Used for retrieving resources.
- **POST** Used for creating resources.
- **PATCH** Used for updating resources with partial JSON data.
- **PUT** Used for replacing resources or entire collections. Fields will be replaced with given values and any fields without given values will be reset to their default. Any fields without defaults, for example ID fields, will remain unchanged.
- **DELETE** Used for deleting resources.

Search Parameters

Some endpoints allow you to use search parameters in order to filter your results. When using parameters, do not add that query string to your HMAC Hash.

Pagination

Pagination of certain resources in list view happens automatically. Information about the total number of resources, like the total count, are provided in headers.

- `X-Resultset-Total` shows the total number of records for the request.
- `X-Resultset-Page` shows the page for the call.
- `X-Resultset-Limit` shows the record limit for the page (i.e. each call). The default is 25, the max is 200.

If there are additional pages to be returned, a `Link` header will be included that has links to the first, previous, next and last pages.

For example, a response from the Site API might return the following in the header, showing that there are a total of 44 records to be returned, this is the first page, and that 25 is the record limit:

```
"http_code": 200,
  "result": {
    "headers": [
      "HTTP/1.1 200 OK",
      "Date: Sat, 25 Jun 2016 00:21:34 GMT",
      "Server: Apache",
      "X-REPOSITORY: repos\myrepos",
      "X-Resultset-Total: 44",
      "X-Resultset-Page: 1",
      "X-Resultset-Limit: 25",
```

When you expect the result set to be greater than 25, use these parameters to determine the pages to return and to increase the limit:

- **page** (int) Which page of results to return. Starts at 1.
- **limit** (int) Number of results per page. Default 25, Max 200. Don't forget that raising the limit may affect performance.

Note: Only the first 25 results are returned by default. If the result set is expected to be over 25, you **must** implement pagination.

For example, this CURL request to the Site API returns the second page of result sets.

```
curl https://api.weeblycloud.com/user/USER_ID/site?page=2&limit=25 \
-H "X-Public-Key: YOURAPIKEY" \
-H "X-Signed-Request-Hash: YOURSECRETHASH"
```

Signing and Authenticating Requests

The API is validated using both a public key and a private HMAC Hash of the request contents. These values must be passed in the request header as X-Public-Key and X-Signed-Request-Hash. To get started making requests, you will need YOURAPIKEY and YOURAPISECRET which can be generated and obtained on the Settings tab in the Weebly Cloud Admin provided for your account. YOURAPISECRET is a shared secret known only to you and Weebly Cloud. Do not share YOURAPISECRET.

```
POST /hosts/user/39793399/site HTTP/1.1
User-Agent: curl/7.30.0
Host: api.weeblycloud.com
Accept: */*
Content-type: application/json
X-Public-Key: YOURAPIKEY
X-Signed-Request-Hash: YOURSECRETHASH
```

In the example above, YOURAPIKEY refers to your API Key. This API Key is used by our system to identify your account. YOURSECRETHASH refers to the base-64 encoded HMAC-SHA256 signature. The signature will need to be computed for each request. You can generate YOURSECRETHASH using

an HMAC-SHA256 hash of the request TYPE (Ex. POST), URL (ex. user/234256/loginLink) and any REQUEST DATA (ex. { 'plan_id': 34 }) using YOURAPISECRET as the key, and base-64 encoding the result. Any requests that do not contain a valid signature will be rejected. Since YOURAPISECRET is shared only by you and Weebly Cloud, we are able to validate your request by generating the same HMAC-SHA256 signature and match it against YOURSECRETHASH.

```
<?php
$base_api = "https://api.weeblycloud.com/";
$api_key = YOURAPIKEY;
$api_secret = YOURAPISECRET;

$request_type = "POST";
$url = "user/39793399/site";
$data = array(
    "domain" => "test-site.com"
);

$content = json_encode($data);
$hash = hash_hmac('SHA256', $request_type . "\n" . $url . "\n" . $content, $api_secret);
$hash = base64_encode($hash);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $base_api.$url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, $request_type);
curl_setopt($ch, CURLOPT_POSTFIELDS, $content);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-type: application/json',
    'X-Public-Key: ' . $api_key,
    'X-Signed-Request-Hash: ' . $hash
));

$out = curl_exec($ch);
$info = curl_getinfo($ch);
curl_close($ch);

$result = json_decode($out);
```

Note the line below which generates the hash for this request.

```
$hash = hash_hmac('SHA256', $request_type . "\n" . $url . "\n" . $content, $api_secret);
```

Errors

When a call can't successfully complete, Weebly throws an error and returns in the JSON response.

```
{
```

```
"error": {  
  "code": 900,  
  "message": "brand_name field missing"  
}
```

Following are the error codes and descriptions:

- 100 Invalid hash provided.
The hash does not match the request
- 101 Unknown API key provided.
Check the Admin portal to verify that your key still exists.
- 102 This Weebly account doesn't have API access.
- 200 This Weebly account doesn't have access to this user.
- 201 This Weebly account doesn't have access to this site.
- 300 Query cannot be routed.
- 900 Invalid request.
- 999 An unknown error occurred.
- 32001 Invalid namespace or function name.
- 32002 Class not found.
- 32404 Invalid request.
- 32600 Invalid request.
- 32601 Method not found.
- 32602 Invalid parameters - too few.

Page API

The Pages API provides methods for reading and updating information about the pages for a given user's site.

Fields

The following table shows all fields that exist for this API, those that are returned when you retrieve a list, and those that are changeable using `PATCH`. All fields are returned when you retrieve a single item.

Name	Description	Type	List	Change-able
page_id	The unique ID of a page	string	X	
title	The title of a given page	string	X	X
	Where this page appears in its of pages, as configured on the Pages tab of the editor. For example: Home - page_order 1 HomeSubPage - page_order 1 HomeSubPage - page_order 2 About - page_order 2 AboutSubPage - page_order 1 Contact - page_order 3 ContactSubPage - page_order 1>			
page_order		integer	X	
page_url	You can determine the hierarchy using the <code>parent_id</code> for the page. Relative URL for the page. You can get the domain from the Site API	string	X	
parent_id	The <code>page_id</code> of the page one level up in the navigation hierarchy. If there is no parent, nothing is returned. The layout header used for the page. Valid values are:	string	X	
layout	<ul style="list-style-type: none">• short-header• tall-header• no-header	string	X	
hidden	Whether or not the page is hidden from the site's navigation. Valid values are: <ul style="list-style-type: none">• true• false	boolean		

	Whether or not the page requires site membership in order to be viewed.	
membership_required	Valid values are: <ul style="list-style-type: none"> • true • false 	boolean
updated_date	When the page was last updated.	Unix timestamp in GMT

Note: Any fields returned that are not documented are currently unsupported and can be safely ignored.

Retrieve a List of Pages for a Given Site

Endpoint URL:

```
GET /user/USER_ID/site/{SITE_ID}/page
```

Returns all pages for the given site. Only list fields are returned.

Query Paramters

Returned values are paginated. You can further filter results using these parameters

Parameter	Description	Type
	Which page of results to return. Start is 1.	
page	<i>Note: Only the first 25 results are returned by default. If the result set is expected to be over 25, you must implement pagination.</i>	integer
limit	Number of results per page to return. Default is 25. Max is 200.	integer
query	Use to retrieve only pages that have a full-text match with the query string.	string
	Field name to set a filter on. Valid values are:	
filterby	<ul style="list-style-type: none"> • page_id • title • layout 	string
filterfor	Value to search the filterby field for.	string
sortby	Field name to sort by	string
	Sort direction. Valid values are:	
sortdir	<ul style="list-style-type: none"> • asc • desc 	string

Example CURL request returning all pages:

```
curl https://api.weeblycloud.com/user/USER_ID/site/SITE_ID/page \
-H "X-Public-Key: YOURAPIKEY" \
-H "X-Signed-Request-Hash: YOURSECRETHASH"
```

Example filtered request:

```
curl https://api.weeblycloud.com/user/USER_ID/site/SITE_ID/page?filterby=layout
-H "X-Public-Key: YOURAPIKEY" \
-H "X-Signed-Request-Hash: YOURSECRETHASH"
```

Example response:

```
[
  {
    "page_id": "131221993403487092",
    "title": "new",
    "page_order": 2,
    "parent_id": "205694454183612584",
    "layout": "tall-header",
    "page_url": "new.html"
  },
  {
    "page_id": "205694454183612584",
    "title": "Blog",
    "page_order": 2,
    "parent_id": null,
    "layout": "tall-header",
    "page_url": "blog1.html"
  },
  {
    "page_id": "267240304658043510",
    "title": "members only",
    "page_order": 4,
    "parent_id": null,
    "layout": "tall-header",
    "page_url": "members-only.html"
  }
]
```