# The Curata Content API

Your Curata site includes an application programming interface (API) which allows you to access any of the content you've published to your Curata site from other sites, content management systems, or programs. Common applications include pulling content from Curata into your Wordpress blog, showing timely articles in a footer or sidebar of a page on your corporate website, or building a highly interactive site for your readers to navigate the content you've published.

# A Quick Example (try it)

Using Curata's Categories and Articles JSON-P API, and a little jQuery, you can build a simple article browser in Javascript.

```
<!DOCTYPE html> <html>     <head>         <title>Using the Curata JSON-P
Content API with jQuery</title>         <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script>         $(document).ready(function()         {             // This
event handler will wait for any <a class="feed"> to appear             // in
the document.             $(document).on("click", "a.feed", function(evt)
{                 evt.preventDefault();                 var feed_url =
$(this).attr("href");                 feed_url = feed_url.replace("/articles?",
"/articles.js?");                 jQuery.ajax(feed_url,                 {
dataType: "jsonp",                     success: function(data, textStatus,
jqXHR)                     {                         var articles =
data.articles,                             article = null;
var $articles = $("#articles");
$("#feed_description").html(data.title);
$articles.html("");                         for (var i = 0, length =
articles.length; i < length; i++)                         {
article = articles[i];                             $articles.append("<li>" +
"<h3><a href='" + article.url + "'>" +
article.title +                                                 "</a></h3>" +
"<p>" +                                                 article.snippet +
"</p>" +                                                 "</li>");
}                     }                 });             });             //
This is the initial JSON-P call to fetch the category groups             // and
categories from a Curata portal.  Note that you can             // use this
from any site on the Internet (not just pages on             // the same Curata
portal).             //             // The most important parameter is the
dataType: "jsonp", which             // tells jQuery's AJAX engine to use
script injection to invoke             // the API in a cross-domain safe way.
jQuery.ajax("http://contentcurationmarketing.com/api/v1/categories.js",
{                 dataType: "jsonp",                 success: function(data,
textStatus, jqXHR)                 {                     var groups =
data.groups,                         group = null,
category = null;                     var $groups = $("#groups"),
$categories = null;                      for (var i = 0, length =
groups.length; i < length; i++)                     {
group = groups[i];                         $groups.append("<li>" +
"<h3><a class='feed' href='" + group.feed_url + "'>" +
group.title +                                        "</a></h3>" +
"<ul class='categories'></ul>" +
"</li>");                        $categories = $groups.find("ul");
for (var j = 0, jlength = group.categories.length; j < jlength; j++)
{                             category = group.categories[j];
$categories.append("<li><a class='feed' href='" + category.feed_url + "'>" +
category.title +                                                "</a></li>");
}                     }                 }             });         });
</script>     </head>      <body>         <h1>Using the Curata JSON-P Content
API with jQuery</h1>         <p>             View Source on this page to see a
documented example             of calling the Curata JSON-P Content API.  Note
that             this example is simplified, and there are many more
properties of the articles returned by the API than             are used in
this example.         </p>          <div id="example">             <div
id="left">                 <h2>Category Groups/Categories</h2>
<ul id="groups"></ul>             </div>             <div id="right">
<h2 id="feed_description">(click a category to retrieve articles)</h2>
<ol id="articles"></ol>             </div>         </div>     </body> </html>
```

## A note about versions

Curata's API is currently at version 1, and while we plan to add more capabilities, fields, and query parameters to version 1, we promise that any new features we add will not break anything you build against the version 1 API. If we find that we do need to make changes that would be incompatible with your implementations by changing field or parameter names, the meaning of a field, or the structure of the URLs, we will add a new version instead of modifying version 1.

## Available formats

The Curata Content API is available over HTTP via the `/api` URL on your Curata site. We currently support three formats:

- RSS (`application/rss+xml`, file extension `.rss`), which is only available for the Articles API;
- JSON (`application/json`, file extension `.json`);
- JSON-P (`application/x-javascript`, file extension `.js`).

You have two options for specifying which format you'd like to use for a given call:

- HTTP's `Accept` header; this is the preferred method for specifying which format you'd like, but it requires that you have control over your HTTP client. This may not be possible if you are giving a Curata API URL to an RSS reader, or making a JSON-P request, for example.
- Adding a file extension to end of the URL, before the query string. For example, to request featured articles in RSS, you would use the

URL `http://yourcuratasite.com/api/v1/articles.rss?is_featured=true`

## Data Types

There are a few simple types of data that the Curata API query parameters recognize:

| type | values | example | notes |
|---|---|---|---|
| number | numbers | 1234 | Generally, negative numbers and decimals probably don't apply. |
| numbers | a list of numbers, separated by commas | 10,11,1234 | |
| date/time | an ISO 8601 date and/or time string | 2012-06-04 or 2012-06-04T16:34:00 | |
| boolean | a string representing true or false | true | "y", "yes", "true", and "t" all represent True. "n", "no", "false", "f" and "" represent False. Omitting a boolean parameter usually means don't filter on it either way. |
| string | any string | hello-world | |
| strings | a list of strings, separated by commas | hello,world | |
| URL | a string representing a location on the Internet | http://contentcuration.hivefire.com/articles/93481/content-curation-and-its-positive-impacts-on-busin/ | |

## The Articles API

## URLs

- `http://yourcuratasite.com/api/v1/articles`, for listing and querying articles
- `http://yourcuratasite.com/api/v1/articles/<article-id>`, for retrieving a single article

## Query Parameters

By default, the articles API without query parameters will return the most recent 40 articles published to your Curata site. You can further refine your query, or modify which page of query results you get with the parameters below. Note that you can use any of these parameters in any order; the results will be the most restrictive combination. For example, if you specify `in_category=1`and `has_topic=joe-smith`, the results will include articles that are **both** in category 1 and mention Joe Smith.

| parameter | data type | description |
|---|---|---|
| count | number | The number of articles to return. |
| start | number | The number of the first article to return (zero-based). |

| parameter | data type | description |
|---|---|---|
| end | number | The number of the last article to return (zero-based). If both `count` and `end` are specified,`count` will override `end`. |
| sort | string | The field on which to order results. May be one of:<br>`approval_date`, when the article was approved on your Curata site (default);<br>`pub_date`, when the article was originally published;<br>`index_date`, when the article was found by Curata. |
| direction | string | Either `DESC` (default) or `ASC`, for descending or ascending sort |
| start_index_time | date | The date/time of the earliest article to return, based on the articles' `index_date`. |
| end_index_time | date | The date/time of the latest article to return, based on the articles' `index_date`. |
| start_publish_time | date | The date/time of the earliest article to return, based on the articles' `pub_date`. |
| end_publish_time | date | The date/time of the latest article to return, based on the articles' `pub_date`. |
| start_approval_time | date | The date/time of the earliest article to return, based on the articles' `approval_date`. |
| end_approval_time | date | The date/time of the latest article to return, based on the articles' `approval_date`. |
| in_category_group | numbers | A list (separated by commas) of category group ids to filter articles. |
| in_category | numbers | A list (separated by commas) of category ids to filter articles. |
| published_by | numbers | A list (separated by commas) of the publisher ids to filter articles. |
| has_topic | strings | A list (separated by commas) of topic slugs to filter articles. |
| is_featured | boolean | True will return only featured articles, False will return only non-featured articles, and omitting the`is_featured` parameter will return both featured and non-featured articles. |
| blog | boolean | True will return only blog articles, False will exclude blog articles, and omitting the `blog`parameter will return both blog and non-blog articles. |
| related | boolean | `related=true` returns a list of related articles keyed by `related_articles`. Simply omit the`related` parameter to disable the return of related articles (identical to `related=false`). This is only available when querying individual articles. (i.e. `/api/v1/articles/1234.json?related=true`) |
| shared_url | strings | If you use Curata's CMS/Social Media sharing framework to post articles to your own CMS, you may need to request articles via the API when you don't have Curata article IDs or URIs available at the point of making the request. In this case, you can use the `shared_url` or`shared_identifier` parameters.<br>`shared_url` refers to the URL of the article as it appears on your CMS. For instance, if you are sharing to a WordPress site, the URL for an article may appear as`http://mysite.example.com/?p=12345` or`http://mysite.example.com/2013/08/30/cool-article/`, depending on your permalink settings. In this case, simply query Curata's API with `?shared_url=http://mysite.example.com/2013/08/30/cool-article/` |

| parameter | data type | description |
|---|---|---|
| shared_identifier | strings | An article must have posted to your CMS in order for Curata to retrieve it by `shared_url` or`shared_identifier`, but it is fine if it has only been posted as a Draft. To confirm the format of the URLs that Curata stores when an article is posted to your CMS, refer to the Share > CMS article queue for any articles that have been Posted.<br>As with `shared_url`, you may also use the database identifier that your CMS has assigned to the article to retrieve it via the Curata API. Please note that if you have multiple CMS channels enabled, there is a possibility that your different CMSes may actually assign the same identifier, perhaps to different articles. Curata will return all matching articles for this query, and you may need to distinguish among the articles returned. |
| contributor | strings | One or more email addresses of contributors to your site. Will return articles contributed by those contributors, including those that have not been approved yet. |
| custom:*field name* | string | Find articles with specific values for their custom fields. For example, if you have configured a custom field named 'My Notes', you may find articles that have the text 'hello world' in their 'My Notes' field with the query `custom:My%20Notes=hello%20world`. The query you provide will be used to search case-insensitively for a substring of the field. |

## Other Options

There are additional options that don't affect which articles are returned, but rather how the results are presented:

| parameter | data type | description |
|---|---|---|
| link | string | Directs the RSS feed to use one of three different links as the link for the article. The options are:<br>`portal` indicates that the article should link to the Curata site's Article Explore page (default);<br>`sharebar` indicates that the article should link to the share page for the article;<br>`direct` indicates that the article should link to the publisher's original article. |
| publisher | string | Directs the RSS feed to use one of two options for the `dc:creator` of the RSS feed. The options are:<br>`name` uses the publisher's name as the `dc:creator` for the article within the feed;<br>`domain` uses the publisher's domain name as the `dc:creator` for the article within the feed. |

## Response

Requesting articles in RSS format will return a valid RSS feed, suitable for adding to a feed reader or other software which can read articles from an RSS feed.

Articles in JSON (and also JSON-P) format include all of the information for an article that would be available on your Curata site, as well as a few fields to help uniquely identify the article and describe it's attributes.

The HTTP response from an Articles API call may return one of the following status codes:

| status code | reason |
|---|---|
| 200 | The request was successful. |
| 400 | There was a problem with one of the query parameters. An informational message will be returned as the body of the response, in `text/plain`. |
| 404 | Single article API only; the article does not exist. |
| 5xx | A temporary error on Curata's system. Try again after a short time. |

An example response,

from http://contentcuration.hivefire.com/api/v1/articles.json?count=2&end_index_time=2012-07-01T10:00:00:

```
{     "title": "Content Curation Marketing",     "description": "Content
Curation Marketing is an online information resource providing the most
relevant news, analysis, blogs and other dynamic content focused on content
curation marketing. The microsite is powered by Curata, an MIT start-up based
in Cambridge, Massachusetts. The company delivers a publishing platform that
enables organizations to increase online marketing effectiveness, sales
conversions and online thought leadership.",     "total": 1862,     "articles":
[          {               "id": 93422,               "uri":
"/api/v1/articles/93422",               "curata_url":
"http://contentcuration.hivefire.com/articles/93422/content-curation-sneaks-up-
on-marketers/",               "share_url":
"http://contentcuration.hivefire.com/articles/share/93422/",             "url":
"http://socialmediatoday.com/ardathalbee/563305/content-curation-sneaks-
marketers?utm_source=feedburner&utm_medium=feed&utm_campaign=The+Social+Custome
r+%28all+posts%29",               "pub_date": "2012-06-30T15:18:54",
"title": "Content Curation Sneaks Up on Marketers",               "snippet":
"<p><span>Ardath Albee gives a few tips on how to curate within Twitter ...",
"image": null,               "thumbnail": null,               "publisher": {
"domain": "socialmediatoday.com",                    "name": "Social Media Today",
"id": 501               },               "authors": [],               "categories": [
{                    "name": "Tips & Best Practices",
"id": 7                    }               ],               "topics": [],
"quotes": [],               "videos": [                    {
"embed_html": "<iframe
src='http://embed.ted.com/talks/marco_tempest_a_cyber_magic_card_trick_like_no_
other.html' width='853'
height='480' frameborder='0' scrolling='no' webkitAllowFullScreen
mozallowfullscreen allowFullScreen>
</iframe>",                         "src_url":
```

```
"http://embed.ted.com/talks/marco_tempest_a_cyber_magic_card_trick_like_no_othe
r.html",                        "dimensions": "853x480",
"content_type": "video/x-embed-ted",                        "video_id":
"marco_tempest_a_cyber_magic_card_trick_like_no_other"                 }
],             "is_featured": false,             "blog": false,
"index_date": "2012-07-01T10:00:00",             "contributors": [
{                       email: "joesmith@example.com",                     name:
"joesmith",                    date: "2013-10-16T11:00:37",
comment: "I loved this article!"                  },                {
email: "janesmith@example.com",                       name: "janesmith",
date: "2013-10-16T11:00:37",                        comment: "You guys will love
this!"                  }              ]          },         {              "id":
93206,              "uri": "/api/v1/articles/93206",              "curata_url":
"http://contentcuration.hivefire.com/articles/93206/the-weeklymonthly-content-
roundup/",              "share_url":
"http://contentcuration.hivefire.com/articles/share/93206/",              "url":
"http://www.velocitypartners.co.uk/our-blog/b2b-content-marketing-models-i-the-
weeklymonthly-content-roundup/",              "pub_date": "2012-06-29T07:30:59",
"title": "The weekly/monthly content roundup",              "snippet": "<p>Great
tips and notes on how to create a round up posted ...",              "image":
"http://www.velocitypartners.co.uk/wp-content/uploads/2012/06/content-roundup-
web1.jpg",              "thumbnail":
"http://contentcuration.hivefire.com/curata_ext/pictures/dgxAZlPgNAU3VOk.jpeg",
"publisher": {                  "domain": "velocitypartners.co.uk",
"name": "B2B Marketing Agency",                  "id": 6884              },
"authors": [],              "categories": [                  {
"name": "Tips & Best Practices",                      "id": 7                  }
],              "topics": [                  {                      "type":
"Topic",                       "name": "Content Marketing",
"slug": "content-marketing"                  },                  {
"type": "Topic",                       "name": "Curation",
"slug": "curation"                  }              ],              "quotes": [],
"videos": [],              "is_featured": false,              "blog": false,
"index_date": "2012-07-01T10:00:00",              "contributors": []          }
] }
```

A successful request to `/api/v1/articles` in JSON will return a JSON object with 4 properties:

| name | data type | description |
|---|---|---|
| title | string | A suggestion for the title of the results; the same as the title for the RSS feed. |
| description | string | The description of your site, as entered in Dashboard -> Share -> Site -> Settings -> About Page Text |
| total | number | The total number of articles matching the given query parameters. Note, this total is **not** the number of articles actually returned, but represents the number of articles available on all pages. To get subequent ranges of articles, use the `start`, `end`, or `count` parameters. |
| articles | array | The list of articles, ordered by `index_date`. |

Note: the ordering of properties of JSON objects is not significant, and may change based on various factors. It it a good practice not to rely on the order of properties, and most available JSON libraries will handle this for you.

A successful request to `/api/v1/articles/<article-id>` will return a JSON object with a single `article` property.

Articles returned via the API are JSON objects with the following properties:

| name | data type | description |
|---|---|---|
| id | number | A unique ID for this article within your Curata site. |
| uri | URL | A unique path in the API for this article. |
| curata_url | URL | The URL of this article on your Curata site. |
| share_url | URL | The URL of this article's share bar page on your Curata site, if enabled (see Dashboard -> Share -> Site -> Settings -> Enable Share Bar). |
| url | URL | The original publisher's URL for this article. |
| pub_date | date | The date this article was published. |
| title | string | The title of this article. |
| snippet | HTML string | The snippet of the article. |
| image | URL | The URL of a representative image for this article. Note, this URL will originate from the publisher's site. |
| thumbnail | URL | The URL of a thumbnailed version of the image for this article. Note, this URL will always originate from your Curata site. |
| publisher | object | The publisher from which this article was curated. |
| authors | list | A list of authors (JSON objects) associated with this article. |
| categories | list | A list of categories (JSON objects) associated with this article. |
| topics | list | A list of topics (JSON objects) mentioned in this article. |
| quotes | list | A list of quotes (JSON objects) extracted from this article, and the person being quoted. |
| videos | list | A list of videos (JSON objects) extracted from this article, and other data related to the videos. Video dimensions are represented as '{width}x{height}' in pixels. |
| is_featured | boolean | Indicates whether this is a featured article. |
| blog | boolean | Indicates whether this article is from your blog. |
| index_date | date | The date this article was found by Curata. |
| approval_date | date | The date this article was approved for publishing by an editor. |
| custom_fields | object | An object of key value pairs where the key is the name of a custom field as it is defined in Curata. |
| contributors | list | A list of objects, with the email address, name, date, and comment from each contributor that submitted this article to your site. |

# The Categories API

## URLs

- `http://yourcuratasite.com/api/v1/categories`, for listing the category groups and categories of your site.

## Query Parameters

By default, the Categories API without query parameters will return all category groups and their associated categories.

| parameter | data type | description |
|---|---|---|
| query | string | An (optional) string to use as a search criteria. |

## Response

The HTTP response from a Categories API call may return one of the following status codes:

| status code | reason |
|---|---|
| 200 | The request was successful. |
| 400 | There was a problem with one of the query parameters. An informational message will be returned as the body of the response, in `text/plain`. |
| 5xx | A temporary error on Curata's system. Try again after a short time. |

An example response, from http://contentcuration.hivefire.com/api/v1/categories/:

```
{     "groups": [         {             "id": 2,            "title": "Curation
Topics"             "feed_url":
"http://contentcuration.hivefire.com/api/v1/articles?in_category_group=2",
"color": "696969",            "categories": [                {
"feed_url":
"http://contentcuration.hivefire.com/api/v1/articles?in_category=10",
"title": "Case Studies",                    "curata_url":
"http://contentcuration.hivefire.com/category/10/case-studies/",
"id": 10,                    "full_name": "Curation Topics : Case Studies",
"slug": "case-studies"                 },                  {
"feed_url":
"http://contentcuration.hivefire.com/api/v1/articles?in_category=9",
"title": "Curation 101",                    "curata_url":
"http://contentcuration.hivefire.com/category/9/curation-101/",
"id": 9,                    "full_name": "Curation Topics : Curation 101",
"slug": "curation-101"                 },                  {
"feed_url":
"http://contentcuration.hivefire.com/api/v1/articles?in_category=6",
"title": "Industry & Business News",                    "curata_url":
"http://contentcuration.hivefire.com/category/6/industry-business-news/",
"id": 6,                    "full_name": "Curation Topics : Industry &
Business News",                    "slug": "industry-business-news"
},                  {                    "feed_url":
"http://contentcuration.hivefire.com/api/v1/articles?in_category=8",
"title": "Industry Trends",                    "curata_url":
"http://contentcuration.hivefire.com/category/8/industry-trends/",
"id": 8,                    "full_name": "Curation Topics : Industry Trends",
"slug": "industry-trends"                 },                  {
"feed_url":
"http://contentcuration.hivefire.com/api/v1/articles?in_category=7",
"title": "Tips & Best Practices",                    "curata_url":
"http://contentcuration.hivefire.com/category/7/tips-best-practices/",
"id": 7,                    "full_name": "Curation Topics : Tips & Best
Practices",                    "slug": "tips-best-practices"                 }
]          }      ] }
```

A successful request to `/api/v1/categories` in JSON will return a JSON object with 1 property, `groups`, which is a list of the enabled category groups of your Curata site.

Each group is a JSON object with 5 properties:

| name | data type | description |
|---|---|---|
| id | number | A unique ID for this category group within your Curata site. |
| title | string | The name of this category group. |
| feed_url | URL | The URL of the corresponding Articles API list of articles for this category group. |
| color | string | The RGB color code you assigned to this category group. |
| categories | list | The list of categories belonging to this group. |

Each category object is a JSON object with 6 properties:

| name | data type | description |
|---|---|---|
| id | number | A unique ID for this category within your Curata site. |
| slug | string | A slugged string identifying this category. |
| title | string | The name of this category. |
| full_name | URL | The name of this category, with it's group name prepended. |
| feed_url | URL | The URL of the corresponding Articles API list of articles for this category. |
| curata_url | URL | The URL of the corresponding category page within your Curata site. |

# The Publishers API

## URLs

- `http://yourcuratasite.com/api/v1/publishers`, for listing the original publishers of articles on your site.

## Query Parameters

By default, the Publishers API without query parameters will return the first 25 publishers in alphabetical order along with the total number of publishers.

| parameter | data type | description |
|---|---|---|
| query | string | An (optional) string to use as a search criteria for the domain or name of publisher. |
| name | string | An (optional) string to use as a search criteria. Matches (case insensitive) name of the publisher. |
| domain | string | An (optional) string to use as a search criteria. Matches (case insensitive) domain name of the publisher. |
| sort | string | An (optional) string by which the resulting publishers are sorted. By default it is sorted by name. Other valid options are domain or id. |
| direction | string | An (optional) string to define the direction in which the resulting publishers are displayed. By default, it is in ascending order. The options are ASC or DESC. |
| start | number | An (optional) integer denoting the starting point of the resulting publishers. |
| limit | string | An (optional) integer denoting the offset from the starting point to be returned. |

## Response

The HTTP response from a Publishers API call may return one of the following status codes:

| status code | reason |
|---|---|
| 200 | The request was successful. |
| 403 | Must log in to perform a `PUT` or `DELETE`. |
| 404 | Publisher was not found. |
| 422 | Publisher name is required. |
| 5xx | A temporary error on Curata's system. Try again after a short time. |

A successful request to `/api/v1/publishers` in JSON will return a JSON object with 2 property, `publishers` and `total`.`publishers` is a list of the returned publishers, and `total` is the total number of publishers that are available per your search criterion.

Each group is a JSON object with 5 properties:

| name | data type | description |
|---|---|---|
| domain | string | The domain name of the publisher. |
| curata_url | string | The path of the Curata URL. |
| uri | string | The path of the publisher. |
| name | string | The name of the publisher. |
| id | number | The id number of the publisher. |

# Email List and Subscriber APIs

Email newsletter mailing lists and subscribers can be managed via the API. For example, you could host a newsletter subscriber form on a non-Curata site whose server then updates Curata newsletter subscriptions. Or, you could use this API to make individual or bulk changes to subscriptions, or export lists of subscribers for use in spreadsheets or other tools.

## The Email List API

## URLs

- `http://yourcuratasite.com/api/v1/email/lists` GET listing email lists on your site, POST to create new lists.
- `http://yourcuratasite.com/api/v1/email/lists/<list-id>` GET for retrieving a single list, PUT to modify a list.

## GET Query Parameters

By default, at GET request to the Email List API will return all active lists. You can further refine your query using the following parameters:

| parameter | data type | description |
|---|---|---|
| status | string | List status. Options are `active` (default), `removed`, or `all`. |
| public | boolean | Show lists open to the public (`true`), visible only to the editor (`false`), or both (`all`, default). |
| automatic | boolean | Show lists set to send an automated newsletter `(true`), not set to send (`false`), or both (`all`, default). |
| default | boolean | Show lists that are checked by default on the new subscription form `(true`), not checked (`false`) or both (`all`, default). |
| query | string | Search lists by name and description |
| count | number | The number of lists to return. If unspecified, will return all lists. |
| start | number | The number of the first list to return (zero-based). |
| end | number | The number of the last list to return (zero-based). If both `count` and `end` are specified, `count` will override `end`. |
| sort | string | The field on which to order results. May be one of:<br>`name` (default)<br>`description`<br>`status`<br>`public`<br>`automatic` |

| parameter | data type | description |
|---|---|---|
| | | `default` |
| direction | string | Either `DESC` or `ASC` (default), for descending or ascending sort |

# GET Response

The Email List API accepts requests for JSON and JSON-P formatted responses.

The HTTP response from an Email Lists API call may return one of the following status codes:

| status code | reason |
|---|---|
| 200 | The request was successful. |
| 400 | There was a problem with one of the query parameters. An informational message will be returned as the body of the response, in `text/plain`. |
| 404 | Single email list API only; the email list does not exist. |
| 5xx | A temporary error on Curata's system. Try again after a short time. |

A successful request to `/api/v1/email/lists` will return a JSON object with three properties:

| name | data type | description |
|---|---|---|
| total | number | The total number of email lists that match the query. This may not necessarily be the same as the number of email lists returned, since you can request batches of lists with the `start`, `end`, or `count` parameters. |
| uri | URL | The Uniform Resource Identifier for this request |
| lists | array | Individual email list objects |

An individual email list object, either when part of a group or requested individually, is a JSON object with eleven properties:

| name | data type | description |
|---|---|---|
| name | string | Name of the email list |
| description | string | Description of the email list |
| id | number | ID number of the list |
| uri | URL | Uniform Resource Identifier for the list |
| subscribers_uri | URL | URI from which to retrieve the actual subscribers to this email list |
| subscriber_count | number | Number of active subscribers |
| opted_out_count | number | Number of subscribers who have opted out of when subscribed to this list |
| status | string | Whether or not the list is `active` or `removed` |
| default | boolean | Whether or not the list is checked by default on the new subsciption from |
| public | boolean | Whether or not the list is open to the public or visible only the editors |
| automatic | boolean | Whether or not the list is configured to send an automatic email newsletter |

# POST and PUT Requests

The Email Lists API will accept POST requests to create new email lists. An individual lemail list URI will also accept a PUT request to edit the existing list. In both cases, the request payload should contain a JSON object similar to that returned by a GET request. The `Content-Type` header should be set to `application/json`.

The writable fields when creating or modifying a list are:

| name | data type | description |
|---|---|---|
| name | string | Name of the email list |
| description | string | Description of the email list |
| status | string | Whether or not the list is `active` or `removed` |

| name | data type | description |
|---|---|---|
| default | boolean | Whether or not the list is checked by default on the new subsciption from |
| public | boolean | Whether or not the list is open to the public or visible only the editors |
| automatic | boolean | Whether or not the list is configured to send an automatic email newsletter |

POST and PUT requests will return a response similar to a GET request, but only containing the lists that were created or modified.

# The Email Subscriber API

## URLs

- `http://yourcuratasite.com/api/v1/email/subscribers` GET listing of email subscribers to your newsletters, POST to create new subscribers and/or modify existing subscribers in bulk.
- `http://yourcuratasite.com/api/v1/email/subscribers/<subscriber-id>` GET for retrieving a single subscriber, PUT to modify a subscriber.

## GET Query Parameters

By default, at GET request to the Subscriber API will return all active subscriber to all lists. You can further refine your query using the following parameters:

| parameter | data type | description |
|---|---|---|
| list | number | Return subscribers to a particular mailing list id. If unspecified, will return all subscribers. To request subscribers who are not on a list, pass `none`. |
| status | string | Show subscribers who are `active` (default), have been deleted (`removed`), or both (`all`). |
| opted_out | boolean | Show subscribers who have opted out. `false` (default) will return only subscribers who have not opted out, `true` will return only those who have opted out, and `all` will return all subscribers, opted out or not. |
| email | string | Show subscribers using a particular email address. Email addresses are unique, so this should only ever return a list of one subscriber. |
| query | string | Search subscribers by email address or username, if they are a registered user of your site. Accepts partial matches, so, for example, `query=@example.com` will return a list of all subscribers with with`example.com` email addresses. |
| count | number | The number of subscribers to return. If unspecified, will return all subscribers. |
| start | number | The number of the first subscriber to return (zero-based). |
| end | number | The number of the last subscriber to return (zero-based). If both `count` and `end` are specified, `count`will override `end`. |
| sort | string | The field on which to order results. May be one of:<br>`email` (default)<br>`opted_out`<br>`timestamp`<br>Any enabled field type, for example `last_name`, `first_name`, `company`, etc. |
| direction | string | Either `DESC` or `ASC` (default), for descending or ascending sort |

## GET Response

The Subscriber API accepts requests for JSON and JSON-P formatted responses. The non-individual resource also will accept Comman Separated Variable (.csv) and plain text (.txt) requests. CSV will return a .csv file with all of the same data fields as returned by the JSON type. The plain text version will return a comma separated list of the email addresses only.

The HTTP response from an Email Lists API call may return one of the following status codes:

| status code | reason |
|---|---|
| 200 | The request was successful. |
| 400 | There was a problem with one of the query parameters. An informational message will be returned as the body of the response, in `text/plain`. |
| 404 | Single subscriber API only; the subscriber does not exist. |
| 422 | Unable to modify a property of a subscriber. An explanation of the problem will be returned as the body of the response in `text/plain`. |
| 5xx | A temporary error on Curata's system. Try again after a short time. |

A successful request to `/api/v1/email/subscribers` will return a JSON object with four properties:

| name | data type | description |
|---|---|---|
| uri | URL | The URI generating this query |
| total | number | Number of subscribers returned by this query |
| enabled_field_types | object | List of the additional subscriber form fields that have been configured for this site. Fields that are required are marked as "R", optional fields ar marked as "O". |
| subscribers | array | List of individual subscribers |

An individual subscriber object, either requested as part of a group or individually, is a JSON object with six or more properties.

| name | data type | description |
|---|---|---|
| uri | URL | URI for the individual subscriber |
| email | string | Email address of the subscriber |
| username | string | If the subscriber has registered as a user of your site, the username associated with that account. Otherwise, an empty string. |
| timestamp | date/time | Date/time that when the user subscribed. |
| opted_out | boolean | Whether or not the subscriber has opted out of email from your site. |
| lists | array | ID numbers of the email lists this subscriber is subscribed to. |
| Additional Fields | string | If you have configured your Curata site to use additional required or optional fields when someone subscribes to your site, such as name, company, title, etc., these will appear as additional properties. |

# POST and PUT Requests

The Subscriber API will accept POST requests to create new subscribers or bulk edit existing subscriptions. An individual subscriber may be modified with a PUT request. In both cases, the request payload should contain a JSON object or array of objects similar to that returned by a GET request. The `Content-Type` header should be set to `application/json`.

When POSTing bulk changes to many subscribers, all email addresses must be valid before any subscription is modified. Any invalid email addresses will be listed in a plain text response. Email address is unique to each subscriber, so if you include existing subcribers email addresses in a POST, the existing subscription will be modified and not duplicated.

The writable fields when creating or modifying a subscriber are:

| name | data type | description |
|---|---|---|
| email | string | Required at create only |
| lists | array | Optional. ID numbers of email lists this subscriber should be subscribed to. To remove |

| name | data type | description |
|---|---|---|
| | | a subscriber from a particular mailing list, remove it from this list of ids. When creating a new subscriber, if this property is not set, the default lists will be used. |
| opted_out | boolean | Optional. Only use this to permanently opt a subscriber out. **Once a subscriber is opted out, you cannot undo it.** |
| Additional Fields | string | If you have configured your Curata site to use additional required or optional fields when someone subscribes to your site, such as name, company, title, etc. you can set these by adding additional parameters. |

PUT and POST requests will return a response similar to a GET request, but only containing the subscribers that were created or modified.

# The Topics API

## URLs

- `http://yourcuratasite.com/api/v1/topics`, for listing the topics your site.

## Query Parameters

By default, the Topics API without query parameters will return all of the topics in your portal.

| parameter | data type | description |
|---|---|---|
| query | string | An (optional) string to use as a search criteria. |
| sort | string | The field on which to order results. May be one of:<br>`name`, the name of the topic. |
| start | number | The number of the first topic to return (zero-based). |
| limit | number | The maximum number of topics to return. |

## Response

The HTTP response from a Topics API call may return one of the following status codes:

| status code | reason |
|---|---|
| 200 | The request was successful. |
| 400 | There was a problem with one of the query parameters. An informational message will be returned as the body of the response, in `text/plain`. |
| 5xx | A temporary error on Curata's system. Try again after a short time. |

An example response, from http://contentcuration.hivefire.com/api/v1/topics/:

```
{     "topics": [          {              "thumbnail": null,              "name":
"ABC",              "id": 13,              "description": "An alphabet song is
any of various songs used to teach children the alphabet, used in
kindergartens, pre-schools and homes around the world.",              "type":
"Topic",              "slug": "abc",              "custom_fields": {},
"aliases": [                  "ABC"              ]          },          {
"thumbnail": null,              "name": "Aberdeen",              "id": 58949,
"description": "Research and analysis of information technology companies and
products. Many free downloads.",              "type": "Organization",
"slug": "aberdeen",              "custom_fields": {},              "aliases": [
"Aberdeen"              ]          },          {              "thumbnail": null,
"name": "Algorithmic Curation",              "id": 21275,
"description": "",              "type": "Topic",              "slug":
```

```
"algorithmic-curation",             "custom_fields": {},             "aliases":
[                  "Algorithmic Curation"              ]          },      ] }
```

A successful request to `/api/v1/topics` in JSON will return a JSON object with 1 property, `topics`, which is a list of the enabled topics on your Curata site.

Each topic is a JSON object with 8 properties:

| name | data type | description |
|---|---|---|
| id | number | A unique ID for this topic within your Curata site. |
| name | string | The name of this topic. |
| description | string | A description of the topic. |
| thumbnail | URL | A link to a thumbnail image associated with the topic. |
| type | string | The type of the topic. The default topic types include `person`, `topic` and `organization`, but custom types can also be added. |
| slug | string | The shortened form of the topic name used for URLs. |
| aliases | array | A list of aliases for the topic, including the value of the `name` property. |
| custom_fields | object | A JSON object containing the names and values of any custom fields associated with the topic. |