LeadConduit HTTP Application Programming Interface (API)

The Basics

The RESTful LeadConduit Statistics Service API is implemented as JSON over HTTPS.

Authentication

When you're using the API, it's always from the perspective of your account in LeadConduit. You are only able to access the statistics you can normally see while logged into LeadConduit. Authentication is done using HTTP Basic with the same credentials you use to log into LeadConduit. If you would rather not use your own email and password, you may create a new user in your LeadConduit account and use that one instead.

Here's how you would call the API using curl:

```
curl -u api@yourcompanyname.com:YourAPIPassword \
https://api.leadconduit.com/stats/financial/today
```

Remember that anyone who has an email address and password in your LeadConduit account can read stats via the API, just as they can when logged into LeadConduit. Protect your password closely. If you fear that your password has been compromised, just reset it in LeadConduit. Calls to the API will have to be updated to use your new password.

Implementation note: Some HTTP Client libraries (Jakarta HttpClient, for example), do not send Basic credentials with every request by default. Instead, they send a request without credentials, and if a HTTP 401 is received, they reissue the request with the credentials. Please make the necessary configuration adjustments to ensure that Basic credentials are sent with every HTTP request.

Financial Statistics API

Using the financial statistics API your software may fetch the number of good and bad leads by date, lead source, lead recipient, or any combination of the three.

What resources are available?

The following simple HTTP resources can be used to access the API. Each of the below calls filters the statistics based on named date or date range. These calls are provided as a convenience, and specific date ranges can be specified (more on that later).

```
GET https://api.leadconduit.com/stats/financial/today
GET https://api.leadconduit.com/stats/financial/yesterday
GET https://api.leadconduit.com/stats/financial/last_7_days
GET https://api.leadconduit.com/stats/financial/this_month
GET https://api.leadconduit.com/stats/financial/last_month
GET https://api.leadconduit.com/stats/financial/YYYY
GET https://api.leadconduit.com/stats/financial/YYYY/MM
GET https://api.leadconduit.com/stats/financial/YYYY/MM/DD
GET https://api.leadconduit.com/stats/financial/all
```

What parameters may be used?

The following parameters may be used to filter the result set: `campaign_id`, `source_id`, and `recipient_id`. For the `/stats/financial/all` resource, you may also use the `start` and `end` parameters. See the parameter glossary for more information on these parameters.

The following additional parameters can be used to further customize the results returned by the financial statistics API:

- `verbose` – (“false”, “true”) Default is “false”. Set to “true” to return statistics results even if there is no lead flow. Setting this parameter to “true” can cause a very large result set to be returned from the API. Omitting the verbose parameter will cause statistics to be returned with 0 (zero) in the total column.
- `prices` – (“all”, “all_rolled_up”) By default (“all”), a new statistics row will be shown for each unique combination of purchase price and sale price. For example, if you lower the payout for one of your affiliates, you will see two rows in the financial results: one for all the leads purchased at the higher price, and one for leads purchased at the lower price. Setting this parameter to “all_rolled_up” will cause one row to be returned. The purchase_price column and the sale_price columns will display the range of prices collected.

What does the response look like?

All calls to the financial statistics API return a JSON hash with two keys: `count` and `items`. The `count` is the number of items being returned. The `items` key contains an array of records.

```
curl -v –uyou@yourcompany.com:yourpassword \
https://api.leadconduit.com/stats/financial/today?source_id=987654321

HTTP/1.1 200 OK
Date: Tue, 03 Mar 2009 21:04:52 GMT
Server: Apache/2.2.3 (Red Hat)
Content-Length: 57262
Connection: close
Content-Type: application/json; charset=utf-8

{“count”: 2,
 “items”:[
   {“campaign”:“Your campaign”,
    “campaign_id”:”123456789”,
    “source”:“Your affiliate”,
    “source_id”:”987654321”,
    “recipient”:”Advertiser”,
    “recipient_id”:”abc123456”,
    “profile”: “”,
    “profile_id”: “”,
    “total”: 15
    “billable”,10,
    “non_billable”:5
    “invalid”:2,
    “rejected”:1,
    “returned”:2,
    “purchase_price”:3.0,
    “sale_price”:5.0,
    “cost”:30.0,
    “revenue”:50.0,
    “profit”:20.0},
   {...}
  ]
}
```

What do `items` represent?

As you can see, each element of the `items` array contains a hash with the following keys:

- `campaign` - the campaign name
- `campaign_id` - the ID of the campaign
- `source` - the lead source (affiliate) name

- `source_id` - the entity ID of the lead source (affiliate)
- `recipient` - the lead recipient (advertiser) name
- `recipient_id` - the entity ID of the lead recipient (advertiser)
- `purchase_price` - cost per lead in USD
- `sale_price` - revenue per lead in USD
- `profile` – if lead profiling is set up, this is the unique profile shown as a concatenation of lead field values.
- `total` - number of good and bad leads provided by the source and sent to the recipient at the purchase and sale prices
- `billable` - number of billable leads provided by the source and sent to the recipient at the purchase and sale prices. Only "good" leads are billable.
- `non_billable` - number of non-billable leads provided by the source and sent to the recipient at the purchase and sale prices. "Bad" leads are not billable.
- `invalid` – number of non-billable leads that failed the campaign's validation criteria and thus were marked bad by LeadConduit.
- `rejected` – number of non-billable leads that were rejected during delivery from LeadConduit to the advertiser's server.
- `returned` – number of non-billable leads that were returned by the advertiser after having been originally accepted.
- `cost` - total cost for the billable leads in USD (purchase_price * billable)
- `revenue` - total revenue for billable leads in USD (sale_price * billable)
- `profit` - total profit for billable leads in USD (revenue - cost)

Each hash can be considered a "row" as you might think of a record in a database. Each hash has a unique combination of the following columns: `campaign_id`, `source_id`, `recipient_id`, `purchase_price`, `sale_price`, and `profile`. That is to say, if the lead pricing changes you will see two hashes with the same campaign, source and recipient, but with different pricing. This feature makes it possible for you to understand how many leads were purchased from whom, and sold to whom at what pricing levels.

How does lead profiling work?

The `profile` is a unique value that's generated from all the profiled campaign fields. So if you're profiling the `debt_amount` and `zip_code` field, then your profile might look like: `25000|78703`. The statistics can be broken out by `profile`, so you can see how many good leads and bad leads came through for each `profile`.

You can control the stats results by providing the `profile_id` parameter. The default value for this parameter is `all`, which will give you one row per unique `profile`. This is equivalent to passing nothing.

To have the API roll up all profiles into one "row," just pass `profile_id=all_rolled_up`. When you do this, you are indicating that you don't care about the profile breakdown.

You can also pass the specific `profile` you're interested in seeing. For example, if you want to see how many leads with $25K in debt from 78703 came from each provider, you could pass `profile_id=25000|78703&source_id=all`.

Leads API

Using the Leads API you can read lead data out of LeadConduit based on campaign, date, and lead status. You can also reject, return, or mark leads as converted and delivered via this API.

Implementation note: This API is not designed to be polled frequently, so please exhibit responsible behavior when calling the API in a loop. A good rule of thumb if you're polling for changes to a group of leads is to limit the necessary calls to once or twice per day. Evening hours are better than mid-day. You may use multiple threads to simultaneously access the API, but please limit the number of concurrent calls to three.

What resources are available?

The following resources are available to read leads out of LeadConduit.

```
GET https://api.leadconduit.com/leads
```

This resource lists leads. When making this call you must provide the `campaign_id` parameter. This resource is automatically paged on your behalf. Please see the section on result paging for more information on how this works.

```
GET https://api.leadconduit.com/leads/#{lead_id}
```

This resource refers to a single lead. You must replace `#{lead_id}` with the identifier of the lead you wish to retrieve. This value refers to the LeadConduit lead ID.

```
GET https://api.leadconduit.com/leads/unsent
```

This resource lists all leads that have not yet been sent to the advertiser. These leads will always be "good." Leads that have been sent but could not be delivered do not show up in this list. This resource is automatically paged.

```
POST   https://api.leadconduit.com/leads/#{lead_id}/delete
DELETE https://api.leadconduit.com/leads/#{lead_id}
```

Deleting lead is an asynchronous operation so the API returns HTTP 202 indicating that the lead will be deleted shortly (any other code indicates failure). Once the lead is deleted, it will no longer be available to you or your lead sources. The statistics in the appropriate campaign will be adjusted in such a way that it will appear that the lead was never submitted. You may only delete leads in campaigns that you own.

```
POST https://api.leadconduit.com/leads/#{lead_id}/reject
```

Rejecting a lead has no effect unless the lead is pending delivery. If the lead has already been delivered, use the lead return API call instead. The `reason` parameter specifies why the lead is being rejected. The `reason` parameter has a 75 character maximum length and will be truncated if it exceeds that length). This is an asynchronous operation so the API returns HTTP 202 indicating that the operation will be completed shortly (any other code indicates failure). Your lead sources, when logged into their LeadConduit accounts, will be able to see a summary of how many leads were rejected by reason.

```
POST https://api.leadconduit.com/leads/#{lead_id}/rollback_rejection
```

Use an HTTP post to the above URL to undo a lead rejection. If the lead rejection is successfully rolled back, the API returns HTTP 200. Any other code indicates that the lead rejection was not rolled back. Only the advertiser or owner of the lead's campaign may rollback a lead rejection and calls to this resource by someone else will generate an HTTP 401 Unauthorized.

```
POST https://api.leadconduit.com/leads/#{lead_id}/return
```

Use an HTTP post to the above URL to return a lead to the lead source that provided it. Use the `reason` parameter to specify why the lead is being returned. The `reason` parameter has a 75 character maximum length and will be truncated if it exceeds that length). If the lead is successfully returned, the API returns HTTP 200. Any other code indicates that the lead was not returned. Only the advertiser or owner of the lead's campaign may return the lead and calls to this resource by someone else will generate an HTTP 401 Unauthorized. Your lead sources, when logged into their LeadConduit accounts, will be able to see a summary of how many leads were returned by reason.

```
POST https://api.leadconduit.com/leads/#{lead_id}/rollback_return
```

Use an HTTP post to the above URL to undo a lead return. If the lead return is successfully rolled back, the API returns HTTP 200. Any other code indicates that the lead return was not rolled back. Only the advertiser or owner of the lead's campaign may rollback a lead return and calls to this resource by someone else will generate an HTTP 401 Unauthorized.

```
POST https://api.leadconduit.com/leads/#{lead_id}/convert
```

Use an HTTP post to the above URL to mark a lead as converted. The optional `date` parameter may be provided in order to specify the date the lead was converted. If no `date` parameter is provided, today's date will be used. Several date formats can be accepted (most callers are using `YYYY-MM-DD`). Dates that cannot be parsed or dates that are in the future will result in an HTTP 400.

Marking the lead as converted will cause it to appear in the lead conversion report, available to the campaign owner and the campaign advertiser. This is a synchronous operation. If the API returns HTTP 200, then the lead has been successfully marked as converted. Any other code indicates the lead was not returned. Only the advertiser or owner of the lead's campaign may return the lead. Calls to this resource by someone else will generate and HTTP 401 Unauthorized. A summary of how many leads were converted is only accessible by the campaign owner or advertiser.

```
POST https://api.leadconduit.com/leads/#{lead_id}/rollback_conversion
```

Use an HTTP post to the above URL to undo a lead conversion. If the lead conversion is successfully rolled back, the API returns HTTP 200. Any other code indicates that the lead conversion was not rolled back. Only the advertiser or owner of the lead's campaign may rollback a lead conversion and calls to this resource by someone else will generate an HTTP 401 Unauthorized.

What parameters may be used?

There are two ways to call the leads API. The first is performs a query for leads based on certain criteria. For these calls, you must provide the `campaign_id` parameter. The `source_id`, `recipient_id`, `start`, and `end` parameters are all optional.

You may also use the `status` parameter to filter leads based on their current disposition. Possible values for `status` are:

- `good` – include leads that have not yet been marked “bad”
- `invalid` – include leads that were marked bad according to campaign constraints when they were posted into LeadConduit
- `rejected` – include leads that were marked rejected during delivery to the advertiser’s server
- `returned` – include leads that were marked bad after delivery to the advertiser’s server

You may specify multiple `status` values in order to fetch various combinations of leads. For example, `status=rejected&status=returned` will fetch all leads that were either rejected when they were delivered to the advertiser, or after they were delivered to the advertiser. Leads that failed campaign validation will not be included.

The second way to list leads is by providing the `source_lead_id` parameter:

```
GET https://api.leadconduit.com/leads?source_lead_id=000000aef
```

This will return a list of leads that were created during a campaign-to-campaign lead delivery, based on the ID of the lead in the “source” campaign. This call is useful if you have a lead ID and want to find associated leads in “destination” campaigns.

The lead query parameters are not available when looking up leads by `source_lead_id`, so providing `campaign_id`, `source_id`, `recipient_id`, etc. will have no effect.

What does the response look like?

All calls to the leads API return a JSON hash with five keys: `count`, `page_count`, `has_more`, `campaign_id`, and `items`. The `count` is the number of items being returned. For more information on `page_count` and `has_more`, see the section on result paging. The `items` key contains an array of lead records.

```
curl -v -uyou@yourcompany.com:yourpassword \
https://api.leadconduit.com/stats/leads?campaign_id=000qxm169&start=2009-03-04&end=2009-03-04

HTTP/1.1 200 OK
Date: Tue, 03 Mar 2009 21:04:52 GMT
Server: Apache/2.2.3 (Red Hat)
Content-Length: 572620
Connection: close
Content-Type: application/json; charset=utf-8

{"campaign_id":"000qxm169",
 "count":277,
 "page_count":6,
 "has_more":false,
 "items":[
   {"id":"011c5g8ch",
    "campaign_id":"000qxm169",
    "campaign_name":"Widgets Offer",
    "recipient_id":"e55369",
    "source_id":"000e7hs65",
    "created_at":"2009/03/04 15:42:19 +0000",
    "good":false,
    "bad":true,
    "returned":true,
    "invalid":false,
    "rejected":false,
    “reasons”:[“email bounced”]
    "delivered":true,
    "delivery_summary_id":1234567890,
    "error_delivery_summary_id":null,
    "values":{ "lname":"Doe",
               "fname":"John",
               "email":"jdoe@yahoo.com" }
    "purchase_price":0.5,
    "sale_price":1.0,
    "profit":0.40,
    "margin":100,
    “source_lead_id”:null,
    "wiped_at":null,
    "pickled_at":null },
    {...},
    {...}, ...
  ],
}
```

Campaigns API

The campaigns API allows you to pull a simple list of all the campaigns which are visible in your account:

```
GET https://api.leadconduit.com/campaigns
```

What does the response look like?

```
curl -v –uyou@yourcompany.com:yourpassword \
https://api.leadconduit.com/campaigns

HTTP/1.1 200 OK
Date: Tue, 03 Mar 2009 21:04:52 GMT
Server: Apache/2.2.3 (Red Hat)
Content-Length: 572620
Connection: close
Content-Type: application/json; charset=utf-8

{"count":3,
 "items":[
    {"id":"000qxm170",
     "name":"Foo Campaign"},
    {"id":"000qxm171",
     "name":"Bar Campaign"},
    {"id":"000qxm172",
     "name":"Baz Campaign"}
  ]
}
```

How can I fetch information about the campaign’s fields?

Examining the list of campaign fields can be accomplished by calling the following resource, where `#{campaign_id}` must be replaced with the ID of the campaign you wish to read:

```
GET https://api.leadconduit.com/campaigns/#{campaign_id}/fields
```

Sources and Recipients API

The campaigns API allows you to pull a simple list of all the lead sources (affiliates) or lead recipients (advertisers) which are visible in your account:

```
GET https://api.leadconduit.com/sources
GET https://api.leadconduit.com/recipients
```

What does the response look like?

The responses for sources and recipients are the same:

```
curl -v –uyou@yourcompany.com:yourpassword \
https://api.leadconduit.com/sources

HTTP/1.1 200 OK
Date: Tue, 03 Mar 2009 21:04:52 GMT
Server: Apache/2.2.3 (Red Hat)
Content-Length: 572620
Connection: close
Content-Type: application/json; charset=utf-8

{"count":2,
 "items":[
    {"id":"000qxm180",
     "name":"Affiliate Number 1"},
    {"id":"000qxm181",
     "name":"Affiliate Number 2"}
 ]
}
```

You may pass the optional `campaign_id` parameter in order to fetch the sources or recipients for a specific campaign.

Paged Responses

Certain API calls return paged data sets. Each page of data contains 50 or fewer results. You may control which page you are reading by passing the `page` parameter as an integer representing the page number. Paged responses contain several special values to help your program more easily utilize this feature.

- `count` – the total number of records across all pages.
- `page_count` – the number of total pages available in this data set. If the data set contains 275 records, page_count will be 6 (50 leads on the first 5 pages, and 25 leads on the 6th page).
- `has_more` – true if there are additional pages beyond the current page, otherwise, false.

API Parameter Glossary

The following parameters are used across various LeadConduit APIs.

- `campaign_id` – Limit the results to this campaign
- `source_id` – Limit the results to this lead source (affiliate)
- `recipient_id` – Limit the results to this lead recipient (advertiser)
- `start` – Limit the results to those after this date (yyyy-mm-dd)
- `end` – Limit the results to those before this date (yyyy-mm-dd)
- `profile` – Limit the results to leads with this profile

A note about parameters

Note that all parameters are case sensitive. Providing `Campaign_ID` or `campaign_Id` instead of `campaign_id` will result in that parameter being disregarded. Similarly, when passing IDs of any kind, failure to provide a valid ID is will cause zero results to be returned.