

Live Feed API

The data provider is required to furnish us with a URL which will accept a HTTP POST. This is a 'callback' target for the analysis results of records which are accepted for processing.

POST URL:

POST to `http://host:port/api/live/import`

Should a record be a duplicate or not verifiable the sending process will receive an immediate negative response. If the record is "accepted for processing", the sending process will receive a response containing a 'reference' field and the final results will be POSTed to the URL you provide at setup. This response will contain also the reference.

Important Information:

Live feed results cannot be 'segmented' in the same manner as batch processing. It is up to the implementor to examine the results and handle their data accordingly.

We recommend only contacting results with the following properties:

1. subscriber_score_bucket of 'high'
2. subscriber_email_status of 'verified'
3. No blacklist hits (examine the following fields: subscriber_email_is_blacklisted, subscriber_email_domain_is_blacklisted, subscriber_signup_domain_is_blacklisted, subscriber_signup_ip_is_blacklisted)

POST Fields:

| Field | Description |
|---------------------------------|--|
| auth | Authentication Token required |
| subscriber_email | Email Address required |
| subscriber_fname | First Name suggested |
| subscriber_mname | Middle Name or Initial(s) |
| subscriber_lname | Last Name suggested |
| subscriber_addr1 | Address Line 1 suggested |
| subscriber_addr2 | Address Line 2 |
| subscriber_phone | Phone Number |
| subscriber_mobile | Mobile Phone Number |
| subscriber_fax | Fax Number |
| subscriber_city | City suggested |
| subscriber_state | State/Region Code suggested |
| subscriber_zip | Zip/Postal Code suggested |
| subscriber_country | Country Code (2 character) suggested |
| subscriber_gender | Gender (M/F) suggested |
| subscriber_dob | Date of Birth. "YYYY-MM-DD" format suggested |
| subscriber_signup_ip | Sign up IP Address required |
| subscriber_signup_tstamp | Sign up Time stamp "YYYY-MM-DD hh:ii:ssZ" Format required |
| subscriber_signup_url | Sign up Web page URL required |
| subscriber_last_activity_tstamp | Last Activity Time stamp "YYYY-MM-DD hh:ii:ssZ" Format suggested |

Values should be correctly encoded using percent encoding as per <http://en.wikipedia.org/wiki/Percent-encoding>

HTTP Response Status Codes:

1. 200 - OK : Record was accepted for processing
2. 400 - Bad Request : There is a problem with the request. See the response.
3. 402 - Payment Required : The account you're trying to send data to cannot accept data at this time
4. 406 - Not Acceptable : Rate limit for API has been hit
5. 408 - Timeout : It took too long to process the request, try again later
6. 409 - Conflict : The record was rejected, see the 'error' and 'msg' fields in the response as to why it was rejected
7. 500 - Internal Server Error : Something went terribly wrong, try again later

HTTP Response Body:

Error Response (not 200), the client should receive a JSON response in the following format.

```
{
  "status"    : STRING("error"),
  "error"     : STRING(ERROR_CODE),
  "msg"       : STRING(ERROR_MESSAGE)
}
```

ERROR_CODE is one of:

1. operation_timeout
2. rejected
3. duplicate
4. bad_parameter
5. rate_limit

ERROR_MESSAGE contains a verbose error message.

Success Response (200), the client should receive a JSON response in the following format. The final accept/reject message will be POSTed to the URL you provided at setup. This response will contain the reference and the email address.

```
{
  "status"    : STRING("accepted_for_processing"),
  "msg"       : STRING(STATUS_MESSAGE),
  "reference" : INTEGER
}
```

Callback:

The results will be POSTed to the URL you provided at setup. The system will attempt to POST the response up to 15 times. The system expects to receive any of the following HTTP status codes: 200, 201 or 202.

The analysis results will be in the following format

```
{
  "reference"      : INTEGER,
  "email"         : STRING(EMAIL_ADDRESS),
  "status"        : STRING(EMAIL_STATUS),
  "msg"           : STRING(MESSAGE),
  "score"         : INTEGER(-infinity to 100),
  "results"       : JSON_OBJECT(REULTS_DATA)
}
```

EMAIL_ADDRESS : The normalized email address send in

EMAIL_STATUS : One of verified, hardbounce, mboxfull, indeterminate

MESSAGE : Explanatory message

SCORE : The higher the score, the better the record

RESULTS : A JSON object containing the detailed result (see below)

```
{
  "subscriber_score_bucket"           : STRING(high | medium | low | bad),
  "subscriber_score"                  : INTEGER(),
  "subscriber_score_reason"           : STRING(),
  "subscriber_email"                  : STRING(EMAIL),
  "subscriber_email_status"           : STRING(EMAIL_STATUS),
  "subscriber_fname"                  : STRING(),
  "subscriber_mname"                  : STRING(),
  "subscriber_lname"                  : STRING(),
  "subscriber_addr1"                  : STRING(),
  "subscriber_addr2"                  : STRING(),
  "subscriber_phone"                  : STRING(),
  "subscriber_mobile"                 : STRING(),
  "subscriber_fax"                    : STRING(),
  "subscriber_city"                   : STRING(),
  "subscriber_state"                  : STRING(),
  "subscriber_zip"                    : STRING(),
  "subscriber_country"                : STRING(),
  "subscriber_gender"                 : STRING(),
  "subscriber_dob"                    : STRING(),
  "subscriber_signup_tstamp"          : STRING(),
  "subscriber_signup_ip"              : STRING(),
  "subscriber_signup_url"             : STRING(),
  "subscriber_signup_category"        : STRING(),
  "subscriber_last_activity_tstamp"   : STRING(),
  "subscriber_email_is_blacklisted"   : BOOLEAN(),
  "subscriber_email_blacklisted_reason" : STRING(),
  "subscriber_email_domain"          : STRING(),
  "subscriber_email_domain_is_blacklisted" : BOOLEAN(),
  "subscriber_email_domain_blacklisted_reason" : STRING(),
  "subscriber_signup_domain_is_blacklisted" : BOOLEAN(),
  "subscriber_signup_domain_blacklisted_reason" : STRING(),
  "subscriber_signup_ip_is_blacklisted" : BOOLEAN(),
  "subscriber_signup_ip_blacklisted_reason" : STRING(),
  "subscriber_email_domain_parked"    : STRING(),
  "subscriber_email_domain_lawsite"   : BOOLEAN(),
  "subscriber_email_count_seen"       : INTEGER(),
  "subscriber_email_has_online_profile" : BOOLEAN(),
  "subscriber_email_has_greylis"     : BOOLEAN(),
  "subscriber_signup_domain_parked"   : STRING(),
  "subscriber_signup_domain_lawsite"  : STRING(),
  "subscriber_signup_ip_geo_country"   : STRING(),
  "subscriber_signup_ip_geo_region"   : STRING(),
  "subscriber_signup_ip_geo_city"     : STRING(),
  "subscriber_signup_ip_geo_zip"      : STRING(),
  "subscriber_reference"              : STRING()
}
```

Example Callback processing script

Below is a simple example of how the callback could be handled on your server. It is most likely that implementor would want to process and store the results and not just echo them out.

```
<?php
// To extract JSON POST data you need to use php://input
$post_data = file_get_contents('php://input');
$json = json_decode($post_data);

$error = "";
if (isset($json->{"error"})) {
    $error = $json->{"error"};
}
$email      = $json->{"email"};
$status     = $json->{"status"};
$msg        = $json->{"msg"};
$score      = $json->{"score"};
$results    = $json->{"results"};

// We want to get the email status, the score bucket and
// all blacklist flag fields from the results section
$score_bucket      = $results->{"subscriber_score_bucket"};
$email_status      = $results->{"subscriber_email_status"};
$email_is_blacklisted = (bool) $results->{"subscriber_email_is_blacklisted"};
$email_domain_is_blacklisted = (bool) $results->{"subscriber_email_domain_is_blacklisted"};
$signup_domain_is_blacklisted = (bool) $results->{"subscriber_signup_domain_is_blacklisted"};
$signup_ip_is_blacklisted = (bool) $results->{"subscriber_signup_ip_is_blacklisted"};

$is_blacklisted = in_array(true, array($email_is_blacklisted,
                                       $email_domain_is_blacklisted,
                                       $signup_domain_is_blacklisted,
                                       $signup_ip_is_blacklisted));

// First case is when the record should be safe to contact
if ($subscriber_score_bucket == 'high'
    && $subscriber_email_status == 'verified'
    && ! $is_blacklisted) {
    // Do whatever you want with the save result
    print "SAFE\n";
    var_dump($json);
} else {
    // result did not meet our safety criteria
    print "NOT SAFE\n";
    var_dump($json);
}
?>
```