Status of Email Registration Requests

eEvidence Web Services

JSON, or JavaScript Object Notation, is an open standard format that uses human-readable text to transmit data objects consisting of **key:value** pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML.

The basic query

The simplest eEvidence JSON query looks like this:

https://secure.eevid.com/srv/jlist/{private.key}/{start.date}/{end.date}, where

/srv/jlist/	Base URI for the JSON web service
/{private.key}	User query private key[1]
/{start.date}	Starting date for the records to be retrieved in YYYYMMDD format
/{end.date}	Ending date for the records to be retrieved in YYYYMMDD format

Example: https://secure.eevid.com/srv/jlist/**0123456789_ABCDEFGH**/**20130301**/**20130530**

Once implemented, the query returns the relevant data for non-delivered emails within the date range, following the following schema:

Key	Value
id:	eEvid ID
type:	"OUT" = request for registering outgoing email, "IN" = request for registering incoming email, "FILE" = request for registering body and attachments only[2]
xfield:	email ID at source[3]
date:	delivery date in YYYY-MM-DD hh:mm:ss format
subject:	email subject
to:	email recipient
af:	attachment filenames
rcode:	SMTP transmission diagnosis code
rlog:	full reply message at destination
status:	"S" = Confirmed, "D"= Confirmed with annotation[4], "P" = Pending delivery[5], "E" = Failed, "H" = Half-done[6]
tx_id:	secondary ID required for downloading the eEvid.Cert via GET.EEVID web service

[1] Each user belonging to an E+ Unlimited license –including the license's administrator–, has a unique and personal {private.key} assigned. The {private.key} can be found by accessing the user private area, under My Account.

[2] Emails sent to register@eevid.com.

[3] Only if the X-eEvid-SourceID field has been found in the email header. Otherwise, this value will be set to "0" (zero).

[4] eEvidence may receive a later notice regarding a delivered email: non delivery postmaster errors, out of office warnings, etc. If a later notice is received, we will keep a copy, forward it back to the user and register it by changing the eEvid status from "S" to "D".

[5] Email delivery requests not transmitted within the first 2 hours are cancelled and bounced back to the sender.

[6] The eEvid.Cert misses the printed version of the email message body for emails with a wrong email formatting successfully delivered to the intended recipients.

The data is returned in raw format, each value identified with its corresponding label. In order to deal with it, the raw data needs to be parsed (e.g. http://json.parser.online.fr). Example:

RAW Data


```
[{"id":"1FDD1282E343","type":"OUT","xfield":"0
","date":"2013-05-14 13:07:20","subject":"Test
1","to":"toanyname@example.com","af":["file01"
,"file02"],"rcode":500,"rlog":" SMTP error
from remote mail server after RCPT TO: host
mx1.example.com [12.34.56.78]: 550 Requested
action not taken: mailbox
unavailable","status":"D","tx_id":"e05b9019-
e67b-47a5-9ª27-
71f5er9813bm"},{"id":"1FDD12C33544",...}]
```


Parsed Data


```
{
    "id":"1FDD1282E343",
    "type":"OUT",
    "xfield":"0",
    "date":"2013-05-14 13:07:20",
    "subject":"Test 1",
    "to":"toanyname@example.com",
    "af":["file01","file02"],
    "rcode":500,
    "rlog":"SMTP error from remote mail
    server  after RCPT TO: host
    mx1.example.com [12.34.56.78]: 550
    Requested action not taken: mailbox
    unavailable",
    "status":"D",
    "tx_id":"e05b9019-e67b-47a5-9ª27-
    71f5er9813bm",
},
{
    "id":"1FDD12C33544",
    . . . . . .
    . . . . . .
    . . . . . .
    . . . . . .
```


Parsing the af data array

The **af** field contains the filenames of the files attached to he original email, if any. This information is provided as a data array, which can be treated as an object.

A python example:

```
jdata=json.loads(data)
for atts in jdata["af"]:
   print atts
```

Understanding the rcode/rlog relationship

These two fields provide you the necessary information to whether the email has been delivered and, if not, why was so.

Through the **rcode** field, we inform you about the SMTP transmission diagnosis code following the RFC 2821 standard. In most cases the **rcode** field may match the first 3 characters of the **rlog** field, but this may not always be the case. Some MTAs may simply not follow the RFC 2821 recommendations on the matter, returning undocumented Reply Codes, or we may decide to assign an **rcode** value when we can't connect with the destination MTA.

Use the **rcode** value as an indication to whether some action should be taken on your side.

rcode: 250 — Positive delivery, no further action required
421 — Temporary reject, you should try later
450 — Aborted, you could try later
500 — negative delivery, not worth retrying

The **rlog** field registers the information provided by the destination MTA following the final Reply Code. For further details on any **rcode** value, review the **rlog** information.

Optional parameters

The most elementary query is limited to the four basic parameters and does not return records for emails successfully delivered and certified.

An optional parameter can be used to retrieve all records, regardless of their status:

/full/	Retrieve all records

Example: https://secure.eevid.com/srv/jlist/0123456789_ABCDEFGH/20130301/20130530/**full**

A reason to retrieve all records would be to make sure you get back data for all expected records. If, whatever the reason is, an email never reached us, you can then detect no value is being returned for that particular record.

Testing JSON

We have in place a demo JSON web service for beta testing:

https://secure.eevid.com/srv/**jdemolist**/{private.key}/{start.date}/{end.date}

Querying the demo JSON web service will return real data, limited to 5 records only.

Searching the status of a given email

None of the above can be used to check for the status of a given record. This can be achieved through a specific JSON web service:

https://secure.eevid.com/srv/**search**/{private.key}/{search_field}/{search_value}

where:

/srv/search/	Base URI for the JSON web service
/{private.key}	User query private key
/{search_field}	"xfield" or "eevid"
/{search_value}	X-eEvid-SourceID or returned eEvid ID

Example: https://secure.eevid.com/srv/**search**/**0123456789_ABCDEFGH**/**xfield**/**28GAH893NJ4**

Managing errors

The JSON web service query may encounter several errors, returned as follows:

When the {private.key} does not match any existing user:
{"id": "-1","error":"1.0 User not found"}

When an eEvid.ID or an X-Field value provided within the search query does not exist:
{"id": "-1","error":"1.1 Record not found","value":"*eEvid.ID or X-Field*"}

When the query contains unexpected values:
{"id": "-1","error":"1.4 Search fields are not valid"}

When the record does not belong to the user, identified with the {private.key}:
{"id": "-1","error":"2.1 Invalid credentials"}

Some limits on the number of queries per hour can be eventually set up in the future. If this is ever implemented, abusing the JSON web service will result in the following error:

{"id": "-1","error":"error description"}

It's highly recommended to foresee this specific errors when implementing the JSON web service capabilities.

Key considerations

Who can query what

The JSON web service will only return records associated to the {private.key} included in the query.

However, when querying for a specific email, the {private.key} can either be the user's who supposedly sent that email, or the license's user administrator's.

Query limitations

For now, the JSON web service can be queried without any limitation on the number of queries per time period. There's either no limit on the date range defined on the query: whatever this is, only a maximum of 10,000 records will be returned, from newest to oldest.

X-Field records matching tip

The returned {xfield} field will contain whatever data we have encountered in the email header, labeled as X-eEvid-SourceID (non standard x-field). This field is meant to allow a quick match between our records and the customer's system records: to do so, the X-eEvid-SourceID label should contain the customer's system's unique ID for that particular email.

{xfield} maximum length	60 characters
{xfield} expected characters	A-Za-z0-9 alphanumeric characters

External references

JSON from the Wikipedia
http://en.wikipedia.org/wiki/JSON