

# Data Router Release 1

## Log Query API

---

### Background

The Data Router (DR) Log Query API is an HTTPS-based API that allows client systems to get information about DR publication and delivery activity. The API is intended primarily for end users of the DR—publishers and subscribers—to access records related to activity on their feeds and subscriptions. While DR operations staff may find the API useful, the API is not designed to be an operations tool and specifically does not aim to replace the use of traditional application log files and application monitoring tools. The API is designed to provide information about the history of activity, not real-time status. Information about an activity may not become available through the API until several minutes after the activity has occurred<sup>1</sup>.

The DR creates three types of records that it exposes through the Log Query API, corresponding to three types of events:

- The DR creates a *publish attempt* record when a publisher makes a PUT or DELETE request using the File Publishing and Delivery API<sup>2</sup>.
- The DR creates a *delivery attempt* record when the DR attempts to deliver a file (using a PUT request) or a retraction (using a DELETE request) to a subscription delivery endpoint using the File Publishing and Delivery API.
- The DR creates an *expiry* record when the DR determines that it should not make any further delivery attempts for a file or a retraction for a subscription.

The Log Query API allows a client system to retrieve these types of records, with the ability to filter the set of retrieved records using query parameters.

### Authentication and Authorization

In DR Release 1 (R1), there is no authentication and authorization associated with Log Query API requests. In later releases, this is likely to change.

---

<sup>1</sup> Each DR node will send its log data periodically to the server hosting the Log Query API. The exact interval has not been set yet, but it is likely to be in the range of 5 to 15 minutes. The results returned by the Log Query API will therefore lag behind the actual activity.

<sup>2</sup> *Data Router R1 File Publishing and Delivery API*, Version 1.1, 2013-03-18.

---

The API will use HTTPS. The DR will use a server certificate signed by Verisign and issued under AT&T Services , Inc.

## API Specification

### Accessing the API

When a user creates a feed using the DR Provisioning API<sup>3</sup>, the DR generates a URL (called the feed log URL and given the symbolic name *<feedLogUrl>*) that can be used to access the Log Query API to get log information associated with the feed. Similarly, when a user creates a subscription to a feed, the DR generates a URL (called the subscription log URL and given the symbolic name *<subLogUrl>*) that can be used to get log information associated with the subscription.

### Making Log Queries

Client systems make log queries by making a HTTP GET request against a feed log URL or a subscription log URL, using query parameters in the request to restrict the set of results returned. In response to any well-formed request, the DR will return a response with body containing a list (possibly empty) of the log records that match the query.

A GET request directed to a feed log URL with no query parameters returns the following records for the 24 hours immediately preceding the query:

- all<sup>4</sup> of the publish attempt records for the feed.
- all of the delivery attempt records for all of the subscriptions to the feed.
- all of the expiry records for all of the subscriptions to the feed.

A GET request directed to a subscription log URL with no query parameters returns the following records for the 24 hours immediately preceding the query:

- all of the publish attempt records for the feed associated with the subscription.
- all of the delivery attempt records for the subscription.
- all of the expiry records for the subscription.

A client system can filter the set of records returned by a request and/or alter the time interval for which records will be retrieved by adding a query string to the Request-URI. Table 1 lists the query parameters supported by the Log Query API. A query string may contain multiple parameters, but only one instance of each parameter name. Each parameter in a query further restricts the set of records selected—that is, the parameters in a query are logically ANDed together.

---

<sup>3</sup> *Data Router Release 1 Provisioning API*, Version 1.1, 2013-03-18.

<sup>4</sup> “All” in this and subsequent bullets means all of the records available to the Log Query API at the time that it receives a GET request. The DR will have a policy governing how long it retains log data, meaning that older records will not be available. Similarly, since the Log Query API server gets periodic log updates rather than a real-time stream of log records, records related to very recent events may not be available.

---

Table 1: Query Parameters

Parameter	Description	Allowed values
type	Selects records of the specified type	<ul style="list-style-type: none"> <li>• pub: publish attempt</li> <li>• del: delivery attempt</li> <li>• exp: delivery expiry</li> </ul>
publishId	Selects records with the specified publish ID (the unique identifier assigned by the DR at the time of the initial publication request and carried in the X-ATT-DR-PUBLISH-ID header in the response to the original publish request and in all delivery requests).	
start	Selects records created at or after the specified date and time. If absent, and if end is also absent, start defaults to the current time minus 24 hours. If absent, and if end is present, start defaults to the time specified by end minus 24 hours.	A date-time expressed in the format specified by RFC 3339 <sup>5</sup> , section 5.6, with the additional restriction that the time must be specified in UTC and the time-offset must be designated by "Z".
end	Selects records created at or before the specified date and time. If absent, and if start is also absent, end defaults to the current time. If absent, and if start is present, end defaults to the time specified by start plus 24 hours.	A date-time with the same restrictions as start.
statusCode	Selects records with the specified statusCode field. (Will never select an exp record, because exp records do not have a status code.)	An HTTP integer status code or one of the following special values: <ul style="list-style-type: none"> <li>• success, which matches any status code in the range 200-299</li> <li>• redirect, which matches any status code in the range 300-399</li> <li>• failure, which matches any status code &gt; 399.</li> <li>• -1, which matches the special code (-1) used when the DR did not receive an HTTP status code. (Applies to del records only.)</li> </ul>

<sup>5</sup> Date and Time on the Internet: Timestamps, RFC 3339, July 2002.

Parameter	Description	Allowed values
expiryReason	Selects records with the specified expiry reason. (Selects only <code>exp</code> records, because <code>pub</code> and <code>del</code> records do not have an expiry reason.)	

## Query Responses

When the DR receives a valid log query, it sends an HTTP response with a status code of 200, a Content-Type of `application/vnd.att-dr.log-list`, and a body containing a list of log records that match the query<sup>6</sup>. The list is represented as a JSON array. Each element of the array is a log record, represented as a JSON object with fields that contain log information. If no log records match the query, the JSON array will be empty.

Table 2 lists the fields that may be present in a JSON object representing a log record. The `type` field identifies the record as being a publication attempt record, a delivery attempt record, or a delivery expiry record. While most fields are present in all log records, some are specific to a particular type of record or to the type of publish/delivery request (PUT or DELETE) that the record is associated with.

**Table 2: Log Record Fields**

Field	Description	In Types
<code>type</code>	Record type: <ul style="list-style-type: none"> <li><code>pub</code>: publication attempt</li> <li><code>del</code>: delivery attempt</li> <li><code>exp</code>: delivery expiry</li> </ul>	All
<code>date</code>	The UTC date and time at which the record was generated, with millisecond resolution, in the format specified by RFC 3339, section 5.6, using "Z" as the time-offset marker. Example: <code>2013-03-15T16:07:18.901Z</code>	All
<code>publishId</code>	The unique identifier assigned by the DR at the time of the initial publication request (carried in the <code>X-ATT-DR-PUBLISH-ID</code> header in the response to the original publish request and in all delivery requests).	All
<code>requestURI</code>	The Request-URI associated with the request	All
<code>method</code>	The HTTP method (PUT or DELETE) for the request	All
<code>contentType</code>	The media type of the payload of the request	All where <code>method</code> is PUT
<code>contentLength</code>	The size (in bytes) of the payload of the request	All where <code>method</code> is PUT
<code>sourceIP</code>	The IP address from which the request originated	<code>pub</code>

<sup>6</sup> Consistent with standard HTTP practice, the DR may apply a `gzip` content coding to the body unless the client has explicitly signaled (through the use of the `Accept-Encoding` header) that it cannot accept this coding.

Field	Description	In Types
endpointId	The identity used to submit a publish request to the DR	pub
deliveryId	The identity used to submit a delivery request to a subscriber endpoint	del
statusCode	The HTTP status code in the response to the request. A value of -1 indicates that the DR was not able to obtain an HTTP status code <sup>7</sup> .	pub, del
expiryReason	The reason that delivery attempts were discontinued: <ul style="list-style-type: none"> <li>notRetryable: The last delivery attempt encountered an error condition for which the DR does not make retries.</li> <li>retriesExhausted: The DR reached its limit for making further retry attempts.</li> </ul>	exp
attempts	Total number of attempts made before delivery attempts were discontinued	exp

## Appendix 1: Error Conditions

The Log Query API uses standard HTTP error status codes to indicate problems in handling a request. Table 3 lists the codes that a client system is most likely to encounter while using the API. This is not an exhaustive list—other codes may be used, and some HTTP libraries that a client system might use sometimes generate error responses locally, outside of the control of the API.

In keeping with HTTP's standard usage of status codes, codes in the range 400-499 indicate a problem with the request. Requests that elicit a response in the 400-499 range should not be retried without first being modified to correct the problem. Status codes in the 500-599 range indicate a problem on the server side. Requests that elicit a response in the 500-599 range can be retried once the server problem is cleared.

The Log Query API never redirects requests and so does not use status codes in the range 300-399.

A response with an error status code may include a body with HTML or plain text content providing a description of the problem.

**Table 3: Error Status Codes**

Code	Meaning	Description
------	---------	-------------

---

<sup>7</sup> A -1 code can appear only in `del` records. It indicates that the delivery attempt failed and that the DR did not receive an HTTP status code. There are many possible reasons for this, among them: (a) failure to find a DNS record corresponding to the delivery host; (b) failure to set up a TCP connection to the delivery host/port, possibly due to networking issues; (c) failure to establish an SSL or TLS session with the delivery host, typically due to issues related to the X.509 certificates. (This is not an exhaustive list; there are numerous additional possible causes.)

Code	Meaning	Description
400	Bad Request	The request is defective in some way. Possible causes: <ul style="list-style-type: none"> <li>Unrecognized parameter name in the query string.</li> <li>Invalid parameter value in the query string.</li> </ul>
404	Not Found	The request was not directed to a feed log URL or subscription log URL known to the system.
405	Method Not Allowed	The HTTP method in the request was something other than GET.
406	Not Acceptable	The request has an Accept header indicating that the requester will not accept a response with <code>application/vnd.att-dr.log-list</code> content.
500	Internal Server Error	The DR API server encountered an internal error and could not complete the request.
503	Service Unavailable	The DR API service is temporarily unavailable. The response may include a <code>Retry-After</code> header indicating when the client system should retry the request.

## Change History

### Version 1.1

- Remove `destinationIP` field from query responses. The DR software does not capture this information.
- Add a note that the `statusCode` field can contain -1, rather than a real HTTP status code, under some conditions, and add the possibility of querying for records with a -1 code.
- Provide correct information about the certificate authority that signs DR server certificates.