Using the Lumin.ai Billing API

Appointment Scheduler Edition

March, 2019

About This Document

This document describes how to use the Lumin.ai Billing API to obtain real-time information about billable events such as scheduled or rescheduled appointments. Use it if:

- You are responsible for populating these billing data into your own company's billing system for subsequent invoicing of your customer or downstream partner, and
- Pricing is based on the number of appointments scheduled by the Lumin.ai system.

API

The interface is a simple REST call that returns all billable events within a given datetime range.

Coordination

Obtain the following information before you begin coding.

- **URI and Token**: These are customer-specific and supplied by Lumin.ai
- **Event types**: For simple pricing based on scheduled appointments only, use '`billing.scheduled`'. If many billing types are needed, separate them with commas (no spaces), e.g., '`billing.scheduled,billing.rescheduled`'.
- **Billing frequency**: This is determined by the contract that governs this billing. Most likely, each invoicing period exactly coincides with one calendar month. Since the API provides real-time data, you can safely query it just seconds after the end of a billing period.

If you need to report any unexpected results, please supply the URI and payload that you used, as well as any error response that you received. Do not email bearer tokens.

Querying

Always permit one request to end before beginning another one.

Use this POST via HTTPS to request billing records. The payload is JSON:

```
{
    "type" : "billing.scheduled",
    "start": "2019-02-01T01:00:00-08:00",
    "end"  : "2019-02-28T23:00:00-08:00"
}
```

Use these headers, substituting the actual token for `<token>`:

```
{
    "Authorization": "Bearer <token>"
}
```

Paging

If the result contains a `next_token`, then you must call the API back for another page of data using a payload like this:

```
{
    "type"       : "billing.scheduled",
    "Start"      : "2019-02-01T01:00:00-08:00",
    "end"        : "2019-02-28T23:00:00-08:00",
    "start_token": <previous value of paging token>,
}
```

Filtering by Host

In Lumin.ai terminology, the word “host” refers to a single appointment-making entity (e.g., one store). Providing a `host` key in the query payload filters the result set by the value of the output field `host_id`.

Successful Response

If a query is successful, then the response code is 200, the key `ok` is `true`, and the key `items` points to an array of dictionaries:

```
{
    "ok": true,
    "next_token": <paging token>,
    "items": [
        {
            "conversation_id": "5735605401026560",
            "host_did": "199234567890",
            "host_id": "default",
            "id": "MDAwMDAwMDAwMDAwMDAwMDYyODQwNTU1NDQ4NTY1NzY=",
            "timestamp": "2019-02-24T21:29:36.798081+00:00",
            "type": "billing.scheduled"
        },
        {
            "conversation_id": "5711381785477120",
            "host_did": "199234567890",
            "host_id": "default",
            "id": "MDAwMDAwMDAwMDAwMDAwMDYzMDY4ODc1OTEwMDIxMTI=",
            "timestamp": "2019-02-24T21:36:01.295808+00:00",
            "type": "billing.scheduled"
        }
    ]
}
```

Output Fields

Each element in the `items` array has these fields:

- `id` (256 chars max, unique): ID of billing record
- `conversation_id`: ID of conversation that generated the billing record
- `datetime` (iso8601): Timestamp of billing record (UTC)
- `host_id` (unicode): Host ID
- `host_did`: DID of host including country code and all digits, with no + and no delimiters
- `user_did`: DID of end user including country code and all digits, same format
- `event_type` (unicode): e.g., `billing.scheduled`