

The Euphoria API is a basic Request and Response type environment. The data that is returned can be cached for up to 30 seconds, to eliminate load under frequent use

## Usage Requirements

* Requests must be HTTPS:
* Must contain Tenant information, and an Authentication Key (see: Obtaining an API Key)
* Must contain Action Name (see: API Action Name Reference)
* XML must be wrapped in <XML></XML> Document Element
* **Must** use the following endpoint - https://api.euphoria.co.za/Euphoria.Api.aspx

The API will require authentication on each request. No requests will be possible without the Tenant Name and Authentication Key. (see: Obtaining an API Key)

### Obtaining an API Key

* Log into the TMS
* Select the PBX the API needs to be on.
* Navigate to the API Manager. (If this option is not available, please contact the support company, and ask for the API Manager to be enabled.).
* Depending on which TMS is being used, the user will select either “API Auth Codes” or “API Manager”.
* Once loaded, the user can click on “Add API Key” and type in a relevant name for the API key.
* The TMS will then provide the user with a new API Authorization code.

### Obtaining a Tenant Name

A Tenant name is a unique reference that is used to identify a specific PBX. Follow the below steps to obtain the Tenant Name:

* Log into the TMS but do not select the account. (Under an account there will either be a field called “Name” or “Account”. In this field there will be a unique one-word name which will be the Tenant Name).
* Log into the TMS, select the PBX the API should be used on.
* Navigate to Extension Manager
* >Extensions
* >Click on any extension
* >Click on the “Send Password” button. (This will open a pop-up window information.)
* One of the fields will provide SIP username. The SIP username is made up of two parts that is separated with a dash (-): Extension Number - Tenant Name

### GetQueueCalls

Returns all the call recording records using a few filters and specific page sizing etc

#### Input

|  |  |  |
| --- | --- | --- |
| Input | Description | Example |
| queueName | Name of the queue | CustomerServiceQueue |
| pageSize | How many entries to show | 30 |
| startAt | startAt | 0 |
| startDate | Start date | 2020-10-27 |
| endDate | End date | 2020-10-28 |

Example:  
<XML>

<Tenant>

<Name>Tenant-Name</Name>

<Auth>Auth-Key</Auth>

</Tenant>

<ActionName>GetQueueCalls</ActionName>

<queueName></queueName>

<pageSize></pageSize>

<startAt></startAt>

<startDate></startDate>

<endDate></endDate>

</XML>

#### Output

Will return current queue calls

|  |  |  |
| --- | --- | --- |
| Output | Description | Example |
| QueueName | Name of the queue | CustomerServiceQueue |
| Direction | Direction of the call | inbound |
| uniqueid | Unique id of the call | pbx20-151685161.15616 |
| Recorded | Is the call recorded. 1 = Yes, 0 = No | 1 |
| RecordedID | Id of the recording | 428532 |
| RecordingFilename | Name of the recording | pbx20-15615618.5183... |
| RecordingFileSize | Size of the recording | 123170 |
| RecordingDeleted | Is the recording deleted | 0 |
| QueueEvent | Every event that occurs, will have it’s separate event entry. This will include when the call entered the queue, was it answered or not, when it was connected, to when it was completed. There might be other entries in between when it was put on hold etc. In the first entry when it enters the queue, there will be an entry for the caller id. | QueueEvent  DateTime  2020-10-28 09:00:00  Event  ENTERQUEUE  url  []  callerid  0105934500  position  1 |