CallMiner Media Ingestion API v2 Developer's Guide

VERSION 2025.01

API

CONTENTS

1 Introduction

The CallMiner Media Ingestion API v2 is a standardized interface into the CallMiner Eureka mining platform. It provides a highly available, scalable way to add and update the media and metadata for all of the audio and text-based contacts in your Eureka system.

There are two versions of the CallMiner Media Ingestion API. Both API versions contain the same routes that enable you to ingest your contacts and their associated metadata, but they have different authentication methods. The original version of the CallMiner Media Ingestion API uses JWT authentication, while the CallMiner Media Ingestion API v2 uses CallMiner's Identity Provider Service and single sign-on (SSO) for authentication.

> **Important**: This document covers how to ingest contacts using the CallMiner Media Ingestion API v2, which uses CallMiner's Identity Provider Service and SSO authentication. If you are using the original CallMiner Media Ingestion API, which uses JWT authentication, see the *CallMiner Media Ingestion API Developer's Guide*.

The CallMiner Media Ingestion API v2 will be referred to in this document as the "CallMiner Media Ingestion API" or the "Ingestion API".

This document explains how to upload media and metadata to the CallMiner Eureka mining platform with the CallMiner Media Ingestion API. It describes the requirements needed to use the CallMiner Media Ingestion API, supported media types, and HTTP status codes.

1.1 | Definitions

- **MEDIA**: The audio file or text transcript for a given contact.
  - **AUDIO MEDIA**: The audio file formatted in a specific MIME type for a given contact. See Supported MIME Types for more information.
  - **TEXT-BASED MEDIA**: The text transcript for a contact formatted as a specific text-based media type. Text media can be ingested in chat, email, survey, Twitter, and Facebook formats.

- **CONTACT**: The individual customer interaction being analyzed. Contacts are made up of the media and metadata for each interaction. They can be phone calls (with audio and a transcript), emails, tweets, chats, etc.
- **METADATA**: The data that identifies a particular contact, such as the agent name or customer account number.

1.2 | Audience

This document is intended for developers who are integrating against the CallMiner Media Ingestion API and ingesting contacts into their Eureka system. This includes:

- Developers using the CallMiner Media Ingestion API interactive documentation to ingest contacts.
- Developers using a third-party application, like Postman, to ingest contacts.

1.3 | Assumptions

This document makes the following assumptions:

- You are familiar with JSON.
- You are aware of metadata mappings in the Initial System Configuration in your Eureka system, and you have a valid Source ID. Each ingestion source has a unique ID. The ingestion source is the method by which CallMiner obtains media and metadata for ingestion, such as via File Transfer Protocol (FTP) or CallMiner Media Ingestion API. A source must be configured prior to sending requests with CallMiner Media Ingestion API.
- You are familiar with the CallMiner Data API. For more information, see the *CallMiner Data API Cookbook*.

1.4 | Requirements

To access and use the CallMiner Media Ingestion API, you must meet the following requirements:

- You must use one of the authentication methods listed in the Login and Authentication section.
- You must pass in valid CallMiner metadata key-value pairs when you upload metadata to the CallMiner Media Ingestion API. To get a list of valid metadata key-value pairs in your Eureka system, see the Retrieve Metadata Configurations section before you begin ingesting contacts.
- You must have a valid name or Source ID for the ingestion source you're using. For a list of available sources in your Eureka system, issue a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and Initial System Configuration in Analyze to view the available sources in the

left pane. A source must be configured prior to sending requests with the CallMiner Media Ingestion API.

- The maximum size of metadata or media files that you can upload to CallMiner is *1 GB* (1,073,741,824 Bytes) per request.
- The maximum length of audio you can send in for a single audio-based contact is *1 hour and 45 minutes*.
- The maximum number of words that you can send in for a single text-based contact is roughly *21,000* words.

1.5 | Conventions Used in This Document

The following conventions are used in this document:

Convention	Example
Clickables and route names are represented in bold text.	Press the **Model** button to open the **GET api/v2/users** route.
Permission names are represented in bold, italicized magenta text.	You need ***Download*** permission to use this route.
The following are represented in purple, bold, monospace font: • Attribute Names • Header Names • Parameter Names • Field Names • HTTP Status Codes	Enter a value for the **`ExampleMetadata`** field.
Attribute example values are represented in italics.	Enter *Agent* as the value for the metadata column attribute.
Required fields are represented in bold, italicized green text with a dashed underline.	metaField (string): ***Required***. The column name of the metadata item.
Code examples are represented in mono space font on a gray background.	`<add key="service name" value="CallMiner Task Adapter"/>`

Convention	Example
Placeholders in code samples that need to be filled in with actual values are represented in bold, capitalized text.	`Authorization: JWT **AUTH_TOKEN**` `Content-Type: application/json; charset=utf-8`
Convention	Example
Clickables and route names are represented in bold text.	Press the **Model** button to open the **GET api/v2/users** route.

1.6 | Feedback

Your feedback plays an important part in improving our technical documentation. Please send any questions, comments, or suggestions for improvements to docfeedback@callminer.com.

2 Login and Authentication

There are a few ways that you can interact with the CallMiner Media Ingestion API. You can use interactive documentation, which allows you to ingest contacts and provides documentation on how to use each route, or you can use a third-party application, like Postman, to ingest contacts into your Eureka system. Depending on which option you choose, you'll need to follow different login and authentication steps.

Choose the login and authentication option below that you'd like to use:

- Logging In and Authenticating with the CallMiner Media Ingestion API Interactive Documentation
- Authenticating with a Third-Party Application

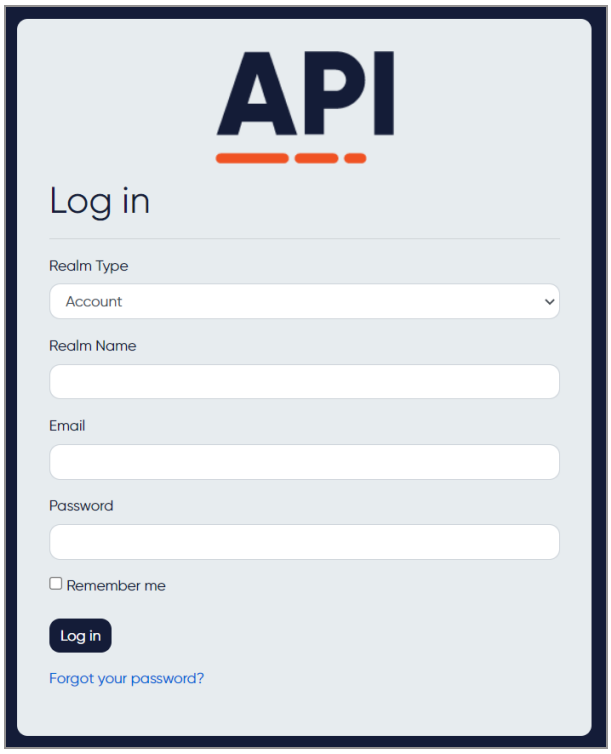
2. 1 | Logging In and Authenticating with the CallMiner Media Ingestion API Interactive Documentation

The CallMiner Media Ingestion API interactive documentation uses CallMiner's Identity Provider Service, an authentication application that employs industry-standard methods for user authentication and application authorization. This authentication process provides Single Sign-On for users when they access the CallMiner Media Ingestion API interactive documentation.

To log in and authenticate with the CallMiner Media Ingestion API interactive documentation:

1. Open your browser and enter the URL that was provided to you by CallMiner for the CallMiner Media Ingestion API interactive documentation. If you do not have the URL, please contact support@callminer.com.

2. You'll be redirected to CallMiner's Identity Provider Service login page.



3. If it's not already selected for you, choose *Account* in the Realm Type drop-down and enter your Realm Name in the appropriate field. Your Realm Name is the company name that you provided to CallMiner during the initial onboarding process.

> **Note**: If these fields are not visible in the Login dialogue box, it means that they have already been set in the query parameters. Skip to the step below.

4. Enter your email address and password into the dialogue boxes. Your email is the email address you provided during the initial onboarding process.

> **Note**: If you haven't activated your account with CallMiner, please see the *Identity Provider Service Administrator Guide* for information on how to do so.

5. Click **Log in**.
6. You are now logged in to the CallMiner Media Ingestion API interactive documentation.

7. Go to CallMiner Media Ingestion API Routes and choose which route you'd like to use first to ingest your contacts.

2. 2 | Authenticating with a Third-Party Application

This section describes how to use a CallMiner-provided Client ID and Secret to generate an access token for making authenticated requests against the CallMiner Media Ingestion API. The CallMiner Media Ingestion API uses CallMiner's Identity Provider Service, an authentication application that employs industry-standard methods for user authentication and application authorization. With this authentication process, you can use a Client ID and Secret to request an access token from the Identity Provider Service. You can then use this token in the `Authorization` header in whatever third-party application you want to use to authenticate each request you make against the CallMiner Media Ingestion API.

> **Note**: Before you can request a token from CallMiner's Identity Provider Service, you'll need to request a Client ID and Secret from support@callminer.com.

To request an access token, complete the following steps:

1. Open the third-party application you'd like to use.
2. Enter the URL for CallMiner's Identity Provider Service. To find your URL, see CallMiner's Identity Provider Service URLs.
3. Create a **POST** request with the **connect/token** route.
4. Set the `Content-Type` request header to *application/x-www-form-urlencoded*.
5. Enter your Client ID, Client Secret, grant type, and scope into the body of the request:
    - **client_id**: Enter the Client ID you received from support@callminer.com.
    - **client_secret**: Enter the Client Secret you received from support@callminer.com.
    - **grant_type**: Enter the text `client_credentials`
    - **scope**: Enter the text `https://callminer.net/auth/platform-ingestion`

    Below is an example of what the body of the request might look like:

```
POST connect/token
Content-Type: application/x-www-form-urlencoded

{
   "client_id": "abcd-1234",
   "client_secret": "abcd-1234",
```

```
   "grant_type": "client_credentials",
   "scope": "https://callminer.net/auth/platform-ingestion"
}
```

6. Send the request. One of the following happens:

   - If an error occurs, a status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the session request is successful, the API server responds with a `200 (OK)` HTTP status code and the access token you requested.

     Below is an example of what the body of the response might look like:

     ```
     {
        "access_token": "abcd123",
        "expires_in": 3600,
        "token_type": "Bearer",
        "scope": "https://callminer.net/auth/platform-ingestion"
     }
     ```

     The response includes:

     - **access_token**: The access token string as issued by the authorization server.
     - **expires_in**: The number of seconds that the token is granted for.

       > **Note**: This authentication token has a lifespan of 1 hour.

     - **token_type**: The type of access token, which is typically *Bearer*.
     - **scope**: The application scope that the token will allow you to access.

7. For the lifetime of this token, you will prefix the token with the word *Bearer* and enter it into the `Authorization` header of each request you make against the CallMiner Media Ingestion API.

   For example:

   ```
   Authorization: Bearer abcd123
   ```

> **Note**: To avoid any authentication errors, before the current token expires, you must repeat the steps outlined above to generate a new access token. If a request is made with an expired token and/or an invalid token, the CallMiner Media Ingestion API will generate an HTTP status code of `401` indicating that it's an Unauthorized request.

8. Now that you have your token, go to CallMiner Media Ingestion API Routes and choose which route you'd like to use first.

3 CallMiner Media Ingestion API Routes

Choose the route below that you'd like to use:

3. 1 | Retrieve Metadata Configurations

This section describes how to retrieve the contact metadata configurations in your Eureka system. This route returns a list of each metadata item in your Eureka system and provides detailed information about each item, such as its column name, friendly name, and description. You will need to know information about specific metadata items, like their column names, to use most routes in the CallMiner Media Ingestion API.

You can use the metadata information returned in this route to:

- Get a list of every metadata configuration for a tenant in your Eureka system.
- Update or add the metadata for a contact that already exists in your Eureka system using the Upload Metadata Only routes. To do this, you will need to know the `ColumnName` returned in this route for each metadata item you want to update, which can be different from the friendly name for the metadata item displayed in Analyze.
- Determine which metadata items you can use to filter contacts by in the Upload Metadata by Filtering route. The `IsIndexed` field returned with each metadata item in this route indicates whether or not you can use the metadata item as a filter with the Upload Metadata by Filtering route.

> **Note**: You need a valid authentication token to complete requests. For more information on getting a token, see Login and Authentication.

To retrieve the contact metadata configurations in your Eureka system, complete the following steps:

1. Create a **GET** request with the **/api/info/callmetadataconfig** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the GET request you will create:

   ```
   GET /api/info/callmetadataconfig HTTP/1.1
   Host: INGESTION_API_URL
   Authorization: Bearer AUTH_TOKEN
   Content-Type: application/json; charset=utf-8
   ```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:

   - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
   - **Content-Type**: Set the value to *application/json; charset=utf-8*.

3. Send the request.

   One of the following will happen:

   - If an error occurs, the API returns a status code in the **400** to **500** range. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API returns a **200** response and a list of all available metadata fields with details.

     The following example shows a partial sample response from a **GET /api/info/callmetadataconfig** request. A full response includes the following information for each metadata item within your account:


```
[
   {
      "ColumnName": "Agent",
      "AttributeOrDimension": "Attribute",
      "FriendlyName": "Agent",
      "Description": "Person who handled the call",
      "DisplayOrder": 0,
      "DimensionBinRoundTo": 0,
      "IsAdvanced": 0,
      "FormatString": "",
      "DimensionMin": 0,
      "DimensionMax": 0,
      "IsIndexed": 1,
      "Max_Char_Length": 50,
      "BaseTypeName": "String",
      "DisplayCoaching": 0,
      "IsCoachAdvanced": 0,
      "CoachingHierarchy": 0,
      "DW_Hierarchy": 0,
      "TypeName": "nvarchar"
   }
]
```


     The response includes the following information:

     - **ColumnName**: The column name for the metadata item in the database. This is the metadata name, or **key**, used in other Ingestion API routes to update or add a metadata item in your Eureka system.
     - **AttributeOrDimension**: Indicates whether the metadata item is an *Attribute* or a *Dimension*. If it's an attribute, the metadata item is based on information submitted along with the contact (for example, the agent name, the customer ID, location, etc.). If the

metadata item is a dimension, it is based on the quantifiable aspects of the contact itself (for example, duration, agitation, percent silence, word count, etc.). If a metadata item is *Undefined*, it can't be used as a filter in the left pane of the search page in Analyze.

- **FriendlyName**: The user-facing name for the metadata item in Analyze. For example, if the column name for a metadata item is *WAVLength*, the friendly name in Analyze might be *Duration*.
- **Description**: A brief definition of the metadata item that describes what it is.
- **DisplayOrder**: An integer indicating the order in which a metadata item displays within search results.
- **DimensionBinRoundTo**: The number of digits to round to for calculated dimension bins.
- **IsAdvanced**: Indicates if the metadata item appears within the drop-down menu of available filters in the Advanced Filter section of Analyze. A value of *1* means that the metadata item does appear as an available filter in the Advanced Filter section of Analyze. A value of *0* means that it doesn't appear in the Advanced Filter section.
- **FormatString**: The format for how the metadata item appears within Analyze. The available options for format are *percent*, *time*, *character*, and a custom .NET string format.
- **DimensionMin**: The minimum value for a metadata column configured as a dimension. This value is stored in the database and updated by scheduled stored procedures. For example, the minimum value for the word count metadata column could be *0* words.
- **DimensionMax**: The maximum value for a metadata column configured as a dimension. This value is stored in the database and updated by scheduled stored procedures. For example, the maximum value for the word count metadata column could be *65000* words.
- **IsIndexed**: Indicates whether or not the metadata item is indexed. If a metadata item is indexed, you can use the item as a filter in the Upload Metadata by Filtering route. A value of *1* indicates that the metadata item is indexed and can be used as a filter in the Upload Metadata by Filtering route, while a value of *0* indicates that it is not indexed and cannot be used as a filter.

> **Note**: The `IsIndexed` setting only determines whether or not you can use this metadata item to filter contacts when you Upload Metadata by Filtering. You can update any metadata item for a specific contact using the Upload Metadata Only routes, regardless of whether or not it's indexed, except for the following metadata items:
>
> - ClientId
> - CreateDate
> - RequestId
> - IsSample
> - AudioSourceName
> - CorrelationId
> - TenantApiKey
> - ParentID

- **Max_Char_Length**: The maximum number of characters that the value for the metadata item may have. For example, if the `Max_Char_Length` for the metadata key Agent Name is *50*, the agent name value must be fewer than 50 characters.
- **BaseTypeName**: The type of the metadata item, such as *String*, *Numeric*, and *Date*. This type is further specified by the `TypeName` attribute below. If you want to update a metadata item, the value you enter needs to match the type specified here.
- **DisplayCoaching**: Indicates if the metadata item is available within the drop-down menu of filters that can be used within Find in the Coach application. If you have Coach configured, a value of *1* indicates that the metadata item appears within the filter drop-down menu in Find. A value of *0* means that it doesn't appear in the drop-down menu of available filters.
- **IsCoachAdvanced**: Indicates if you can further filter the metadata item in the Find filter drop-down menu using the free-text search box. A value of *1* means you can filter the metadata item using the free-text search box in Find. A value of *0* means that the metadata item cannot be filtered further using the free-text search box. For example, if the Agent metadata item is available to use as a drop-down filter in Coach and has a value of *1* for the `IsCoachAdvanced` field, you could enter a specific agent's name in the free-text search box (e.g. John Smith) to filter your results by that specific agent.
- **CoachingHierarchy**: Indicates if the metadata item is available as an option for use in Coach's hierarchy tree. A value of *1* means the metadata item is available to use in

Coach's hierarchy tree. A value of *0* means that it isn't available as an option for use in the hierarchy tree.

- **DW_Hierarchy**: This attribute is only used by MyEureka. This doesn't affect metadata items in Analyze or Coach.
- **TypeName**: The type of the metadata item, which further specifies the `BaseTypeName`. For example, if the `BaseTypeName` is *Date*, the `TypeName` might be *smalldatetime*.

3. 2 | Upload Audio Media and Metadata

This section describes how to upload paired audio media and metadata into your Eureka system. Using this route, you can ingest a single audio contact, paired with its associated metadata, into your Eureka system for processing. This route gives you two options when uploading an audio contact. You can:

- Upload a single audio contact with its associated metadata.
- Upload a single audio contact that you want to split into separate contacts. This option allows you to segment a call at places in the call where there are changes to particular metadata values, such as when one agent transfers a call to another agent. After segmenting, the segments of the call are then analyzed independently and appear in Eureka Analyze as individual contacts.
- Upload a single audio contact that you want to split into separate contacts because the original contact exceeds the maximum allowed contact length.

> **Note**: There is a maximum contact length of 1 hour and 45 minutes for ingestion. If your contact is too long, you will receive an error message telling you that you have surpassed the limit and that the contact cannot be ingested.

Uploading paired audio media and metadata is a two-step process:

1. Start a session in one of the following ways:
    - Start a Session to Upload Audio Media and Metadata: Choose this option when uploading a single audio file with corresponding metadata.
    - Start a Session to Upload Audio Media and Metadata Segments: Choose this option when uploading a single audio file that you want to split into segments. Note that *audio/m4a* and *video/mp4* files are not currently supported with this route.
2. Upload Audio Media.

> **Note**: You need a valid authentication token to complete requests. For more information on getting a token, see Login and Authentication.

3. 2. 1 | Start a Session to Upload Audio Media and Metadata

This section describes how to start a session to upload audio media and metadata for a single audio file. This is an example of the POST request you will create:

```
POST /api/session/metadatamedia HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8
{
   "Metadata": [
   {
      "Key": "METADATA_KEY",
      "Value": "METADATA_VALUE"
   }
   ],
   "TotalMediaLength": FILE_SIZE,
   "MediaType": "MIME_TYPE",
   "ClientCaptureDate": "RECORDING_DATE",
   "VoicePrintIdentifier": "AGENT_NAME",
   "AudioDecryptionKey": "DECRYPTION_KEY",
   "SourceId": "AUDIO_SOURCE",
   "CorrelationId": "UNIQUE_ID"
}
```

To start a session to upload paired audio media and metadata for a single audio file, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatamedia** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.
2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **TotalMediaLength** (integer): The number of bytes in the media file to be uploaded in this session. Set this value to the size of the media file you are uploading.
    - **MediaType** (string): The MIME type of the media file you are uploading. You can upload any CallMiner-supported media MIME type, such as *audio/wav*. See Supported MIME types for more information.
    - **ClientCaptureDate** (string): The localized date and time that the audio contact was originally recorded. The `ClientCaptureDate` should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See DateTime Formatting for more information about ISO 8601 format.

- **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The `Metadata` array must contain at least one metadata item. Each metadata item consists of a `key` and a `value`, representing a single piece of metadata to be applied to a given contact:
    - **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
    - **Value** (string): The literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **VoicePrintIdentifier** (string) (optional): The VoicePrint identifier used in Speaker Separation. This field tells the system which agent is speaking in the contact. If you have Speaker Separation, set the field for `VoicePrintIdentifier` to the same value that corresponds to the Agent metadata key. For example, if you have a metadata item that has the `key` set to *Agent* and the `value` set to *John Smith*, you would enter *John Smith* into the `VoicePrintIdentifier` value as well. This value is only required if Speaker Separation is enabled for the particular Source ID.
- **AudioDecryptionKey** (string) (optional): The key for decrypting encrypted audio. This value is only required if Audio Encryption is enabled for the particular SourceId. Currently, only Aspect AQM encryption is supported. If you're unsure if you need to set this key, contact support@callminer.com.

- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string) (optional): Typically a Universally Unique Identifier (UUID) data type, though any kind of string value is acceptable. Its purpose is:
  - To uniquely identify an interaction within the context of your CallMiner account. You can track the success or failure of media and metadata using the Correlation ID in the Mining Summary section of Analyze or via the SystemMetricsController routes in the CallMiner Data API.
  - To provide a means of correlating the disparate parts of an interaction, such as media and metadata, with each other. For example, if you send the media for a contact in one request and you want to add or update the contact's associated metadata at a later time with a separate request, you use the Correlation ID to tell your Eureka system that the metadata should be attached to the original contact.

> **Note**: If you choose not to supply the Correlation ID for a given request, a UUID value will be auto-generated for you. Please note that this value should uniquely identify a particular contact. If you supply a Correlation ID that matches another ID in your system, the contact will fail to publish in your Eureka system. If your recording system offers a unique identifier for each contact, use this value as the Correlation ID.

The following example shows a sample request to start a new audio media session with all values, other than the Ingestion API URL and authorization token, populated. In this request, a session is created for a WAV file that is 300,000 bytes in length:

```
POST /api/session/metadatamedia HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "Metadata": [
   {
      "Key": "Agent",
```

```
      "Value": "John Smith"
    }
    ],
    "TotalMediaLength": 300000,
    "MediaType": "audio/wav",
    "ClientCaptureDate": "2019-08-01T12:01:22.674Z",
    "VoicePrintIdentifier": "John Smith",
    "AudioDecryptionKey": "",
    "SourceId": "ExampleSourceId",
    "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

4. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** HTTP status code and the body of the response includes the Correlation ID and the Session ID.

   The following example shows a sample response:

```
{
  "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4",
  "SessionId": "4b2bef4e-f1e4-4427-81ff-bd3aa7a7a9b5"
}
```

   A successful response includes the following:

   - **CorrelationId**: The UUID string value that pairs an audio file with any applicable metadata files.
   - **SessionId**: The system-generated UUID that identifies the media upload session and is used to properly associate an uploaded media file with the proper session settings.

5. Record the Session ID. You will need to enter this value to upload the media in the next steps.
6. Record the Correlation ID. If you did not supply a Correlation ID during the creation of the session, record the system-generated value for future metadata updates to this contact or to track it through the mining system via the Analyze **Admin** tab or the CallMiner Data API.
7. Skip to the Upload Audio Media section and follow the steps there.

3. 2. 2 | Start a Session to Upload Audio Media and Metadata Segments

This section describes how to start a session to upload audio media and metadata in segments. This enables you to take in a single piece of audio and split it into specified segments, and it allows you to upload segments of a call that differ by metadata, such as when one agent transfers a call to another agent.

This is an example of the POST request you will create:

```
POST /api/session/metadatamedia/segments HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "SegmentData": [
      {
            "Start": FIRST_SEGMENT_START_TIME,
            "End": FIRST_SEGMENT_END_TIME,
            "Metadata": [
            {
               "Key": "METADATA_KEY",
               "Value": "METADATA_VALUE"
            },
         ]
         },
         {
            "Start": SECOND_SEGMENT_START_TIME,
            "End": SECOND_SEGMENT_END_TIME,
            "Metadata": [
            {
               "Key": "METADATA_KEY",
               "Value": "METADATA_VALUE"
            },
         ]
      }
   ],
   "TotalMediaLength": FILE_SIZE,
   "MediaType": "MIME_TYPE",
   "ClientCaptureDate": "RECORDING_DATE",
   "VoicePrintIdentifier": "AGENT_NAME",
   "AudioDecryptionKey": "DECRYPTION_KEY",
   "SourceId": "AUDIO_SOURCE",
   "CorrelationId": "UNIQUE_ID"
}
```

To start a session to segment a contact and upload it as separate contacts, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatamedia/segments** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.
2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **SegmentData** (array): Identifies the start time, end time, and metadata of each segment associated with this session. You will make one `SegmentData` array for each segment that you want to turn into its own individual contact.

      > **Note**: Any audio or metadata outside of these designated segments will be discarded.

        - **Start** (real): The start time of the audio segment in seconds. Enter the value as a number with up to one decimal point. For example, you could set the Start time for the segment to *30.1* seconds.
        - **End** (real): The end time of the audio segment in seconds. Enter the value as a number with up to one decimal point. For example, you could set the End time for the segment to *180.2* seconds.
        - **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The `Metadata` array must contain at least one metadata item. Each metadata item consists of a `key` and a `value`, representing a single piece of metadata to be applied to a given contact:
            - **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
            - **Value** (string): The literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **TotalMediaLength** (integer): The number of bytes in the media file to be uploaded in this session. Set this value to the size of the media file you are uploading.
- **MediaType** (string): The MIME type of the media file you are uploading, such as *audio/wav*. See Supported MIME types for more information. Note that *audio/m4a* and *video/mp4* files are not currently supported with this route.
- **ClientCaptureDate** (string): The localized date and time that the audio contact was originally recorded. The `ClientCaptureDate` should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See DateTime Formatting for more information about ISO 8601 format.
- **VoicePrintIdentifier** (string) (optional): The VoicePrint identifier used in Speaker Separation. This field tells the system which agent is speaking in the contact. If you have Speaker Separation, set the field for `VoicePrintIdentifier` to the same value that corresponds to the Agent metadata key. For example, if you have a metadata item that has the `key` set to *Agent* and the `value` set to *John Smith*, you would enter *John Smith* into the `VoicePrintIdentifier` value as well. This value is only required if Speaker Separation is enabled for the particular Source ID.
- **AudioDecryptionKey** (string) (optional): The key for decrypting encrypted audio. This value is only required if Audio Encryption is enabled for the particular Source ID. Currently, only Aspect AQM encryption is supported. If you're unsure if you need to set this key, contact support@callminer.com.

- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string) (optional): The unique identifier for the contact. Typically a UUID data type, though any kind of string value is acceptable. Its purpose is:
    - To uniquely identify an interaction within the context of your CallMiner account. You can track the success or failure of media and metadata using the Correlation ID in the Mining Summary section of Analyze or via the SystemMetricsController routes in the CallMiner Data API.
    - To provide a means of correlating the disparate parts of an interaction, such as media and metadata, with each other. For example, if you send the media for a contact in one request and you want to add or update the contact's associated metadata at a later time with a separate request, you would use the Correlation ID to tell your Eureka system that the metadata should be attached to the original contact.

> **Note**: If you choose not to supply the Correlation ID for a given request, a UUID value will be auto-generated for you. Please note that this value should uniquely identify a particular contact. If you supply a Correlation ID that matches another ID in your system, the contact will fail to publish in your Eureka system. If your recording system offers a unique identifier for each contact, use this value as the Correlation ID.

> **Note**: When audio is segmented, a number is appended to the Correlation ID to make each segment uniquely identifiable. For example, the two segments created from the original contact could be assigned *123456_1* and *123456_2* as their Correlation IDs.

The following example shows a sample request to start a new audio media session with all values, other than the Ingestion API URL and authorization token, populated. In this request, a session is created for a WAV file that is 300,000 bytes in length with two segments (one from 0 to 21.1 seconds and another from 35.1 to 64.5 seconds). Each segment has its own metadata.

```
POST /api/session/metadatamedia/segments HTTP/1.1
Host: INGESTION_API_URL
```

```
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "SegmentData": [{
      "Start": 0,
      "End": 21.1,
      "Metadata": [{
         "Key": "Agent",
         "Value": "Bob"
      }]
   },{
      "Start": 35.1,
      "End": 64.5,
      "Metadata": [{
         "Key": "Agent",
         "Value": "Tim"
      }]
   }],

   "TotalMediaLength": 300000,
   "MediaType": "audio/mp3",
   "ClientCaptureDate": "2018-10-04T15:02:27.886Z",
   "VoicePrintIdentifier": "Bob",
   "AudioDecryptionKey": "",
   "SourceId": "AudioSource1",
   "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

4. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** HTTP status code and the body of the response includes the Correlation ID and the Session ID.

   The following shows a sample response:

```
{
  "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4",
  "SessionId": "4b2bef4e-f1e4-4427-81ff-bd3aa7a7a9b5"
}
```

   A successful response includes the following:

- **CorrelationId**: The string value that pairs an audio file with any applicable metadata files.
- **SessionId**: The system-generated UUID that identifies the media upload session and is used to properly associate an uploaded media file with the proper session settings.

5. Record the Session ID. You will need to enter this value to upload the media in the next steps.
6. Record the Correlation ID. If you did not supply a Correlation ID during the creation of the session, record the system-generated value. If you have segmented your original contact, take the Correlation ID returned in the response and add a *_1* to the end of it for the first segment, *_2* for the second, etc. to record the Correlation IDs for the new contacts you have made. For example, the Correlation ID for the first segment could be *1836dd5c-0291-4775-8009-85a117f932a4_1* and the one for the second segment could be *1836dd5c-0291-4775-8009-85a117f932a4_2*.Recording the Correlation IDs will allow you to make future metadata updates to these contacts or to track them through the mining system via the Analyze **Admin** tab or the CallMiner Data API.
7. Follow the steps in the next section, Upload Audio Media.

3. 2. 3 | Upload the Audio Media File

Once you've started a session for your paired audio media and metadata or your audio media and metadata segments, you can upload your audio media and metadata into your Eureka system. To do so, complete the following steps:

1. Create a **POST** request with the **/api/media/{sessionId}** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   The following is an example of the POST request you will create:

   ```
   POST /api/media/{sessionId} HTTP/1.1
   Host: INGESTION_API_URL
   Authorization: Bearer AUTH_TOKEN
   Content-Type: AUDIO_TYPE
   Content-Length: LENGTH_IN_BYTES

   BINARY_MEDIA_FILE
   ```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:

   - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
   - **Content-Type**: The MediaType value you entered in the previous section. For a list of all MIME types, see Supported MIME types.

- **Content-Length**: The number of bytes in the media file. This value should match the `TotalMediaLength` you provided when you started your session in the previous section. It should also match the length of the binary media data in the file.

  > **Note**: Most third-party applications will auto-fill this header and value for you.

3. Replace the `{sessionID}` query parameter with the Session ID value that you received from the API in the previous section.
4. Add the binary file content for the media you are uploading.

   The following shows a sample request for an audio/wav file that is 319990 bytes in length. The request has a Session ID of *101491* and everything but the authorization token, Ingestion API URL, and binary file populated:

```
POST /api/media/101491 HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: audio/wav
Content-Length: 319990

BINARY_MEDIA_FILE
```

5. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a `201 (Created)` HTTP status code and the body of the response.

   The following shows a sample response:

```
{
  "CurrentMediaLength": 319990,
  "TotalMediaLength": 319990,
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

   The response includes the following information:

- **CurrentMediaLength**: The number of bytes that have been saved to disk for this media file during this session.
- **TotalMediaLength**: The total number of bytes that need to be saved to disk to completely upload the media file, which you specified during the request.
- **MiningId**: The ID created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

3. 3 | Upload Metadata Only

This section describes how to add or change existing metadata for a contact. Some common uses for this request are:

- You have a CRM system and you want to add an ID or other identifiable information from the CRM system to the contact metadata after the contact has been ingested.
- You have post-contact survey data that you want to add to a contact's metadata, so you can track post-contact customer satisfaction per agent over time.
- You have multiple contacts with a customer over a sales cycle, and you want to be able to identify the contacts that helped toward the sale. Your goal is to find trends in those contacts and improve agent training. You can use the Ingestion API to upload a metadata key that identifies the sale status, such as "Sale Success", and a value that identifies that the contact led to a sale, such as "Yes". You can then use Analyze to find those contacts and identify trends.

  > **Note**: Any metadata updates that affect a category filter or score in Analyze will cause that score to be recalculated.

You can upload metadata in two different ways:

- Upload Metadata with a Correlation ID: Choose this option if you want to add or update metadata for a specific contact and you have the Correlation ID for the contact.
- Upload Metadata by Filtering: Choose this option if you want to add or update metadata for a contact or contacts where you know the contact(s) contain certain metadata values, but you do not know the exact unique ID numbers for the contact(s).

> **Note**: You need a valid authentication token to complete requests. For more information on getting a token, see Login and Authentication.

3. 3. 1 | Upload Metadata with a Correlation ID

This section describes how to upload metadata to a specific contact using a Correlation ID.

This is an example of the POST request you will create:

```
POST /api/session/metadata HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
```

```
Content-Type: application/json; charset=utf-8

{
  "Metadata": [
    {
      "Key": "METADATA_KEY",
      "Value": "METADATA_VALUE"
    },
    {
      "Key": "METADATA_KEY",
      "Value": "METADATA_VALUE"
    }
  ],
  "SourceId": "AUDIO_SOURCE",
  "CorrelationId": "UNIQUE_ID"
}
```

To upload metadata, complete the following steps:

1. Create a **POST** request with the **/api/session/metadata** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.
2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The `Metadata` array contains at least one metadata item. Each metadata item consists of a `key` and a `value`, representing a single piece of metadata to be applied to a given contact:
        - **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.

- **Value** (string): The value of a metadata item is the literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a `key` that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if the attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

> **Note**: When you are updating metadata, the values of the following metadata keys cannot be updated:
>
> - ClientCaptureDate
> - CorrelationId
> - RequestId
> - SourceName
> - MessageTotal

- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string): The unique identifier of the contact that you want to upload metadata for. If you supplied a Correlation ID when you originally ingested the contact, enter that value here. If you did not provide a Correlation ID, enter the Correlation ID value you received in the response when you originally ingested the contact. You may also find this value in

Analyze once the contact has gone through the mining system.

> **Note**: If fields were given different friendly names in your Eureka system, the field you see in Analyze called *Correlation ID* may not be the same Correlation ID that was used to ingest the contact's media. Check with your system administrator to find which field and value you should be using to pair the metadata with the appropriate media.

The following example shows a sample request to upload metadata with all values, other than the Ingestion API URL and authorization token, populated:

```
POST /api/session/metadata HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
  "Metadata": [
    {
      "Key": "Agent",
      "Value": "Chris Ferguson"
    },
    {
      "Key": "AgentGroup",
      "Value": "STS-135"
    },
    {
      "Key": "ANI",
      "Value": "6625811326"
    },
    {
      "Key": "Department",
      "Value": "Collections"
    },
    {
      "Key": "Direction",
      "Value": "Outgoing"
    },
    {
      "Key": "DNS",
      "Value": "3297553961"
    }
  ],
  "SourceId": "AudioSource1",
  "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
```

```
}
```

4. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a `200` HTTP status code and the body of the response includes the Correlation ID and the Mining ID.

   The following example shows a sample response:

```
{
  "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4",
  "MiningId": "4b2bef4e-f1e4-4427-81ff-bd3aa7a7a9b5"
}
```

   The response includes the following information:

   - **CorrelationId**: The string value that pairs an audio file with any applicable metadata files.
   - **MiningId**: This ID is used by CallMiner to track the contact through the mining system. The Mining ID is created after sending the request to the CallMiner mining system and links the session to a particular mining job.

3. 3. 2 | Upload Metadata by Filtering

This section describes how to update one or more metadata values for a particular contact or contacts using metadata items and dates to filter for the contact(s) you want to update.

> **Note**: You may want to use the **GET /api/info/callmetadataconfig** route to retrieve a list of contact metadata configurations for your account to help you choose what metadata items to use to filter contacts. For more information about this request, see Retrieve Metadata Configurations.

Within the Upload Metadata by Filtering route, you have three filter options to choose from:

- Filtering by ExactMatch: Use this array with the `Updates` array when you know the exact metadata item values for the contact you want to update. The Ingestion API will use those metadata values to filter the contacts in your system and update the most recent contact

that exactly matches the metadata values you provided.

- Filtering by ProximityMatch: Use this array with the **ExactMatch** and **Updates** arrays if you want to update a single contact and you have an approximate estimate for when the contact was originally recorded. You can use the **ClientCaptureDate** and associated metadata item values to give the Ingestion API a search window in which to find and update the contact that most closely matches the metadata and date criteria you provided.
- Filtering by DateRange: Use this array with the **ExactMatch** and **Updates** arrays if you want to add or update metadata values to the latest contact within the date range you provided or you want to add or update metadata to the latest contact within the range and any contacts associated with that contact through a customer journey. You can also add or update metadata values to all the contacts within the provided range or to all the contacts within the provided range and any contacts associated with the range of contacts through a customer journey.

> **Note**: In order to use a metadata item to filter contacts, the metadata item must be indexed. Metadata items may be indexed in the **Admin** section of Analyze. For more information, see the Analyze User Guide, which is available within the application. If you used the **GET callmetadataconfig** route to retrieve all your metadata items, any metadata item with a value of *1* for the **IsIndexed** field is available to use to filter contacts with the **POST metadatafiltered** route.

Upload Metadata by Filtering Using ExactMatch

Use this array with the **Updates** array when you know the exact metadata item values for the contact you want to update. The Ingestion API will use those metadata values to filter the contacts in your system and update the most recent contact that exactly matches the metadata values you provided.

To upload metadata using the **ExactMatch** filter, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatafiltered** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.
2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:

- **MetadataItems** (array): This is the element container that includes the metadata items used to filter the contacts in your Eureka system and the metadata items to update for the contact (s).
    - **ExactMatch** (array): ***Required***. An array containing key-value pairs of metadata used to identify the contact you want to update. The `ExactMatch` allows you to filter your contacts by metadata items like agent name, department, location, etc., so the API knows which contact to apply your metadata updates to. This filter uses AND logic, meaning if you use the agent *John Smith* and the *Sales Department* as your metadata filters, the API will only look for and update contacts that were made by the Sales Department *AND* John Smith.

      > **Note**: Be sure to use the most specific metadata item combinations possible as filters in this route. For example, you might filter using multiple metadata values like a specific location, department, agent name, etc. to accurately choose the particular contact you want to update. By using very specific and targeted metadata combinations, it ensures that you're updating the correct contact in your Eureka system.

      > **Note**: The API processes the key-value pairs in order, so it's recommended to list broader metadata items, like location, before specific items, like agent name.

      > **Note**: The key-value pairs entered here do not have to be the same metadata items you want to update in the Updates element.

        - **Key** (string): The metadata item you want to use to filter contacts. For example, you can enter *Location* to filter for only the contacts with a specific location.
        - **Value** (string): The corresponding metadata value to filter by. For example, you can enter *New York* to filter for only the contacts whose location is New York.
    - **Updates** (array): ***Required***. The metadata updates you want to make to the contact. Each `Updates` array contains key-value pairs for the metadata items you want to add or update for the contact in your Eureka system.
        - **Key** (string): The key component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka

system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.

- **Value** (string): The literal value for the metadata item that corresponds to the associated `key`. Enter the value that you want to update the metadata item with. This will replace any existing value for the metadata item. For example, you might enter the `value` *Bob Smith* with the *Agent* `key` to update a particular agent's name from Robert Smith to Bob Smith.

  > **Note**: The following metadata items cannot be updated:
  >
  > - ClientId
  > - CreateDate
  > - RequestId
  > - IsSample
  > - AudioSourceName
  > - CorrelationId
  > - TenantApiKey
  > - ParentID

- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.

The following example shows a sample request to upload metadata using the `ExactMatch` filter with all values, other than the Ingestion API URL and authorization token, populated:

```
POST /api/session/metadatafiltered HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8
{
   "MetadataItems": [
```

```
      {
         "ExactMatch": [
            {
               "Key": "Location",
               "Value": "Fort Myers"
            },
            {
               "Key": "Agent",
               "Value": "Johnny Johnson"
            }
         ],
         "Updates": [
            {
               "Key": "Agent",
               "Value": "John Johnson"
            }
         ]
      }
   ],
   "SourceId": "AudioSourceName"
}
```

4. Send the request.

   One of the following will happen:

   - If an error occurs, the API returns a status code in the **400** to **500** range. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** status code, and the body of the response includes the Request ID.

   The following example shows a sample response:

```
[
   {

         "Success": true,
         "RequestId": "5295431-3fy1-4gf2-b998-2e464de33652",
         "Message": []

   }
]
```

   The response includes the following information:

- **Success**: Indicates whether or not an individual request was successful. If *true*, the individual request was successful. If *false*, the individual request was not successful.
- **RequestId**: The Request ID is the same thing as a Mining ID. It is used by CallMiner to track the individual contact through the mining system. The Mining ID is created after sending the request to the CallMiner mining system and links the session to a particular mining job.
- **Message**: If the request wasn't successful, this field will provide reasons why the request failed.

Upload Metadata by Filtering Using ProximityMatch

This array, sent in a request with the `ExactMatch` and `Updates` arrays, allows you to update a single contact using an approximate estimate for when the contact was originally recorded. You can use the `ClientCaptureDate` and associated metadata item values to give the Ingestion API a search window in which to find and update the contact that most closely matches the metadata and date criteria you provided.

To upload metadata using the `ProximityMatch` filter, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatafiltered** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.
2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **MetadataItems** (array): This is the element container that includes the metadata items used to filter the contacts in your Eureka system and the metadata items to update for the contact (s).
        - **ExactMatch** (array): ***Required***. An array containing key-value pairs of metadata used to identify the contact you want to update. The `ExactMatch` allows you to filter your contacts by metadata items like agent name, department, location, etc., so the API knows which contact to apply your metadata updates to.

> **Note**: Be sure to use the most specific metadata item combinations possible as filters in this route. For example, you might filter using multiple metadata values like a specific location, department, an agent name, etc. to more narrowly choose the particular contact you want to update. By using very specific and targeted metadata combinations, it ensures that you're updating the correct contact in your Eureka system.

> **Note**: The API processes the key-value pairs in order, so it's recommended to list broader metadata items, like location, before specific items, like agent name.

> **Note**: The key-value pairs entered here do not have to be the same metadata items you want to update in the Updates element.

    - **Key** (string): The metadata item you want to use to filter contacts. For example, you can enter *Location* to filter for only the contacts with a specific location.
    - **Value** (string): The corresponding metadata value to filter by. For example, you can enter *New York* to filter for only the contacts whose location is New York.

- **ProximityMatch** (array): Used to identify the contact you want to update. This array contains the date and time that corresponds to the `ClientCaptureDate` and a proximity limit value that creates a range around the datetime. When using `ProximityMatch`, the API will update the contact that most closely matches the parameters set below.
    - **Value** (string): The ISO 8601 datetime value that corresponds to the `ClientCaptureDate`. This value is used to filter for contact(s) within the specified `ProximityLimit` of the datetime you enter. See DateTime Formatting for more information about ISO 8601 format.
    - **ProximityLimit** (integer): The time in seconds that is used in conjunction with the `Value` to identify a contact. The `ProximityLimit` creates a range around the datetime entered for `Value` to help find the contact you want to update. The maximum value is *3600*.
- **Updates** (array): ***Required***. The metadata updates you want to make to the contact. Each `Updates` array contains key-value pairs for the metadata items you want to add or update for the contact in your Eureka system.

- **Key** (string): The key component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
- **Value** (string): The literal value for the metadata item that corresponds to the associated key. Enter the value that you want to update the metadata item with. This will replace any existing value for the metadata item. For example, you might enter the `value` *Bob Smith* with the *Agent* `key` to update a particular agent's name from Robert Smith to Bob Smith.

  > **Note**: The following metadata items cannot be updated:
  >
  > - ClientId
  > - CreateDate
  > - RequestId
  > - IsSample
  > - AudioSourceName
  > - CorrelationId
  > - TenantApiKey
  > - ParentID

- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.

The following example shows a sample request to upload metadata using the `ProximityMatch` filter with all values, other than the Ingestion API URL and authorization token, populated:

```
POST /api/session/metadatafiltered HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
```

```
Content-Type: application/json; charset=utf-8
{
   "MetadataItems": [
      {
         "ExactMatch": [
            {
               "Key": "Location",
               "Value": "Fort Myers"
            },
            {
               "Key": "Agent"
               "Value": "Johnny Johnson"
            }
         ],
         "ProximityMatch": [
            {
               "Value": "2019-04-23T15:52:19.398Z",
               "ProximityLimit": 1800
            }
         ],
         "Updates": [
            {
               "Key": "Agent",
               "Value": "John Johnson"
            }
         ]
      }
   ],
   "SourceId": "AudioSourceName"
}
```

4. Send the request.

   One of the following will happen:

   - If an error occurs, the API returns a status code in the **400** to **500** range. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** status code, and the body of the response includes the Request ID.

   The following example shows a sample response:

```
[
   {

         "Success": true,
```

```
        "RequestId": "5295431-3fy1-4gf2-b998-2e464de33652",
        "Message": []

    }
]
```

The response includes the following information:

- **Success**: Indicates whether or not an individual request was successful. If *true*, the individual request was successful. If *false*, the individual request was not successful.
- **RequestId**: The Request ID is the same thing as a Mining ID. It is used by CallMiner to track the individual contact through the mining system. The Mining ID is created after sending the request to the CallMiner mining system and links the session to a particular mining job.
- **Message**: If the request wasn't successful, this field will provide reasons why the request failed.

Upload Metadata by Filtering Using DateRange

This array, sent in a request with the **ExactMatch** and **Updates** arrays, allows you to add or update metadata to an individual contact or multiple contacts within a date range. The **MatchOption** and **MatchType** fields within this filter offer four different update options. You can:

- Add or update metadata to the latest contact within the date range you provided.
- Add or update metadata to the latest contact within the range and any contacts associated with that contact through a customer journey.
- Add or update metadata to all the contacts within the provided range.
- Add or update all the contacts within the provided range and any contacts associated with the range of contacts through a customer journey.

See the Note under the **MatchType** field for more information on how to configure each option.

To upload metadata using the **DateRange** filter, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatafiltered** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.
2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:

- **MetadataItems** (array): This is the element container that includes the metadata items used to filter the contacts in your Eureka system and the metadata items to update for the contact (s).
  - **ExactMatch** (array): ***Required***. An array containing key-value pairs of metadata used to identify the contact(s) you want to update. The `ExactMatch` allows you to filter your contacts by metadata items like agent name, department, location, etc., so the API knows which contact(s) to apply your metadata updates to.

    > **Note**: Be sure to use the most specific metadata item combinations possible as filters in this route. For example, you might filter using multiple metadata values like a specific location, department, an agent name, etc. to more narrowly choose the particular contact(s) you want to update. By using very specific and targeted metadata combinations, it ensures that you're updating the correct contact(s) in your Eureka system.

    > **Note**: The API processes the key-value pairs in order, so it's recommended to list broader metadata items, like location, before specific items, like agent name.

    > **Note**: The key-value pairs entered here do not have to be the same metadata items you want to update in the Update element.

    - **Key** (string): The metadata item you want to use to filter contacts. For example, you can enter *Location* to filter for only the contacts with a specific location.
    - **Value** (string): The corresponding metadata value to filter by. For example, you can enter *New York* to filter for only the contacts whose location is New York.
  - **DateRange** (array): Used to identify the contact(s) you want to update. This array contains the start and end dates that you wish to use to filter contacts. When using `DateRange`, based on your `MatchType` setting, the API will update either the most recent contact within the range or all contacts within the range that match the metadata items in the `ExactMatch` array. Based on your `MatchOption` setting, the API will update either the individual contacts that match the metadata items you provided in the `ExactMatch` array or all of the contacts associated to those matching contacts through a customer journey. See the Note under `MatchType` for more information about how to configure each option.

- **StartTime** (string): The beginning of the datetime range of contacts in ISO 8601. See DateTime Formatting for more information about ISO 8601 format.
- **EndTime** (string): The ending of the datetime range of contacts in ISO 8601.

> **Note**: The difference between `StartTime` and `EndTime` cannot exceed 30 days.

- **MatchOption** (string): Choose from *Journey* and *Normal*. If *Normal*, updates one contact within the range with the metadata updates. If *Journey*, updates all matching contacts within the datetime range and all contacts associated to those matching contacts through a customer journey with the metadata updates.
- **MatchType** (string): Determines whether to update all contacts within the date range or only the latest contact within the date range. Enter *Latest* to update the most recent contact within the range, or enter *All* to update all contacts within the range. The default value is *Latest*.

> **Warning**: Remember to use specific and targeted metadata value combinations as filters on this route! If you use a common metadata value as the filter in the `ExactMatch` array, it is possible to update every contact in your system within the range you provided, depending on how you configure the `MatchOption` and `MatchType` attributes. For example, if you use a *New York* Location metadata value as the only filter and you set `MatchType` to *All*, you could update every contact in your Eureka system that was recorded at your New York location within the provided range. Once you send the request, it may be difficult or even impossible to reverse the changes.

> **Note**: The `MatchOption` and `MatchType` fields give you four options for updating your contacts:
>
> - Set `MatchType` to *Latest* and `MatchOption` to *Normal* to add or update metadata to the latest contact within the date range you provided.
> - Set `MatchType` to *Latest* and `MatchOption` to *Journey* to add or update metadata to the latest contact within the date range and any contacts associated with that contact through a customer journey.
> - Set `MatchType` to *All* and `MatchOption` to *Normal* to add or update metadata to all the contacts within the provided date range.
> - Set `MatchType` to *All* and `MatchOption` to *Journey* to add or update metadata to all the contacts within the provided date range and any contacts associated with the range of contacts through a customer journey.

- **Updates** (array): ***Required***. The metadata updates to make to the contact(s). Each `Updates` array contains key-value pairs for the metadata items you want to add or update for the contact(s) in your Eureka system.
  - **Key** (string): The key component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
  - **Value** (string): The literal value for the metadata item that corresponds to the associated `key`. Enter the value that you want to update the metadata item with. This will replace any existing value for the metadata item. For example, you might enter the `value` *Bob Smith* with the *Agent* `key` to update a particular agent's name from Robert Smith to Bob Smith.

> **Note**: The following metadata items cannot be updated:
>
> - ClientId
> - CreateDate
> - RequestId
> - IsSample
> - AudioSourceName
> - CorrelationId
> - TenantApiKey
> - ParentID

- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.

The following example shows a sample request to upload metadata using the `DateRange` filter with all values, other than the Ingestion API URL and authorization token, populated:

```
POST /api/session/metadatafiltered HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8
{
   "MetadataItems": [
      {
         "ExactMatch": [
            {
               "Key": "Agent",
               "Value": "Johnny Johnson"
            }
         ],
         "DateRange": {
            "StartTime": "2019-04-1T15:52:19.398Z",
            "EndTime": "2019-04-7T15:52:19.398Z",
            "MatchOption": "Normal",
            "MatchType": "Latest"
         },
```

```
         "Updates": [
            {
               "Key": "Agent",
               "Value": "John Johnson"
            }
         ]
      }
   ],
   "SourceId": "AudioSourceName"
}
```

4. Send the request.

   One of the following will happen:

   - If an error occurs, the API returns a status code in the **400** to **500** range. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** status code, and the body of the response includes the Request ID.

   The following shows a sample response:

```
[
   {
         "Success": true,
         "RequestId": "5295431-3fy1-4gf2-b998-2e464de33652",
         "Message": []
   }
]
```

   The response includes the following information:

   - **Success**: Indicates whether or not an individual request was successful. If *true*, the individual request was successful. If *false*, the individual request was not successful.
   - **RequestId**: The Request ID is the same thing as a Mining ID. It is used by CallMiner to track the individual contact through the mining system. The Mining ID is created after sending the request to the CallMiner mining system and links the session to a particular mining job.
   - **Message**: If the request wasn't successful, this field will provide reasons why the request failed.

3. 4 | Upload Text-Based Media and Metadata

The CallMiner Media Ingestion API enables you to upload chat, survey, email, Twitter, Facebook, and generic text-based contacts into the CallMiner Eureka mining platform. Using the routes in this section, you are able to upload the various text-based interactions between your agents and your customers so that you can gain business intelligence and insights into what's going on in your contacts.

You have the following options to choose from when ingesting text-based media and metadata:

- Upload All Types of Text-Based Media and Metadata: This route allows you to ingest chat, survey, email, Twitter, Facebook, and generic text-based contacts into the CallMiner Eureka mining platform.
- Upload Email-Based Media and Metadata: This route is email-specific and allows you to upload more email-specific information (such as recipients, attachment names, etc.) to help further analyze your email-based contacts.
- Update an Email Contact: This route allows you to append new emails to an email contact thread that already exists in your Eureka system.

3. 4. 1 | Upload All Types of Text-Based Media and Metadata

The Transcript route is a simple, scalable way to add your text-based contacts and related metadata to the Eureka mining system for processing. You can use this route to upload chat, survey, email, Twitter, Facebook, and generic text-based contacts. Once you have uploaded your text-based contacts to the Eureka system, you can then analyze them and gain useful insights into your customer interactions.

> **Note**: You need a valid authentication token to complete requests. For more information on getting a token, see Login and Authentication.

To upload text-based contacts and metadata, complete the following steps:

1. Create a **POST** request with the **/api/transcript** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the POST request you will create with everything except for the Ingestion API URL and authorization token populated:

```
POST /api/transcript HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "Transcript": [
   {
      "Speaker": 0,
      "Text": "Hello, how are you today?",
      "PostDateTime": "2019-06-12T14:21:04.6747897Z",
      "TextInformation": "agent@example.com"
   },
   {
      "Speaker": 1,
      "Text": "I'm good
      Thank you

      🙂",
      "PostDateTime": "2019-06-12T14:22:07.5682357Z",
      "TextInformation": "customer@example.com"
   }
   ],
   "Metadata": [
      {
```

```
            "Key": "Agent",
            "Value": "Bob"
        }
    ],
    "MediaType": "Chat",
    "ClientCaptureDate": "2020-02-11T16:38:16.8041674Z",
    "SourceName": "ExampleSourceName",
    "CorrelationId": "e4b69e6f-72b7-4d74-891e-36b81ab9b576"
}
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **Transcript** (array) (optional): The list of raw transcript items of the media file you are uploading in the session. A `Transcript` item represents one speaker's message in the overall contact. For example, if you're uploading a chat between your agent and customer, you could have one `Transcript` item for your agent's message and another for your customer's message. Together, the two `Transcript` items combine to form the overall contact. Each `Transcript` item in the contact may have some or all of the following information:
        - **Speaker** (integer): The Speaker ID assigned to the speaker in your Eureka system. For example, your agent might be labeled as *1* and the customer might be labeled as *2*.

          > **Note**: To get a list of speakers in your Eureka system and their associated IDs, use the **GET /api/v2/speakers** route in the Data API.

        - **Text** (string): The text written in the raw transcript. This also includes any punctuation, special characters, emojis, and hashtags written in the original transcript string.
            - Some rules about the `Text` string:
                - You must escape all special characters in the transcript that are also used in JSON, such as quotation marks ("), backslashes (\), forward slashes (/), etc., by adding a backslash (\) before each character in the JSON document.

> **Example**: If you want to ingest a transcript with the word "technically" in quotes, you would enter it into the JSON document as `\"technically\"`.

- Any emojis you have in the text of the raw transcript should be formatted as the literal emoji, not the Unicode character representation of the emoji. If you send the Unicode character representation of an emoji into the Ingestion API, the emoji will not render correctly in Analyze.
- Your Eureka system supports most emojis up to version 11.0 and most Unicode characters up to version 5.0.
- If your transcript is too large, it will be reflected as a mining failure in the Mining Summary report, and you can resubmit the transcript in smaller increments.

> **Note**: After you send in the text for each message in the text contact, your Eureka system will transform the formatting of the text behind the scenes for display purposes. For more information on what this formatting entails, see How CallMiner Transforms Text.

  - **PostDateTime** (string): The date and time that the transcript you are uploading was originally posted. It should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See ISO 8601 DateTime Formatting for more information.
  - **TextInformation** (string): The information that further identifies the speaker in the transcript event. This includes things like a speaker's username, handle, email, etc.

- **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The **`Metadata`** array must contain at least one metadata item. Each metadata item consists of a **`key`** and a **`value`**, representing a single piece of metadata to be applied to a given contact:
  - **Key** (string): The **`key`** component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The **`ColumnName`** attribute for a metadata item in that route is the metadata **`key`** you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data

API **/api/v2/metadata** route.

- **Value** (string): The literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **MediaType** (string): The type of text-based contact you are uploading. You can upload any CallMiner-supported text-based contact type by choosing one of the options below and entering it into the body of the request within quotation marks:
  - ***Chat***: Chat-based contacts, such as SMS, MMS, or online chat interactions.
  - ***Email***: Email contacts.

    > **Note**: You can also use the Upload Email-Based Media and Metadata route to upload emails. That route is email-specific and it allows you to upload more email-specific metadata--such as recipient info, attachment names, etc.--to help further identify and analyze your email-based contacts.

  - ***Survey***: Survey contacts that gather feedback from users.
  - ***TwitterMedia***: Twitter contacts, such as tweets, replies, reply chains, mentions, etc.
  - ***FacebookMedia***: Facebook contacts, such as posts, messages, comments, etc.
  - ***Generic***: Other contact types that may or may not have user turns (e.g. legal documents).
- **ClientCaptureDate** (string): The localized date and time that the text contact was originally posted. The `ClientCaptureDate` should be entered in ISO 8601 format, including time zone off-

set. If no time zone offset is supplied, the date is assumed to be in UTC. See ISO 8601 DateTime Formatting for more information.

- **SourceName** (string): The name of the ingestion source. The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available sources, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and Initial System Configuration in Analyze to view the available sources in the left pane. A source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string) (optional): The unique identifier for the contact. Typically a UUID data type, though any kind of string value is acceptable. Its purpose is:
    - To uniquely identify an interaction within the context of your CallMiner account. You can track the success or failure of media and metadata using the Correlation ID in the Mining Summary section of Analyze or via the SystemMetricsController routes in the CallMiner Data API.
    - To provide a means of correlating the disparate parts of an interaction, such as media and metadata, with each other. For example, if you send the media for a contact in one request and you want to add or update the contact's associated metadata at a later time with a separate request, you would use the Correlation ID to tell your Eureka system that the metadata should be attached to the original contact.

> **Note**: For email contacts, CallMiner recommends using the value of the Message ID header for the first email in the email chain as the Correlation ID for the contact. The Message ID is a unique identifier for an email. Any subsequent emails in the email contact chain will contain the original email's Message ID in the References header. You can use the subsequent emails' References headers to track which emails make up the larger email contact chain, and because the Message ID of the email is stored in the email headers, you don't have to record the Message ID (Correlation ID) on your own system. When you need the Correlation ID to append new emails to the original contact thread, you can always find the Correlation ID in the References header of each subsequent email. See the Common Internet Message Headers for more information.

> **Note**: If you choose not to supply the Correlation ID for a given request, a UUID value will be auto-generated for you. Please note that this value should uniquely identify a particular contact. If you supply a Correlation ID that matches another ID in your system, the contact will fail to publish in your Eureka system. If your recording system offers a unique identifier for each contact, use this value as the Correlation ID.

4. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** HTTP status code.

     A successful response includes the following:

     - **CorrelationId**: The string value that pairs a text transcript with any applicable metadata files.
     - **MiningId**: The ID created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

   The following example shows a sample response:

```
{
  "CorrelationId": "b41738cd-bcbb-42f2-8d4c-7066d7d204ed",
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

5. Record the Correlation ID. If you did not supply a Correlation ID during the creation of the session, record the system-generated value to track this contact through the mining system via the Analyze **Admin** tab or the CallMiner Data API.

3. 4. 2 | Upload Email-Based Media and Metadata

> **Note**: This route only allows you to upload email contacts. To upload other types of text-based contacts, see Upload Text-Based Media and Metadata.

The CallMiner Media Ingestion API enables you to upload email contacts into the CallMiner Eureka mining platform. An email contact--otherwise known as an email thread--can be made up of a single email or multiple emails that comprise a conversation chain. The email transcript route is a simple way to add your email contacts and their related metadata into the mining system for processing.

This route is specifically designed to ingest emails and email threads to provide you with the ability to analyze complex interactions between your business and your customers.

> **Example**: A common use-case for this route is a customer emails your business and one of your agents replies, requesting more information from the customer. Once the customer replies, your agent forwards the customer's email to another agent in your organization who is better suited to answer the customer's questions. The second agent replies, and the customer's questions are resolved. This interaction, which is made up of more than five individual emails from three separate participants all combine to form one email thread.

This route allows you to ingest this complex email thread as a single contact with a single Contact ID, while also maintaining the identifying information for each individual email that makes up the overall contact. The route preserves the recipient information, the attachment names, etc. for each individual email within the email contact thread. That way, you can perform complex analysis on the entire interaction between your customer and your agents, from the first email in the thread to the last.

> **Note**: You need a valid authentication token to complete requests. For more information on getting a token, see Login and Authentication.

To upload email contacts and metadata, complete the following steps:

1. Create a **POST** request with the **/api/transcript/email** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the POST request you will create with everything except for the Ingestion API URL and authorization token populated:

```
POST /api/transcript/email HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "Metadata": [
      {
         "Key": "Agent",
         "Value": "John Smith"
      }
   ],
   "ClientCaptureDate": "2020-01-30T12:57:22.7195023Z",
   "SourceName": "ExampleSourceName",
   "CorrelationId": "b304a537-4bbf-4e4e-b5ef-01ac74642fc2",
   "Transcript": [
      {
         "AttachmentNames": [
            "WelcomePacket.pdf",
            "NewCustomerForms.docx"
         ],
         "RecipientBCC": [
            "SupportTicket@example.com"
         ],
         "RecipientCC": [],
         "RecipientTo": [
            "customer@example.com"
         ],
         "Subject": "Just saying hi",
         "Speaker": 0,
         "Text": "Hello, how are you today? Please review the WelcomePacket pdf
         and then fill out and return the NewCustomerForms document. Thanks!",
         "PostDateTime": "2020-01-30T12:57:22.7195023Z",
         "From": "agent@example.com"
      },
      {
         "AttachmentNames": [
            "NewCustomerForms.docx"
         ],
         "RecipientBCC": [],
         "RecipientCC": [],
         "RecipientTo": [
            "agent@example.com"
         ],
         "Subject": "RE: Just saying hi",
         "Speaker": 1,
```

```
            "Text": "I am good, thank you! I filled out the NewCustomerForms and
            attached it.",
            "PostDateTime": "2020-01-30T12:57:22.7195023Z",
            "From": "customer@example.com"
        }
    ],
    "EmailThreadComplete": true
}
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The **`Metadata`** array must contain at least one metadata item. Each metadata item consists of a **`key`** and a **`value`**, representing a single piece of metadata to be applied to a given contact:
        - **Key** (string): The **`key`** component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The **`ColumnName`** attribute for a metadata item in that route is the metadata **`key`** you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
        - **Value** (string): The literal value that corresponds to the associated **`key`**. For example, if the **`key`** for a metadata item is *Agent*, the **`value`** could be *John Smith*.

> **Note**: The metadata **`key`** you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **ClientCaptureDate** (string): The localized date and time that the text contact was originally posted. The `ClientCaptureDate` should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See ISO 8601 DateTime Formatting for more information.
- **SourceName** (string): The name of the ingestion source. The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available sources, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and Initial System Configuration in Analyze to view the available sources in the left pane. A source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string) (optional): The unique identifier for the contact. Typically a UUID data type, though any kind of string value is acceptable. Its purpose is:
    - To uniquely identify an interaction within the context of your CallMiner account. You can track the success or failure of media and metadata using the Correlation ID in the Mining Summary section of Analyze or via the SystemMetricsController routes in the CallMiner Data API.
    - To provide a means of correlating the disparate parts of an interaction, such as media and metadata, with each other. For example, if you send the media for a contact in one request and you want to add or update the contact's associated metadata at a later time with a separate request, you would use the Correlation ID to tell your Eureka system that the metadata should be attached to the original contact.

> **Note**: For email contacts, CallMiner recommends using the value of the Message ID header for the first email in the email chain as the Correlation ID for the contact. The Message ID is a unique identifier for an email. Any subsequent emails in the email contact chain will contain the original email's Message ID in the References header. You can use the subsequent emails' References headers to track which emails make up the larger email contact chain, and because the Message ID of the email is stored in the email headers, you don't have to record the Message ID (Correlation ID) on your own system. When you need the Correlation ID to append new emails to the original contact thread, you can always find the Correlation ID in the References header of each subsequent email. See the Common Internet Message Headers for more information.

> **Note**: If you choose not to supply the Correlation ID for a given request, a UUID value will be auto-generated for you. Please note that this value should uniquely identify a particular contact. If you supply a Correlation ID that matches another ID in your system, the contact will fail to publish in your Eureka system. If your recording system offers a unique identifier for each contact, use this value as the Correlation ID.

- **Transcript** (array) (optional): The list of individual emails that compose the larger email contact thread that you are uploading in the session. Each `Transcript` item represents one speaker's individual email in the larger email thread contact. Each email in the email thread may have some or all of the following information:
    - **AttachmentNames** (array): The file names of attachments on the email. Attachment names should be written as a list of string values.
    - **RecipientBCC** (array): The BCC recipients of the email. BCC recipients should be provided as a list of string values.
    - **RecipientCC** (array): The CC recipients of the email. CC recipients should be provided as a list of string values.
    - **RecipientTo** (array): The direct recipients of the email. Direct recipients should be written as a list of string values.
    - **Subject** (string): The subject line of the email.
    - **Speaker** (integer): The Speaker ID assigned to the speaker in your Eureka system. For example, your agent might be labeled as *1* and the customer might be labeled as 2.

> **Note**: To get a list of speakers in your Eureka system and their associated IDs, use the **GET /api/v2/speakers** route in the Data API.

- **Text** (string): The text written in the raw transcript. This also includes any punctuation, special characters, emojis, and hashtags written in the original transcript string.
    - Some rules about the `Text` string:
        - You must escape all special characters in the transcript that are also used in JSON, such as quotation marks ("), backslashes (\), forward slashes (/), etc., by adding a backslash (\) before each character in the JSON document.

          > **Example**: If you want to ingest a transcript with the word "technically" in quotes, you would enter it into the JSON document as `\"technically\"`.

        - Any emojis you have in the text of the raw transcript should be formatted as the literal emoji, not the Unicode character representation of the emoji. If you send the Unicode character representation of an emoji into the Ingestion API, the emoji will not render correctly in Analyze.
        - Your Eureka system supports most emojis up to version 11.0 and most Unicode characters up to version 5.0.
        - If your transcript is too large, it will be reflected as a mining failure in the Mining Summary report, and you can resubmit the transcript in smaller increments.

          > **Note**: After you send in the text for each message in the text contact, your Eureka system will transform the formatting of the text behind the scenes for display purposes. For more information on what this formatting entails, see How CallMiner Transforms Text.

- **PostDateTime** (string): The date and time that the transcript you are uploading was originally posted. It should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See ISO 8601 DateTime Formatting for more information.

- **From** (string): The email address for the speaker of the transcript event.

- **EmailThreadComplete** (boolean): If *true*, the email conversation has finished. If *false*, the email contact will not be processed until you update the contact to say it is complete or a certain configurable time period elapses. This allows you to make additional metadata and transcript updates before the contact is mined. Default is *true*.

4. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** HTTP status code.

     A successful response includes the following:

     - **CorrelationId**: The string value that pairs a text transcript with any applicable metadata files.
     - **MiningId**: The ID created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

   The following example shows a sample response:

```
{
  "CorrelationId": "b41738cd-bcbb-42f2-8d4c-7066d7d204ed",
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

3. 4. 3 | Update an Email Contact

> **Note**: This route allows you to append new emails to an email contact thread that already exists in your Eureka system. See Upload Email-Based Media and Metadata or Upload Text-Based Media and Metadata if you'd like to upload a new email contact into your Eureka system.

The CallMiner Media Ingestion API enables you to upload email contacts into the CallMiner Eureka mining platform. An email contact--otherwise known as an email thread--can be made up of a single email or multiple emails that comprise a conversation chain.

This route provides you with the ability to update your existing email contacts, so you can perform complex analysis on the entire interaction between your customer and your agents, from the first email in the thread to the last. With this route, you can take new emails that your agents send or receive after you've ingested the original email contact thread and append those new emails to the existing thread. That way, if your agents continue to interact with your customer, you can add the latest emails to the chain and expand the original email contact to keep track of the growing interaction.

You can also use this route to append a new email to the email contact chain while simultaneously updating the metadata for the larger email contact. This is useful if, for instance, a new agent has been assigned to handle the interaction, or if the customer sent the first email in the chain and no agent was assigned to the interaction until a follow-up email was sent. When you append the latest email to the chain, you can update the agent metadata information at the same time.

> **Note**: You need a valid authentication token to complete requests. For more information on getting a token, see Login and Authentication.

To append a new email to an existing email contact thread, complete the following steps:

1. Create a **PUT** request with the **/api/transcript/email** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the PUT request you will create with everything except for the Ingestion API URL and authorization token populated:

   ```
   PUT /api/transcript/email HTTP/1.1
   Host: INGESTION_API_URL
   Authorization: Bearer AUTH_TOKEN
   Content-Type: application/json; charset=utf-8
   ```

```
{
   "EmailThreadComplete": true,
   "EmailInformation": [
      {
         "AttachmentNames": [
            "NewCustomerForms.docx"
         ],
         "RecipientBCC": [
            "supportTicket@example.com"
         ],
         "RecipientCC": [],
         "RecipientTo": [
            "agent@example.com"
         ],
         "Subject": "RE: Just saying hi",
         "Speaker": 0,
         "Text": "Did you receive the attachments that I sent you earlier?

         Best,
         Fred Jones",
         "PostDateTime": "2020-02-26T15:56:21.322039Z",
         "From": "customer@example.com"
      }
   ],
   "Metadata": [
      {
      "Key": "Agent",
      "Value": "John Smith"
      }
   ],
   "ClientCaptureDate": "2020-02-26T15:56:21.322039Z",
   "SourceName": "ExampleSourceName",
   "CorrelationId": "2fd72fa1-3452-4ca7-b7ad-fa4ae0f53bb2",
   "UpdateType": "Append"
}
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **EmailThreadComplete** (boolean): If *true*, the email conversation has finished. If *false*, the email contact will not be processed until you update the contact to say it is complete or a

certain configurable time period elapses. This allows you to make additional metadata and transcript updates before the contact is mined. Default is *true*.

- **EmailInformation** (array) (optional): The list of individual emails you want to append to the existing email contact thread. Each `EmailInformation` item represents one speaker's individual email in the larger email thread contact. Each email in the email thread may have some or all of the following information:
    - **AttachmentNames** (array): The file names of attachments on the email. Attachment names should be written as a list of string values.
    - **RecipientBCC** (array): The BCC recipients of the email. BCC recipients should be provided as a list of string values.
    - **RecipientCC** (array): The CC recipients of the email. CC recipients should be provided as a list of string values.
    - **RecipientTo** (array): The direct recipients of the email. Direct recipients should be written as a list of string values.
    - **Subject** (string): The subject line of the email.
    - **Speaker** (integer): The Speaker ID assigned to the speaker in your Eureka system. For example, your agent might be labeled as *1* and the customer might be labeled as *2*.

      > **Note**: To get a list of speakers in your Eureka system and their associated IDs, use the **GET /api/v2/speakers** route in the Data API.

    - **Text** (string): The text written in the raw transcript. This also includes any punctuation, special characters, emojis, and hashtags written in the original transcript string.
        - Some rules about the `Text` string:
            - You must escape all special characters in the transcript that are also used in JSON, such as quotation marks ("), backslashes (\), forward slashes (/), etc., by adding a backslash (\) before each character in the JSON document.

              > **Example**: If you want to ingest a transcript with the word "technically" in quotes, you would enter it into the JSON document as `\"technically\"`.

- Any emojis you have in the text of the raw transcript should be formatted as the literal emoji, not the Unicode character representation of the emoji. If you send the Unicode character representation of an emoji into the Ingestion API, the emoji will not render correctly in Analyze.
- Your Eureka system supports most emojis up to version 11.0 and most Unicode characters up to version 5.0.
- If your transcript is too large, it will be reflected as a mining failure in the Mining Summary report, and you can resubmit the transcript in smaller increments.

> **Note**: After you send in the text for each message in the text contact, your Eureka system will transform the formatting of the text behind the scenes for display purposes. For more information on what this formatting entails, see How CallMiner Transforms Text.

- **PostDateTime** (string): The date and time that the transcript you are uploading was originally posted. It should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See ISO 8601 DateTime Formatting for more information.
- **From** (string): The email address for the speaker of the transcript event.

- **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The `Metadata` array contains at least one metadata item. Each metadata item consists of a `key` and a `value`, representing a single piece of metadata to be applied to a given contact:
  - **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
  - **Value** (string): The value of a metadata item is the literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a `key` that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if the attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

> **Note**: When you are updating metadata, the values of the following metadata keys cannot be updated:
>
> - ClientCaptureDate
> - CorrelationId
> - RequestId
> - SourceName
> - MessageTotal

- **ClientCaptureDate** (string): The localized date and time that the text contact was originally posted. The `ClientCaptureDate` should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See DateTime Formatting for more information about ISO 8601 format.
- **SourceName** (string): The name of the ingestion source. The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available sources, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and Initial System Configuration in Analyze to view the available sources in the left pane. A source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string): The unique identifier for the contact. This tells your Eureka system which original email contact thread the new email should be appended to. If you supplied a Correlation ID when you originally ingested the contact, enter that value here. If you did not provide a Correlation ID, enter the Correlation ID value you received in the response when you originally ingested the contact. You may also find this value in Analyze once the contact

has gone through the mining system.

> **Note**: If fields were given different friendly names in your Eureka system, the field you see in Analyze called *Correlation ID* may not be the same Correlation ID that was used to ingest the contact's media. Check with your system administrator to find which field and value you should be using to pair the metadata with the appropriate media.

> **Note**: If you used the Message ID of the original email in the email thread as the Correlation ID for the contact, enter the Message ID value here, which can be found in the References header of the email you are trying to append to the original contact chain.

- **UpdateType** (string): The type of update being performed on the contact. Enter *Append* in this field.

4. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** HTTP status code.

     A successful response includes the following:

     - **CorrelationId**: The string value that pairs a text transcript with any applicable metadata files.
     - **MiningId**: The ID created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

   The following example shows a sample response:

```
{
  "CorrelationId": "b41738cd-bcbb-42f2-8d4c-7066d7d204ed",
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

3. 5 | Upload Video Media and Metadata

This section describes how to upload video media and metadata into your Eureka system. Using this section, you can ingest a single video contact, paired with its associated metadata, into your Eureka system for processing. Once you have uploaded your video contact to the Eureka system, you can then analyze it and gain useful insights into your customer interactions. With this section, you can:

- Upload a single video contact with its associated metadata.
- Take a video meeting that you have recorded and ingest it into your Eureka system for analysis.
- Ingest a video meeting between your sales department and your customer and then analyze the interaction in Eureka to see how you can improve your sales process.

Uploading your video contacts into your Eureka system is a two-step process. First, you will start a session using the **POST /api/session/metadatamedia** route. Then, you will upload your video media file using the **POST /api/media/{sessionId}** route.

> **Important**: If your video file is more than 128 megabytes in size, you must use a third-party application to ingest the contact. The interactive documentation does not support video file ingestion if the file exceeds 128 megabytes. With a third-party application, however, you can ingest video contacts that are up to one gibibyte in size.

3. 5. 1 | Start a Session to Upload Video Media and Metadata

To start a session to upload a video contact, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatamedia** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the POST request you will create, with everything except for the Ingestion API URL and authorization token populated:

```
POST /api/session/metadatamedia HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "Metadata": [
   {
      "Key": "Agent",
```

```
      "Value": "John Smith"
    }
    ],
    "TotalMediaLength": 300000,
    "MediaType": "video/mp4",
    "ClientCaptureDate": "2019-08-01T12:01:22.674Z",
    "VoicePrintIdentifier": "John Smith",
    "AudioDecryptionKey": "",
    "SourceId": "ExampleSourceId",
    "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **TotalMediaLength** (integer): The number of bytes in the media file to be uploaded in this session. Set this value to the size of the media file you are uploading.

      > **Note**: If your video file is more than 128 megabytes in size, you must use a third-party application to ingest the contact. The interactive documentation does not support video file ingestion if the file exceeds 128 megabytes. With a third-party application, however, you can ingest video contacts that are up to one gibibyte in size.

    - **MediaType** (string): The MIME type of the media file you are uploading. Enter *video/mp4* for video contacts. If you have extracted the audio from your video contact and made it into an M4A file, enter *audio/m4a* for this value. See Supported MIME types for more information.
    - **ClientCaptureDate** (string): The localized date and time that the video contact was originally recorded. The **`ClientCaptureDate`** should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See DateTime Formatting for more information about ISO 8601 format.
    - **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The **`Metadata`** array must contain at least one metadata item. Each metadata item consists of a **`key`** and a **`value`**, representing a single piece of metadata to be applied to a given contact:

- **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
- **Value** (string): The literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **VoicePrintIdentifier** (string) (optional): The VoicePrint identifier used in Speaker Separation. This field tells the system which agent is speaking in the contact. If you have Speaker Separation, set the field for `VoicePrintIdentifier` to the same value that corresponds to the Agent metadata key. For example, if you have a metadata item that has the `key` set to *Agent* and the `value` set to *John Smith*, you would enter *John Smith* into the `VoicePrintIdentifier` value as well. This value is only required if Speaker Separation is enabled for the particular Source ID.
- **AudioDecryptionKey** (string) (optional): The key for decrypting encrypted audio. This value is only required if Audio Encryption is enabled for the particular Source ID. Currently, only Aspect AQM encryption is supported. If you're unsure if you need to set this key, contact support@callminer.com.
- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial**

System Configuration in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.

- **CorrelationId** (string) (optional): Typically a Universally Unique Identifier (UUID) data type, though any kind of string value is acceptable. This value should uniquely identify a particular contact. If your recording system offers a unique identifier for each contact, use this value as the Correlation ID. The purpose of the Correlation ID is:
  - To uniquely identify an interaction within the context of your CallMiner account. You can track the success or failure of media and metadata using the Correlation ID in the Mining Summary section of Analyze or via the SystemMetricsController routes in the CallMiner Data API.
  - To provide a means of correlating the disparate parts of an interaction, such as media and metadata, with each other. For example, if you send the media for a contact in one request and you want to add or update the contact's associated metadata at a later time with a separate request, you use the Correlation ID to tell your Eureka system that the metadata should be attached to the original contact.

> **Note**: If you choose not to supply the Correlation ID for a given request, a UUID value will be auto-generated for you. If you supply a Correlation ID that matches another ID in your system, the contact will fail to publish in your Eureka system.

4. Send the request.

   One of the following will happen:

   - If an error occurs, an HTTP status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a **200** HTTP status code.

     A successful response includes the following:

     - **SessionId**: The system-generated UUID that identifies the media upload session and is used to properly associate an uploaded media file with the correct session.
     - **CorrelationId**: The string value that pairs a video file with the applicable metadata.

     The following example shows a sample response:

```
{
   "SessionId": "4b2bef4e-f1e4-4427-81ff-bd3aa7a7a9b5",
```

```
    "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

5. Record the Session ID. You will need to enter this value to upload the media in the next steps.
6. Record the Correlation ID. If you did not supply a Correlation ID during the creation of the session, record the system-generated value for future metadata updates to this contact or to track it through the mining system via the Analyze **Admin** tab or the CallMiner Data API.
7. Go to the next section and follow the steps there to upload your contact's video media.

3. 5. 2 | Upload the Video Media File

Once you've started a session for your video media and metadata, you can upload the video file for the contact and ingest the contact into your Eureka system. To do so, complete the following steps:

1. Create a **POST** request with the **/api/media/{sessionId}** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   The following is an example of the POST request you will create:

```
POST /api/media/{sessionId} HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: MIME_TYPE
Content-Length: LENGTH_IN_BYTES

BINARY_MEDIA_FILE
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
   - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
   - **Content-Type**: The MediaType value you entered in the previous section. Enter *video/mp4*.
   - **Content-Length**: The number of bytes in the media file. This value should match the `TotalMediaLength` you provided when you started your session in the previous section. It should also match the length of the binary media data in the file.

     > **Note**: Most third-party applications will auto-fill this header and value for you.

3. Replace the `{sessionID}` query parameter with the Session ID value that you received from the API in the previous section.
4. Add the binary file content for the media you are uploading.

   The following shows a sample request for a *video/mp4* file that is *319990* bytes in length. The request has a Session ID of *101491* and everything but the authorization token, Ingestion API URL, and binary file populated:

```
POST /api/media/101491 HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: video/mp4
Content-Length: 319990

BINARY_MEDIA_FILE
```

5. Send the request.

   One of the following will happen:

   - If an error occurs, an HTTP status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a `201 (Created)` HTTP status code and the body of the response.

   The following shows a sample response:

```
{
  "CurrentMediaLength": 319990,
  "TotalMediaLength": 319990,
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

   The response includes the following information:

   - **CurrentMediaLength**: The number of bytes that have been saved to disk for this media file during this session.
   - **TotalMediaLength**: The total number of bytes that need to be saved to disk to completely upload the media file, which you specified during the request.
   - **MiningId**: The unique identifier created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

You have now successfully uploaded a video contact. After the contact has gone through the mining system, you can analyze it in Eureka and gain useful insights into your customer interactions.

3. 6 | Upload a Contact with Multiple Media Files

This section describes how to upload a contact with multiple media files into your Eureka system. Using this section, you can ingest a single *audio/wav* file with the metadata for a contact, as well as a *video/fbs* file associated with the contact.

With this section, you can:

- Upload audio and video files for a contact with associated metadata.
- Take the audio and screen-recording files you recorded for a meeting and ingest them into your Eureka system for analysis.
- Ingest a screen-recorded meeting between your sales department and your customer and then analyze the interaction in Eureka to see how you can improve your sales process.

Uploading a contact with multiple media files into your Eureka system is a two-step process. First, you will start a session using the **POST /api/session/metadatamedia** route. Then, you will upload your media files using the **POST /api/media/multiple/{sessionId}** route.

3. 6. 1 | Start a Session to Upload Multiple Media Files and Metadata

To start a session to upload a contact with multiple media files, complete the following steps:

1. Create a **POST** request with the **/api/session/metadatamedia** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the POST request you will create, with everything except for the Ingestion API URL and authorization token populated:

```
POST /api/session/metadatamedia HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "Metadata": [
      {
         "Key": "Agent",
         "Value": "John Smith"
      }
   ],
   "TotalMediaLength": 300000,
   "MediaType": "audio/wav",
   "ClientCaptureDate": "2019-08-01T12:01:22.674Z",
```

```
    "VoicePrintIdentifier": "John Smith",
    "AudioDecryptionKey": "",
    "SourceId": "ExampleSourceId",
    "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4",
    "AdditionalMedia": [{
       "TotalMediaLength": 12121212,
       "MediaType": "video/fbs"
    }]
}
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **TotalMediaLength** (integer): The size (in bytes) of the audio media file to be uploaded in this session.

      > **Note**: You will enter the size and type of the video media file for the contact in the `AdditionalMedia` array below.

    - **MediaType** (string): Enter *audio/wav* for this value.

      > **Important**: *audio/wav* is the only supported value here when uploading a contact with multiple media files.

    - **ClientCaptureDate** (string): The localized date and time when the contact was originally recorded. The `ClientCaptureDate` should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See DateTime Formatting for more information about ISO 8601 format.
    - **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The `Metadata` array must contain at least one metadata item. Each metadata item consists of a `key` and a `value`, representing a single piece of metadata to be applied to a given contact:

- **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.
- **Value** (string): The literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **VoicePrintIdentifier** (string) (optional): The VoicePrint identifier used in Speaker Separation. This field tells the system which agent is speaking in the contact. If you have Speaker Separation, set the field for `VoicePrintIdentifier` to the same value that corresponds to the Agent metadata key. For example, if you have a metadata item that has the `key` set to *Agent* and the `value` set to *John Smith*, you would enter *John Smith* into the `VoicePrintIdentifier` value as well. This value is only required if Speaker Separation is enabled for the particular Source ID.
- **AudioDecryptionKey** (string) (optional): The key for decrypting encrypted audio. This value is only required if Audio Encryption is enabled for the particular Source ID. Currently, only Aspect AQM encryption is supported. If you're unsure if you need to set this key, contact support@callminer.com.
- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial**

System Configuration in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.

- **CorrelationId** (string) (optional): The unique identifier for the contact. Typically a Universally Unique Identifier (UUID) data type, though any kind of string value is acceptable. If your recording system offers a unique identifier for each contact, use this value as the Correlation ID. The purpose of the Correlation ID is:
    - To uniquely identify an interaction within the context of your CallMiner account. You can track the success or failure of media and metadata using the Correlation ID in the Mining Summary section of Analyze or via the **SystemMetricsController** routes in the CallMiner Data API.
    - To provide a means of correlating the disparate parts of an interaction, such as media and metadata, with each other. For example, if you send the media for a contact in one request and you want to add or update the contact's associated metadata at a later time with a separate request, you use the Correlation ID to tell your Eureka system that the metadata should be attached to the original contact.

    > **Note**: If you choose not to supply the Correlation ID for a given request, a UUID value will be auto-generated for you. If you supply a Correlation ID that matches another ID in your system, the contact will fail to publish in your Eureka system.

- **AdditionalMedia** (array): The information about the additional video media file for the contact.

    > **Note**: You may only enter one *video/fbs* file in this `AdditionalMedia` array.

    - **TotalMediaLength** (integer): The size (in bytes) of the video media file to be uploaded in this session.
    - **MediaType** (string): Enter *video/fbs* for this value.

        > **Important**: *video/fbs* is the only supported file type for `AdditionalMedia`.

4. Send the request.

One of the following will happen:

- If an error occurs, an HTTP status code in the **400** or **500** range is returned. Refer to HTTP Status Codes for more information.
- If the request is successful, the API server responds with a **200** HTTP status code.

  A successful response includes the following:

  - **SessionId**: The system-generated UUID that identifies the media upload session and is used to properly associate an uploaded media file with the correct session.
  - **CorrelationId**: The string value that pairs the media files with the applicable metadata.

  The following example shows a sample response:

```
{
   "SessionId": "4b2bef4e-f1e4-4427-81ff-bd3aa7a7a9b5",
   "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

5. Record the Session ID. You will need to enter this value to upload the media files in the next steps.
6. Record the Correlation ID. If you did not supply a Correlation ID during the creation of the session, record the system-generated value for future metadata updates to this contact or to track it through the mining system via the Analyze **Admin** tab or the CallMiner Data API.
7. Go to the next section and follow the steps there to upload your contact's media.

3. 6. 2 | Upload the Media Files

Once you've started a session for uploading your media and metadata, you can upload the multiple media files for the contact and ingest the contact into your Eureka system. To do so, complete the following steps:

1. Create a **POST** request with the **/api/media/multiple/{sessionId}** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   The following is an example of the POST request you will create:

```
POST /api/media/multiple/{sessionId} HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
```

```
Content-Disposition: form-data; name="MainMedia"; filename="FILE_NAME"
Content-Type: audio/wav

(file data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="AdditionalMedia"; filename="FILE_NAME"
Content-Type: video/fbs

(file data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
   - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
3. Replace the `{sessionId}` URL path parameter with the Session ID value that you received from the API in the previous section.
4. (Skip this step if using a third-party application.) Upload the media files using the interactive documentation:
   a. For the audio media file you are uploading:
      i. Press the **Choose File** button for the `MainMedia` file.
      ii. A file manager window will open.
      iii. Choose the binary file for the audio media you are uploading. For example, *AudioFile.wav*.
      iv. Press **Open**.
   b. For the video media file you are uploading:
      i. Press the **Choose File** button for the `AdditionalMedia` file.
      ii. A file manager window will open.
      iii. Choose the binary file for the video media you are uploading. For example, *ScreenRecording.fbs*.
      iv. Press **Open**.
5. (Skip this step if using the interactive documentation.) Upload the media files using a third-party application. To do so:

a. Create a request body that contains a multipart form (`content-type: multipart/form-data`) with the following two sections:

   i. **Section 1**: Must be named **`MainMedia`** and contain the binary file for the audio media you are uploading.

   > **Important**: The **`content-type`** value for this section must match the **`MediaType`** value you entered for the audio file in the **POST /api/session/metadatamedia** route.

   ii. **Section 2**: Must contain the binary file for the video media you are uploading. While there is no set name requirement for this section, we recommend naming it **`AdditionalMedia`**.

   > **Important**: The **`content-type`** value for this section must match the **`MediaType`** value you entered for the video file in the **POST /api/session/metadatamedia** route.

The following shows a sample request that has a Session ID of *101498* and everything but the authorization token and Ingestion API URL populated:

```
POST /api/media/multiple/101498 HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN

Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="MainMedia"; filename="AudioFile.wav"
Content-Type: audio/wav

(file data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="AdditionalMedia";
filename="VideoFile.fbs"
Content-Type: video/fbs

(file data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

6. Send the request.

   One of the following will happen:

   - If an error occurs, an HTTP status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a `201 (Created)` HTTP status code and the body of the response.

   The following shows a sample response:

```
{
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

   The response includes the following information:

   - **MiningId**: The unique identifier created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

You have now successfully uploaded a contact with multiple media files. After the contact has gone through the mining system, you can analyze it in Eureka and gain useful insights into your customer interactions.

3. 7 | Upload a Contact and Append a Video or Screen Capture File to it at a Later Time

This section describes how to upload a contact with multiple media files into your Eureka system using separate media upload requests. Using this section, you can ingest a single contact (a media file and its associated metadata) in one media upload request and then you can return later and upload the associated video or screen capture file for the contact. This is especially helpful for contacts where the audio file may be available for ingestion immediately, while the associated video or screen capture file may take longer to be ready for upload.

> **Note**: To use this feature, your tenant must have `VideoPlaybackEnabled` set to *true*.

Uploading a contact with multiple media files into your Eureka system is a multi-step process. To upload your contact, complete the following in order:

3. 7. 1 | Start a Session to Upload Audio Media and Metadata

The first thing you need to do to upload your contact is start a session. To do so:

1. Create a **POST** request with the **/api/session/metadatamedia** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   This is an example of the POST request you will create, with everything except for the Ingestion API URL and authorization token populated:

```
POST /api/session/metadatamedia HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: application/json; charset=utf-8

{
   "Metadata": [
      {
         "Key": "Agent",
```

```
            "Value": "John Smith"
        }
    ],
    "TotalMediaLength": 8962048,
    "MediaType": "audio/wav",
    "ClientCaptureDate": "2019-08-01T12:01:22.674Z",
    "VoicePrintIdentifier": "John Smith",
    "AudioDecryptionKey": "",
    "SourceId": "ExampleSourceId",
    "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
    - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
    - **Content-Type**: Set the value to *application/json; charset=utf-8*.
3. Set the values for the following elements in the body of the request:
    - **TotalMediaLength** (integer): The size (in bytes) of the first media file to be uploaded in the next section (Upload the First Media File for the Contact).
    - **MediaType** (string): The MIME type of the first media file you are uploading. You can upload any CallMiner-supported media MIME type, such as *audio/wav*. See Supported MIME types for more information.
    - **ClientCaptureDate** (string): The localized date and time when the contact was originally recorded. The ClientCaptureDate should be entered in ISO 8601 format, including time zone offset. If no time zone offset is supplied, the date is assumed to be in UTC. See DateTime Formatting for more information about ISO 8601 format.
    - **Metadata** (array): The data that describes a contact, such as the agent name or customer account number. The `Metadata` array must contain at least one metadata item. Each metadata item consists of a `key` and a `value`, representing a single piece of metadata to be applied to a given contact:
        - **Key** (string): The `key` component of a metadata item is derived from a set of CallMiner metadata keys. To obtain a list of the metadata keys in your Eureka system, use the Retrieve Metadata Configurations route. The `ColumnName` attribute for a metadata item in that route is the metadata `key` you will enter in this route. A complete list of acceptable metadata keys can also be obtained by issuing a **GET** request to the Data API **/api/v2/metadata** route.

- **Value** (string): The literal value that corresponds to the associated `key`. For example, if the `key` for a metadata item is *Agent*, the `value` could be *John Smith*.

> **Note**: The metadata `key` you provide has to correspond to a key that CallMiner recognizes, or the metadata item will be ignored. This means that you have to first map your metadata to CallMiner's metadata before uploading a contact or changing metadata. For more information on how to do this, see Retrieve Metadata Configurations.

> **Note**: If you incorrectly format the `key` or `value` attribute for a metadata item, it may prevent the contact from being mined properly. For example, if a metadata attribute is marked as an *(integer)*, you must enter a whole digit number, like *1*. If you enter a word, like *one*, into this field, it may cause an invalid format error in the mining system and the contact may not mine properly.

- **VoicePrintIdentifier** (string) (optional): The VoicePrint identifier used in Speaker Separation. This field tells the system which agent is speaking in the contact. If you have Speaker Separation, set the field for `VoicePrintIdentifier` to the same value that corresponds to the Agent metadata key. For example, if you have a metadata item that has the `key` set to *Agent* and the `value` set to *John Smith*, you would enter *John Smith* into the `VoicePrintIdentifier` value as well. This value is only required if Speaker Separation is enabled for the particular Source ID.
- **AudioDecryptionKey** (string) (optional): The key for decrypting encrypted audio. This value is only required if Audio Encryption is enabled for the particular Source ID. Currently, only Aspect AQM encryption is supported. If you're unsure if you need to set this key, contact support@callminer.com.
- **SourceId** (string): The name of the ingestion source (otherwise known as an audio source). The ingestion source is the method by which CallMiner obtains media and metadata for ingestion. For a list of available ingestion sources and their respective source names, send a **GET** request to the **/api/v2/audiosources** route in the Data API or open **Admin** and then **Initial System Configuration** in Analyze to view the available ingestion sources. An ingestion source must be configured prior to sending requests with the CallMiner Media Ingestion API.
- **CorrelationId** (string) (optional): The unique identifier for the contact. If your recording system offers a unique identifier for each contact, we highly recommend that you use this value as the Correlation ID so it is easier to keep track of the contact between your system and

CallMiner's. It will also make appending the video or screen capture file to the contact easier in a later step.

4. (Skip this step if using a third-party application.) Remove the `AdditionalMedia` array from the request or leave the array empty.
5. Send the request.

   One of the following will happen:

   - If an error occurs, an HTTP status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a `200` HTTP status code.

     A successful response includes the following:

     - **SessionId**: The system-generated UUID that identifies the media upload session and is used to properly associate an uploaded media file with the correct session.
     - **CorrelationId**: The string value that pairs the media file with the applicable metadata.

     The following example shows a sample response:

```
{
   "SessionId": "4b2bef4e-f1e4-4427-81ff-bd3aa7a7a9b5",
   "CorrelationId": "1836dd5c-0291-4775-8009-85a117f932a4"
}
```

6. Record the Session ID. You will need to enter this value to upload the media files in the next steps.
7. Record the Correlation ID. You will need to enter this value to upload the video or screen capture file in a future step.
8. Go to the next section and follow the steps there to upload your contact's media.

3. 7. 2 | Upload the First Media File for the Contact

Once you've started a session for uploading your media and metadata, you can upload the first media file for the contact and ingest the contact into your Eureka system. To do so, complete the following steps:

1. Create a **POST** request with the **/api/media/{sessionId}** route and the CallMiner Media Ingestion API URL. To find your URL, see CallMiner Media Ingestion API v2 URLs.

   The following is an example of the POST request you will create:

```
POST /api/media/{sessionId} HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: AUDIO_TYPE
Content-Length: LENGTH_IN_BYTES

BINARY_MEDIA_FILE
```

2. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:
   - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
   - **Content-Type**: The MediaType value you entered in the previous section. For a list of all MIME types, see Supported MIME types.
   - **Content-Length**: The number of bytes in the media file. This value should match the TotalMediaLength you provided when you started your session in the previous section. It should also match the length of the binary media data in the file.

     > **Note**: Most third-party applications will auto-fill this header and value for you.

3. Replace the `{sessionID}` query parameter with the Session ID value that you received from the API in the previous section.
4. Add the binary file content for the media you are uploading.

   The following shows a sample request for an audio/wav file that is 8962048 bytes in length. The request has a Session ID of *101491* and everything but the authorization token, Ingestion API URL, and binary file populated:

```
POST /api/media/101491 HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type: audio/wav
Content-Length: 8962048

BINARY_MEDIA_FILE
```

5. Send the request.

   One of the following will happen:

- If an error occurs, a status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
- If the request is successful, the API server responds with a `201 (Created)` HTTP status code and the body of the response.

The following shows a sample response:

```
{
  "CurrentMediaLength": 8962048,
  "TotalMediaLength": 8962048,
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

The response includes the following information:

- **CurrentMediaLength**: The number of bytes that have been saved to disk for this media file during this session.
- **TotalMediaLength**: The total number of bytes that need to be saved to disk to completely upload the media file, which you specified during the request.
- **MiningId**: The ID created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

6. Go to the next section and follow the steps there to upload the video or screen capture file for the contact.

3. 7. 3 | Upload the Video or Screen Capture File

Once the video or screen capture file is ready for upload, return to the API and complete the following:

1. Choose which route you want to use to append the second file to the contact. Each route transforms the overall contact in slightly different ways:
   a. **Video Route** (**PUT /api/media/video/{correlationId}**): If you append a video file to an audio contact, the system will combine the two files and convert the audio contact into a full video contact in the back-end.
   b. **Screen Capture Route** (**PUT /api/media/screencapture/{correlationId}**): If you append a screen capture file to an audio contact, the contact will stay an audio contact, but the system will add a metadata flag to the contact indicating that it has a screen capture file associated with it. You can then use this metadata flag in Analyze to filter by contacts that have screen capture files associated with them.

2. Create a PUT request with your chosen route and the CallMiner Media Ingestion API URL (to find your URL, see CallMiner Media Ingestion API v2 URLs). For example:

   - The following is an example of the PUT request you might create if you are uploading a video file:

```
PUT /api/media/video/{correlationId} HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type:VIDEO_TYPE
Content-Length: LENGTH_IN_BYTES

BINARY_MEDIA_FILE
```

   - The following is an example of the PUT request you might create if you are uploading a screen capture file:

```
PUT /api/media/screencapture/{correlationId} HTTP/1.1
Host: INGESTION_API_URL
Authorization: Bearer AUTH_TOKEN
Content-Type:VIDEO_TYPE
Content-Length: LENGTH_IN_BYTES

BINARY_MEDIA_FILE
```

3. (Skip this step if using the interactive documentation.) Set the following HTTP request headers:

   - **Authorization**: Set the value to the word *Bearer* followed by a space and then the authorization token. For example, *Bearer abc1234*.
   - **Content-Type**: The MediaType value you entered in the previous section. You can choose either *video/fbs* or *video/mp4*. For a list of all MIME types, see Supported MIME types.
   - **Content-Length**: The number of bytes in the media file.

     > **Note**: Most third-party applications will auto-fill this header and value for you.

4. Replace the `{correlationId}` query parameter with the Correlation ID value that you received from the API in the previous Start a Session to Upload Audio Media and Metadata section.

5. Add the binary file content for the video or screen capture file you are uploading.

   > **Important**: Your file must either be a *video/fbs* or a *video/mp4* file.

6. Send the request.

   One of the following will happen:

   - If an error occurs, a status code in the `400` or `500` range is returned. Refer to HTTP Status Codes for more information.
   - If the request is successful, the API server responds with a `200 (Success)` HTTP status code and the body of the response.

   The following shows a sample response:

```
{
  "MiningId": "c7c5acb0-710a-43e6-b35e-f6238d9aa5f7"
}
```

   The response includes the following information:

   - **MiningId**: The ID created after sending the request to the CallMiner mining system. The Mining ID links the contact to a particular mining job. This is used by CallMiner for internal tracking.

You have now successfully uploaded a contact with multiple media files. After the contact has gone through the mining system, you can analyze it in Analyze or Coach and gain useful insights into your customer interactions.

Appendix A

CallMiner Media Ingestion API v2 URLs

Find your location below and use the appropriate link to make requests against the CallMiner Media Ingestion API v2 with a third-party application:

Location	Ingestion API v2 URL
United States (Primary)	https://api.callminer.net/ingestion
United States (FISMA/HITRUST)	https://apif.callminer.net/ingestion
United Kingdom	https://apiuk.callminer.net/ingestion
Australia	https://apiaus.callminer.net/ingestion
Canada	https://apica.callminer.net/ingestion
Europe (EU)	https://apiew.callminer.net/ingestion

If you would like to use the interactive documentation instead to access and use the CallMiner Media Ingestion API v2, choose the appropriate link below:

Location	Ingestion API v2 Interactive Documentation URL
United States (Primary)	https://api.callminer.net/ingestion/swagger
United States (FISMA/HITRUST)	https://apif.callminer.net/ingestion/swagger
United Kingdom	https://apiuk.callminer.net/ingestion/swagger
Australia	https://apiaus.callminer.net/ingestion/swagger
Canada	https://apica.callminer.net/ingestion/swagger
Europe (EU)	https://apiew.callminer.net/ingestion/swagger

Appendix B

CallMiner's Identity Provider Service URLs

The CallMiner Media Ingestion API v2 uses CallMiner's Identity Provider Service, an authentication application that employs industry-standard methods for user authentication and application authorization. With this authentication process, you can use a Client ID and Secret to request an access token from the Identity Provider. You can then use this token in the `Authorization` header in whatever third-party application you want to use to authenticate each request you make against the CallMiner Media Ingestion API v2.

Find your location below and use the appropriate link to request a token in the "Authenticating with a Third-Party Application" section of Login and Authentication:

Location	CallMiner's Identity Provider Service URL
United States (Primary)	https://idp.callminer.net
United States (FISMA/HITRUST)	https://idpf.callminer.net
United Kingdom	https://idpuk.callminer.net
Australia	https://idpaus.callminer.net
Canada	https://idpca.callminer.net
Europe (EU)	https://idpew.callminer.net

Appendix C

Supported MIME Types

The Multipurpose Internet Mail Extensions (MIME) type is a standard way to indicate the nature and format of a document. In the table below, the MIME type in the left column corresponds to the MediaType value on the right that should be used in requests to the CallMiner Media Ingestion API. If you want to upload a WAV file, for example, the value for the Content-Type query parameter should be set to *audio/wav* in the request.

MIME Type	MediaType Value
WAVE, WAV	audio/wav
MP3	audio/mp3
OGG	audio/ogg
VOX6, VOX8, VOX11	audio/vox
WMA	audio/wma
PCM	audio/pcm
M4A	audio/m4a
MP4	video/mp4

Appendix D

HTTP Status Codes

HTTP status codes are standard response codes given by web servers. They indicate whether or not a request has been successfully completed. Below is a list of common HTTP status codes that you may receive when you use the Ingestion API:

Code	Reason
200	OK - The GET, PUT, PATCH or DELETE request was successful. Can also be used for a POST request that doesn't result in a creation.
201	Created - The POST request was successful and a new resource was created as a result.
401	Unauthorized - The request requires authentication. If you get this error, double check that you've included the correct authorization token without typos. If the error persists, contact support@callminer.com.
415	Unsupported Media Type - The server is refusing to service the request because the media format for the requested data is not supported by the server.
429	Rate Limit Exceeded - The user has sent too many requests in a given amount of time. Please note that the CallMiner Media Ingestion API v2 has a default rate limit of 200 requests per minute per tenant. **Pro-tip**: If the program or script you are using to make requests against the Ingestion API supports HTTP persistent connection through the `KeepAlive` property, we recommend setting `KeepAlive` to *false*. Disabling this setting will improve performance and increase the number of requests you can make against the Ingestion API over the 200 requests per minute per tenant default. For more information, see https://learn.microsoft.com/en-us/keepalive.
500	Internal Server Error - The server encountered an unexpected condition, which prevented it from fulfilling the request. If the error persists, contact support@callminer.com.
504	Gateway Timeout - The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server or some other auxiliary server it needed to access to complete the request. If the error persists, contact support@callminer.com.

For a complete list of HTTP status codes, see https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.

Appendix E

ISO 8601 DateTime Formatting

To standardize the date and time format in CallMiner API requests, we use ISO 8601 format. ISO 8601 should be written in *yyyy-mm-ddThh:mm:ss.fffZ* format:

- **yyyy**: Indicates the year.
- **mm**: Indicates the month.
- **dd**: Indicates the day of the month.
- **T**: Indicates that a time value follows the data.
- **hh**: Indicates the hour in a 24-hour format.
- **mm**: Indicates the minutes.
- **ss**: Indicates the seconds.
- **fff**: Represents the number of milliseconds.
- **Z**: Indicates that the time is in UTC time.

For example, the fully formatted ISO 8601 DateTime for a contact recorded on April 10, 2019 at 2:10 PM would appear as *2019-04-10T14:10:00.000Z*.

Appendix F

How CallMiner Transforms Text

After you send in the `text` for each message in the text contact, your Eureka system will transform the formatting of the text behind the scenes. The Eureka system will make the following formatting changes:

Text Type	How It Is Transformed	Example

HTML tags	All HTML tags will be removed. Anything between **<** and **>** will be removed during processing, regardless of whether or not it is valid HTML.	If the `text` contains this:
		`<p>This is the best product.</p>`
		The final transcript in Eureka will look like this:
		`This is the best product.`

Multiple, consecutive new lines	If there are more than two consecutive new lines, they will be condensed into just two new lines.	If the `text` contains this:
		`Thank you.`
		
		`That's all I needed.`
		The final transcript in Eureka will look like this:
		`Thank you.`
		
		`That's all I needed.`

Multiple, consecutive spaces	Multiple consecutive spaces will be condensed down to a single space.	If the `text` contains this:

Text Type	How It Is Transformed	Example
		<pre>1 This is an example.</pre> The final transcript in Eureka will look like this: <pre>1 This is an example.</pre>
Emojis, punctuation, and numbers	Behind the scenes, your Eureka system breaks emojis, punctuation, and individual numerical digits into separate words to allow for complex searching in Analyze. Your total word count will also include words that have been redacted. This means that the word count you have calculated for a transcript may differ from the system-calculated word count for the same transcript.	If the `text` contains this: <pre>1 There were 100 products</pre> Instead of only counting as 4 words, the final transcript in Eureka will split up each numerical digit into it's own word and the final word count will be 6 for this message.