Adding Live captions and live translations

Last Modified on 10/15/2024 7:24 pm IDT

In this article we provide examples of calls and responses relative to Live Captions jobs in Reach integration flows.

Retrieving a Live captions job and associated information

Retrieving a Live captions job

Example of a request to retrieve a Live caption job using entryVendorTask.getJobs - Kaltura VPaaS API Documentation:

```
curl -X POST https://www.kaltura.com/api_v3/service/reach_entryvendortask/action/getJobs \
    -d "ks=$KALTURA_SESSION" \
    -d "filter[objectType]=KalturaEntryVendorTaskFilter"
```

Example of a response after requesting a Live caption job using entryVendorTask.getJobs - Kaltura VPaaS API Documentation:

```json
{
  "totalCount": 1,
  "objects": [
    {
      "id": "464420602",
      "partnerId": 5831172,
      "vendorPartnerId": 5173642,
      "createdAt": 1727259369,
      "updatedAt": 1727259369,
      "queueTime": 1727259369,
      "entryId": "1_0k5j8ixt",
      "status": 1,
      "reachProfileId": 287163,
      "catalogItemId": 33962,
      "price": 0,
      "userId": "test@test.com",
      "accessKey": "fjJ8NDgyMTRE2MnwM92Psr0JTe3xsIQMbsjdVBR2YJFebuqECTCoAhr-8xUJ9SteASydS1qHf1_4qblWTCPyRKRBcQCq
      "version": 0,
      "creationMode": 1,
      "taskJobData": {
        "entryDuration": 3600000,
        "startDate": 1727265600,
        "endDate": 1727269200,
        "scheduledEventId": 58812112,
        "objectType": "KalturaScheduledVendorTaskData"
      },
      "expectedFinishTime": 1727864169,
      "serviceType": 2,
      "serviceFeature": 8,
      "turnAroundTime": -1,
      "objectType": "KalturaEntryVendorTask"
    }
  ],
  "objectType": "KalturaEntryVendorTaskListResponse"
}
```

> 💡 Note the access key which will be used as ks value for further API calls.

> 💡 There can be multiple jobs for one same scheduled event.

Retrieving a Live captions task

Example of a request to retrieve a Live caption task using entryVendorTask.get - Kaltura VPaaS API Documentation:

```
curl -X POST https://www.kaltura.com/api_v3/service/reach_entryvendortask/action/get \
   -d "ks=$KALTURA_SESSION" \
   -d "id=464420602"
   -d "responseProfile[systemName]=reach_vendor"
```

Example of a response after requesting a Live caption job using
entryVendorTask.getJobs - Kaltura VPaaS API Documentation:

```json
{
  "relatedObjects": {
    "reach_vendor_catalog_item": {
      "totalCount": 1,
      "objects": [
        {
          "serviceType": 2,
          "serviceFeature": 8,
          "turnAroundTime": -1,
          "engineType": "OpenCalaisReachVendor.OPEN_CALAIS",
          "sourceLanguage": "English",
          "allowResubmission": false,
          "stage": 2,
          "contract": "",
          "createdBy": "",
          "notes": "",
          "enableSpeakerId": false,
          "minimalRefundTime": 30,
          "minimalOrderTime": 30,
          "durationLimit": 480,
          "objectType": "KalturaVendorLiveCaptionCatalogItem"
        }
      ],
      "objectType": "KalturaVendorCatalogItemListResponse"
    },
    "reach_vendor_profile": {
      "totalCount": 1,
      "objects": [
        {
          "name": "Default test profile",
          "defaultOutputFormat": 1,
          "enableMetadataExtraction": true,
          "enableSpeakerChangeIndication": false,
          "enableAudioTags": false,
          "enableProfanityRemoval": true,
          "maxCharactersPerCaptionLine": 26,
          "labelAdditionForMachineServiceType": "",
          "labelAdditionForHumanServiceType": "",
          "contentDeletionPolicy": 2,
          "flavorParamsIds": "",
          "vendorTaskProcessingRegion": 1,
          "objectType": "KalturaReachProfile"
        }
      ],
      "objectType": "KalturaReachProfileListResponse"
    }
  },
  "id": "464420602",
  "partnerId": 5831172,
  "vendorPartnerId": 5173642,
  "createdAt": 1727259369,
  "entryId": "1_0k5j8ixt",
  "status": 1,
  "reachProfileId": 287163,
  "catalogItemId": 33962,
  "accessKey": "fjJ8NDgyMTRE2MnwM92Psr0JTe3xsIQMbsjdVBR2YJFebuqECTCoAhr-8xUJ9SteASydS1qHf1_4qblWTCPyRKRBcQCqcMl
  "version": 0,
  "taskJobData": {
    "objectType": "KalturaScheduledVendorTaskData"
  },
  "objectType": "KalturaEntryVendorTask"
}
```

![Kaltura logo]

> 💡 Note that the task includes the `sourceLanguage` information.

Example of a response after requesting a Live caption job that was cancelled:

```
{
  "code": "ENTRY_VENDOR_TASK_NOT_FOUND",
  "message": "Entry vendor task item with id provided not found [468571812]",
  "objectType": "KalturaAPIException",
  "args": {
    "ID": "468571812"
  }
}
```

Retrieving catalog item details

Example of a request to retrieve catalog item details using vendorCatalogItem.get - Kaltura VPaaS API Documentation:

```
curl -X POST https://www.kaltura.com/api_v3/service/reach_vendorcatalogitem/action/get \
    -d "ks=$KALTURA_SESSION" \
    -d "id=33962"
```

Example of a response after requesting catalog item details using vendorCatalogItem.get - Kaltura VPaaS API Documentation:

```
{
  "id": 33962,
  "vendorPartnerId": 5173642,
  "name": "test-live-captions-eng-withTimeFields",
  "systemName": "test-live-captions-eng-withTimeFields",
  "createdAt": 1727259141,
  "updatedAt": 1727259141,
  "status": 2,
  "serviceType": 2,
  "serviceFeature": 8,
  "turnAroundTime": -1,
  "engineType": "OpenCalaisReachVendor.OPEN_CALAIS",
  "sourceLanguage": "English",
  "allowResubmission": false,
  "stage": 2,
  "contract": "",
  "createdBy": "",
  "notes": "",
  "enableSpeakerId": false,
  "fixedPriceAddons": 0,
  "minimalRefundTime": 30,
  "minimalOrderTime": 30,
  "durationLimit": 480,
  "objectType": "KalturaVendorLiveCaptionCatalogItem"
}
```

💡 Note the information ab out the catalog item (also found in the task):

- `minimalRefundTime` - minimum number of minutes that a Reach end-user can cancel a job order before captioning start time and still get refunded

- `minimalOrderTime` - minimum time (in minutes) to place an order before captioning session start

- `durationLimit` - maximum captioning session duration

Retrieving a scheduled event

Example of a request to retrieve a scheduled event using scheduleEvent.get - Kaltura VPaaS API Documentation:

```
curl -X POST https://www.kaltura.com/api_v3/service/schedule_scheduleevent/action/get \
  -d "ks=$KALTURA_SESSION" \
  -d "scheduleEventId=58812112"
```

Example of a response after requesting a scheduled event using scheduleEvent.get - Kaltura VPaaS API Documentation:

```
{
  "id": 58812112,
  "partnerId": 5831172,
  "summary": "test townhall 3",
  "description": "",
  "status": 2,
  "startDate": 1727265600,
  "endDate": 1727269200,
  "classificationType": 1,
  "organizer": "889800133a387a16d64e5bc83a6d351b8cbfd07ac429035108755612c27493fc",
  "ownerId": "kmsAdminServiceUser",
  "sequence": 11,
  "recurrenceType": 0,
  "duration": 3600,
  "tags": "",
  "createdAt": 1727259266,
  "updatedAt": 1727259369,
  "templateEntryId": "1_0k5j8ixt",
  "blackoutConflicts": [],
  "projectedAudience": 0,
  "preStartTime": 0,
  "postEndTime": 0,
  "liveFeatures": [
    {
      "systemName": "LiveCaptionFeature-reach-464420602",
      "preStartTime": 0,
      "postEndTime": 0,
      "language": "eng",
      "objectType": "KalturaLiveCaptionFeature"
    }
  ],
  "objectType": "KalturaLiveStreamScheduleEvent"
}
```

> 💡 As one scheduled event can have multiple tasks (for example multiple live translations in addition to live captions), these different tasks are listed as individual `liveFeatures`.
>
> For each task and `liveFeature` the vendor will update the RTMP endpoint and Websocket information.
>
> Note the `systemName` under `liveFeatures` : the suffix value corresponds to the Vendor task id retrieved in the task.
>
> The vendor can parse or match the suffix of the `systemName` field to identify a specific task.

Delivering a Live captions or Live translations job

Updating a scheduled event

Example of a request to update a scheduled event using scheduleEvent.updateLiveFeature - Kaltura VPaaS API Documentation:

```
curl -X POST https://www.kaltura.com/api_v3/service/schedule_scheduleevent/action/updateLiveFeature \
    -d "ks=$KALTURA_SESSION" \
    -d "scheduledEventId=58812112" \
    -d "featureName=LiveCaptionFeature-reach-464420602" \
    -d "liveFeature[objectType]=KalturaLiveCaptionFeature" \
    -d "liveFeature[captionToken]=qfdfdsqfdsfqdsfsqfdsq" \
    -d "liveFeature[captionUrl]=http%3A%2F%2Ftest.test.com%3A43251" \
    -d "liveFeature[mediaKey]=iouzralkdsfqmjldsq" \
    -d "liveFeature[mediaUrl]=http%3A%2F%2Fstream.stream.com%3A32090"
```

💡 Note the `featureName` value maps to the `systemName` value under `liveFeatures` returned when retrieving the scheduled event.

Example of a response after updating a scheduled event (and adding RTMP endpoint and Websocket information) using scheduleEvent.updateLiveFeature - Kaltura VPaaS API Documentation:

```
{
  "id": 58812112,
  "partnerId": 5831172,
  "summary": "test townhall 3",
  "description": "",
  "status": 2,
  "startDate": 1727265600,
  "endDate": 1727269200,
  "classificationType": 1,
  "organizer": "889800133a387a16d64e5bc83a6d351b8cbfd07ac429035108755612c27493fc",
  "ownerId": "kmsAdminServiceUser",
  "sequence": 11,
  "recurrenceType": 0,
  "duration": 3600,
  "tags": "",
  "createdAt": 1727259266,
  "updatedAt": 1727260995,
  "templateEntryId": "1_0k5j8ixt",
  "blackoutConflicts": [],
  "projectedAudience": 0,
  "preStartTime": 0,
  "postEndTime": 0,
  "liveFeatures": [
    {
      "systemName": "LiveCaptionFeature-reach-464420602",
      "preStartTime": 0,
      "postEndTime": 0,
      "mediaUrl": "http%3A%2F%2Fstream.stream.com%3A32090\"",
      "mediaKey": "iouzralkdsfqmjldsq",
      "captionUrl": "http%3A%2F%2Ftest.test.com%3A43251",
      "captionToken": "qfdfdsqfdsfqdsfsqfdsq",
      "language": "eng",
      "objectType": "KalturaLiveCaptionFeature"
    }
  ],
  "objectType": "KalturaLiveStreamScheduleEvent"
}
```

Updating a job status to Scheduled

Example of a request to update a job using entryVendorTask.updateJob - Kaltura VPaaS API Documentation:

```
curl -X POST https://www.kaltura.com/api_v3/service/reach_entryvendortask/action/updateJob \
   -d "ks=$KALTURA_SESSION" \
   -d "id=471386462" \
   -d "entryVendorTask[status]=9" \
   -d "entryVendorTask[objectType]=KalturaEntryVendorTask"
```

 Example of a response after updating a job status:

```
{
 "id": "471386462",
 "partnerId": 4811563,
 "vendorPartnerId": 3973671,
 "createdAt": 1729006886,
 "updatedAt": 1729007748,
 "queueTime": 1729006886,
 "entryId": "1_a2eizyd3",
 "status": 9,
 "reachProfileId": 287163,
 "catalogItemId": 33962,
 "price": 0,
 "userId": "889800133a387a16d64e5bc83a6d151b8cbfd07ac429035108745612c27493fc",
 "accessKey": "djJ8EDgyMTE2Mnyfnpl506_cBE-64pO7j-Fy6qyVDZn-5YwCPMG93RiyrHFxqzEGS7RBpsrDfb0p433FS8Of7zWSS2hDjRp
 "version": 0,
 "creationMode": 1,
 "taskJobData": {
  "entryDuration": 3600000,
  "startDate": 1729166400,
  "endDate": 1729170000,
  "scheduledEventId": 59073162,
  "objectType": "KalturaScheduledVendorTaskData"
 },
 "expectedFinishTime": 1729611686,
 "serviceType": 2,
 "serviceFeature": 8,
 "turnAroundTime": -1,
```

Websocket response - schema

Responses returned by the Streaming Speech Recognition services should have the following schema:

```
{
    "response": {
        "id": string (UUID),
        "type": "transcript" | "captions",
        "service_type": "transcription" | "translation",        "language_code": string,
        "start": float,
        "end": float,
        "start_pts": float,
        "start_epoch": float,
        "is_final": boolean,
        "is_end_of_stream": boolean,
        "speakers": [
            {
                "id": string (UUID),
                "label": string | null
            }
        ],
        "alternatives": [
            {
                "transcript": string,
                "start": float,
                "end": float,
                "start_pts": float,
                "start_epoch": float,
                "items": [
                    {
                        "start": float,
                        "end": float,
                        "kind": "text" | "punct",
                        "value": string,
                        "speaker_id": string (UUID)
                    }
                ]
            }
        ]
    }
}
```

Websocket response - fields description

- `"response"` - The root element in the response JSON

  - `"id"` - A unique identifier of the response (UUID)

  - `"type"` - The response type. Can be either "transcript" or "captions" (See explanation in below note).

  - `"service_type"` can be either " `transcription"` or " `translation"` and is related to which kind of service generated this response:

    - `"transcription"` means that the response was generated by performing speech-to-text on the audio speech (i.e. it will always be performed on the original audio language).

- "translation" means that the response was generated by performing machine translation on the text generated in the "transcription" responses.

- `"language_code"` - can be any ISO language code (e.g. "en-US", "es-ES", "pt-BR", etc.) and represents the language of the text in the response.

  - For responses of " `transcription`" service type, the `"language_code"` field will always be equal to the language of the original audio language (as was specified in the order).

  - For responses of " `translation`" service type, the `"language_code"` field will be one of the requested translation languages (as was specified in the order).

- `"start"` - The start time of the utterance. Measured in seconds from the beginning of the media stream.

- `"end"` - The (current) end time of the utterance. Measured in seconds from the beginning of the media stream.

- `"start_pts"` - The pts value corresponding to the `"start"` of the response, as received from the input media stream. Measured in seconds.

  - Note: if the input media stream doesn't provide pts values, this field will have the same value as `"start"` .

- `"start_epoch"` - The epoch timestamp at which the media corresponding to the `"start"` of the response was received.

- `"is_final"` - A boolean denoting whether the response is the final one for the utterance (See explanation in below note). For a "captions" response, this is always set to `"true"` , since captions are not incrementally updated (thus, each "captions" response is final).

- `"is_end_of_stream"` - A boolean denoting whether the response is the last one for the entire media stream

- `"speakers"` - A list of objects representing speakers in the media stream, as identified by the speech recognition service.

  - `"id"` - A unique identifier of the speaker (UUID)

  - `"label"` - A string representing the speaker. Only available in sessions with human transcribers in the loop. This field is set to `null` by default.

- `"alternatives"` - A list of alternative transcription hypotheses. At least one alternative is always returned.

    - `"transcript"` - A textual representation of the alternative in the current response.

    - `"start"` - Same as `["response"]["start"]`.

    - `"end"` - Same as `["response"]["end"]`.

    - `"start_pts"` - Same as `["response"]["start_pts"]`.

    - `"start_epoch"` - Same as `["response"]["start_epoch"]`.

    - `"items"` - A list containing textual items (words and punctuation marks) and their timings.

        - `"start"` - The start time of the item. Measured in seconds from the beginning of the media stream.

        - `"end"` - The end time of the item. Measured in seconds from the beginning of the media stream.

        - `"kind"` - The item kind. Can be either "text" or "punct" (a punctuation mark).

        - `"value"` - The item textual value

        - `"speaker_id"` - The unique identifier of the speaker that this item is associated with. Corresponds with an `"id"` of one of the speakers in the `"speakers"` field.

Websocket - Response types

There are two types of responses - "transcript" and "captions":

1. **Transcript**: this type of response contains the recognized words since the beginning of the current utterance. Like in real human speech, the stream of words is segmented into utterances in automatic speech recognition. An utterance is recognized incrementally, processing more of the incoming audio at each step. Each utterance starts at a specific start-time and extends its end-time with each step, yielding the most updated result. Note that sequential updates for the same utterance will overlap, each response superseding the previous one - until a response signaling the end of the utterance is

received (having `is_final == True`). The `alternatives` array might contain different hypotheses, ordered by confidence level.

2. **Captions**: this type of response contains the recognized words within a specific time window. In contrast to the incremental nature of "transcript"-type responses, the "captions"-type responses are non-overlapping and consecutive. Only one "captions"-type response covering a specific time-span in the audio will be returned (or none, if no words were uttered). The `is_final` field is always `True` because no updates will be output for the same time-span. The `alternatives` array will always have only one item for captions.

Responses on silent audio segments

It should be noted that "transcript" and "captions" responses behave differently when the audio being transcribed is silent:

- "transcript" responses are sent regardless of the audio content, in such a way that the entire audio duration is covered by "transcript" responses. In case of a silent audio segment, "transcript" responses will be sent with an empty word list, but with timestamps which mark the portion of the audio that was transcribed.
- "captions" responses are sent only when the speech recognition output contains words. In case of a silent audio segment, no "captions" responses will be sent, since a caption doesn't make sense without any words. Therefore, "captions" responses will not necessarily cover the entire audio duration (i.e. there may be "gaps" between "captions" responses).

Websocket - Service types

The received responses may originate in one of two types of service:

1. **Transcription**: responses generated by a Speech Recognition service which converts speech in the input media stream to text. Responses with "transcription" service type will have the same language of the input media stream (as specified in the order).

2. **Translation**: responses generated by a Machine Translation service which translates "transcription" responses (in the input language) to one or more target languages (as specified in the order).

> 💡 Note: Due to natural differences between languages, translated responses may/can diverge in word count and word order. Since translated words were never really uttered in the original audio, they do not have "real" timings. Therefore, words in translation responses are assigned timings which are expected to be heuristically distributed within the time boundaries of the source language utterance. Heuristic timings may be used for synchronization purposes like displaying translated content in alignment with the media.

Websocket - Error handling and recovery

Initial connection

In case the Kaltura WebSocket client fails to establish the initial connection with the service, e.g. due to temporary unavailability, it will perform exponential retry, up to a configurable max value.

During a session

In case the connection to the service is dropped during a session, the behavior of the Kaltura WebSocket client will attempt to reconnect in case the connection was closed prematurely, as many times as needed, until the final response is received (or some non-retryable error occurrs).

Websocket - Idle streams

As the customer's media stream is re-streamed to the vendor via RTMP, there may be times when no messages are sent over the WebSocket. For example:

- The external media source hasn't started yet,

- The media stream is silent and only "captions" responses were requested.

In case no message is sent over the WebSocket for more than 10 minutes, the connection can be dropped by the vendor, and will need to be re-established. To prevent these undesired disconnections, the Kaltura WebSocket client will send a "ping" message at least once every 10 minutes.

It is expected that the vendor will respond with a "pong" message that the Kaltura WebSocket client will handle.

Websocket - Connection duration limit

As the customer's media stream is re-streamed to the vendor via RTMP, the maximum allowed connection duration is 2 hours. After that time, the vendor will drop the connection with a "Going Away" (code: 1001) close message. In such cases, it is Kaltura's responsibility to reconnect.